11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

COMPARING PARALLEL TECHNOLOGIES BASED ON *GPU* AND *CPU* IN NUMERICALLY SOLVING SINGLE PHASE FLOW PROBLEMS

DANY S. DOMINGUEZ[*], ESBEL. V. ORELLANA[*], BRUNO P. SANTOS[†] AND SUSANA M. IGLESIAS[*]

[*] Pós-graduação em Modelagem Computacional,
Universidade Estadual de Santa Cruz,
Rodovia Jorge Amado km 16, CEP 45662-900, Ilhéus-Bahia
e-mail: dsdominguez@gmail.com, evalero@uesc.br, smiglesias@uesc.br,
www.nbcgib.uesc.br/ppgmc

[†] Departamento de Ciências Exatas e Tecnológicas
Universidade Estadual de Santa Cruz,
Rodovia Jorge Amado km 16, CEP 45662-900, Ilhéus-Bahia
email: bruno.ps@live.com, www.uesc.br

Key Words: *Parallel computing, GPU, CPU, Single phase flow.*

Abstract. A numerical solution for a single phase flow in a porous medium problem is relevant in many fields of science and engineering. Due to the high computational cost these problems should be solved using parallel processing techniques. Initially, the MPI and OpenMP models were used for CPU architecture as the main parallel technique. Recently the CUDA model for graphic processing units has settled as the main technique for solving large scientific problems. This work presents a numerical solution for a single phase flow in a porous medium problem using GPU-CUDA technology. The mathematical model uses the Poisson mixed equation. The numerical solution is based in the Raviart-Thomas finite element method and a domain decomposition method. In this work were implemented three parallel codes: the first using the MPI model, the second using an MPI-OPenMP hybrid model and the last one using CUDA. Were made performance tests for various spatial meshes in CPU and GPU architectures and performance metrics were calculated. The applied tests showed that CUDA and GPU are excellent alternatives to solve single phase flow problems in porous medium.

1 INTRODUCTION

The solution of single phase flow through porous media is of special interest in several fields of science and engineering, such as primary exploration of hydrocarbon reservoirs [1], groundwater aquifer simulation and heat transfer in pebble bed gas cooled nuclear reactors [2] among other disciplines. All those problems present high computational cost and generally are solved using parallel processing techniques. The first of these techniques used to solve

this type of problems was based in Central Processing Unit (CPU) using Message Passing Interface (MPI) and Open Multi Processing (OpenMP) models. The MPI model was developed to use in distributed memory environments. The OpenMP model is suitable for multi-core architectures with shared-memory. In the last years the use of Graphics Processing Unit (GPU) and Compute Unified Architecture (CUDA) platform has settled as the main technique for solving large scientific problems [1, 3].

This work presents a numerical solution for a single phase flow in a porous medium problem using GPU-CUDA technology. The mathematical modeling of the problem uses a mixed Poisson equation and the numerical solution is implemented using the Raviart-Thomas finite element method [4, 5] and a domain decomposition method [6]. In [7] were used the MPI and OpenMP models to solve the problem treated in this work in distributed memory and in shared memory respectively. A hybrid implementation is offered in [8] to solve the problem in computer clusters where each node has a multi-core architecture. The efficiency of the present GPU - CUDA approach is compared with previous results [7, 8] that used CPU architectures to solve the same problem.

In the next section we offer the mathematical model´s basis. Details about the algorithm and the parallelization mechanisms used are described in section 3. The results obtained in the GPU-CUDA implementation, are shown in section 4 and finally in section 5, are offered the conclusions and suggestions for future works.

2 MATHEMATICAL FUNDAMENTALS

The mathematical modeling of the problem is a mixed Poisson equation in two-dimensional Cartesian geometry. Considering a rectangular domain D, with length Lx and width Ly, and zero-flux boundary conditions so we have

$$\vec{q}(x,y) = -k(x,y)\vec{\nabla}p(x,y), \tag{1}$$

$$\nabla\vec{q}(x,y) = f(x,y), \tag{2}$$

$$\vec{q}(x,y)\bullet\vec{n}_{\Gamma} = 0, \tag{3}$$

where $\vec{q}$ is the speed, k is the permeability coefficient, p is the pressure, f is the source function, $\vec{n}$ is the normal vector at the boundary Γ, x and y represent an spatial coordinates being, $0 \le x \le L_x$, and $0 \le y \le L_y$.

To obtain a unique solution of the equation system in (1-3), has imposed a normalization condition in the distribution of pressure fields, in the form

$$\iint_D p(x,y) = 0. \tag{4}$$

The equations (1-3) are discretized in conventional square finite elements of Raviart-Thomas (Ω_{ij}) of size *h*. Inside the element Ω_{ij}, the permeability *k* and source function *f*, are constant. The geometry and relevant quantities of each element are illustrated in figure 1.

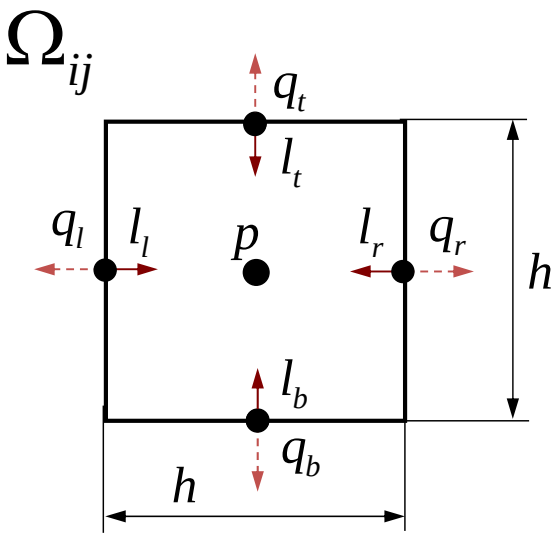

Figure 1: Finite element Ω_{ij} and relevant quantities

Considering a Raviart-Thomas discretization [4] and the domain decomposition method [6] the equation system (1-2) for each inner element of domain *D* would be written as

$$q_r + q_t + q_l + q_b = fh\,, \tag{5}$$

$$q_\alpha = -\frac{2k}{h}\left(l_\alpha - p\right), \qquad \alpha = r,\ t,\ l,\ b\ ; \tag{6}$$

$$l_\alpha = \beta_{\alpha,\alpha'}\left(q_\alpha + q_{\alpha'}\right) + l_{\alpha'}\ , \quad (\alpha,\alpha') = (t,b),(r,l),(b,t)(l,r)\,, \tag{7}$$

where $\{r,\ t,\ l,\ b\}$ represents each edge of the element (t = top, l = left, b = bottom and r = right), l_α are the Lagrange multipliers, and $\beta_{\alpha,\alpha'}$ are the coefficients of the Robin condition. The equations (5-7) represent a linear algebraic system of nine equations in nine unknowns for each inner element. For the boundary elements, equations (6) and (7), must be replaced for the corresponding boundary condition. The resulting algebraic equations and normalization condition (4) allow the Gauss-Seidel´s iterative scheme implementation [9] to solve the problem. In [6] is possible to find a detailed description of the problem´s numerical formulation.

3 GPU-CUDA IMPLEMENTATION

The computational algorithm for the numerical implementation of the single phase flow problem in porous media is shown in figure 2. This algorithm is divided into four stages: (i) the input data reading, (ii) the preliminary calculations, (iii) the iterative process and (iv) the data output. Of these stages, the iterative process (in orange in the figure 2) is the step with the highest computational cost being relevant the parallelization using CUDA-GPU model.

Input data
Read initial data
Memory allocation
Preliminary calculations
Auxiliary parameters in spatial discretization
Computing Robin coefficients $\beta_{\alpha,\alpha'}$
Auxiliary calculations for iterative process
Iterative process
Initial approximation or update for q_α , p , l_α
Identify elements (inner or boundary)
Compute q_α and p values
Update Lagrange multiplicators l_α
Compute normalization condition
Verify convergence criteria
Print results
Print q_α and p values

Figure 2: Computational algorithm.

The kernel configuration used in parallel code execution has the necessary number of threads to support the number of elements used in the discretization. For example, in a 32 × 32 mesh the kernel will be configured with 1024 threads. So, each thread is responsible for the processing of an element. In the first iteration the thread assigns the initial approximation and identify the element type. The element can be internal or one of the edge type element: top edge, left edge, bottom edge, right edge, top left corner, top right corner, bottom left corner or bottom right corner. For each element type, the discretized equations (5-7), to be computed by the thread are different. Then the thread calculates the velocity in each edge, the pressures and Lagrange multipliers. The normalization condition´s calculation and the convergence criteria tests are made in a reduction operation. The iterative process is repeated until the maximum relative deviation of pressure distribution between two consecutive iterations for any mesh element does not exceed the limit value.

4 RESULTS AND DISCUSSION

Were implemented four codes for the problem solution (i) a sequential code (ii) an MPI code [7], (iii) a hybrid code (combining MPI-OpenMP) [8] and (iv) a CPU-CUDA code. The major difficulty in the code parallelization was the speed dependencies of the neighbor elements in the pressure calculation. Were made performance tests for various mesh sizes in CPU and GPU architectures. Details of the hardware present in the parallel machines used (CPU and GPU architectures) are offered in table 1. The GPU workstation that appears in column 3 table 1 includes an Nvidia graphic card. Were made numerical experiments with three different Nvidia graphic cards. The technical specifications of GPU cards used appear in table 2.

Table 1: Hardware specifications of parallel machines

Parallel machine	**CPU Cluster**	**GPU Workstation**
Node number	8	1
Core number	2	4
Processor	Core 2 Duo E6550	Core™ i7-2600
Processor clock (GHz)	2.3	3.4
Cachê memory (MB)	4	8
RAM memory (GB)	2	8
GPU Card	No	Yes

Table 2: Specifications of GPU cards

Specification	**GPU card 1**	**GPU card 2**	**GPU card 3**
Name	GeForce GT 520	GeForce 9800 GT	GeForce GTX 560 Ti
GPU cores	48	112	384
GPU memory (MB)	1024	512	1024
Graphics clock (MB)	810	600	822
Processor clock (MHz)	1620	1500	1645
Memory interface	64 bits DDR3	256 bits DDR3	256 bits DDR3
CUDA version	2.1	1.1	2.1

For meshes of 64, 128, 256, 512 and 1024 elements in each spatial direction were executed numerical experiments. The MPI and hybrid implementations were executed in the cluster, and the CUDA implementation was executed on the workstations with GPU for each graphic card available. The bigger mesh size used was limited by the available memory on the GPU card, in this case the 512 MB available in the accelerator card 2. Each numerical experiment runs one code, five times and collects the execution time. The outliers were neglected and with another 3 remaining values we compute the arithmetic mean of execution time. Using this value we calculate the speedup of each implementation in each mesh. The results are shown in the figure 3.

In figure 3 we can observe that the CPU-based implementations reaches a relatively low speedup. The speedup order is approximately the number of nodes available in the parallel machine. The use of hybrid implementation does not increase significantly, the speedup compared to MPI implementation. This is due to the strong influence of communication time in the total execution time. The CUDA implementation supported by the GPU, even using domestic GPUs, reaches high speedups (between 16 and 70, for mesh with 1024 elements). The speedup values increase with the number of GPU cores. This occurs, because, as higher is the core number, higher will be the number of elements (threads) that can be calculated simultaneously.

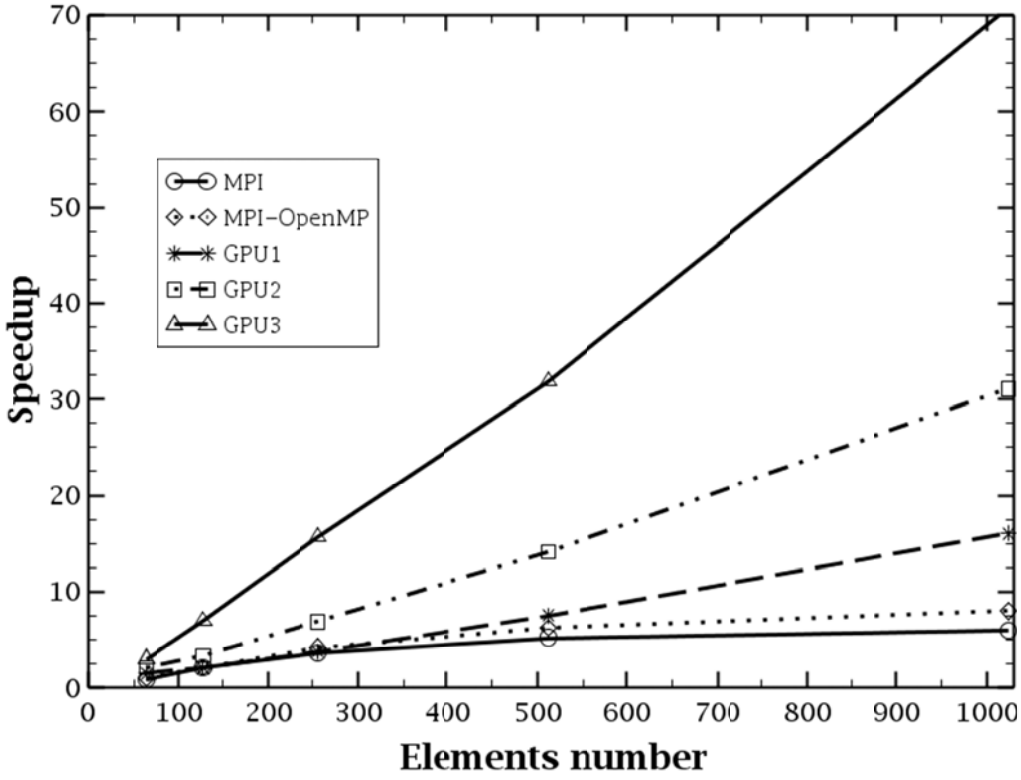

Figure 3: Speedup values for MPI, hybrid (MPI-OpenMP) and CUDA implementations

5 CONCLUSIONS AND FUTURE WORKS

In this work we use the Raviart-Thomas finite element discretization and a domain decomposition method to numerically solve single phase flow problems in porous medium. The solution method is parallelized using the GPU-CUDA model. This solution allows the Nvidia graphic card use in scientific calculus applications. The speedup values were obtained and compared with MPI and MPI-OpenMP implementations in CPU architectures.

The results show that the speedup reached by the CUDA implementation are much better than the values offered by the CPU-based implementations. This result demonstrates the great potential and low cost of the parallel techniques based on GPU-CUDA to obtain a numerical solution of the single phase flow in porous mediums

The future developments of this research guides us to obtain a hybrid MPI-CUDA implementation. In this implementation, we can use a cluster, where each node is a GPU workstation. We hope that this implementation achieves highest speedups and the limitations of the mesh size forced by the GPU memory are mitigated.

ACNOWLEDGMENTS

This work was partially supported by Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB), Brazil. The authors appreciate the computational support offered by Centro de Armazenamento de dados e Computação Avançada da Uesc (CACAU).

REFERENCES

[1] Pejman Tahmasebi, Muhammad Sahimi, Gregoire Mariethoz and Ardeshir Hezarkhani. Accelerating geostatistical simulations using graphics processing units (GPU*).Computers & Geosciences* (2012) **46:**51-59.

[2] Xianke Meng, Zhongning Sun and Guangzhan Xu. Single-phase convection heat transfer characteristics of pebble-bed channels with internal heat generation. *Nuclear Engineering and Design* (2012) **252**:121-127.

[3] M. Papadrakakis, G. Stavroulakis and A. Karatarakis. A new era in scientific computing: Domain decomposition methods in hybrid CPU–GPU architectures. *Comput. Methods Appl. Mech. Engrg.* (2011) **200**:1490–1508.

[4] P.A. Raviart, J.M. Thomas. A mixed finite element method for 2nd order elliptic problems. *Lecture Notes in Math.* (1977) **606**:292-315.

[5] Yanping Chena, Zuliang Lub and Yunqing Huang. Superconvergence of triangular Raviart–Thomas mixed finite element methods for a bilinear constrained optimal control problem. *Computers & Mathematics with Applications* (2013) **66**:1498-1513.

[6] J. J. Douglas, F. Pereira, L. M. Yeh and P. J. P. Leme. Domain decomposition for immiscible displacement in single porosity systems. *Lecture Notes in Pure and Applied Mathematics* (1994) **164**:191-199.

[7] Esbel T. V. Orellana, Dany S. Dominguez and Adriano M. dos Santos. Utilização de MPI e Open-MP para Resolução da Equação de Poisson em Arquiteturas Multicore. *Proceedings of XII Encontro de Modelagem Computacional (EMC 2009)*, (2009) Rio de Janeiro, Brazil (in portuguese).

[8] Esbel T. V. Orellana, Dany S. Dominguez,Adriano M. dos Santos and Susana M. Iglesias. Paradigma Híbrido MPI-OpenMP na Resolução da Equação de Poisson em Arquiteturas Multicore. *Proceedings of 30 Congresso Ibero-Latino-Americano de Métodos Computacionais em Engenharia (CILAMCE 2009)*, (2009) Armação de Búzios, Brazil (in portuguese).

[9] J. D. Hoffman. *Numerical Methods for Engineers and Scientists*. 2nd Edition, Marcel Deker, (2001).