Mitigating the Energy Crisis using Simulation for Enhanced Oil Recovery: Analyzing oil fields grain by grain

David W. Holmes* John R. Williams† Peter Tilke‡

August 15, 2008

Abstract

In this paper, we address some of the very real challenges facing a world with an ever increasing appetite for oil. Finite global oil reserves pose an impending threat to our way of life and improving on the current inefficiency of the oil recovery process is increasingly becoming the primary research focus of the petroleum industry. Numerical simulation is a particularly powerful research tool and we address the need for developing codes which will be capable of improving our understanding of oil reservoir dynamics.

1 Problem Statement

In an oil reservoir, $20-40\%$ of the oil can be recovered by primary development techniques. The rest remains trapped in the rock pores. Enhanced Oil Recovery (EOR) techniques, such as water flooding, gas injection, chemical injection and thermal stimulation, optimistically recover an additional $10-20\%$ of the oil. This still leaves almost half of the oil trapped in the rock pores. The Department of Energy (DOE) has estimated that if "next generation" EOR is applied, the United States could generate an additional 240 billion barrels of recoverable oil resources - over 30 years supply at the present US consumption rate of 20 million barrels per day. For comparison, the Middle East holds an estimated 685 billion barrels that are recoverable and the tar sands of Alberta 300 billion recoverable barrels of "heavy" oil, with over a trillion barrels potentially recoverable using enhanced methods.

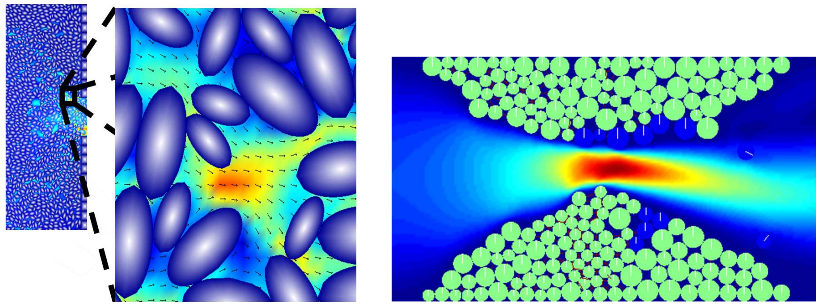
Designing new EOR technologies is inhibited by our poor understanding of the fundamental physical processes within a reservoir, particularly at the pore scale. Some of the modeling challenges include:

1. Predicting multi-phase fluid flow of oil, water and gas through complex pore networks (Fig. 1 (a)),
2. Predicting the properties of the rock and fluids from indirect down-hole measurement techniques, such as electrical resistivity and acoustic wave propagation,
3. Understanding the hydro-fracturing of rocks,

*Postdoctoral Associate, Civil and Environmental Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, Email: dholmes@mit.edu

†Director MIT Auto-ID Lab, Professor of Information Engineering, Civil and Environmental Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307

‡Scientific Advisor, Department of Mathematics and Modeling, Schlumberger-Doll Research Center, 1 Hampshire Street, Cambridge, MA 02139-1578



(a) (b)

Figure 1: (a) Lattice-Boltzmann simulation of fluid flow through porous media, (b) fluid flow through a pore throat with sand production.

4. Understanding the mechanisms of sand production and borehole collapse (Fig. 1 (b)),
5. Mastering the computational challenges, which stem from reasoning about the positions and contacts of large numbers of objects, and
6. Architecting integrated multi-physics software systems that can run on multi-core/multi-machine architectures.

2 Developing an Advanced Simulation Framework

The complexity of oil reservoir dynamics mean that the actual numerical tools used to simulate reservoir phenomena must be capable of optimally utilizing available computing power. Traditionally, multi-machine, cluster type computing has been the hardware architecture chosen for large scale numerical modeling, however with the move of companies like Intel and A.M.D. towards multi-core chip architectures and projects such as Intel's Kiefer project planning to release a 32 core chip by 2009/2010 [1], there is significant potential benefit to developing a simulation framework capable of taking advantage of the new direction in CPU technology.

Microsoft's *Concurrency and Coordination Runtime* (CCR)[1] [2, 3] has made significant progress towards addressing the challenge of software development for multi-core platforms. There is, however, little published work on the CCR's application to numerical simulation problems. In this research we have developed a spatial distribution algorithm (to be published elsewhere) which has allowed us to implement the CCR within a more specific simulation framework. By using the CCR's 'events based' concurrent execution and task scheduling and by adding a spatial distribution mechanism allowing events based load balancing and ensuring functional completion before synchronization, the framework is capable of 100% parallelization and near-linear[2] *speed-up* within multi-core architectures, see Fig 2. Additionally, the framework makes optimal use of shared memory with the largest 700,000 particle SPH model executing while using less than 1.5GB of memory.

It is our opinion that the continuing advancement of science and engineering simulation tools must take advantage of the new multi-core programming environments. Unfortunately, a great majority of the legacy

[1]presently distributed as part of the Microsoft Robotics Studio package

[2]The 3% sub-linearity is potentially accounted for by hardware inefficiency.

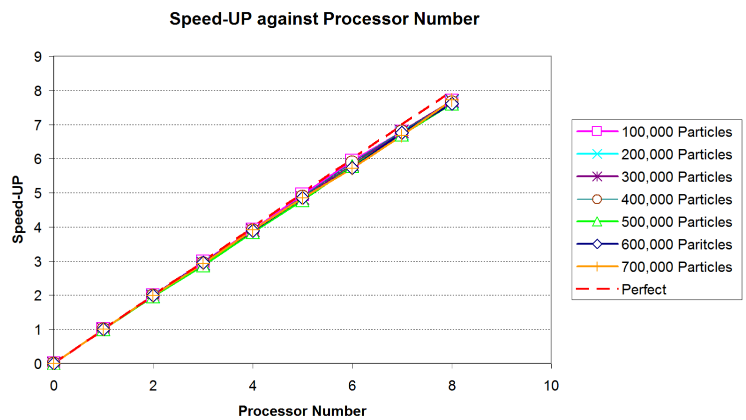


Figure 2: Results for *speed-up* of the developed simulation framework. Results from CFD simulations using Smoothed Particle Hydrodynamics on a Dell Server PE2900 with 8-core Intel Xeon E5345 CPU.

serial and cross-machine numerical codes being supported by industry will prove to be very challenging to convert to multi-core without significant cost. This poses many with some particulary difficult choices to make in the near future. An entirely new way of looking at numerical software development is necessary and those that delay may well be left behind.

3 Conclusion

The problem of oil recovery efficiency is something which has the potential to have global effect in the not so distant future. By developing numerical software which is capable of simulating complex multi-physics, multi-scale phenomena we will greatly enhance our understanding of oil reservoir dynamics and will greatly improve our ability to develop new EOR techniques. Such numerical tools will require significant computing power and, as such, must develop with advances in hardware architectures. We have provided here a brief overview of what can be expected of the future of numerical simulation.

References

[1] P. Schmid, The project "Keifer" awakening, Tom's Hardware.
URL http://www.tomshardware.com/reviews/project-keifer-32-core,1280.html

[2] G. Chrysanthakopoulos, S. Singh, An asynchronous messaging library for C#, in: Proceedings of the Workshop on Synchronization and Concurrency in Object-Oriented Languages, OOPSLA 2005, San Diego, 2005, pp. 89–97.
URL http://research.microsoft.com/~tharris/scool/papers/sing.pdf

[3] G. Chrysanthakopoulos, Concurrency runtime: An asynchronous messaging library for C# 2.0, Channel9 Wiki Microsoft.
URL http://channel9.msdn.com/wiki/concurrencyruntime/