

SRC-6 Implementation of Contrast Agent Flow and Diffusion Model for Macromolecules in Tumors

David Meixner¹, Nicholas Szrama², Michael Aref^{2,3,4}, Volodymyr Kindratenko¹

¹ National Center for Supercomputing Applications, UIUC

² Department of Nuclear, Plasma, and Radiological Engineering, UIUC

³ College of Medicine, UIUC

⁴ Biomedical Imaging Center, Beckman Institute, UIUC

{dmeixner|szrama|maref|kindrtnk}@uiuc.edu

1. Introduction

Dynamic contrast-enhanced (DCE) magnetic resonance imaging has the potential for greater diagnostic accuracy using high spatial resolution estimates of contrast agent pharmacokinetics. The heterogeneous microenvironment of the tumor exerts convective and diffusive forces on contrast agent molecules that are hypothesized to be related to histopathologically important features such as neovascularization. Therefore, it is necessary to better understand the transport of fluids and macromolecules inside tumors.

One reason for the non-uniform distribution of contrast agent is the rich vascular outer rim of tumors compared with the necrotic core. In addition, contrast agent distribution is governed by interstitial pressure gradients inside tumors. This causes a reduction of transcapillary exchange for fluids and macromolecules. Also, it creates a radially outward convection flux, making it harder for macromolecules to effectively penetrate inside the tumor. However, there are no direct measurements to support this theory, so a mathematical model has been created to describe the macroscopic fluid and solute transport in tumors [1].

In this work, we report on the results of implementing the microscopic fluid and solute transport in tumors model on an SRC-6 reconfigurable computer. Originally, this model was implemented in MATLAB. While convenient to implement, the run time of such an implementation is prohibitively long for any practical experiments or clinical application. A faster implementation was created by translating the MATLAB model to C, and even faster results were obtained using an SRC-6 reconfigurable computer [4]. In this work, we outline the process of implementing the code on SRC-6 and discuss future plans.

2. The Model

The model of macroscopic fluid and solute transport in tumors is based on the convection diffusion equation [1], [2]:

$$\frac{\partial [CA]_i}{\partial t} + \nabla \cdot (\vec{v}_i [CA]_i) = \nabla \cdot (D_i \nabla [CA]_i) + g_i$$

- $[CA]_i$ is the concentration of the contrast agent in the i^{th} compartment
- v_i is velocity of the contrast agent within the compartment of interest
- D_i is the diffusion coefficient in the i^{th} compartment
- g_i is the net volumetric gain of contrast agent concentration (i.e. sources - sinks)

For the purposes of this implementation, the tumor is modeled as a two-dimensional rectangle. Ideally there are 200×200 compartments making up one time point and the solution is a finite difference approximation using a time-step of 10^{-6} seconds with 300,000,000 steps altogether [3]. Instead of storing the frame at every step, only one out of every 100,000 frames will be stored for final analysis.

3. Model Implementation on SRC-6

MATLAB is a popular tool used for modeling systems and image processing. Due to the processing of code within MATLAB, it is computationally slow, so if fast results are desired, a different implementation is needed. The SRC-6 reconfigurable computer is used in this study to obtain increased performance. Conveniently, the SRC-6 can be programmed in a C-like language. The compiler then translates that code into Verilog for implementation on an FPGA. Efficiently implementing the considered algorithm on

SRC-6 however requires algorithm restructuring in order to take a full advantage of the hardware.

The algorithm implemented in this paper consists of three nested loops: a time-step loop and two loops for each of the x- and y-dimensions. Since the computational core of the algorithm being implemented contains 65 floating-point multiplication operations and 34 floating-point division operations, a limiting factor in the SRC-6 implementation is the available Field-Programmable Gate Array (FPGA) resources needed to implement these operators. Therefore, both FPGAs (Virtex-II Pro XC2VP100) in the SRC-6 MAP Series E processor are used, where the first performs some initialization and boundary conditions calculations whereas the second performs the rest of the calculations at each time-step (Figure 1).

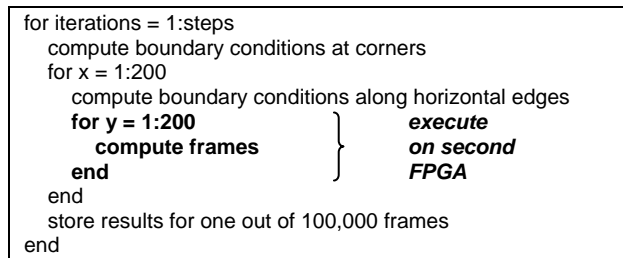


Figure 1: Loop structure

Once the first FPGA is done performing its calculations, values computed along the horizontal edges are stored in the on-board memory (OBM) banks and are used by the second FPGA to compute the rest of the frame. Only the loop carried out on the second FPGA is pipelined.

4. Results

The algorithm was tested using a frame size of 100×100 , 3000 steps, and a step size of 10^{-5} . The performance results from MATLAB, C, and the SRC-6 are shown in Table 1. It is clear that both C and the SRC-6 show considerable speedups over the MATLAB implementation, and the SRC-6 showed about a $6 \times$ speedup over the C code.

	MATLAB	C	SRC-6
Time (sec)	82.21	12.83	2.15

Table 1: Performance results for frame size of 100×100 , 3000 steps, and $\Delta t = .00001$.

5. Discussion

Ideally, the parameters used should be: a frame size as large as 200×200 , 300,000,000 steps, and a time step of 10^{-6} . From a programming point-of-view, this is not difficult to implement as long as the available FPGA hardware resources enable such an implementation. The test was run using the smaller parameters so the execution in MATLAB would finish in a short number of hours.

The SRC-6 has only eight separate on-board memory banks, so only eight read/writes can be performed simultaneously. Because there are several arrays that must be accessed, and they each must be accessed multiple times in the same loop, the inner-most loop on the second FPGA is delayed by four clock cycles needed to access the required data. This feature limits the performance of the current implementation.

Another limiting factor is the amount of available on-board memory that can be used to store the frames data. Thus, our current implementation is effectively limited to a 100×100 frame rather than required 200×200 .

Despite these limitations, the SRC-6 showed considerable speed-up over MATLAB, and showed a measurable speed-up over the same algorithm implemented in C.

Improved performance is expected on the SRC-7 when it becomes available: a $1.5 \times$ speedup is expected due to the increased FPGA operational frequency (150 MHz on SRC-7 instead of 100 MHz on SRC-6); a $4 \times$ speedup is expected due to the wider memory access (16 OBM banks on SRC-7 instead of 8 on SRC-6 so the inner-most loop can be fully pipelined with only one clock per iteration). Thus, even without considering any algorithmic changes, we expect a $6 \times$ speedup over current SRC-6 implementation.

References

- [1] L. Baxter, and R. Jain, "Transport of Fluid and Macromolecules in Tumors I. Role of Interstitial Pressure and Convection", *Microvascular Research*, 1989, pp. 77-104.
- [2] P. Szymczak, and A. J. C. Ladd, "Boundary Conditions for Stochastic Solutions of the Convection Diffusion Equation", *Physical Review E*, 2003, 68, pp. 1-12.
- [3] Duderstadt, J., and Hamilton, L. (1976). *Nuclear Reactor Analysis*. New York: John Wiley & Sons, Inc.
- [4] SRC Computers Inc., SRC Systems and Servers Datasheet, Colorado Springs, CO, 2005.