



High-Performance Computing in Simulating Carbon Dioxide Geologic Sequestration Doctoral Qualification Exam Report

Eduardo J. Sanchez, Christopher P. Paolini, and Jose E. Castillo

July 2013

Publication Number: CSRCR2013-01

Computational Science &
Engineering Faculty and Students
Research Articles

Database Powered by the
Computational Science Research Center
Computing Group & Visualization Lab

COMPUTATIONAL SCIENCE & ENGINEERING



**SAN DIEGO STATE
UNIVERSITY**

Computational Science Research Center
College of Sciences
5500 Campanile Drive
San Diego, CA 92182-1245
(619) 594-3430



High-Performance Computing in Simulating Carbon Dioxide Geologic Sequestration

Doctoral Qualification Exam Report

Eduardo J. Sanchez*, Christopher P. Paolini† and Jose E. Castillo‡

July 2, 2013

Abstract

Carbon Dioxide (CO₂) Capture, Utilization and Sequestration is a collection of technologies that seek to minimize the environmental impact of greenhouse gases. Specifically, to study and simulate the long-term effects of geologic CO₂ sequestration in terms of concentration profiles, numerical water-rock interaction and reactive transport models are employed. Traditionally, numerical codes that simulate water-rock interaction and reactive transport sequentially solve an elemental mass-balance equation for a given lithology containing some fraction of brine, and a number of charged aqueous solute species. Pressure, temperature, and solute concentrations are then sequentially solved in separate modules and coupled through an iterative process, until a convergence criteria is satisfied. Coupling is achieved by iterating between the discretization of the conservation of mass and energy equations, together with equations modeling kinetic and equilibrium reactions. When solving for the concentration of solutes species, mass-transfer coefficient matrices constructed from formation and injectant water velocities and solute concentrations, derived from a previous iteration, are structured and then solved at each grid point, using direct methods for the solution of systems of linear equations. However, this formulation is not well suited for execution on distributed-memory computer clusters. In this work, we present a numerical scheme whereby all solute concentrations in all control volumes are solved simultaneously by constructing a large block-banded sparse matrix of rank $N_a \times n_x$, where N_a is the number of solutes species, and n_x is the number of control volumes. The generated matrices are then factored using tools provided by different APIs implementing direct methods for the solution of systems of linear equations. Performance metrics are considered to compare our large block-banded matrix scheme against a sequential implementation on `blackbox.sdsu.edu`, a local cluster located at *San Diego State University*, as well as in the $\sim 10K$ cores *XSEDE* cluster `trestles.sdsc.edu`. Simulations based on the Frio Formation pilot test case are studied with respect to achieved speedup, efficiency, scalability, and grid refinement.

*Computational Science Research Center, 5500 Campanile Dr, San Diego State University, College of Sciences, San Diego, California, USA, 92182-1245 - esanchez@sciences.sdsu.edu.

†Research Adviser. Computational Science Research Center, 5500 Campanile Dr, San Diego State University, College of Sciences, San Diego, California, USA, 92182-1245 - paolini@engineering.sdsu.edu

‡Academic Adviser. Computational Science Research Center, 5500 Campanile Dr, San Diego State University, College of Sciences, San Diego, California, USA, 92182-1245 - jcastillo@mail.sdsu.edu.

This page has been intentionally left blank for printing purposes.

Contents

| | | |
|-----------|---|-----------|
| 1 | Introduction | 7 |
| 1.1 | Carbon Dioxide Capture, Utilization and Sequestration | 7 |
| 1.2 | The importance of simulating the long-term evolution of the sequestered carbon dioxide | 8 |
| 2 | Organization of the article | 9 |
| 3 | Water-rock interaction and reactive mass transport models | 9 |
| 3.1 | The geology of the processes in sequestering carbon dioxide | 9 |
| 3.2 | The physicochemical properties of carbon dioxide | 10 |
| 3.3 | Mathematical modeling of WRI and reactive transport in geologic media | 11 |
| 4 | The algorithmics of simulating the long-term evolution of the sequestered carbon dioxide | 13 |
| 5 | Reference pilot test case: gas-water-rock interaction in the Frio Formation | 14 |
| 6 | Results for the sequential implementation | 14 |
| 6.1 | Considered hardware platforms and <i>CSRCnet</i> | 14 |
| 6.2 | Attained physical results | 15 |
| 6.3 | A profile analysis of the simulation software | 15 |
| 6.4 | Improving the sequential solvers | 15 |
| 7 | A Block-defined, Global and Sparse (BloGS) matrix storage scheme for the numerical resolution of many partial differential equations on distributed-memory computers | 18 |
| 7.1 | A simplified prototype test case: a calcite dissolution reaction | 20 |
| 7.2 | Results of the sequential implementation | 23 |
| 7.3 | Results of the distributed implementation | 26 |
| 8 | The performance of <i>WebSym.C</i> using the BloGS scheme | 30 |
| 9 | Concluding remarks and directions of future work | 30 |
| 10 | Acknowledgments | 31 |
| A | A generalized (BloGS) matrix with examples | 35 |
| B | The Mimetic Methods Toolkit | 41 |

List of Tables

| | | |
|---|--|----|
| 1 | Summary of the notational conventions adopted in this work. | 6 |
| 2 | Comparison of considered hardware platforms in terms of performance characteristics. See §6.1. | 15 |
| 3 | Top 10 percentages of invested computation time (in seconds) per routine in <i>WebSym.C</i> , as computed by <i>GNU gprof</i> [Fenlason, 1993] in <code>blackbox.sdsu.edu</code> . See §6.3 and Figure 6. The average of 5 instances of 100 cells each of the pilot test case described in §5 was considered for this profiling study. | 16 |
| 4 | Attained execution times (in minutes) from replacing the reference sequential solver with those discussed in §6.4. The averages of 5 executions were taken per each case. | 18 |
| 5 | Comparison of selected solvers to work with. See §7.3. | 26 |
| 6 | Execution times (in seconds) for the proposed test case using SuperLU_DIST on <code>blackbox.sdsu.edu</code> . See §7.3. | 26 |

| | | |
|---|--|----|
| 7 | Execution times (in seconds) using ScaLAPACK on <code>blackbox.sdsu.edu</code> . See §7.3. | 26 |
| 8 | Attained qualities for low-rank matrices. See §7.3 and Figure 14a. | 27 |
| 9 | Attained qualities for high-rank matrices. See §7.3 and Figure 14b. | 28 |

List of Figures

| | | |
|----|--|----|
| 1 | Conceptualization of the process of CCUS (See §1.1) highlighting active research areas. See §1.2. Source: [NETL, 2011]. | 8 |
| 2 | Summary of the major compositional divisions of planet Earth, as a function of depth. See §3.1. Source: Adapted from information given in [Walther, 2009]. | 10 |
| 3 | $p-T$ diagram for CO_2 . See §3.2. Source: Created from models given in [Marini, 2006]. | 11 |
| 4 | Schematics of the algorithmics of <i>WebSym.C</i> present at the numerical core called <i>Sym.8</i> . See §3.3. Source: [Park, 2009]. | 13 |
| 5 | Reference solution at 5 years after injection, computed in <code>blackbox.sdsu.edu</code> . We considered the reference implementation of the LU factorization, which is provided in [Press et al., 1988]. For different grid resolutions, we computed the concentration of CO_2 , H^+ , and Fe^{++} , as a function of distance from the injection well. Reference results are the same for <code>trestles.sdsc.edu</code> | 16 |
| 6 | Highest 10 percentages of invested computation time (in seconds) per routine in the original version of <i>WebSym.C</i> , as computed by <i>GNU gprof</i> in <code>blackbox.sdsu.edu</code> . See §6.3 and Table 3. The average of 5 instances of 100 cells each of the pilot test case described in §5 was considered for this profiling study. | 17 |
| 7 | Behavior of the rank of the BloGS matrices, as a function of the number of solutes N_a and the number of nodes, n_x . See §7. Coloring is simply a result of the plotting software used; it means nothing special except that it varies proportionally to the quantity of being plotted. | 19 |
| 8 | Bandwith and (absolute) density of the BloGS matrices, which are properties dependant on the chosen order of accuracy, ω . See §7. Coloring is simply a result of the plotting software used; it means nothing special except that it varies proportionally to the quantity of being plotted. | 21 |
| 9 | Projections depicting the behavior of the important properties of a BloGS matrix. See §7. Coloring means nothing. | 22 |
| 10 | Attained results for the MATLAB R2008a prototype driver for the solution of a BloGS system. See §7.2. | 24 |
| 11 | Behavior of the condition number of the BloGS matrices, as a function of the rank $r(n_x)$, which is defined by the number of nodes, n_x . See §7. | 24 |
| 12 | Analytical and computed solutions for the LAPACK and SuperLU_SEQ prototype drivers for the solution of a BloGS system. See §7.2. Coloring distinguishes the analytical and computed solution. Specifically, dots depict the computed solution whereas the line connecting the hollow circles depict the analytical solution. | 25 |
| 13 | Analytical and computed solutions for the LAPACK and SuperLU_SEQ prototype drivers for the solution of a BloGS system. See §7.3. p stands for number of processing cores. | 28 |
| 14 | Attained qualities for the speedup under a more comprehensive (r, β) -space (rank and bandwidth). See §7.3 and Tables 8 and 9. Coloring is simply a result of the plotting software used; it means nothing special except that it is useful to visualize differentiate the different collection of values being plotted. | 29 |
| 15 | Current architecture of <i>WebSym.C</i> . See §5 and 4. | 30 |

Notational conventions

In this work, we shall take notational conventions very seriously. Notation is our only interface with the already complex world of the abstract theories we will be dealing with, so why not give it its importance?

1. We shall denote both integer and continuous scalar-valued quantities, say rank of a system, temperature, or pressure, with the default math pseudo-italicized font, using combinations of both lower and uppercase Latin letters and lowercase Greek letters: $a, \dots, z, A, \dots, Z, \alpha, \dots, \omega$. Discretized instances shall be identified with a tilde accent, and will be assumed to be computationally implemented as row-wise-defined arrays.
2. We shall denote continuous vector-valued quantities using boldfaced lowercase Latin letters: $\mathbf{a}, \dots, \mathbf{z}$. Discretized instances shall be identified with a tilde accent.
3. We shall denote matrices using boldfaced uppercase Latin and Greek letters: $\mathbf{A}, \dots, \mathbf{Z}, \mathbf{\Gamma}, \dots, \mathbf{\Omega}$.
4. We shall denote continuous tensor-valued quantities using scripture-styled uppercase Latin letters: $\mathcal{A}, \dots, \mathcal{Z}$. Discretized instances shall be identified with a tilde accent.
5. We shall denote continuous differential operators using standard notation from Calculus. When it comes to their discrete matrix analog operators, we will use boldfaced uppercase Latin Letters with a tilde accent, thus emphasizing the approximation to a continuous operator they intent to. However, those operators built by means of the Castillo–Grone Mimetic Discretization Method shall be identified with a breve accent. As a side note, this notational convention is supported by the fact that, on German cartography, a breve accent placed over two letters is often used in abbreviated place names that end in “*bg*”, as a short for “*burg*”, a common suffix originally meaning “Castle”, which is English for “Castillo”. This prevents misinterpretation since “*berg*” is another common suffix in place names, which means “mountain”. Thus, for example, “*Freibg*” stands for “*Freiburg*”, not “*Freiberg*”. Furthermore, its is also mnemonic, since it resembles a letter ‘C’.
6. We shall denote sets using italic uppercase Greek letters: A, \dots, Ω . Discretized instances shall be identified with a tilde accent. Numerical sets will be denoted with blackboard boldfaced Latin uppercase letters: $\mathbb{A}, \dots, \mathbb{Z}$.
7. Chemical species for which to keep track of their phase is important will be subscripted using the following symbols: {so, li, ga, sc}, which denote solid, liquid, gaseous, and supercritical phase, respectively.

Table 1 summarizes our notational conventions. An important thing to notice is that, when accessed by means of indexing the elements they may contain, the objects shall not conserve their typographical style, thus yielding a default math pseudo-italicized font. For example, notice that tensors loose their typographical style, in this case, their scripture style, thus yielding a default math pseudo-italicized font uppercase Latin letter. This can be depicted in columns three and four of Table 1. However, when objects are indexed as being part of a enumerable set, they will preserve their typographical style.

| Object | Continuous domain | Discrete domain | Indexed |
|----------|--|--|--|
| Scalar | $a, \dots, z, A, \dots, Z, \alpha, \dots, \omega$ | $\tilde{a}, \dots, \tilde{z}, \tilde{A}, \dots, \tilde{Z}, \tilde{\alpha}, \dots, \tilde{\omega}$ | $a_i, \dots, z_i, A_i, \dots, Z_i, \alpha_i, \dots, \omega_i$ |
| Vector | $\mathbf{a}, \dots, \mathbf{z}$ | $\tilde{\mathbf{a}}, \dots, \tilde{\mathbf{z}}$ | a_i, \dots, z_i |
| Matrix | $\mathbf{A}, \dots, \mathbf{Z}, \mathbf{\Gamma}, \dots, \mathbf{\Omega}$ | $\mathbf{A}, \dots, \mathbf{Z}, \mathbf{\Gamma}, \dots, \mathbf{\Omega}$ | $a_{ij}, \dots, z_{ij}, \alpha_{ij}, \dots, \omega_{ij}$ |
| Tensor | $\mathcal{A}, \dots, \mathcal{Z}$ | $\tilde{\mathcal{A}}, \dots, \tilde{\mathcal{Z}}$ | A_{ij}, \dots, Z_{ij} |
| Operator | $\nabla, \nabla \cdot, \dots$ | $\tilde{\mathbf{A}}, \dots, \tilde{\mathbf{Z}}$ or $\check{\mathbf{A}}, \dots, \check{\mathbf{Z}}$ | $\tilde{A}_{ij}, \dots, \tilde{Z}_{ij}$ or $\check{A}_{ij}, \dots, \check{Z}_{ij}$ |
| Set | A, \dots, Ω or $\mathbb{A}, \dots, \mathbb{Z}$ | $\tilde{A}, \dots, \tilde{\Omega}$ | $\tilde{A}_i, \dots, \tilde{\Omega}_i$ |

Table 1: Summary of the notational conventions adopted in this work.

1 Introduction

Nowadays, the conversion of energy into electrical energy, or production of electrical energy, is a very important challenge that mankind has to face. Different production methods exist, each with different approaches for both, usage of resources and interaction with the environment. Based on the diversity of these approaches, current production methods can be classified according to an important dichotomy that has been somehow established, as a matter of course. Production methods can be classified as “green” energy production methods or “gray” energy production methods.

Green energy production methods comprise production technologies based on renewable natural resources. Examples include solar, eolic, hydrological, and geothermal energy production, among others. These methods are characterized for having a very efficient interaction with the environment, in the sense that the anthropogenic impact of implementing these methods is relatively low. However, renewable natural resources are highly dependent on both geographical features and climatological conditions; thus minimizing their reliability.

Gray energy production methods comprise production technologies based on naturally occurring fossil fuels, such as oil, natural gas, or coal. These methods are generally more reliable than green methods, since they utilize resources that are naturally abundant; however, their environmental impact is stronger than that of green methods, given the fact that greenhouse gas (GHG) emissions are a direct consequence of burning fossil fuels [EPA, 2012a].

The steady accumulation of GHGs from the combustion of fossil fuels has led to an increase in the amount of solar radiation trapped between the atmosphere and the earth [Shakun, 2012]. This increased radiation raises the temperature of the earth’s atmosphere and ocean systems. Many researchers believe that the continuing increment in temperature will lead to catastrophic changes in weather conditions around the globe [White et al., 2003]. Therefore, with Carbon Dioxide (CO_2) being the most abundant GHG, many efforts are underway to reduce the levels of CO_2 entering the atmosphere. For example, in April of 2012, the United States of America (USA) Environmental Protection Agency (EPA) proposed a regulatory legislative framework to limit GHG emissions from new fossil fuel-fired power plants by limiting CO_2 emissions [EPA, 2012b].

1.1 Carbon Dioxide Capture, Utilization and Sequestration

Carbon Dioxide Capture, Utilization and Sequestration (CCUS) is a collection of technologies that intend to minimize the environmental impact of GHGs that arise from the combustion of fossil fuels. Specifically, these technologies seek to first separate and capture the CO_2 from flue gases expelled by coal-fired power plants; the collected CO_2 is then transported (if required) to the injection site, where it is compressed above the critical pressure. As it is injected into underground formations, such as depleted oil reservoirs, gas reservoirs, or deep brine aquifers, the geothermal gradient heats the CO_2 to a above the critical temperature, thus taking it to a supercritical phase ($\text{CO}_{2(\text{sc})}$) [White et al., 2003](In §3.2, we will summarize the most important aspects of the physicochemical properties of CO_2).

Once in the underground reservoir, the CO_2 can be utilized for further purposes. An example of CO_2 utilization, which is key for the economic appeal of CCUS, is established by the assortment of Enhanced Oil/Gas Recovery (EOR/EGR) methods in hydrocarbons extraction [Ewing, 1983]. Several methodologies for oil extraction have been broadly studied and implemented thus far. Unfortunately, most of these techniques are still not effective and significant amounts of hydrocarbons often remain in the reservoir (50% or more). In order to recover more of the residual hydrocarbons, several EOR methods involving complex chemical and thermal effects have been developed. One method for EOR is based on the injection of gases like CO_2 , which mix with the resident hydrocarbon with a phase change to form a single fluid phase. If complete mixing or miscibility is attained, the fluids will flow together in one phase, eliminating distinction between possible phases, thus making complete hydrocarbon recovery theoretically possible, since the miscible phase flows more readily than the oil.

Once properly utilized (if so), the CO_2 is to remain sequestered in the chosen underground formation, staying away from the atmosphere. The success of the sequestration is inferred by means of constant monitoring efforts, for which the proper technology is used.

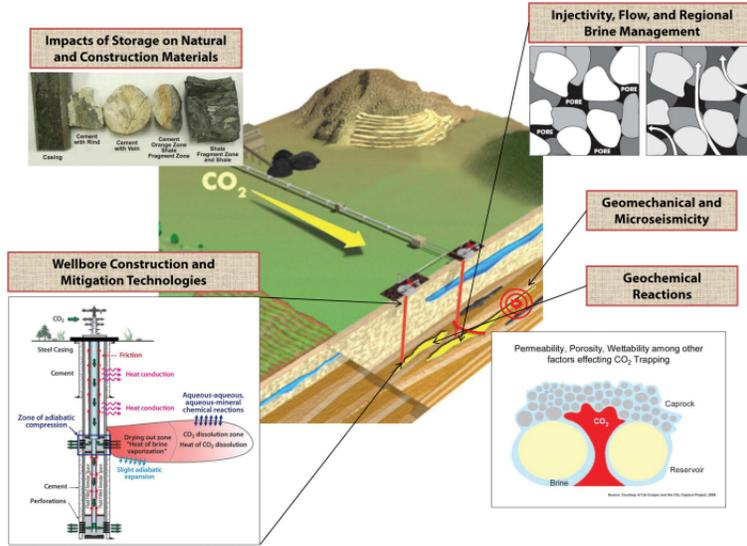


Figure 1: Conceptualization of the process of CCUS (See §1.1) highlighting active research areas. See §1.2. Source: [NETL, 2011].

1.2 The importance of simulating the long-term evolution of the sequestered carbon dioxide

As it was already mentioned, CCUS represents a promising alternative to help mitigating global warming. However, if significant amounts of CO_2 are to be sequestered in underground reservoirs, it is clear that the geochemical implications have to be analyzed. Figure 1 shows an schematic of the process of CCUS (process explained in §1.1), in where the need for the study of the geochemical reactions following injection is depicted as an active research area [NETL, 2011, Jun et al., 2013].

An example of one important topic is the study of large-scale pressure build-ups in response to the injection, and how would they limit the storage capacity of suitable formations. Over-pressurization may fracture the caprock, thus causing leakage and induced seismicity [Zhou and Birkholzer, 2011, Song and Zhang, 2013]. The chance of leaking represents a significant risk [Harvey et al., 2013]. A frequently cited example is the disaster occurred on August, the 21st of 1986, when roughly one cubic kilometer of gaseous CO_2 escaped into the atmosphere from the floor of Lake Nyos in the hilly jungle terrain of western Africa. By sunrise, more than 1,700 people and 3,200 animals had died of asphyxiation [Pentland, 2008]. However, the benefits of CCUS make it hard to ignore, since it has been showed that power plants equipped with CCUS technology produce about 80% to 90% less CO_2 than those without it. Also, CCUS could reduce the cost of climate stabilization by 30% [Pentland, 2008], and it is believed that CO_2 can remain sequestered in such formations, depending on the chemical and mechanical characteristics of the underground resident water and rock constituents, for at least one thousand years.

In this work, we will study the evolution of the concentration of the injected CO_2 , in order to analyze the potential for CO_2 sequestration in time. Therefore, we will only be concerned with the sequestration stage of the CCUS process pipeline described in §1.1. Plenty of work is being done in other fields of CCUS. Research on CO_2 capture is addressed in [Soong et al., 2012, Northington et al., 2012, Golombek et al., 2009], and research on numerical simulations can be found in [Zaman et al., 2012, Liu and Wilcox, 2013, Movagharnjad and Akbari, 2011]. Research on transport [Ji and Zhu, 2013, McCoy and Rubin, 2008, ZEP, 2011], Enhanced Oil/Gas Recovery methods [Jaramillo et al., 2009, Suebsiri et al., 2006, Khoo and Tan, 2006], and post-injection CO_2 monitoring [Bao et al., 2013, Seto and McRae, 2011, McAlexander et al., 2011], has also been undergone.

2 Organization of the article

This work is organized as follows: In §1 we provide an introduction to CCUS (§1.1), as well as some of the important research that has been undergone, which emphasizes the importance of simulating the long-term evolution of the sequestered CO₂ (§1.2). In §3, we discuss the mathematics and the geology of water-rock interaction and reactive mass transport. Specifically, §3.1 focuses on the geological context of CO₂ sequestration, §3.2 studies the physicochemical aspects of CO₂, and §3.3 presents the mathematics modeling these processes. In §4, we explain the algorithmical methodology we implement to study the sequestration scenarios, as well as the related numerical analysis and discretization methods, and we present *WebSym.C* as the software implementing such methodology (§4). Once the complete theoretical background has been discussed, in §5, we present the pilot test case upon which we have validated our numerical results. The results of improving *WebSym.C* by means of replacing the solver in its sequential version are summarized in §6. Specifically, we summarize the most important aspects of all of the considered hardware platform in this work (§6.1), and we show the computational results for the simulation of the chosen pilot test case (§6.2), for which we perform a profile study, thus identifying the source of the most intense computational burden (§6.3). By means of comparing the execution times of the different selected sequential solvers against the reference implementation, we conclude the attained performance of the sequential version of *WebSym.C* (§6.4). Motivated by the further improvement of *WebSym.C*, in §7, we introduce the BloGS scheme for the parallel global solution of the involved solutes in CO₂ sequestration scenarios. We explain its most important properties, and we show some examples (Appendix A). We present a simplified geochemical example (§7.1), through which we show the feasibility of this approach for solving multiple PDEs by means of a single system of equations, with properties that make it suitable for high-performance distributed solvers. We present results for this example, using sequential (§7.2) and distributed solvers (§7.3). Finally, we present the applications of this scheme to *WebSym.C* (§8). Concluding remarks and directions of future work are given in §9.

3 Water-rock interaction and reactive mass transport models

In this work, we are interested in analyzing the long-term behavior of the sequestered CO₂ in deep brine aquifers, since it has been claimed that, deep saline brine aquifers will likely become preferred geologic storage sites, because of their estimated geologic storage sites [Kharaka et al., 2006]. In order to perform this analysis, water-rock interaction (WRI) and reactive mass transport models are an invaluable resource [NETL, 2011]. In this section, we begin by presenting important background knowledge regarding the geology of the processes in CO₂ sequestration (§3.1), as well as important knowledge on the physicochemical properties of CO₂ (§3.2). We then provide an overview of the mathematical implications of studying WRI, as well as the reactive transport of mass in geologic media (§3.3).

3.1 The geology of the processes in sequestering carbon dioxide

In this section, we intend to present a geological background to assist in understanding where are the processes of CO₂ sequestration performed. This work is intended to be addressed to an interdisciplinary pool of researchers; therefore, we will invest some effort in presenting basic terminology in each considered field, thus maximizing the interdisciplinary scope of this research. Figure 2 shows a summary of the major compositional divisions of planet Earth, as a function of depth. If we assume that these compositional changes occur uniformly throughout the entire planet, then we can average these compositional layers, therefore attaining a general perspective of the geochemistry of planet Earth as a function of depth. Specifically, in Figure 2, we see (on the left diagram), that the Earth can be divided into the crust, the mantle, and the core, each of which possess different geochemical composition. The diagram on the right of Figure 2 presents a detailed perspective of the near surface region, in where the rheologic division that indicates the relative rigidity of the layers is also shown. We can see that the oceanic crust averages 7 km in thickness, whereas the continental crust averages around 36 km to 40 km in thickness.

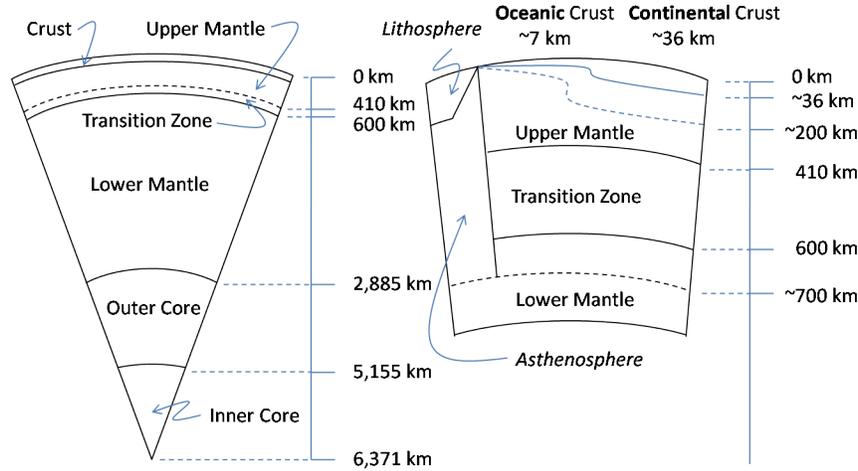


Figure 2: Summary of the major compositional divisions of planet Earth, as a function of depth. See §3.1. Source: Adapted from information given in [Walther, 2009].

Given the nature of the pilot test cases we have considered in order to validate our numerical results, in this work, we will restrain ourselves to sequestration scenarios in the continental crust (see §5). The continental crust can be divided into upper and lower continental crust. If we assume the average thickness of the crust to be of 40 km, then the upper crust would take 20.9%, or 53 km in thickness [Walther, 2009]. In this work, we are interested in CO₂ geologic sequestration scenarios that take place in the percentage of the uppercrust which is composed mostly of sedimentary rocks, i.e., the upper 14% of the upper crust, or the first 5.6 km of depth. We will refer to this region as the sedimentary region of the Earth’s crust or as the subsurface. Sedimentary rocks are rocks that are formed as a consequence of the deposition of material at the Earth’s surface.

In the sedimentary region, the composition is based on shales and siltstones (44.0%), as well as sandstones, greywackes (both accounting for 20.9% of the composition), and mafic volcanic rocks (20.3%). Shales are fine-grained sedimentary rocks composed of a mix of flakes of clay minerals, and tiny fragments of other minerals, as for example, quartz and calcite. Siltstones are sedimentary rocks which have a grain size finer than sandstone, which are also a type of sedimentary rock composed mainly of sand-sized minerals. Volcanic rocks are rocks formed from magma erupted from a volcano. These are termed mafic, since these are rich in both magnesium and iron. Greywackes are a variety of sandstone, generally characterized by its hardness and dark color. These are composed mainly by quartz and feldspar. Carbonates are also present in the sedimentary region, except that in lower concentrations (14.5%), as well as evaporites, which account for 0.1% of the composition [Walther, 2009]. Evaporites are water-soluble mineral sediments, that result from concentration and crystallization by evaporation from aqueous solutions.

The importance of the knowledge of the composition of the sedimentary region of the crust, lies in the role that sedimentary rocks play in the processes of CO₂ geologic sequestration. Effective CO₂ sequestration is achieved by the overlying caprock, which prevents CO₂ migration into up-hole intervals, into shallow freshwater, and ultimately to the atmosphere [Bennion and Bachu, 2007]. The confining properties of the caprock are due to its very low permeability, and to capillary pressure effects that prevent any significant flow of CO₂ through it [Bennion and Bachu, 2007].

3.2 The physicochemical properties of carbon dioxide

Carbon Dioxide was first identified around the middle of the 18th century by Joseph Black, during his studies at the University of Edinburgh [Marini, 2006]. CO_{2(ga)} was originally called “fixed air” because it was fixed in solid form by magnesia and quicklime. Therefore, it can be said that Black was the first one who realized experiments of CO₂ production and sequestration.

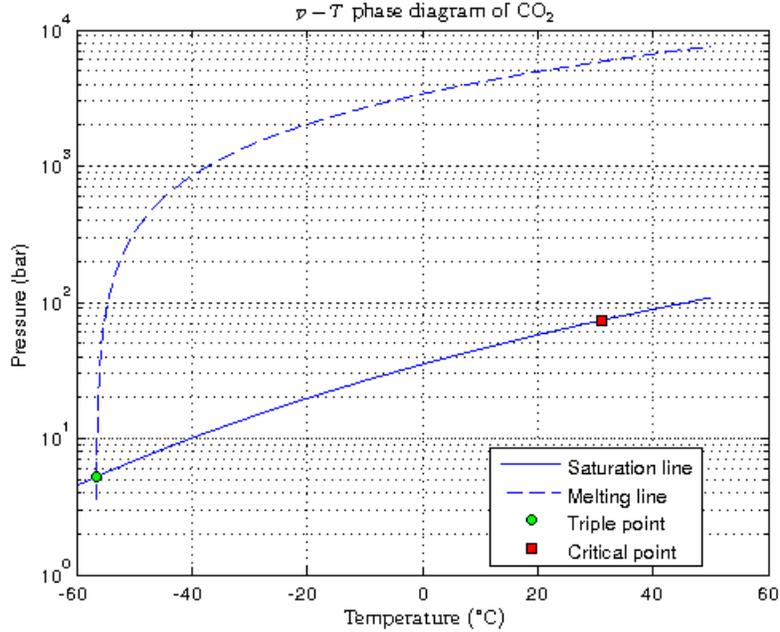


Figure 3: $p - T$ diagram for CO_2 . See §3.2. Source: Created from models given in [Marini, 2006].

The physicochemical properties of any species, can be summarized by the pressure - temperature diagram, or $p - T$ diagram, which describes the variation of pressure as a function of temperature. Figure 3 presents the $p - T$ diagram of CO_2 , created under the assumption that [Marini, 2006]:

$$\log(p_{\text{sat}}) = \frac{-863.6}{T_{\text{sat}}} + 4.705, \quad (1)$$

as well as that [Marini, 2006]:

$$p_{\text{melt}} = 523.18 - 51.547T_{\text{melt}} + 0.22695T_{\text{melt}}^2, \quad (2)$$

where both temperatures are assumed to be given in Celsius, and the resulting pressures are assumed to be given in bars. One important characteristic of Figure 3 is the triple point, which can be taken to be located at $(-56.57 \pm 0.03 \text{ }^\circ\text{C}, 5.185 \pm 0.005 \text{ bar})$ [Marini, 2006]. In this point, all of the three macroscopic states of CO_2 , i.e., solid, liquid, and gaseous coexist. Another aspect depicted in Figure 3, which is very important for CO_2 sequestration studies is the critical point, which can be taken to be located at $(31.03 \pm 0.04 \text{ }^\circ\text{C}, 73.80 \pm 0.15 \text{ bar})$ [Marini, 2006]. Passing this point, the CO_2 is said to be in a supercritical state, i.e., it can be thought as a gas that cannot be liquefied regardless of the exerted pressure.

3.3 Mathematical modeling of WRI and reactive transport in geologic media

In this section, we provide the fundamentals of the mathematical modeling of the problem of interest. We present the Partial Differential Equation (PDE) we study, and we also discuss some important aspects of the geochemistry of the processes hereby analyzed. This section summarizes the mathematics introduced in both [Park, 2009] and [Paolini et al., 2011]. Similarly, we introduce the chosen discretization method that was selected in order to attained the reference results. We explain how the reference discretization was performed, and what are the most important implications of such selection.

In the simulator under study, the interaction of prime interest is that of the water that already exists in the reservoir (formation water), and the solutes-charged water that is being injected (injected water). The goal of the simulations is to study the occurring reactions between solutes in the water and the minerals defining the lithology of the reservoir. The core computation is that of the concentration of such solutes. Mass transfer in a porous media flow, of a given known porosity ϕ , accounting for contributions of both diffusive and advective nature, as well as for the contributions from the reactive terms is given by [Paolini et al., 2011]:

$$\underbrace{\frac{\partial}{\partial t} e_\beta}_{\text{Time rate of change}} = \underbrace{\phi \Omega \nabla \cdot (D_\alpha \nabla c_\alpha)}_{\text{Diffusive component}} - \underbrace{\phi \Omega \nabla \cdot (\mathbf{u} c_\alpha)}_{\text{Advective component}} - \underbrace{\sum_{\gamma=1}^M \nu_{\beta\gamma} \rho_\gamma A_\gamma G_\gamma}_{\text{Reactive component}}, \quad (3)$$

where the following notation holds:

1. e_β : Total mass of element β in the reservoir. Units of [g].
2. Ω : Operator for the computation of the elemental mass from the set of solutes [Park, 2009].
3. D_α : Coefficient of diffusivity for the α -th solute. Units of [cm²/s].
4. c_α : Molar concentration of solute species α . Units of [mol/L].
5. \mathbf{u} : Velocity of the injected water. Units of [cm/s].
6. M : Number of mineral species in the formation.
7. $\nu_{\beta\gamma}$: The number of atoms of element β in the γ -th mineral species.
8. ρ_γ : Density of the γ -th mineral species. Units of [g/cm³].
9. A_γ : Surface area of the γ -th mineral species. Units of [cm²].
10. G_γ : Reaction rate for the γ -th mineral species.

The nature of the presented numerical results has its foundation on the computation of the diffusivity coefficients. In the software we will study, (see Figure 4), this value is approximated using a linear function of reservoir temperature:

$$D_\alpha = 10^{-6}(T_{c,\alpha} + T_{f,\alpha}T), \quad (4)$$

where the values $T_{c,\alpha}$ and $T_{f,\alpha}$ for the α -th solute are discussed in [Paolini et al., 2011] and in [Boudreau, 1996]. From these works, we learn that the diffusivity coefficient of H⁺ is an order of magnitude greater than the diffusivity of metal ions such as Fe⁺⁺ and Mg⁺⁺.

Following the work in [Paolini et al., 2011, Park, 2009], we modeled the reservoir as a 1D horizontally oriented 100 m long sandstone lithology comprised of six minerals with volume fractions and grain sizes given in [Paolini et al., 2011, Park, 2009]. The elemental mass of each solute was solved using Equation (3) and the concentration of each solute was consequently computed with respect to time and space per each control volume in the discretized domain.

The mass-conservation equation (3) is discretized in space and time as:

$$\frac{1}{\phi} \frac{e_\beta(t + \Delta t) - e_\beta(t)}{\Delta t} = D_\alpha \left(\frac{c_{\alpha,i+1} - 2c_{\alpha,i} - c_{\alpha,i-1}}{\Delta x^2} - u_x \frac{c_{\alpha,i} - c_{\alpha,i-1}}{\Delta x} \right) - \sum_{\gamma=1}^M \nu_{\beta\gamma} \rho_\gamma A_\gamma G_\gamma, \quad (5)$$

for which the solution can be obtained through an efficient matrix reduction routine, such as the LU factorization method [Park, 2009].

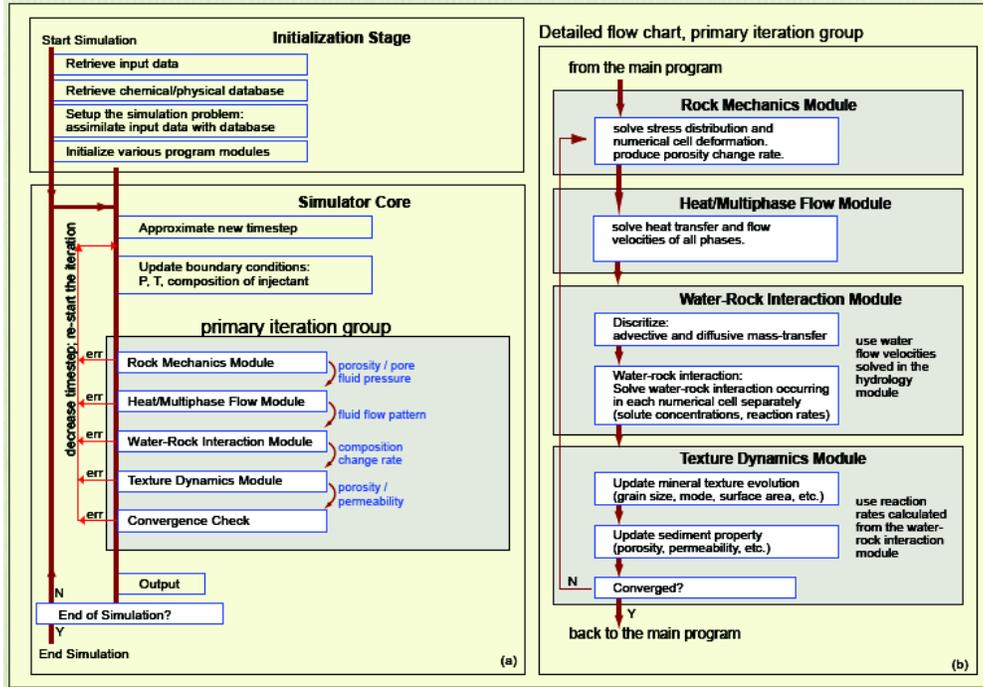


Figure 4: Schematics of the algorithmics of *WebSym.C* present at the numerical core called *Sym.8*. See §3.3. Source: [Park, 2009].

4 The algorithmics of simulating the long-term evolution of the sequestered carbon dioxide

This work, was conducted over the algorithmic framework (*Sym.8*) of *WebSym.C*, a water-rock interaction and reactive mass transport simulator [Park, 2009, Paolini et al., 2011], built with the intention of simulating the short- and long-term chemical, structural, and seismic consequences of the injected $\text{CO}_{2(\text{sc})}$ in deep geologic water-rock systems. This simulator uses elemental mass-balance (Equation 3), explicit mass-transfer with reaction coupling methods, multi-phase and heat flow, support for both $\text{CO}_{2(\text{sc})}$ and oil, fracture mechanics, anisotropic permeabilities, rheological rock mechanics based on incremental stress theory, and a composite petrophysics model capable of describing changing rock composition and properties [Park, 2009, Paolini et al., 2011]. The modules representing these processes are solved using a layered iteration method, with the goal of capturing the nonlinear feedback among all of the processes. Figure 4, shows an schematic diagram representing the algorithmics of *WebSym.C*.

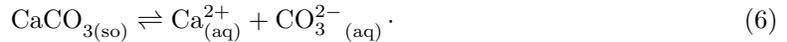
In Figure 4 we see, on the left side, that two main algorithmic stages are defined: first, the initialization stage, and second, the simulator core. On the later, output is given per time step, as well as updating of the boundary conditions occur. Within this second stage lies the primary iteration group, in which the required computations for the hydrology of the problem, as well as for the water-rock interaction and texture dynamics, are solved in different modules. These modules are iteratively solved until consistent solutions to all of the involved variables are achieved [Park, 2009].

On the right side, we present a more detailed description of the primary iteration group. Specifically, we see that the hydrology stage focuses on solving the gas and water flow velocity within the specified simulation domain. After, the discretization of the advective and diffusive mass-transfer components occurs, and the concentration profiles are computed. This module is the main concern of this work. It is important to notice that the algorithmics specify that the solution of the water-rock interaction occurs in each numerical grid separately. Finally, the mineral textures and the properties of all the sediments are computed, and convergence checking is performed.

5 Reference pilot test case: gas-water-rock interaction in the Frio Formation

In this work, we will consider as a test case the one which was the first experiment conducted in the United States in which CO_2 was sequestered. This experiment took place on September of 2004; 1,600 t of CO_2 were injected into a mile-deep well at the Frio Brine Pilot experimental location, located 30 miles northeast of Houston, in the South Liberty Oilfield [Paolini et al., 2011]. The injection well at the Frio formation is 5,753 feet deep and the anticipated injection zone ranged from 5,033 to 5,073 feet and consists of a brine-sandstone system with a top seal of 200 feet of Anahuac shale. The injection began on September 4, 2004, and ran for several days.

The significant findings of the post injection analysis were that injected CO_2 caused the brine at the injection depth to become acidic [Kharaka et al., 2006]. Specifically, acidic brine will dissolve some of the rock and other minerals the brine comes into contact with, adding iron and other metals to the salt water. Specifically, the increased acidity caused the dissolution of carbonate rock:



The latter reaction will play an important role in the subsequent study. Furthermore, as it is stated in [Paolini et al., 2011], this particular injection process can be conceptualized as the downstream arrival of different fluidic fronts: downstream of the injection well, the formation first experiences the arrival of an acidic front, followed by the arrival of a bicarbonate front.

In this work, we will first make sure that, in our attempt to improve the sequential version of *WebSym.C*, we obtain these reference results. We will then inquire regarding further potential for improvement by establishing a simplified test scenario based on Equation (6).

6 Results for the sequential implementation

In this section, we summarize the results of the sequential executions of the simulator. We begin by presenting a quick summary of the most important aspects of the considered hardware platforms that were considered for this study (§6.1). We then present a summary of the reference physical solutions, which will be considered when studying the attained computational performance throughout this work (§6.2). We then present a profiling study of the sequential version of the simulator to establish the fact that most of their computational burden is focused on the resolution of the conservation of mass equation for computing the concentration of all the solutes (§6.3). Finally, we will present the results summarizing the advantages of considering different solvers, in order to achieve the best possible sequential instance of the simulator (§6.4).

6.1 Considered hardware platforms and *CSRCnet*

In this work, the tests were performed on two hardware platforms. The first platform is a relatively small Linux cluster called `blackbox.sdsu.edu`. This cluster is a local resource in *San Diego State University (SDSU)*, which was chosen given its high computational potential per node. Summarized architectural specifications for `blackbox.sdsu.edu` are shown in Table 2. Similarly, since we are interested in studying the achieved performance in highly distributed environments, we have also performed numerical tests in `trestles.sdsc.edu`, which is a well-known *XSEDE* resource, located at the *San Diego Supercomputer Center (SDSC)* [SDSC, 2012]. Summarized architectural specifications for `trestles.sdsc.edu` can also be found in Table 2, and in full detail at [SDSC, 2012].

Communication of physical and performance results between local resources at SDSU and the SDSC was achieved by means of *CSRCnet*. CSRCnet is a specialized, high-speed research network that provides researchers at the *Computational Science Research Center (CSRC)* the ability to transfer data between the two campuses at 10 Gbps. In this work, physical results, profiling results, including text and graphics, were communicated through CSRCnet. Remote source code editing and systems/APIs building and configurations were also achieved by means of CSRCnet.

| Resource | <code>blackbox.sdsu.edu</code> | <code>trestles.sdsc.edu</code> |
|--------------------------------|--------------------------------|--------------------------------|
| Processor model name | Intel® Xeon® Processor E5420 | AMD® Magny-Cours |
| Compute cores per compute node | 8 | 32 |
| Clock frequency | 2493.775 MHz | 2400.043 MHz |
| Cache size | 6144 kB | 512 kB |
| Sockets | 2 | 4 |
| Stepping | 6 | 1 |
| Memory capacity | 32 GB | 64 GB |
| Total compute nodes | 8 | 324 |
| Total compute cores | 64 | 10368 |
| Total memory | 0.24 TB | 20.7 TB |
| Operating system | Red Hat Ent. Server 5.8 | CentOS 5.5 (Final) |
| Kernel release | 2.6.18-274.17.1.el5 | 2.6.18-194.17.4.el5 |

Table 2: Comparison of considered hardware platforms in terms of performance characteristics. See §6.1.

6.2 Attained physical results

In this section, we present the attained physical results from the performed simulations for the reference pilot test case, which is explained in detail in [Kharaka et al., 2006, Paolini et al., 2011] and in [Park, 2009]. A more detailed description of the chemical system can be found in [Park, 2009] while Figure 5 shows the reference solution, which depicts the advection fronts at 5 years after injection, as computed in both `blackbox.sdsu.edu` and `trestles.sdsc.edu`. The reference implementation of the LU factorization is that described and implemented in [Press et al., 1988]. In this example, for different grid resolutions, we computed the concentration of $\text{CO}_{2(\text{li})}$, H^+ , and Fe^{++} , as a function of distance from the injection well. These results are consistent with those presented in [Paolini et al., 2011]; therefore, they can be used as a reference set of solutions.

6.3 A profile analysis of the simulation software

In this section, we present a profile study of the original sequential simulator *WebSym.C*. The purpose of this analysis is to locate the computational tasks which account for the highest execution time within *WebSym.C*. Table 3 presents the top 10 highest percentages of invested computation time (in seconds) per routine in the original version of *WebSym.C*, as computed by *GNU gprof* [Fenlason, 1993] in `blackbox.sdsu.edu`. For this profiling study, the average of 5 instances of 100 cells each of the pilot test case described in §5 were considered.

The results are also depicted in Figure 6. As it can be seen, the main sink of computational time is the LU decomposition routine, which is defined and explained in [Press et al., 1988]. Specifically, up to 32.68% of the time spent per simulation is invested in LU factorization. Furthermore, in general, the top 10 percentages shown in Table 3 are related to the resolution of the conservation of mass equation, for the computation of the solute concentrations.

6.4 Improving the sequential solvers

In the previous section (§6.3), we established the fact that most of the computational burden, in terms of execution time, is focused on solving for the conservation of mass equation, in order to compute the concentration of the injected solutes, in any given sequestration scenario. It is well known, that for any parallel implementation to be properly studied, the fastest known sequential version of such implementation has to be considered [Pacheco, 1997]. Therefore, in this section, we present the attained computational performance when trying to substitute the sequential reference solver [Press et al., 1988], which is intended for pedagogical purposes, with solvers that are indeed oriented to achieve the highest computational performance possible.

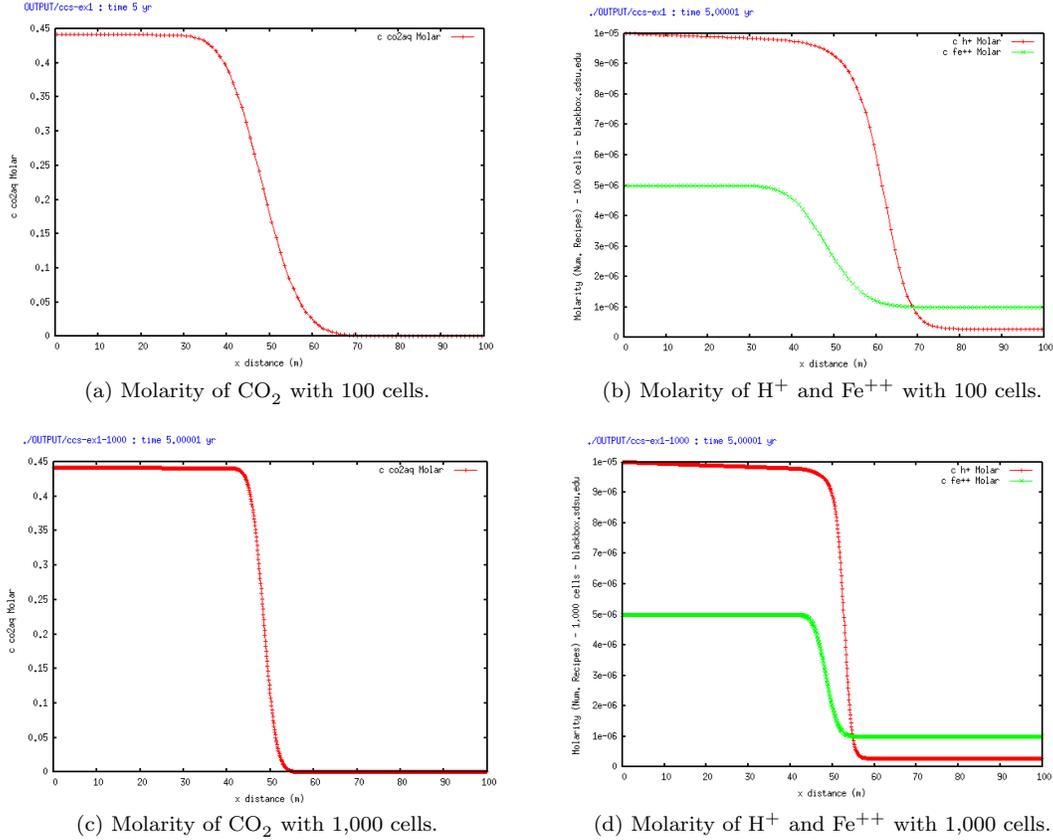


Figure 5: Reference solution at 5 years after injection, computed in `blackbox.sdsu.edu`. We considered the reference implementation of the LU factorization, which is provided in [Press et al., 1988]. For different grid resolutions, we computed the concentration of CO_2 , H^+ , and Fe^{++} , as a function of distance from the injection well. Reference results are the same for `trestles.sdsc.edu`.

| Invoked routine | Percentage of time | Cumulative time | Time per call |
|--|--------------------|-----------------|---------------|
| <code>ludcmp</code> | 32.68 | 621.43 | 621.43 |
| <code>rxn_csolver_std</code> | 32.19 | 1233.56 | 612.13 |
| <code>ionic_strength_correction</code> | 12.97 | 1480.27 | 246.71 |
| <code>rxn_saturation</code> | 4.23 | 1560.65 | 80.38 |
| <code>lubksb</code> | 3.65 | 1630 | 69.35 |
| <code>fdmx_discretize_diffusive</code> | 2.01 | 1668.17 | 38.17 |
| <code>sediment_moles_chem</code> | 1.78 | 1702.11 | 33.94 |
| <code>rxn_rate_driver</code> | 1.65 | 1733.46 | 31.35 |
| <code>rxn_1DX</code> | 1.21 | 1756.52 | 23.06 |
| <code>set_terms_mass_transfer</code> | 1.07 | 1776.79 | 20.27 |

Table 3: Top 10 percentages of invested computation time (in seconds) per routine in *WebSym.C*, as computed by *GNU gprof* [Fenlason, 1993] in `blackbox.sdsu.edu`. See §6.3 and Figure 6. The average of 5 instances of 100 cells each of the pilot test case described in §5 was considered for this profiling study.

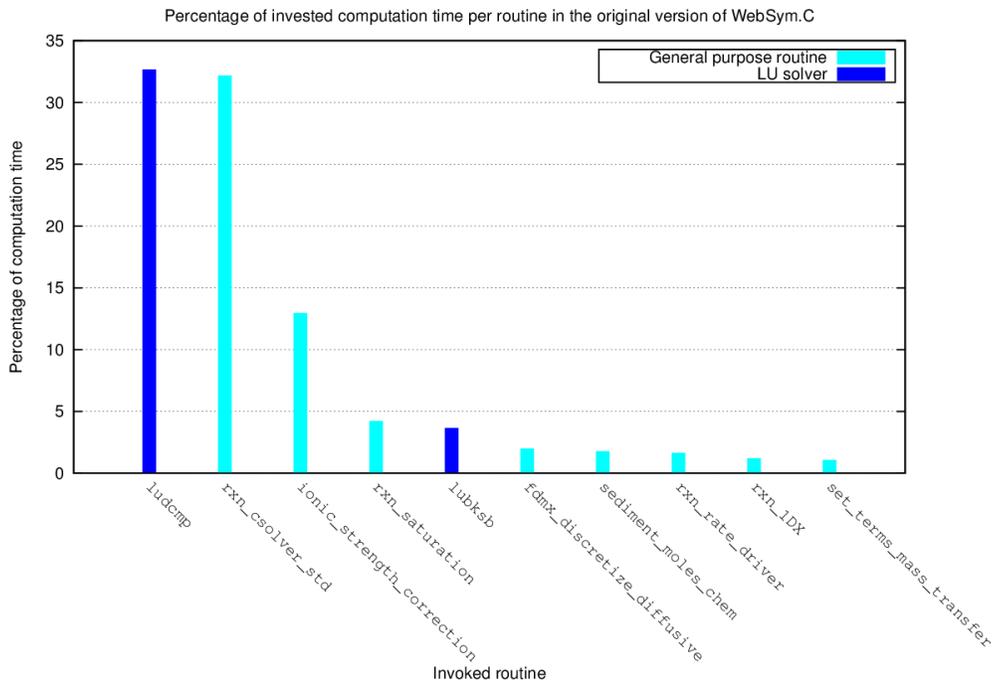


Figure 6: Highest 10 percentages of invested computation time (in seconds) per routine in the original version of *WebSym.C*, as computed by *GNU gprof* in `blackbox.sdsu.edu`. See §6.3 and Table 3. The average of 5 instances of 100 cells each of the pilot test case described in §5 was considered for this profiling study.

The first solver we consider is the *Linear Algebra PACKage (LAPACK)* [Anderson et al., 1999]; specifically, we consider the routines for the solution of banded systems of equations. The second solver we will consider in this work, is *SuperLU-SEQ* [Demmel et al., 1999, Li, 2005]. One of the most important differences between each solver, besides the algorithmic approach they consider in order to solve the problem of implementing the solution of a linear system of equations through a LU factorization, is the necessity of different representations for the matrices. Different data structures that are required for these representations had to be properly implemented. In this work, we have used the *Mimetic Methods Toolkit (MTK)* [Castillo and Miranda, 2013, Sanchez, 2012, Sanchez et al., 2012].

Table 4 presents the runtimes from replacing the reference solver, described in [Press et al., 1988], with the previously discussed high-performance sequential solvers. The impact of the features between the selected architectures can be depicted. Since this is a sequential code, results are tighten to the imposed constraints for execution on a single processor on both systems. For example, executions in `blackbox.sdsu.edu` are faster than those in `trestles.sdsc.edu`, because `blackbox.sdsu.edu` has better processors (higher stepping number). Another example is that in `trestles.sdsc.edu`, the queue manager runtime constraint did not allow for the completion of greater instances of grid refinement (1,000 and 10,000 cells). Specifically, `trestles.sdsc.edu` only allows a maximum of 18 hours worth of wall time per compute core. The solvers behaved as expected except for *SuperLU-SEQ*, which required some extra processing time given the conversion to the Compressed Column Storage (CCS) storage format, required to be compatible with *WebSym.C*. The reason for this overhead lies on the algorithmic implications when adding elements to a matrix represented in the CCS format. An explanation on the CCS format can be found in [Li et al., 1999].

| Resource | blackbox.sdsu.edu | | | trestles.sdsc.edu | | |
|-------------------|-------------------|---------|----------|-------------------|----------|--------|
| Number of cells | 100 | 1,000 | 10,000 | 100 | 1,000 | 10,000 |
| Numerical Recipes | 0:33:18 | 2:48:31 | 35:49:11 | 1:33:17 | 10:00:33 | - |
| LAPACK | 0:30:46 | 2:14:17 | 16:42:03 | 1:29:19 | 8:59:17 | - |
| SuperLU_SEQ | 0:44:34 | 3:11:51 | 34:39:21 | 1:55:36 | - | - |

Table 4: Attained execution times (in minutes) from replacing the reference sequential solver with those discussed in §6.4. The averages of 5 executions were taken per each case.

7 A Block-defined, Global and Sparse (BloGS) matrix storage scheme for the numerical resolution of many partial differential equations on distributed-memory computers

In this section, we introduce the mathematics of a blocks-defined, global and sparse (BloGS) matrix storage scheme, for the parallel computation of the concentration of all the involved solutes, in a given WRI and reactive mass transport scenario. This work was originally proposed in []

Let N_a be the number of solutes for which we are interested in computing their concentrations. Consider a discretized one-dimensional domain $\Omega = [a, b]$, which is discretized using any ω -th order of accuracy (ω even) discretization method, resulting in a uniform grid with n_x nodes. As we learned from the algorithmic layout of *WebSym.C* (see §4), the concentration for each of the N_a solutes is computed per each node in the grid, per each time step, which implies the solution of a small system to be performed many times. This is not suited for an execution on distributed-memory computer clusters, because the small rank of the systems prevents distributed algorithms from scaling, given the overhead introduced by inter-processes communication.

Based on this, we propose to arrange the coefficients that arise from the discretized form of the conservation of mass equation, into a block-defined matrix for the global solution of all the solutes, (which is sparse, thus being referred to as a BloGS matrix), and which for a given even order of accuracy ω , is denoted as $\mathbf{B}(\omega)$. The general form the BloGS is given in Appendix A's Equation (37).

An example BloGS matrix for $\omega = 2$, that is a second order accurate discretization method, looks like:

$$\mathbf{B}(2) = \begin{bmatrix} \mathbf{W}_{1,1} & \mathbf{W}_{1,2} & \mathbf{W}_{1,3} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \mathbf{B}_{2,3} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{3,2} & \mathbf{B}_{3,3} & \mathbf{B}_{3,4} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & & \ddots & \ddots & & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}_{n_x-2, n_x-3} & \mathbf{B}_{n_x-2, n_x-2} & \mathbf{B}_{n_x-2, n_x-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{B}_{n_x-1, n_x-2} & \mathbf{B}_{n_x-1, n_x-1} & \mathbf{B}_{n_x-1, n_x} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{E}_{n_x, n_x-2} & \mathbf{E}_{n_x, n_x-1} & \mathbf{E}_{n_x, n_x} \end{bmatrix}, \quad (7)$$

for which, if we are interested in solving for 2 solutes (for example), then $N_a = 2$, and each block will have dimensions 2×2 . An example for $\omega = 4$ is given in Appendix A.

We selected the Finite Difference Discretization Method (FDM) because that is the selected discretization method in *WebSym.C*, as it is explained in §3.3. However, it is noteworthy to state that the BloGS scheme can be applied with different discretization methods. In fact, in §9, we will mention the future study of this scheme, with Mimetic Discretization Methods [Castillo and Grone, 2003, Castillo and Miranda, 2013]. In this work we are mostly interested in the performance aspects of solving BloGS systems using High-Performance distributed clusters through proper APIs, therefore, we will restrict our result for a second order implementation ($\omega = 2$), which is already an improvement to *WebSym.C*, since originally, it implements a first order upwind scheme for the advective component of the conservation of mass equation. However, we will describe the properties of such matrices, for general ω , N_a , and n_x .

Behavior of the rank as a function of both
the number of solutes and the number of nodes

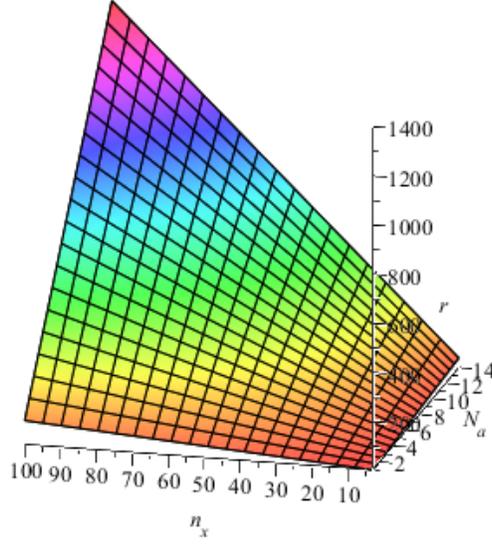


Figure 7: Behavior of the rank of the BloGS matrices, as a function of the number of solutes N_a and the number of nodes, n_x . See §7. Coloring is simply a result of the plotting software used; it means nothing special except that it varies proportionally to the quantity of being plotted.

The first important property we will describe about the BloGS matrices is their rank, r , as a function of both the number of solutes N_a and the number of nodes n_x :

$$r(N_a, n_x) = N_a n_x, \quad (8)$$

where, if we let $\Omega = [a, b]$ denote our one-dimensional domain, discretized with Δx as the step size, then:

$$n_x = \left\lceil \frac{b - a}{\Delta x} \right\rceil. \quad (9)$$

Figures 7, 9a, and 9b depict this relationship. Given current restrictions within the memory management within *WebSym.C*, the number of solutes we can solve for is bounded by 30. i.e., an static array is declared for storing only 30 solutes. However, the number of nodes, n_x , is a consequence of the chosen grid step size; therefore, BloGS matrices can get very large, thus making them suitable for the use of distributed solvers. A well known restriction for the number of nodes, based on the selected order of accuracy is $n_x \geq \hat{n}_x$, where $\hat{n}_x = \omega + 1$.

The second important property to analyze is the bandwidth of the attained matrices. This property has proven to be vital in terms of achieving scalability of the distributed solution of banded systems [Cleary and Dongarra, 1997], as it will be discussed in §7.3. The bandwidth β of these matrices is the sum of the number of diagonals, which is a consequence of the number of solutes determining the dimensions of each block, N_a , and the required order of accuracy, ω . Let kl and ku denote the number of lower and upper diagonals, respectively, then:

$$\beta = kl + 1 + ku. \quad (10)$$

When using general Robin's boundary conditions (given rate of change at the boundary), then $kl = ku$ —as long as we do not assume Dirichlet boundary conditions—thus:

$$kl = ku = N_a \omega, \quad (11)$$

therefore:

$$\beta = 2N_a\omega + 1. \quad (12)$$

Figure 8 depicts the two properties which depend on the chosen order of accuracy, thus informing about the distribution of the elements within the BloGS matrices. Specifically, Figure 8a shows the behavior of the bandwidth, and Figures 9c and 9d, show the related projections.

The final property we will discuss is the density of the attained matrices. For this, we must first compute the number of non-null elements, η . It is important to mention that the term “non-null” is used instead of the term “non-zero”, since some of the elements can actually be zero, but still lie within the scope of the bandwidth, β . An example of this can be depicted when using Dirichlet boundary conditions, for which zero appears as a placeholder for the stencil values that would not be zero if other type of boundary condition were considered instead of Dirichlet’s. Another example can be depicted with $\omega \geq 4$, where diagonals filled with zero values arise, (thus increasing the sparsity of the band) but are still part of the band. In order for us to compute η , we will consider the number of upper and lower diagonals. We must first define a number that imposes an ordering scheme within the diagonals. We will call these numbers the lower- and upper- diagonals indices, kli and kui , respectively. By convention, both kli and kui will equal zero for the main diagonal, which is known to possess r elements. From here, we will sum the terms in each diagonal, subtracting one element as the indices increase:

$$\eta = \sum_{kli=1}^{kl} (r - kli) + \sum_{klu=1}^{ku} (r - kui). \quad (13)$$

We can define the actual number of non-zero values, z , by realizing that each row r_i , $0 \leq i \leq r$, possesses the information for an approximation of ω -th order of accuracy, therefore, if we assume no Dirichlet boundary conditions:

$$z = r(\omega + 1). \quad (14)$$

Based on these two values, η and z , we can compute two different values to help describe the density of the matrix. The first value is d_z , which is the absolute density of the matrix (see Figure 8b):

$$d_z = \frac{z}{r^2}, \quad (15)$$

which implies the following definition for the absolute sparsity of the matrix:

$$\sigma_z = 1 - d_z = 1 - \frac{z}{r^2}. \quad (16)$$

Finally, we can define the density of the matrix in relation to the density of its band, or relative density, as follows:

$$d_\eta = \frac{\eta}{r^2}, \quad (17)$$

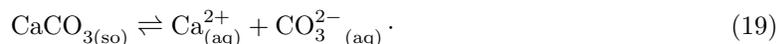
which implies the following definition for the relative sparsity of the matrix:

$$\sigma_\eta = 1 - d_\eta = 1 - \frac{\eta}{r^2}. \quad (18)$$

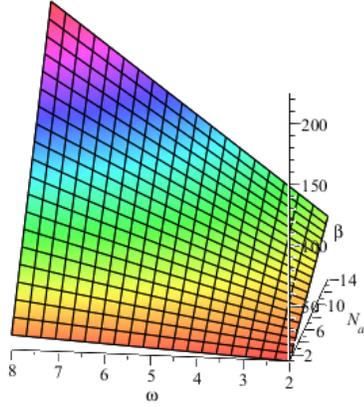
Figures 9e and 9f both show the projections of the behavior of the absolute density, which is also important in terms of achieving any scalability in execution time. Specifically, the size of the matrices, their bandwidth and their absolute sparsity, as previously defined, will determine the nature of the selected high-performance distributed solver, as it will be discussed in §7.3.

7.1 A simplified prototype test case: a calcite dissolution reaction

In order for us to explore the feasibility of reaching the solution of each one of the solute concentrations accurately, by means of the proposed scheme explained in §7, we will consider an example given in [Park, 2009]. The proposed example shows a calcite-water interaction consisting of only one kinetic (calcite dissolution) reaction:

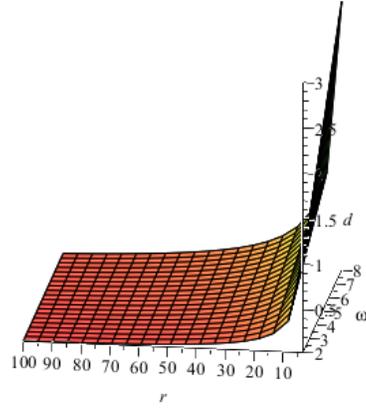


Behavior of the bandwidth as a function of both the number of solutes and the order of accuracy



(a) Bandwith of the BloGS matrices.

Behavior of the density as a function of both the order of accuracy and the rank



(b) Absolute density of the BloGS matrices.

Figure 8: Bandwith and (absolute) density of the BloGS matrices, which are properties dependant on the chosen order of accuracy, ω . See §7. Coloring is simply a result of the plotting software used; it means nothing special except that it varies proportionally to the quantity of being plotted.

Equation 3 can be seen as a generalized diffusion-advection-reaction equation, which accounts for all of the important properties for reactive mass transport in porous media, and which shows a coupling of the terms based on the stoichiometric coefficients and the reactive terms. For the purpose of validating our previously developed scheme we will neglect these physical implications, we will concentrate on the effect of the chosen discretization method and the distribution of the related coefficients in a BloGS matrix.

We will select boundary conditions which are general enough to be proven useful in this context, but which will yield an actual analytic solution, thus allowing us to study the attained accuracy when solving for solute concentration using the BloGS scheme. Based on this, letting c_1 and c_2 be the concentrations of interest, we can define the following problem for each one of the concentrations of interest:

$$\frac{\partial c_i}{\partial x} = \frac{\partial^2 c_i}{\partial x^2} - 1, \quad (20)$$

for $i \in \{1, 2\}$ and $x \in [0, 1]$. Subject to:

$$c_i(0) = 1 \quad (21)$$

$$c_i(1) - c'_i(1) = 0. \quad (22)$$

Based on this, we can then state the following analytical solution to the problem of interest to be:

$$c_i(x) = e^x - x. \quad (23)$$

If we assume a second order, centered finite difference method, we will attain the following form for the discretized PDEs, for a given step size Δx :

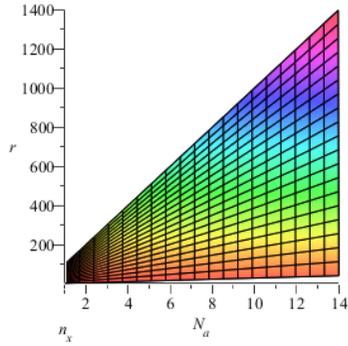
$$\left(\frac{1}{\Delta x^2} + \frac{1}{2\Delta x} \right) c_{i_{j-1}} - \frac{2}{\Delta x^2} c_{i_j} + \left(\frac{1}{\Delta x^2} - \frac{1}{2\Delta x} \right) c_{i_{j+1}} = 1, \quad (24)$$

for which, we will consider the following discrete form for the boundary conditions:

$$c_{i_1} = 1, \quad (25)$$

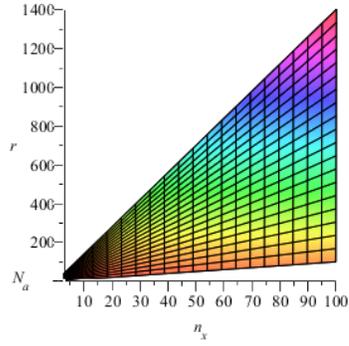
$$-c_{i_{n_x-2}} + 4c_{i_{n_x-1}} + c_{i_{n_x}} = 0, \quad (26)$$

Behavior of the rank as a function of the number of solutes



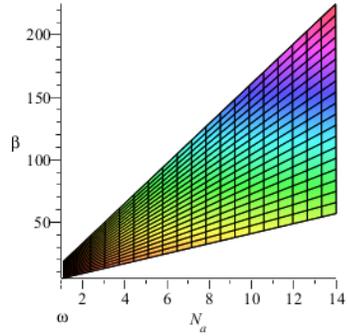
(a) Projection for $r(N_a)$.

Behavior of the rank as a function of the number of nodes



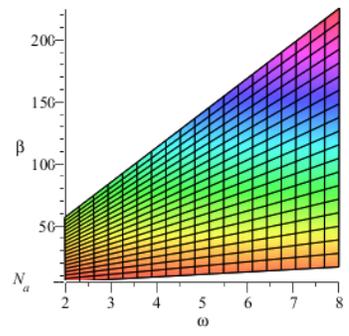
(b) Projection for $r(N_x)$.

Behavior of the bandwidth as a function of the number of solutes



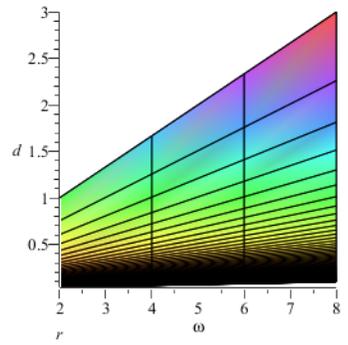
(c) Projection for $\beta(N_a)$.

Behavior of the bandwidth as a function of the order of accuracy



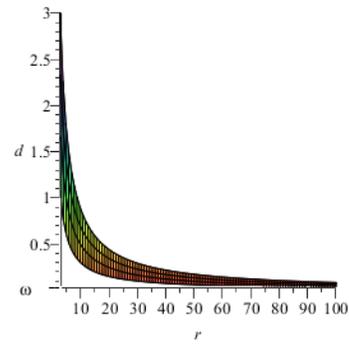
(d) Projection for $\beta(\omega)$.

Behavior of the density as a function of the order of accuracy



(e) Projection for $d(\omega)$.

Behavior of the density as a function of the rank



(f) Projection for $d(r)$.

Figure 9: Projections depicting the behavior of the important properties of a BloGS matrix. See §7. Coloring means nothing.

where n_x , as previously defined, denotes the number of nodes that arises from the discretization based on a step size Δx . Based on this, the entries in Equation (37) take the following forms:

$$\mathbf{W}_{1,1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (27)$$

whereas $\mathbf{W}_{1,2} = \mathbf{W}_{1,3} = \mathbf{0}$. The blocks containing the discretization coefficients for the interior of the grid are defined as follows:

$$\mathbf{B}_{i,j-1} = \begin{bmatrix} \left(\frac{1}{\Delta x^2} + \frac{1}{2\Delta x}\right) & 0 \\ 0 & \left(\frac{1}{\Delta x^2} + \frac{1}{2\Delta x}\right) \end{bmatrix} \quad (28)$$

$$\mathbf{B}_{i,j} = \begin{bmatrix} -\frac{2}{\Delta x} & 0 \\ 0 & -\frac{2}{\Delta x} \end{bmatrix} \quad (29)$$

$$\mathbf{B}_{i,j+1} = \begin{bmatrix} \left(\frac{1}{\Delta x^2} - \frac{1}{2\Delta x}\right) & 0 \\ 0 & \left(\frac{1}{\Delta x^2} - \frac{1}{2\Delta x}\right) \end{bmatrix}, \quad (30)$$

for $i \in [2, n_x - 1]$ and $j \in [2, n_x - 1]$. Finally, the blocks for the discretization of the east boundary, will be defined as follows:

$$\mathbf{E}_{n_x, n_x - 2} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad (31)$$

$$\mathbf{E}_{n_x, n_x - 1} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \quad (32)$$

$$\mathbf{E}_{n_x, n_x} = \begin{bmatrix} 2\Delta x - 3 & 0 \\ 0 & 2\Delta x - 3 \end{bmatrix}. \quad (33)$$

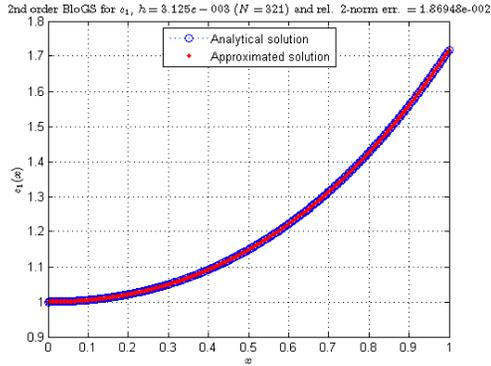
Notice that since we are considering two different concentrations, i.e. $N_a = 2$, then the dimension of each block is $N_a \times N_a = 2 \times 2$. An example of a complete BloGS matrix and its related system of equations, for $n_x = 6$ is given in Appendix A.

7.2 Results of the sequential implementation

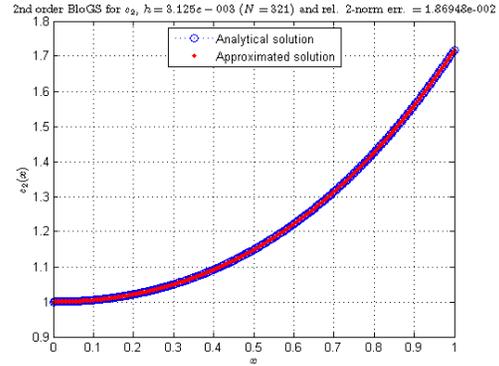
In this section, we study the behavior of the BloGS scheme when sequentially solving for the presented problem. For this study, we implement a prototype driver in MATLAB R2008a, which was useful to study the attained condition number of the BloGS matrix. We also developed two drivers using both LAPACK's banded solvers and SuperLU_SEQ.

The results of the system using the MATLAB prototype are summarized in Figure 10. These results show the feasibility of achieving the solution of the 2 PDEs, under the same system of equations, which properties make it suitable for high-performance distributed solvers. An important result also computed through this MATLAB prototype is the condition number of the matrix for $N_a = 2$, as a function of its rank. Since the condition number of a matrix measures the sensitivity of the solution of a system of linear equations to errors in the data, this result gives an indication of the accuracy of the results from matrix inversion and the linear equation solution, as the matrix increases in size. Figure 11 shows the behavior of this quantity.

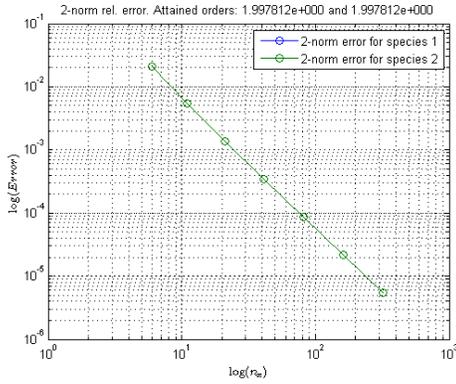
The results achieved using SuperLU_SEQ are depicted in Figure 12. For these results, the MTK API was utilized in order to encode the matrices using the Compressed Column Storage (CCS) sparse matrix format. Similarly, the MTK was also used to provide the required data structures for the manipulation of the banded matrices to be used with LAPACK. Both LAPACK and SuperLU_SEQ do not provide interfaces for these data structures, thus the motivation for the creation of them within the MTK.



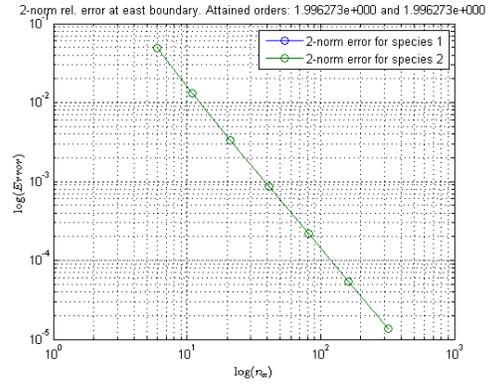
(a) Known and computed solution for species 1.



(b) Known and computed solution for species 2.



(c) Attained order in the interior of the grid.



(d) Attained order in the east boundary of the grid.

Figure 10: Attained results for the MATLAB R2008a prototype driver for the solution of a BloGS system. See §7.2.

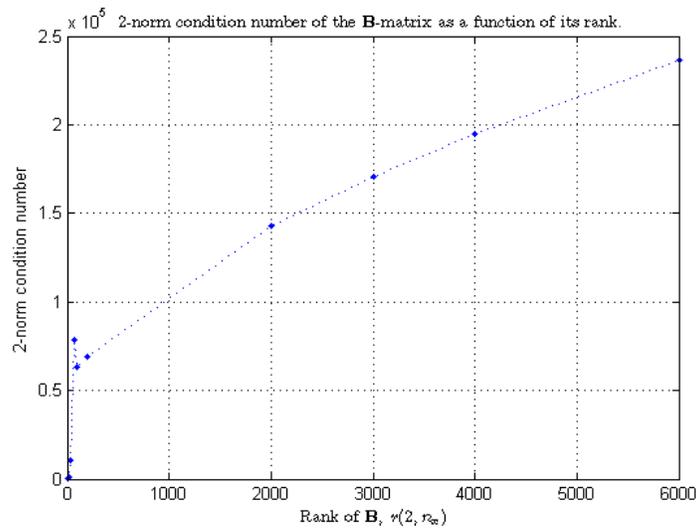
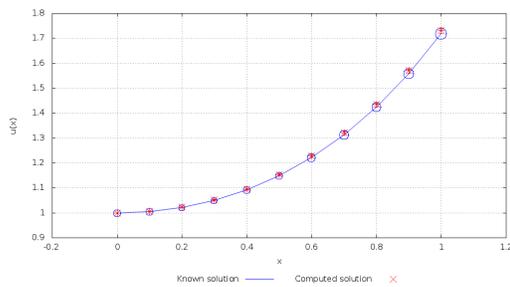
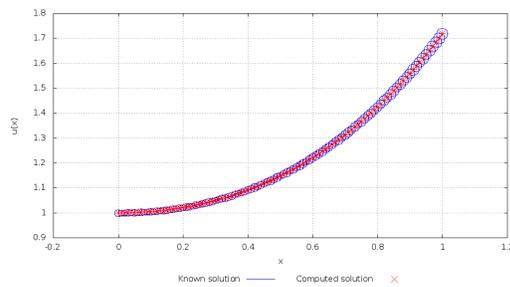


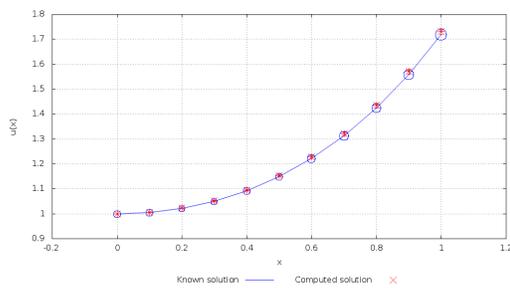
Figure 11: Behavior of the condition number of the BloGS matrices, as a function of the rank $r(n_x)$, which is defined by the number of nodes, n_x . See §7.



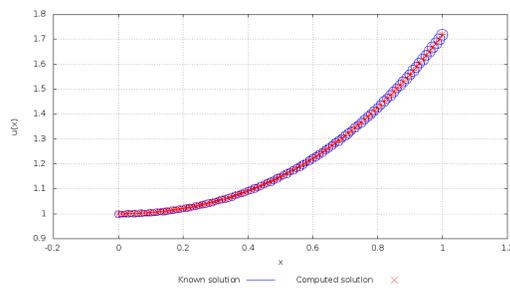
(a) LAPACK solution, $r = 6$.



(b) LAPACK solution, $r = 106$.



(c) SuperLU_SEQ, $r = 6$.



(d) SuperLU_SEQ, $r = 106$.

Figure 12: Analytical and computed solutions for the LAPACK and SuperLU_SEQ prototype drivers for the solution of a BloGS system. See §7.2. Coloring distinguishes the analytical and computed solution. Specifically, dots depict the computed solution whereas the line connecting the hollow circles depict the analytical solution.

| | |
|------------------------------|---------------------------|
| Sequential solver | Distributed counterpart |
| ATLAS' LAPACK SuperLU_SEQ | ScaLAPACK SuperLU_DIST |

Table 5: Comparison of selected solvers to work with. See §7.3.

| (r, β) | \mathbf{p} | | | |
|----------------------|--------------|--------|--------|--------|
| | (1, 1) | (2, 1) | (4, 1) | (8, 1) |
| $(2 \times 10^6, 6)$ | 27.593 | 25.240 | 23.992 | 36.461 |
| $(4 \times 10^6, 6)$ | 53.276 | 50.457 | 56.279 | 73.612 |

Table 6: Execution times (in seconds) for the proposed test case using SuperLU_DIST on `blackbox.sdsu.edu`. See §7.3.

7.3 Results of the distributed implementation

For the sequential version, the first selection was that of utilizing LAPACK. Clearly, the distributed counterpart is the ScaLAPACK [Pacheco, 1997, Choi et al., 1994]. Similarly, for the case of SuperLU_SEQ, its distributed counterpart is SuperLU_DIST [Li and Demmel, 2003]. However, the type of problem these solvers intend to solve becomes of importance for their distributed counterparts. In the sequential case, their different nature impacted the manipulation of memory, i.e., SuperLU_SEQ is intended for the solution of generally sparse matrices, thus yielding the necessity of CCS data structures. With “generally”, we mean those sparse matrices, for which the sparsity pattern follows no general form, as in the case, for example, for banded systems. Analogously, LAPACK is intended for the solution of banded systems. This yields the necessity of data structures for the manipulation of banded matrices.

The intrinsic intended nature for these solvers becomes of importance for their distributed counterparts. In the sequential case, their different nature impacted the manipulation of memory, i.e., SuperLU_SEQ is intended for the solution of generally sparse matrices, thus yielding the necessity of CCS data structures. Analogously, LAPACK is intended for the solution of banded systems, which are a kind of sparse. This yields the necessity of data structures for the manipulation of banded matrices.

However, in the case of their distributed memory counterparts, the necessity of different data structures is not the only concern. We face the problem of achieving scalable speedup. This is, SuperLU_DIST, may fail in scaling for certain instances of the BloGS systems, that do not satisfy the expected properties the solver may assume in order to properly scale. A similar situation may occur for the ScaLAPACK. An example of this is depicted in Tables 6 and 7. Such tables show the execution of two instances of a BloGS-analog matrices, described in terms of parameters r , and β . In such tables, the vector \mathbf{p} denotes the configuration of the process grid for SuperLU_DIST. For ScaLAPACK, given the nature of the algorithms it implements, we have that $\mathbf{p} = p$, that is, one single processor is required.

Based on this, we are interested in understanding when will each particular solver scale. For this, it is clear that the properties of rank and bandwidth become important, since a criteria for the quality of the achieved scalability has to be devised thus allowing for a decision to be made, based on the aforementioned and algebraically known properties. More specifically, the question to

| (r, β) | p | | | |
|----------------------|---------|---------|---------|---------|
| | 1 | 2 | 4 | 8 |
| $(2 \times 10^6, 6)$ | 0.23882 | 0.17029 | 0.13778 | 0.09892 |
| $(4 \times 10^6, 6)$ | 1.18211 | 0.85833 | 0.69183 | 0.49475 |

Table 7: Execution times (in seconds) using ScaLAPACK on `blackbox.sdsu.edu`. See §7.3.

| | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1,024 |
|---------|--------|--------|-------|-------|-------|-------|-------|-------|--------|--------|
| 10,000 | -1.259 | -0.882 | 0.201 | 0.735 | 0.500 | 0.700 | 1.453 | 0.645 | -0.118 | -0.292 |
| 25,000 | -0.954 | -0.098 | 0.415 | 3.479 | 2.892 | 1.262 | 0.201 | 1.135 | 0.316 | -0.259 |
| 50,000 | 0.392 | 1.909 | 2.916 | 5.155 | 4.871 | 3.026 | 0.801 | 0.668 | 0.779 | -0.171 |
| 75,000 | 1.806 | 3.826 | 2.993 | 4.156 | 5.889 | 4.000 | 1.252 | 0.851 | -0.047 | 0.012 |
| 100,000 | 3.822 | 5.679 | 4.235 | 3.497 | 6.520 | 4.486 | 2.071 | 1.533 | 0.525 | - |

Table 8: Attained qualities for low-rank matrices. See §7.3 and Figure 14a.

be asked is, for which neighborhood of the (r, β) -parameter space, is the quality of speedup good enough for each solver? For this, we will introduce the concept of “quality of speedup.”

Such a concept can be outline as follows: Let P be the set of feasible domain decompositions, which should somehow describe to the set of available physical processors on a given parallel computing environment. In the case of `blackbox.sdsu.edu`, a sample set P could be given by:

$$P = \{1, 2, 4, 8, 16, 32, 64\}. \quad (34)$$

Let $L(P)$ be the set of ideal linear speedup values attained when considering a higher granularity for the domain decomposition, based on P . Clearly, $L(P) = P$. Similarly, let $S(P, r, \beta)$ be the set of actual attained speedups, for a given BloGS matrix with rank and bandwidth r and β , respectively. Define also:

$$\hat{s}(r, \beta) = \max_{i \in P} S(i, r, \beta), \quad (35)$$

as the best attained speedup for the pair (r, β) . Based on this, we define and denote the **quality of the speedup for** (r, β) as

$$q(r, \beta) = \hat{s}(r, \beta) \times p(P, S(P, r, \beta)), \quad (36)$$

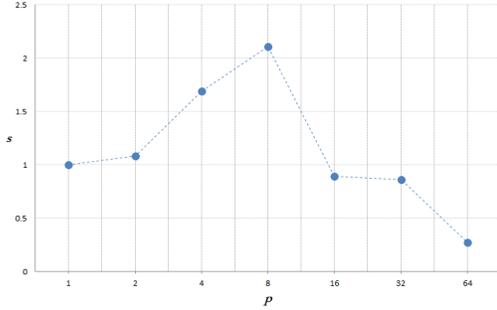
where $p(P, S(P, r, \beta))$ denotes the **Pearson linear correlation coefficient** for the samples given in sets P and $S(P, r, \beta)$, $p \in [-1, 1] \subset \mathbb{R}$.

Equation (36) should be intuitively discussed. In such equation, we are weighting the best attained speedup, against how close to linear it is; since linear speedup is considered to be ideal. We do not take the worst speedup into account, since this may be a misleading number, because of the existence of “sweet spots” in the execution time (See Figures 13a to 13c). Taking the maximum speedup, allows us to account for these spots, in where speedup is maximum, but then decreases for factors such as insufficient problem size yielding process intercommunication overhead. When executed for a more comprehensive (r, β) -parameter space, we obtain the results depicted in Figures 13 and 14.

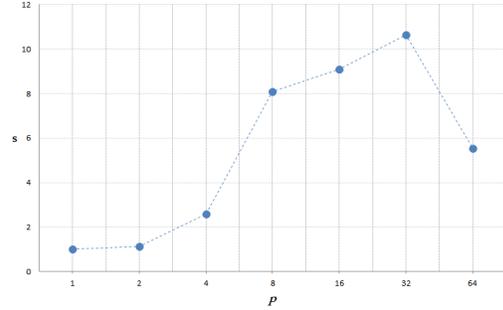
When executed for a more comprehensive (r, β) -parameter space, we obtain the results depicted in Figures 13 and 14.

These results are very important in explaining the differences in terms of the algorithmic nature of the solvers, and their impact on the goal of attaining scalable speedup. As it can be seen in Figure 13, the ScaLAPACK solver scales properly for narrow banded large matrices, as it is expected, given its algorithmic nature discussed in [Cleary and Dongarra, 1997]. Specifically, an insufficient rank yields overhead based on process intercommunication, as it can be seen in Figures 13a to 13c, where a “sweet spot”, or the maximum speedup, appears at 8 and 32 processors. For higher-rank matrices, scalable speedup in consistently achieved (Figure 13d).

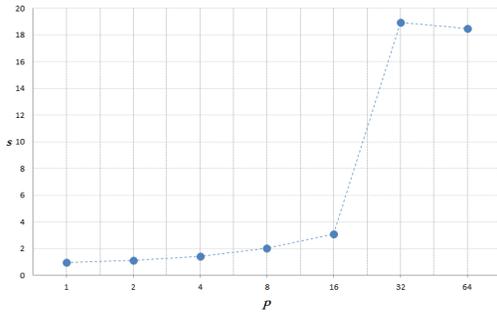
Figure 14 depicts the broader scenario. Specifically, Figure 14a shows the attained speedup for relatively low-rank matrices. As it can be seen, scenarios of narrow bands with relation to the rank, show decent scalability as the rank increases. However, as the bandwidth increases, we loose performance. For large matrices, the available memory imposes a stronger restriction, as the bandwidth increases, up to the point in which for the case of $r = 10,000,000$, some cases of wider band could not be executed. This is graphically represented as very low quality data point, however it is better explained numerically in Table 9.



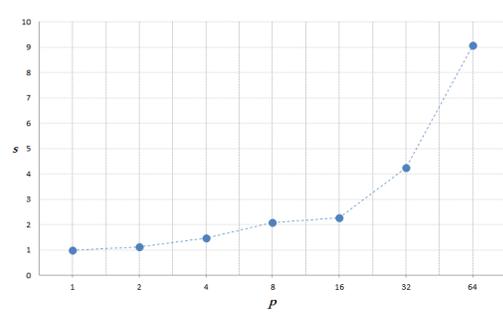
(a) Attained speedup for $(r, \beta) = (10000, 2)$.



(b) Attained speedup for $(r, \beta) = (100000, 2)$.



(c) Attained speedup for $(r, \beta) = (500000, 2)$.

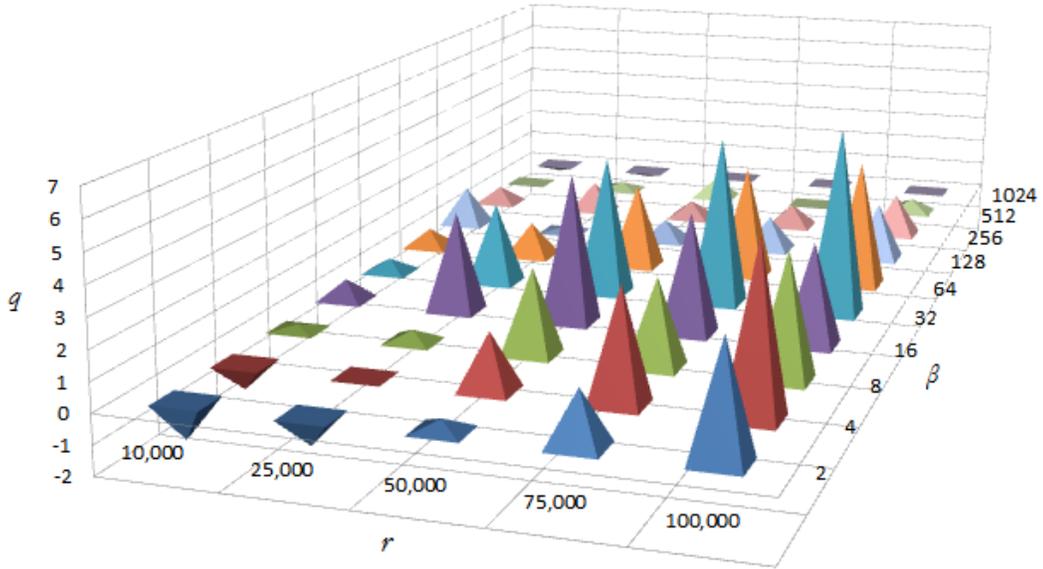


(d) Attained speedup for $(r, \beta) = (1000000, 2)$.

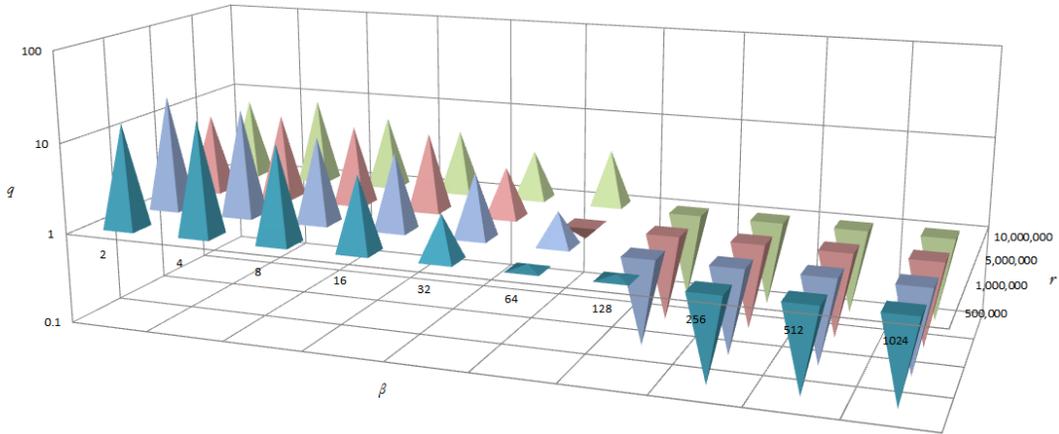
Figure 13: Analytical and computed solutions for the LAPACK and SuperLU_SEQ prototype drivers for the solution of a BloGS system. See §7.3. p stands for number of processing cores.

| | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1,024 |
|------------|--------|--------|--------|-------|-------|-------|-------|-----|-----|-------|
| 500,000 | 14.578 | 18.324 | 12.020 | 6.724 | 3.106 | 1.145 | 1.117 | - | - | - |
| 1,000,000 | 19.229 | 15.742 | 9.070 | 7.257 | 5.163 | 2.366 | - | - | - | - |
| 5,000,000 | 7.688 | 8.925 | 7.678 | 7.513 | 3.656 | 0.673 | - | - | - | - |
| 10,000,000 | 7.707 | 9.010 | 6.459 | 5.181 | 3.540 | 4.274 | - | - | - | - |

Table 9: Attained qualities for high-rank matrices. See §7.3 and Figure 14b.



(a) Attained qualities for low-rank matrices.



(b) Attained qualities for high-rank matrices.

Figure 14: Attained qualities for the speedup under a more comprehensive (r, β) -space (rank and bandwidth). See §7.3 and Tables 8 and 9. Coloring is simply a result of the plotting software used; it means nothing special except that it is useful to visualize differentiate the different collection of values being plotted.

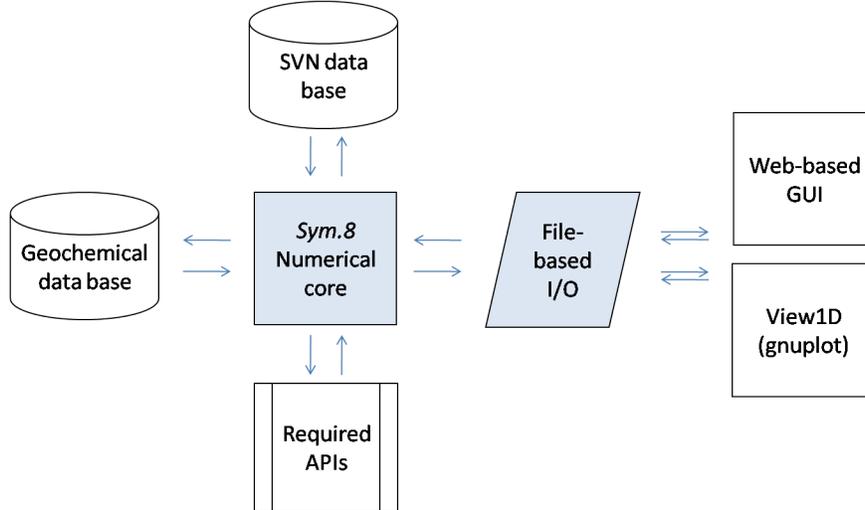


Figure 15: Current architecture of *WebSym.C*. See §5 and 4.

8 The performance of *WebSym.C* using the BloGS scheme

Based on the results presented in §7.3, we realized that a selection criteria between the considered solvers makes no sense to be implemented within the current version of *WebSym.C*. Its numerical core (*Sym.8*), which algorithmics we have explained in §3.3, in its current state of development released to the authors of this work, supports only one-dimensional simulation scenarios. Instances of the BloGS matrices, for these types of scenarios possess a very narrow band, even for the biggest case possible in our reference test case, discussed in §3.3. This case proposes the computation of the concentration profiles for only 12 different solutes. From Equation (12), the largest bandwidth possible for the obtained BloGS matrices, assuming a value $\omega = 2$, will be of only 50 elements. However, Table 6 shows the attained scalability from utilizing SuperLU_DIST on these types of matrices. It shows that given the narrow band, regardless of the rank being large, a scalable speedup can't be achieved, given the nature of this solver. Furthermore, even if a different test case were to be studied and used for the validations of these results, the memory constraint of the simulator of allowing only 30 solutes, would still yield matrices for which the bandwidth, would not reach the required size. Therefore, the matrix would still be considered banded, thus discarding SuperLU_DIST as an option.

9 Concluding remarks and directions of future work

In this work we have presented the theoretical development and preliminary tests of a general storage scheme for any application that requires repeated, independent solutions of a linear system. Specifically, we studied its suitability to exploit High-Performance computing resources. As an application example, we considered the *WebSym.C*, a general water-rock and reactive mass transport simulator. We explored its potential for improvement at the sequential level. Motivated by these results and by the understanding of the algorithmics of *WebSym.C*, we presented the development of the BloGS matrices, which are described through a theory that allows for the description of the intended storage scheme, where the concentrations of all the solutes, are computed in parallel. This led to the discussion and comparison of the performance of several classic APIs for the solution of this problem which clearly showed the constraints in terms of performance of these tools with relation to the problem they intend to solve. To study the attained scalability of the solvers, as a function of specific problem parameters (rank and bandwidth), we introduced the concept of “quality of speedup”. This allowed us to quantify the effect of these parameters that describe instances of BloGS matrices, in a way that we could numerically describe the suitability of a given solution approach (as implemented by different solvers) in a surface defined over the parameter space of

interest.

The concept of “quality of speedup” utilizes the Pearson linear correlation coefficient. Based on this we can compute the quality of the speedup, for any parameter space that properly describes any problem of interest. This quantitative approach allows for the description of the suitability of any distributed memory approach, in terms of achieving a scalable speedup.

We showed that, for one-dimensional scenarios, the BloGS matrices can be represented by banded matrices, which can be solved using banded solvers, thus allowing for scalable speedup. However, for these matrices to possess a structure that allows for more general solvers, these would have to be implemented in higher-dimensional scenarios. As an immediate direction of future work, we intend to apply the BloGS scheme to higher-dimensional problems. This effort will be accompanied with the study of using more generally sparse solvers, such as SuperLU_DIST. We also intend to exploit the concept of quality of speedup in higher-dimensional scenarios, in where several solution approaches become an option depending on the parameter space. This would allow for the creation of a heuristic decision criteria, based on the parameter and quality of the speedup we intend to achieve. Also, the creation of parallel solver tailored to BloGS’ inherent properties is also appealing.

These results, when considering the algorithmic nature of *Sym.8*, justify the creation of a new simulator. Specifically, we require for this new simulator to be able to address higher-dimensional scenarios, thus making the use of SuperLU_DIST an efficient choice. We propose this new simulator to consider an fully distributed domain decomposition from its inception, thus depicting truth scalability, not only at the concentration solver stage, but at the entire process. We intend to also study the implementation of Mimetic Discretization Methods, since they should provide a higher fidelity numerical solutions.

10 Acknowledgments

The author of this work would like to extend their gratitude to Dr. Anthony J. Park for his assistance in understanding the algorithmics of *Sym.8*, *WebSym.C*’s numerical core.

This work was performed under the following financial support:

- Computational Science Research Center at San Diego State University.
- Department of Energy (DOE) National Energy Technology Laboratory (NETL): *Web-based CO₂ Subsurface Modeling, Geologic Sequestration Training and Research Funding* . Opportunity Number: DE-FOA-0000032, Simulation and Risk Assessment.

References

- [Anderson et al., 1999] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999). *LAPACK Users’ Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition.
- [Bao et al., 2013] Bao, B., Melo, L., Davies, B., Fadaei, H., Sinton, D., and Wild, P. (2013). Detecting Supercritical CO₂ in Brine at Sequestration Pressure with an Optical Fiber Sensor. *Environmental Science and Technology*, 47(1):306–313.
- [Bennion and Bachu, 2007] Bennion, D. and Bachu, S. (2007). Permeability and Relative Permeability Measurements at Reservoir Conditions for CO₂ -Water Systems in Ultra Low Permeability Confining Caprocks. *Society of Petroleum Engineers*.
- [Boudreau, 1996] Boudreau, B. (1996). *Diagenetic Models and Their Implementation*. Springer, 1 edition.
- [Castillo and Grone, 2003] Castillo, J. and Grone, R. (2003). A matrix analysis approach to higher-order approximations for divergence and gradients satisfying a global conservation law. *Siam J. Matrix Anal. Appl.*, 25:128–142.

- [Castillo and Miranda, 2013] Castillo, J. and Miranda, G. (2013). *Mimetic Discretization Methods*. CRC Press, 1st edition.
- [Choi et al., 1994] Choi, J., Dongarra, J. J., Ostrouchov, L. S., Petitet, A. P., Walker, D. W., and Whaley, R. C. (1994). The Design and Implementation of the ScaLAPACK LU, QR, and Cholesky Factorization Routines. Technical Report 80, LAPACK Working Note.
- [Cleary and Dongarra, 1997] Cleary, A. and Dongarra, J. (1997). Implementation in ScaLAPACK of Divide-and-Conquer Algorithms for Banded and Tridiagonal Linear Systems. Technical report, Center for Research on Parallel Computation, 6100 South Main Street Houston, TX 77005.
- [Demmel et al., 1999] Demmel, J. W., Eisenstat, S. C., Gilbert, J. R., Li, X. S., and Liu, J. W. H. (1999). A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Analysis and Applications*, 20(3):720–755.
- [EPA, 2012a] EPA (2012a). Inventory of U.S. Greenhouse Gas Emissions and Sinks: 1990-2010. Technical report, Environmental Protection Agency (EPA).
- [EPA, 2012b] EPA (2012b). Standards of Performance for Greenhouse Gas Emissions for New Stationary Sources: Electric Utility Generating Units. *Nature*, 77(72):1–50.
- [Ewing, 1983] Ewing, R., editor (1983). *The Mathematics of Reservoir Simulation*. The Society for Industrial and Applied Mathematics, Philadelphia, 1 edition.
- [Fenlason, 1993] Fenlason, J. (1993). GNU gprof.
- [Golombek et al., 2009] Golombek, R., Greaker, M., Sverre, A., Kittelsen, O., and Aune, F. (2009). Carbon capture and storage technologies in the European power market. Technical Report 603, Statistics Norway, Research Department.
- [Harvey et al., 2013] Harvey, O. R., Qafoku, N. P., Cantrell, K. J., Lee, G., Amonette, J. E., and Brown, C. F. (2013). Geochemical Implications of Gas Leakage associated with Geologic CO₂ Storage: A Qualitative Review. *Environmental Science and Technology*, 47(1):23–36.
- [Jaramillo et al., 2009] Jaramillo, P., Griffin, W. M., and McCoy, S. T. (2009). Life Cycle Inventory of CO₂ in an Enhanced Oil Recovery System. *Environmental Science and Technology*, 43(21):8027–8032. PMID: 19924918.
- [Ji and Zhu, 2013] Ji, X. and Zhu, C. (2013). Predicting Possible Effects of H₂S Impurity on CO₂ Transportation and Geological Storage. *Environmental Science and Technology*, 47(1):55–62.
- [Jun et al., 2013] Jun, Y. S., Giammar, D. E., and Werth, C. J. (2013). Impacts of Geochemical Reactions on Geologic Carbon Sequestration. *Environmental Science and Technology*, 47(1):3–8.
- [Kharaka et al., 2006] Kharaka, D., Cole, S., Hovorka, W., Gunter, K., Knauss, B., and Freifeld, Y. (2006). Gas-water-rock interactions in Frio Formation following CO₂ injection: Implications for the storage of greenhouse gases in sedimentary basins. *Geology*, 34(7):577–580.
- [Khoo and Tan, 2006] Khoo, H. H. and Tan, R. B. H. (2006). Environmental Impact Evaluation of Conventional Fossil Fuel Production (Oil and Natural Gas) and Enhanced Resource Recovery with Potential CO₂ Sequestration. *Energy and Fuels*, 20(5):1914–1924.
- [Li et al., 1999] Li, X., Demmel, J., Gilbert, J., Grigori, i., Shao, M., and Yamazaki, I. (1999). SuperLU Users’ Guide. Technical Report LBNL-44289, Lawrence Berkeley National Laboratory. <http://crd.lbl.gov/~xiaoye/SuperLU/>. Last update: August 2011.
- [Li, 2005] Li, X. S. (2005). An Overview of SuperLU: Algorithms, Implementation, and User Interface. *ACM Transactions on Mathematical Software*, 31(3):302–325.

- [Li and Demmel, 2003] Li, X. S. and Demmel, J. W. (2003). SuperLU_DIST: A Scalable Distributed-Memory Sparse Direct Solver for Unsymmetric Linear Systems. *ACM Trans. Mathematical Software*, 29(2):110–140.
- [Liu and Wilcox, 2013] Liu, Y. and Wilcox, J. (2013). Molecular Simulation Studies of CO₂ Adsorption by Carbon Model Compounds for Carbon Capture and Sequestration Applications. *Environmental Science and Technology*, 47(1):95–101.
- [Marini, 2006] Marini, L. (2006). Chapter 3 Carbon dioxide and CO₂ H₂O mixtures. In *Geological Sequestration of Carbon Dioxide Thermodynamics, Kinetics, and Reaction Path Modeling*, volume 11 of *Developments in Geochemistry*, pages 27–51. Elsevier.
- [McAlexander et al., 2011] McAlexander, I., Rau, G. H., Liem, J., Owano, T., Fellers, R., Baer, D., and Gupta, M. (2011). Deployment of a Carbon Isotope Ratiometer for the Monitoring of CO₂ Sequestration Leakage. *Analytical Chemistry*, 83(16):6223–6229.
- [McCoy and Rubin, 2008] McCoy, S. and Rubin, E. (2008). An engineering-economic model of pipeline transport of CO₂ with application to carbon capture and storage. *International journal of greenhouse gas control*, 20:219–229.
- [Movagharnejad and Akbari, 2011] Movagharnejad, K. and Akbari, M. (2011). Simulation of CO₂ Capture Process. *Engineering and Technology*, 58.
- [NETL, 2011] NETL (2011). Enhancing the Success of Carbon Capture and Storage Technologies. Technical report, U.S. Department of Energy (DOE), National Energy Technology Laboratory (NETL).
- [Northington et al., 2012] Northington, J., Morton, F., and Yongue, R. (2012). Advanced Technology Testing at the National Carbon Capture Center. In *Proceedings of the 29th Annual International Pittsburgh Coal Conference*.
- [Pacheco, 1997] Pacheco, P. S. (1997). *Parallel Programming with MPI*. Morgan Kaufmann, first edition.
- [Paolini et al., 2011] Paolini, C. P., Binter, C. P., Park, A. J., and Castillo, J. E. (2011). An Investigation of the Variation in the Sweep and Diffusion Front Displacement as a Function of Reservoir Temperature and Seepage Velocity with Implications in CO₂ Sequestration. *Proceedings International Energy Conversion Engineering Conference*.
- [Park, 2009] Park, A. i. p. (2009). Water-Rock Interaction and Reactive Transport Modeling Using Elemental Mass-Balance and Explicitly Coupled Iteration Methods: I. the Methodology. *American Journal of Science*.
- [Pentland, 2008] Pentland, W. (2008). The Carbon Conundrum. *Forbes.com*.
- [Press et al., 1988] Press, W., Teukolsky, S., Flannery, B., and Vetterling, W. (1988). *Numerical Recipes in C*. Cambridge University Press, 1 edition.
- [Sanchez, 2012] Sanchez, E. (2012). The Mimetic Methods Toolkit (MTK) online website.
- [Sanchez et al., 2012] Sanchez, E., Paolini, C., and Castillo, J. (2012). Research Advances for Fall 2012: Mimetic Methods Toolkit: An object-oriented API implementing Mimetic Discretization Methods. Technical report, Computational Science Research Center at San Diego State University.
- [SDSC, 2012] SDSC (2012). SDSC Trestles User Guide.
- [Seto and McRae, 2011] Seto, C. J. and McRae, G. J. (2011). Reducing Risk in Basin Scale CO₂ Sequestration: A Framework for Integrated Monitoring Design. *Environmental Science and Technology*, 45(3):845–859.

- [Shakun, 2012] Shakun, J. D. (2012). Global warming preceded by increasing carbon dioxide concentrations during the last deglaciation. *Nature*, 484:49–54.
- [Song and Zhang, 2013] Song, J. and Zhang, D. (2013). Comprehensive Review of Caprock-Sealing Mechanisms for Geologic Carbon Sequestration. *Environmental Science and Technology*, 47(1):9–22.
- [Soong et al., 2012] Soong, Y., Gray, M., Siriwardane, R., and Champagne, K. (2012). Novel Amine Enriched Solid Sorbents for Carbon Dioxide Capture. *Journal of Energy & Environmental Research*, 1(1).
- [Suebsiri et al., 2006] Suebsiri, J., Wilson, M., and Tontiwachwuthikul, P. (2006). Life-Cycle Analysis of CO₂ EOR on EOR and Geological Storage through Economic Optimization and Sensitivity Analysis Using the Weyburn Unit as a Case Study. *Industrial and Engineering Chemistry Research*, 45(8):2483–2488.
- [Walther, 2009] Walther, J. V. (2009). *Essentials of Geochemistry*. Jones and Bartlett Publishers, LLC, 2nd edition.
- [White et al., 2003] White, C., Strazisar, B., Granite, E., Hoffman, J., and Pennline, H. (2003). Separation and capture of CO₂ from large stationary sources and sequestration in geological formations-Coalbeds and deep saline aquifers. *Journal of the Air and Waste Management Association*, 53:645–715.
- [Zaman et al., 2012] Zaman, M., Lee, J., and Gani, R. (2012). Carbon Dioxide Capture Processes: Simulation, Design and Sensitivity Analysis. *Proceedings 12th International Conference on Control, Automation and Systems*.
- [ZEP, 2011] ZEP (2011). The costs of CO₂ transport. Technical report, The European Technology Platform for Zero Emission Fossil Fuel Power Plants.
- [Zhou and Birkholzer, 2011] Zhou, Q. and Birkholzer, J. T. (2011). On scale and magnitude of pressure build-up induced by large-scale geologic storage of CO₂. *Greenhouse Gases Science and Technology*, 1(11-20).

A A generalized (BloGS) matrix with examples

In this Appendix, we intend to show the general form of a BloGS matrix, as well as some instantiated examples for both $\omega = 2$, and $\omega = 4$. This should help the reader to understand their general structure, while clarifying the application of this scheme to an specific discretization example.

The general form the BloGS matrices follows:

where

$$\mathbf{W}(\omega) = \begin{bmatrix} \mathbf{W}_{1,1} & \mathbf{W}_{1,2} & \cdots & \mathbf{W}_{1,\omega+1} \\ \mathbf{W}_{2,1} & \mathbf{W}_{2,2} & \cdots & \mathbf{W}_{2,\omega+1} \\ \vdots & & \ddots & \vdots \\ \mathbf{W}_{\omega/2,1} & \mathbf{W}_{\omega/2,2} & \cdots & \mathbf{W}_{\omega/2,\omega+1} \end{bmatrix}, \quad (38)$$

represents the collection of west boundary blocks, which are defined in terms of the order of accuracy ω , and

$$\mathbf{E}(\omega) = \begin{bmatrix} \mathbf{E}_{n_x - (\frac{\omega}{2} - 1), n_x - \omega} & \cdots & \mathbf{E}_{n_x - (\frac{\omega}{2} - 1), n_x - 1} & \mathbf{E}_{n_x - (\frac{\omega}{2} - 1), n_x} \\ \vdots & \ddots & & \vdots \\ \mathbf{E}_{n_x - 1, n_x - \omega} & \cdots & \mathbf{E}_{n_x - 1, n_x - 1} & \mathbf{E}_{n_x - 1, n_x} \\ \mathbf{E}_{n_x, n_x - \omega} & \cdots & \mathbf{E}_{n_x, n_x - 1} & \mathbf{E}_{n_x, n_x} \end{bmatrix}, \quad (39)$$

represents the collection of east boundary blocks, which are also defined in terms of ω . Each block is strictly diagonal and it has dimensions of $N_a \times N_a$.

In §7, we presented an example for $\omega = 2$. See Equation (7). Such example, when instantiated with a second order, centered finite difference discretization method, in order to solve Equation (20) with $n_x = 6$, yields the matrix:

$$B(2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \left(\frac{1}{\Delta x^2} + \frac{1}{2\Delta x}\right) & 0 & -\frac{2}{\Delta x} & 0 & \left(\frac{1}{\Delta x^2} - \frac{1}{2\Delta x}\right) & 0 & 0 & 0 & 0 & 0 \\ 0 & \left(\frac{1}{\Delta x^2} + \frac{1}{2\Delta x}\right) & 0 & -\frac{2}{\Delta x} & 0 & \left(\frac{1}{\Delta x^2} - \frac{1}{2\Delta x}\right) & 0 & 0 & 0 & 0 \\ & \ddots & & & \ddots & & & \ddots & & \\ & & \left(\frac{1}{\Delta x^2} + \frac{1}{2\Delta x}\right) & 0 & -\frac{2}{\Delta x} & 0 & \left(\frac{1}{\Delta x^2} - \frac{1}{2\Delta x}\right) & 0 & 0 & 0 \\ & & 0 & \left(\frac{1}{\Delta x^2} + \frac{1}{2\Delta x}\right) & 0 & -\frac{2}{\Delta x} & 0 & \left(\frac{1}{\Delta x^2} - \frac{1}{2\Delta x}\right) & 0 & 0 \\ & & -1 & 0 & 4 & 0 & 2\Delta x - 3 & 0 & 0 & 0 \\ & & 0 & -1 & 0 & 4 & 0 & 2\Delta x - 3 & 0 & 0 \end{bmatrix}. \quad (40)$$

The related system will have the following form:

$$\mathbf{B}(2)\mathbf{c} = \mathbf{r} = \mathbf{B}(2) \begin{bmatrix} c_{1,1} \\ c_{2,1} \\ c_{1,2} \\ c_{2,2} \\ c_{1,3} \\ c_{2,3} \\ c_{1,4} \\ c_{2,4} \\ c_{1,5} \\ c_{2,5} \\ c_{1,6} \\ c_{2,6} \end{bmatrix} = \begin{bmatrix} r_{1,1} \\ r_{2,1} \\ r_{1,2} \\ r_{2,2} \\ r_{1,3} \\ r_{2,3} \\ r_{1,4} \\ r_{2,4} \\ r_{1,5} \\ r_{2,5} \\ r_{1,6} \\ r_{2,6} \end{bmatrix}. \quad (41)$$

It is noteworthy that, given the nature of the system, both the solution vector and the vector containing the terms for the reactive components of the equations are collated, thus some minor processing is required once the system has been solved, in order to get the independent solutions for the system.

An example for $\omega = 4$ follows:

$$\mathbf{B}(4) = \begin{bmatrix}
\mathbf{W}_{1,1} & \mathbf{W}_{1,2} & \mathbf{W}_{1,3} & \mathbf{W}_{1,4} & \mathbf{W}_{1,5} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \mathbf{B}_{2,3} & \mathbf{B}_{2,4} & \mathbf{B}_{2,5} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{B}_{3,2} & \mathbf{B}_{3,3} & \mathbf{B}_{3,4} & \mathbf{B}_{3,2} & \mathbf{B}_{3,2} & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \ddots & & \ddots & \ddots & \ddots & & \ddots & \vdots \\
\mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}_{n_x-2, n_x-5} & \mathbf{B}_{n_x-2, n_x-4} & \mathbf{B}_{n_x-2, n_x-3} & \mathbf{B}_{n_x-2, n_x-2} & \mathbf{B}_{n_x-2, n_x-1} & \mathbf{0} \\
\mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{B}_{n_x-1, n_x-4} & \mathbf{B}_{n_x-1, n_x-3} & \mathbf{B}_{n_x-1, n_x-2} & \mathbf{B}_{n_x-1, n_x-1} & \mathbf{B}_{n_x-1, n_x} \\
\mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{E}_{n_x, n_x-4} & \mathbf{E}_{n_x, n_x-3} & \mathbf{E}_{n_x, n_x-2} & \mathbf{E}_{n_x, n_x-1} & \mathbf{E}_{n_x, n_x}
\end{bmatrix}. \tag{42}$$

B The Mimetic Methods Toolkit

All of the related research efforts are summarized in the following CSRC research report:

http://www.csrc.sdsu.edu/research_reports/CSRCR2012-05.pdf