UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

Efficient Solution of Heat and Mass Transfer Equations using Programmable General Purpose Graphical Processing Unit under Natural Boundary Conditions in Capillary Porous Solid and Hollow Cylinder

Hira Narang[1], Fan Wu[2], Abdul Rafae Mohammed[3]

Professor & Head, Department of Computer Science, Tuskegee University, Alabama, USA[1]

Professor, Department of Computer Science, Tuskegee University, Alabama, USA[2]

Graduate Research Student, Department of Computer Science, Tuskegee University, Alabama, USA[3]

Abstract: With the recent developments in computing technology, increased efforts have gone into simulation of various scientific methods and phenomenon in engineering fields. One such case is the simulation of heat and mass transfer equations which is becoming more and more important in analyzing various scenarios in engineering applications. Analyzing the heat and mass transfer phenomenon in a thermal environment requires us to simulate it. However, this process of numerical solution of heat and mass transfer equation is very much time consuming. Therefore, this paper aims at utilizing one of the acceleration techniques developed in the graphics community that exploits a graphics processing unit (GPU) which is applied to the numerical solutions of heat and mass transfer equations. The nVidia Compute Unified Device Architecture (CUDA) programming model caters a good method of applying parallel computing to program the graphical processing unit. This paper shows a good improvement in the performance while solving the heat and mass transfer equations for solid and hollow capillary porous cylinder with the Natural boundary conditions and initial conditions numerically running on GPU. This heat and mass transfer simulation is implemented using CUDA platform on nVidia Quadro FX 4800 graphics card. Our experimental results depict the drastic performance improvement when GPU is used to perform heat and mass transfer simulation. GPU can significantly accelerate the performance with a maximum observed speedup of more than 8 fold. Therefore, the GPU is a good approach to accelerate the heat and mass transfer simulation.

Keywords: Numerical Solution; Heat and Mass Transfer; General Purpose Graphical Processing Unit; CUDA.

I. INTRODUCTION

During the last half century, many scientists and engineers working in Heat and Mass Transfer processes have put lots of efforts in finding solutions both analytically/numerically, and experimentally. To precisely analyze physical behaviors of heat and mass environment, to simulate several heat and mass transfer phenomena such as heat conduction, convection, and radiation are very important.

A heat transfer simulation is accomplished by utilizing parallel computer resources to simulate such heat and mass transfer phenomena. With the helps from computer, initially the sequential solutions were found, and later when high-end computers became available, fast solutions were obtained to heat and mass transfer problems.

However, the heat and mass transfer simulation requires much more computing resources than the other simulations. Therefore, acceleration of this simulation is very essential to implement a practical big data size heat and mass transfer simulation.

This paper utilizes the parallel computing power of GPUs to speed up the heat and mass transfer simulation. GPUs are very efficient considering theoretical peak floating-point operation rates [1]. Therefore, comparing with super-computer, GPUs is a powerful co-processor on a common PC which is ready to simulate a large-scale heat and mass transfer at a less resources. The GPU has several advantages over CPU architectures, such as highly parallel, computation intensive workloads, including higher bandwidth, higher floating-point throughput. The GPU can be an attractive alternative to clusters or super-computer in high performance computing areas.

UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

CUDA [2] by nVidia already proved its effort to develop both programming and memory models. CUDA is a new parallel, C-like language programming Application program interface (API), which bypasses the rendering interface and avoids the difficulties from using GPGPU. Parallel computations are expressed as general-purpose, C-like language kernels operating in parallel over all the points in a application.

This paper develops the numerical solutions to Two-point Initial-Boundary Value Problems (TIBVP) of Heat and Mass with the first boundary and initial conditions in capillary porous cylinder. These problems can be found some applications in drying processes, space science, absorption of nutrients, transpiration cooling of space vehicles at re-entry phase, and many other scientific and engineering problems.

Although some traditional approaches of parallel processing to the solutions of some of these problems have been investigated, no one seems to have explored the high performance computing solutions to heat and mass transfer problems with compact multi-processing capabilities of GPU, which integrates multi-processors on a chip. With the advantages of this compact technology, we developed algorithms to find the solution of TIBVP with the first boundary and initial conditions and compare with some existing solutions to the same problems. All of our experimental results show significant performance speedups. The maximum observed speedups are about 10 times.

The rest of the paper is organized as follow: Section II briefly introduces some closely related work; Section III describes the basic information on GPU and CUDA; Section IV presents the mathematical model of heat and mass transfer and numerical solutions to heat and mass transfer equations; Section V presents our experimental results; And Section VI concludes this paper and give some possible future work directions.

II. RELATED WORK

The simulation of heat and mass transfer has been a very hot topic for many years. And there is lots of work related to this field, such as fluid and air flow simulation. We just refer to some most recent work close to this field here.

Soviet Union was in the fore-front for exploring the coupled Heat and Mass Transfer in media, and major advances were made at Heat and Mass Transfer Institute at Minsk, BSSR. Later England and India took the lead and made further contributions for analytical and numerical solutions to certain problems.

Narang [4-9] explored the wavelet solutions to heat and mass transfer equations and Ambethkar [10] explored the numerical solutions to some of these problems.

Krüger et al. [11] computed the basic linear algebra problems with the feathers of programmability of fragments on GPU, and further computed the 2D wavelets equations and NSEs on GPU. Bolz et al. [12] matched the sparse matrix into textures on GPU, and utilized the multigrid method to solve the fluid problem. In the meantime, Goodnight et al. [13] used the multigrid method to solve the boundary value problems on GPU. Harris [14, 15] solved the PDEs of dynamic fluid motion to get cloud animation.

GPU is also used to solve other kinds of PDEs by other researchers. Kim et al. [16] solved the crystal formation equations on GPU. Lefohn et al. [17] matched the level-set iso surface data into a dynamic sparse texture format.

Another creative usage was to pack the information of the next active tiles into a vector message, which was used to control the vertices and texture coordinates needed to send from CPU to GPU. To learn more applications about general-purpose computations GPU, more information can be found from here [18].

III. AN OVERVIEW OF CUDA ARCHITECTURE

The GPU that we have used in our implementations is nVidia's Quadro FX 4800, which is DirectX 10 compliant. It is one of nVidia's fastest processors that support the CUDA API and as such all implementations using this API are forward compatible with newer CUDA compliant devices.

All CUDA compatible devices support 32-bit integer processing. An important consideration for GPU performance is its level of occupancy. Occupancy refers to the number of threads available for execution at any one time. It is normally desirable to have a high level of occupancy as it facilitates the hiding of memory latency.

The GPU memory architecture is shown in figure 1.

UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

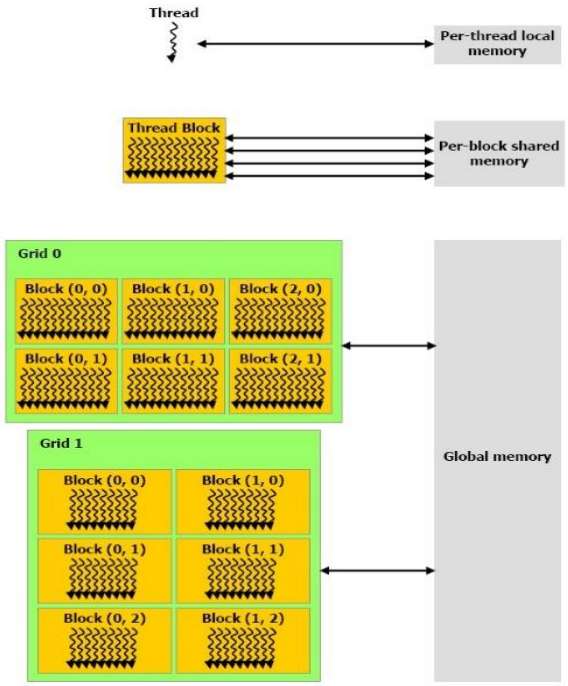

Figure 1: CUDA Memory Architecture

IV. MATHEMATICAL MODEL AND NUMERICAL SOLUTIONS OF HEAT AND MASS TRANSFER

A. Mathematical Model

Consider the Heat and Mass Transfer through a capillary porous solid and hollow cylinder with Natural boundary conditions. Let the z-axis be directed upward along the capillary porous solid and hollow cylinder and the r-axis radius of the capillary porous solid and hollow cylinder. Let u and v be the velocity components along the z- and r- axes respectively. Then the heat and mass transfer equations in the Boussinesq's approximation, are:

$$\frac{\partial T}{\partial t} = k_1 \left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2}\right) + k_2 \left(\frac{\partial C}{\partial t}\right) \quad (1)$$

$$\frac{\partial C}{\partial t} = k_3 \left(\frac{\partial^2 C}{\partial r^2} + \frac{1}{r}\frac{\partial C}{\partial r} + \frac{\partial^2 C}{\partial z^2}\right) + k_4 \left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2}\right) \quad (2)$$

$0 < z < ¥, a < r < b^*, t > 0$

$^*a = 0, b = ¥$ for capillary porous cylinder.

Initial Conditions:

$$T(r, z, 0) = 0$$
$$C(r, z, 0) = 1 \quad (3)$$

Boundary Conditions:

For Solid Cylinder:

$$K_1(T(r, 0, t) - T_\infty) = k_h \cdot \frac{\partial T}{\partial r}$$
$$K_2(C(r, 0, t) - C_\infty) = k_m \cdot \frac{\partial C}{\partial r} \quad (4)$$

$$K_1(T(1, z, t) - T_\infty) = k_h \cdot \frac{\partial T}{\partial r}$$
$$K_2(C(1, z, t) - C_\infty) = k_m \cdot \frac{\partial C}{\partial r} \quad (5)$$
$$T(r, \infty, t) = T_\infty = 0$$
$$C(r, \infty, o) = C_\infty = 1 \quad (6)$$

UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

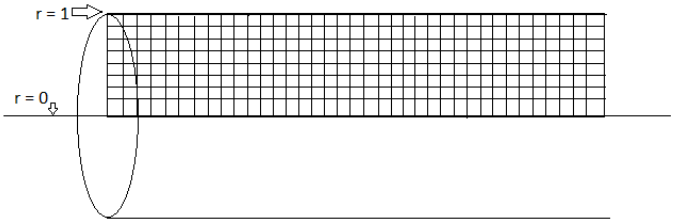

Fig 2: Solid Cylinder with grid

For Hollow Cylinder:
Consider the Hollow cavity in the cylinder has a radius of k.

$$K_1(T(r,0,t) - T_\infty) = k_h.\frac{\partial T}{\partial r}$$
$$K_2(C(r,0,t) - C_\infty) = k_m.\frac{\partial C}{\partial r} \quad (4a)$$

At r = 1:

$$K_1(T(1,z,t) - T_\infty) = k_h.\frac{\partial T}{\partial r}$$
$$K_2(C(1,z,t) - C_\infty) = k_m.\frac{\partial C}{\partial r} \quad (5a)$$

At r = k:

$$K_1(T(k,z,t) - T_\infty) = k_h.\frac{\partial T}{\partial r}$$
$$K_2(C(k,z,t) - C_\infty) = k_m.\frac{\partial C}{\partial r} \quad (5b)$$

$$T(r,\infty,t) = T_\infty = 0$$
$$C(r,\infty,o) = C_\infty = 1 \quad (6a)$$

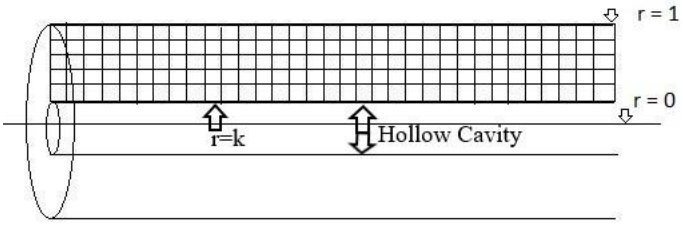

Fig 3: Hollow Cylinder with Grid

Since the solid and hollow cylinder is assumed to be capillary porous, μ_1 is the velocity of the fluid, T_p the temperature of the fluid near the capillary porous solid and hollow cylinder, T_∞ the temperature of the fluid far away from the capillary porous solid and hollow cylinder, C_p the concentration near the capillary porous solid and hollow cylinder, C_∞ the concentration far away from the capillary porous solid and hollow cylinder, g the acceleration due to gravity, β the coefficient of volume expansion for heat transfer, β' the coefficient of volume expansion for concentration, ν the kinematic viscosity, σ the scalar electrical conductivity, ω the frequency of oscillation, k the thermal conductivity.

From Equation (1) we observe From Equation (1) we observe that v_1 is independent of space co-ordinates and may be taken as constant. We define the following non-dimensional variables and parameters.

$$t = \frac{t_1 V_0^2}{4\nu}, z = \frac{V_0 z_1}{4\nu} \quad (7)$$

UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

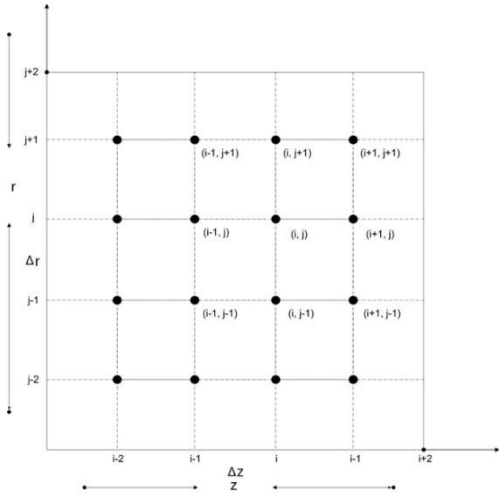

Figure 4: Finite Difference Grid for solid Cylinder

$$u=\frac{u_1}{V_0},T=\frac{T_1-T_\infty}{T_P-T_\infty},C=\frac{C_1-C_\infty}{C_P-C_\infty},P_r=\frac{v}{k},S_c=\frac{v}{D'}$$

$$M=\frac{\sigma B_0^2 v}{\rho V_0^2},G_r=\frac{vg\beta(T_P-T_\infty)}{V_0^3}$$

$$G_m=\frac{vg\beta'(C_P-C_\infty)}{V_0^3},\omega=\frac{4v\omega_i}{V_0^2}$$

Now taking into account Equations (5), (6) and (7), equations (1) and (2) reduce to the following form:

$$\frac{\P T}{\P t}+\frac{\P^2 T}{\P r^2}-4\frac{\P C}{\P t}+\frac{1}{r}\frac{\P T}{\P r}=\frac{4}{P_r}\frac{\P^2 T}{\P z^2} \quad (8)$$

$$\frac{\P C}{\P t}+\frac{\P^2 C}{\P r^2}-4\frac{\P T}{\P t}+\frac{1}{r}\frac{\P C}{\P r}=\frac{4}{P_r}\frac{\P^2 C}{\P z^2} \quad (9)$$

B. Numerical Solutions

Here we sought a solution by finite difference technique of implicit type namely Crank-Nicolson implicit finite difference method which is always convergent and stable. This method has been used to solve Equations (8), and (9) subject to the conditions given by (4), (5) and (6). To obtain the difference equations, the region of the heat is divided into a gird or mesh of lines parallel to z and r axes. Solutions of difference equations are obtained at the intersection of these mesh lines called nodes. The values of the dependent variables T, and C at the nodal points along the plane $x=0$ are given by $T(0,t)$ and $C(0,t)$ hence are known from the boundary conditions.

In the figure 2, Δz, Δr are constant mesh sizes along z and r directions respectively. We need an algorithm to find single values at next time level in terms of known values at an earlier time level. A forward difference approximation for the first order partial derivatives of T and C. And a central difference approximation for the second order partial derivative of T and C are used.

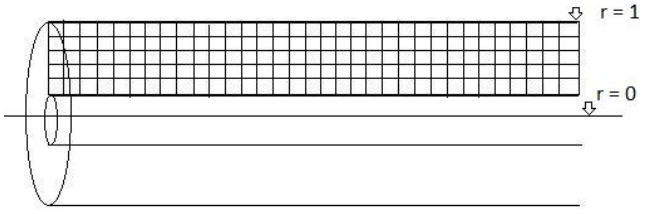

Figure 5 : Position of Grid in Hollow Cylinder

UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

In case of the Solid cylinder the grid in consideration both the inner axis at centre of cylinder and outer surface boundary of the Solid Cylinder i.e. between the axis which passes through the centre i.e. at r=0 and the outer surface of the cylinder. Whereas in case of Hollow Cylinder the grid in consideration both the inner surface boundary and outer surface boundary of the hollow cylinder, hence we apply the boundary conditions on both the inner boundary and outer boundary of the hollow cylinder i.e. the top and bottom boundary of the grid. The position of the grid in both the hollow and solid capillary porous cylinder can be described as below in figure and respectively:

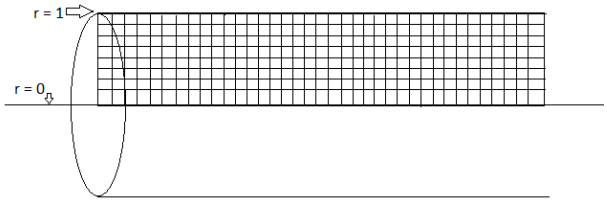

Figure 6: Position of Grid in Solid Cylinder

On introducing finite difference approximations for:
For the purposes of coming up with a numerical solution for the problem, the radius of the capillary porous solid and hollow cylinder is 1.0

$$\left(\frac{\partial^2 T}{\partial z^2}\right)_{i,j} = \frac{T_{i+1,j} - T_{i-1,j} + T_{i+1,j+1} - T_{i-1,j+1} - 2T_{i,j}}{2(\Delta z)^2}$$

$$\left(\frac{\partial^2 T}{\partial r^2}\right)_{i,j} = \frac{T_{i+1,j} - T_{i-1,j} + T_{i+1,j+1} - T_{i-1,j+1} - 2T_{i,j}}{2(\Delta r)^2}$$

$$\left(\frac{\partial T}{\partial r}\right)_{i,j} = \frac{T_{i+1,j} - T_{i-1,j} + T_{i+1,j+1} - T_{i-1,j+1}}{4(\Delta r)}$$

$$\left(\frac{\partial T}{\partial t}\right)_{i,j} = \frac{T_{i,j+1} - T_{i,j}}{\Delta t}, \left(\frac{\partial C}{\partial t}\right)_{i,j} = \frac{C_{i,j+1} - C_{i,j}}{\Delta t}, \left(\frac{\partial u}{\partial t}\right)_{i,j} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t}$$

$$\left(\frac{\partial C}{\partial t}\right)_{i,j} = \frac{C_{i+1,j} - C_{i-1,j} + C_{i+1,j+1} - C_{i-1,j+1}}{4(\Delta t)}$$

$$\left(\frac{\partial^2 C}{\partial z^2}\right)_{i,j} = \frac{C_{i+1,j} - C_{i-1,j} + C_{i+1,j+1} - C_{i-1,j+1} - 2C_{i,j}}{2(\Delta z)^2}$$

$$\left(\frac{\partial^2 C}{\partial r^2}\right)_{i,j} = \frac{C_{i+1,j} - C_{i-1,j} + C_{i+1,j+1} - C_{i-1,j+1}}{2(\Delta r)^2}$$

$$\left(\frac{\partial C}{\partial r}\right)_{i,j} = \frac{C_{i+1,j} - C_{i-1,j} + C_{i+1,j+1} - C_{i-1,j+1}}{4(\Delta r)}$$

The finite difference approximation of Equations (8) and (9) are obtained with substituting Equation into Equations (8) and (9) and multiplying both sides by Δt and after simplifying, we let $\frac{Dt}{(Dz)^2} = r' = 1$ (method is always stable and convergent), under this condition the above equations can be written as:

$$\frac{\partial C}{\partial t} = \frac{1}{2}\left(\frac{U+V-2(T_{i,j}+C_{i,j})}{(\Delta r)^2} + \frac{U+V}{2r(\Delta r)} + \left(\frac{U+V-2(T_{i,j}+C_{i,j})}{(\Delta z)^2}\right)\right.$$

$$\frac{\partial T}{\partial t} = \frac{1}{2}\left(\frac{2U+V-2(2T_{i,j}+C_{i,j})}{(\Delta r)^2} + \frac{2U+V}{2r(\Delta r)} + \frac{2U+V-2(2T_{i,j}+C_{i,j})}{(\Delta z)^2}\right)$$

UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

Where U = $T_{i+1,j} - T_{i-1,j} + T_{i+1,j+1} - T_{i-1,j+1}$

Where V = $C_{i+1,j} - C_{i-1,j} + C_{i+1,j+1} - C_{i-1,j+1}$

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Setup and Device Configuration

The experiment was executed using the CUDA Runtime Library, Quadro FX 4800 graphics card, Intel Core 2 Duo. The programming interface used was Visual Studio.

The experiments were performed using a 64-bit Lenovo ThinkStation D20 with an Intel Xeon CPU E5520 with processor speed of 2.27 GHZ and physical RAM of 4.00GB. The Graphics Processing Unit (GPU) used was an NVIDIA Quadro FX 4800 with the following specifications:

CUDA Driver Version:	3.0
Total amount of global memory:	1.59 Gbytes
Number of multiprocessors:	24
Number of cores:	92
Total amount of constant memory:	65536 bytes
Total amount of shared memory per block:	16384 bytes
Total number of registers available per block:	16384
Maximum number of threads per block:	512

Bandwidth:
Host to Device Bandwidth: 3412.1 (MB/s)
Device to Host Bandwidth: 3189.4 (MB/s)
Device to Device Bandwidth: 57509.6 (MB/s)

In the experiments, we considered solving heat and mass transfer differential equations in capillary porous solid and hollow cylinder with Natural boundary conditions using numerical methods. Our main purpose here was to obtain numerical solutions for Temperature T, and concentration C distributions across the various points in a capillary porous solid and hollow cylinder as heat and mass are transferred from one end of the capillary porous solid and hollow cylinder to the other. For our experiment, we compared the similarity of the CPU and GPU results. We also compared the performance of the CPU and GPU in terms of processing times of these results.

In the experimental setup, we are given the initial temperature T_0 and concentration C_0 at point z = 0 on the capillary porous cylinder. Also, there is a constant temperature and concentration N_0 constantly working the surface of the capillary porous cylinder. The temperature at the other end of the capillary porous cylinder where $z = \infty$ is assumed to be ambient temperature (assumed to be zero). Also, the concentration at the other end of the capillary porous cylinder where $z = \infty$ is assumed to be negligible (≈ 0). Our initial problem was to derive the temperature T_1 and concentration C_1 associated with the initial temperature and concentration respectively. We did this by employing the finite difference technique. Hence, we obtained total initial temperature of $(T_0 + T_1)$ and total initial concentration of $(C_0 + C_1)$ at z = 0. These total initial conditions were then used to perform calculations.

For the purpose of implementation, we assumed a fixed length of the capillary porous cylinder and varied the number of nodal points N to be determined in the capillary porous cylinder. Since N is inversely proportional to the step size Δz, increasing N decreases Δz and therefore more accurate results are obtained with larger values of N. For easy implementation in Visual Studio, we employed the Forward Euler Method (FEM) for forward calculation of the temperature and concentration distributions at each nodal point in both the CPU and GPU. For a given array of size N, the nodal points are calculated iteratively until the values of temperature and concentration become stable. In this experiment, we performed the iteration for 10 different time steps. After the tenth step, the values of the temperature and concentration became stable and are recorded. We run the tests for several different values of N and Δz and the error between the GPU and CPU calculated results were increasingly smaller as N increased. Finally, our results were normalized in both the GPU and CPU.

B. Experimental Results

The normalized temperature and concentration distributions at various points in the capillary porous solid cylinder are depicted in Table 1 and Table 2 whereas the temperature and concentration distribution at different points of a hollow cylinder are depicted in Table 3 and Table 4 respectively. We can immediately see that, at each point in the capillary porous solid and hollow cylinders, the CPU and GPU computed results are similar.

UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

In addition, the value of temperature is highest and the value of concentration is lowest at the point on the capillary porous solid and hollow cylinder where the heat resource and mass resource are constantly applied. As we move away from this point, the values of the temperature decrease and concentration increase. At a point near the designated end of the capillary porous hollow cylinder, the values of the temperature approach zero and concentration approach one.
Figure 7a and Figure 7b Shows the temperature and concentration distribution in the capillary porous solid cylinder with 4 different radiuses.

TABLE 1. COMPARISON OF GPU AND CPU RESULTS FOR CAPILLARY POROUS CYLINDER (CONCENTRATION)

Z	CPU RESULTS	GPU RESULTS
1	0.014427618	0.012206036
2	0.015767461	0.01220366
3	0.015781525	0.01220304
4	0.015785111	0.012202833
5	0.015790976	0.012202833
6	0.015795956	0.012202833
7	0.015799412	0.012202833
8	0.015831817	0.012202833
9	0.015855711	0.012202833
10	0.015735704	0.012202833
11	0.062063104	0.029042556
12	0.16301576	0.103385715
13	0.270609296	0.194035479
14	0.475217373	0.392405129
15	1	1

TABLE 2. COMPARISON OF GPU AND CPU RESULTS FOR CAPILLARY POROUS CYLINDER (TEMPERATURE)

Z	CPU RESULTS	GPU RESULTS
1	1	1
2	0.47679379	0.44083877
3	0.1675914	0.14347097
4	0.0725641	0.03819127
5	0.03429739	0.00212915
6	0.02349273	0.00071069
7	0.02032908	0.00030919
8	0.02032908	0.00030919
9	0.01721083	0.00019713
10	0.01720659	0.00016623
11	0.01720545	0.0001578
12	0.01720525	0.00015553
13	0.01720514	0.00015492
14	0.01792388	0.00015476
15	0.01792388	0.00015476

TABLE 3. COMPARISON OF GPU AND CPU RESULTS FOR CAPILLARY POROUS HOLLOW CYLINDER (CONCENTRATION)

Z	CPU RESULTS	GPU RESULTS
1	0.014693094	0.014099094
2	0.01478963	0.014102179
3	0.014854123	0.014131103
4	0.014889632	0.01415243
5	0.014802567	0.014045314
6	0.015069344	0.055396047
7	0.148974568	0.145503982
8	0.297854123	0.271468762
9	0.437584296	0.413414192
10	1	1

UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

TABLE 4. COMPARISON OF GPU AND CPU RESULTS FOR CAPILLARY POROUS HOLLOW CYLINDER (TEMPERATURE)

Z	CPU RESULTS	GPU RESULTS
1	1	1
2	0.50145236	0.46394996
3	0.18457896	0.17794177
4	0.08032568	0.0756855
5	0.03365478	0.03477666
6	0.02458896	0.02382101
7	0.02145236	0.02061316
8	0.02114526	0.02061316
9	0.01871425	0.01745133
10	0.01736547	0.01744703

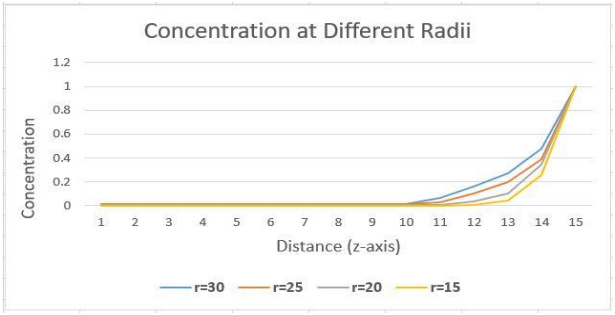

Figure 7a: Temperature Curve of Solid Cylinder with Natural boundary conditions

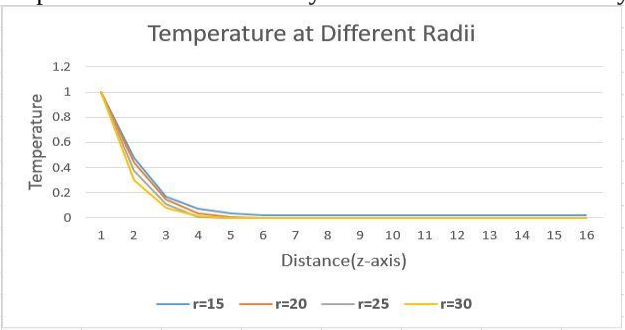

Figure 7b: Concentration Curve of Solid Cylinder with Natural Boundary Conditions

Figure 8a and Figure 8b Shows the temperature and concentration distribution in the capillary porous hollow cylinder with 4 different radiuses.

Furthermore, we also evaluated the performance of the GPU (NVIDIA Quadro FX 4800) in terms of solving heat and mass transfer equations by comparing its execution time to that of the CPU (Intel Xeon E5520).

For the purpose of measuring the execution time, the same functions were implemented in both the device (GPU) and the host (CPU), to initialize the temperature and concentration and to compute the numerical solutions. In this case, we measured the processing time for different values of N. The graph in Figure 5 depicts the performance of the GPU versus the CPU in terms of the processing time. We run the test for N running from 10 to 599 with increments of 30 and generally, the GPU performed the calculations a lot faster than the CPU.

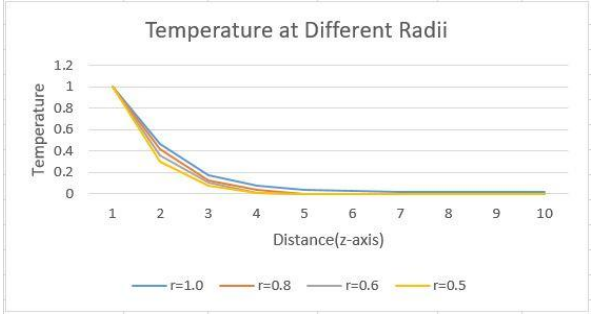

Figure 8a: Temperature Curve of Solid Cylinder with Natural Boundary Conditions

UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

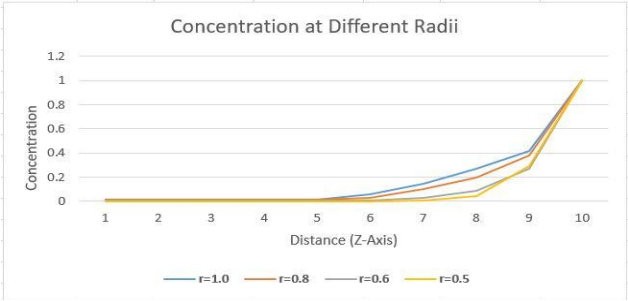

Figure 8b: Concentration Curve of Solid Cylinder with Natural Boundary Conditions

- When N was smaller than 16, the CPU performed the calculations faster than the GPU.
- For N larger than 16 the GPU performance began to increase considerably

Figure 9a and 9b show some of our experimental results for both capillary porous cylinder.

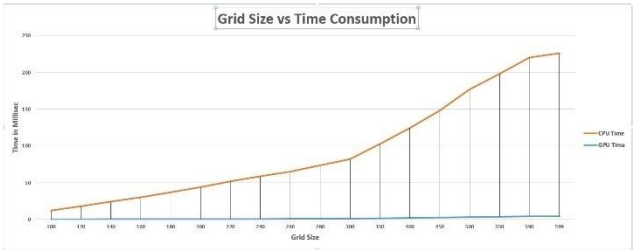

Figure 9a: Performance of GPU and CPU Implementations for capillary porous cylinder

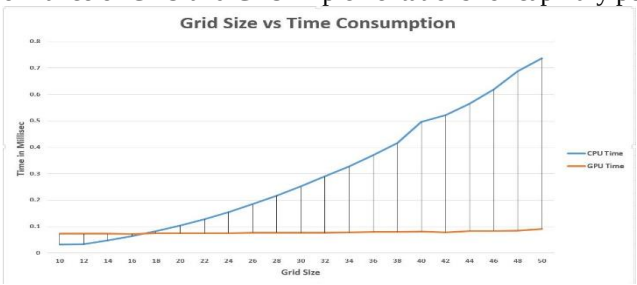

Figure 9b: Performance of GPU and CPU Implementations for capillary porous cylinder with incremental number of nodes

Figure 10a and 10b show some of our experimental results for both capillary porous hollow cylinder.

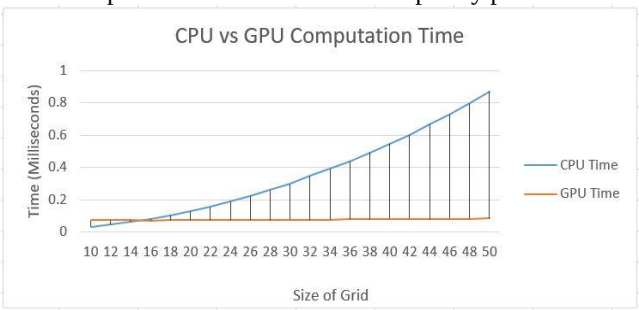

Figure 10a: Performance of GPU and CPU Implementations for capillary porous hollow cylinder

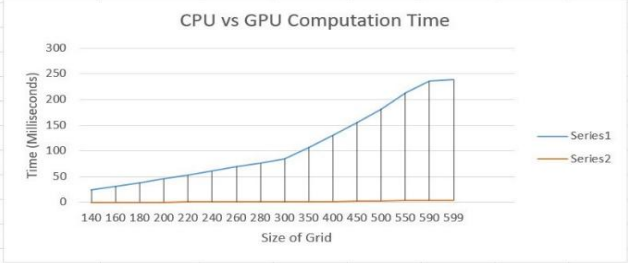

Figure 10b: Performance of GPU and CPU Implementations for capillary porous hollow cylinder with incremental number of nodes

UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

Finally, the accuracy of our numerical solution was dependent on the number of iterations we performed in calculating each nodal point, where more iteration means more accurate results. In our experiment, we observed that after 9 or 10 iterations, the solution to the heat and mass equation at a given point became stable. For optimal performance, and to keep the number of iterations the same for both CPU and GPU, we used 10 iterations and experimental results for capillary porous solid and hollow cylinder show about 8 times speed-up in the simulation.

VI. CONCLUSION AND FUTURE WORK

We have presented our numerical approximations to the solution of the heat and mass transfer equation with the Natural boundary conditions and initial conditions for capillary porous solid and hollow cylinder using finite difference method on GPGPUs. Our conclusion shows that finite difference method is well suited for parallel programming. We implemented numerical solutions utilizing highly parallel computations capability of GPGPU on nVidia CUDA. We have demonstrated GPU can perform significantly faster than CPU in the field of numerical solution to heat and mass transfer. Experimental results for capillary porous solid and hollow cylinder indicate that our GPU-based implementation shows a significant performance improvement over CPU-based implementation and the maximum observed speedups are about 8 times.

There are several avenues for future work. We would like to test our algorithm on different GPUs and explore the new performance opportunities offered by newer generations of GPUs. It would also be interesting to explore more tests with large-scale data set. Finally, further attempts will be made to explore more complicated problems both in terms of boundary and initial conditions as well as other geometry. We also aim to simulate this equation for a composite material i.e. composed of materials with different conductivities.

REFERENCES

[1] J.D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krger, A.E. Lefohn, T.J. Purcell,: A survey of general-purpose computation on graphics hardware. Computer Graphics Forum 26(1) (2007) 80-113.
[2] NVIDIA Corporation. NVIDIA Programming Guide 2.3. Retrieved July, 2009. www.nvidia.com.
[3] A. V. Luikov. Heat and Mass Transfer in Capillary Porous Bodies, Pergamon Press, 1966.
[4] Hira Narang and Rajiv Nekkanti. Wavelet-based Solution to Time-dependent Two-point Initial Boundary Value Problems with Non-Periodic Boundary Conditions, Proceedings of the IATED International Conference Signal Processing, Pattern Recognition & Applications July 3-6 2001, Rhodes, Greece.
[5] Hira Narang and Rajiv Nekkanti. Wavelet-based Solution of Boundary Value Problems involving Hyperbolic Equations, Proceedings from the IATED International Conference Signal Processing, Pattern Recognition & Applications June 25-26, 2002
[6] Hira Narang and Rajiv Nekkanti. Wavelet-based solutions to problems involving Parabolic Equations, Proceedings of the IATED International Conference Signal Processing, Pattern Recognition & Applications 2001, Greece.
[7] Hira Narang and Rajiv Nekkanti. Wavelet-Based Solution to Elliptic Two-Point Boundary Value Problems with Non-Periodic Boundary Conditions, Proceedings from the WSEAS international conference in Signal, Speech, and Image processing Sept 25-28, 2002
[8] Hira Narang and Rajiv Nekkanti. Wavelet-Based Solution to Some Time-Dependent Two-Point Initial Boundary Value Problems with Non-Linear Non-Periodic Boundary Conditions, International Conference on Scientific computation and differential equations, SCICADE 2003, Trondheim, Norway, June 30. July 4, 2003.
[9] Hira Narang and Rajiv Nekkanti. Wavelet based Solution to Time-Dependent Two Point Initial Boundary Value Problems with Non-Periodic Boundary Conditions involving High Intensity Heat and Mass Transfer in Capillary Porous Bodies, IATED International Conference proceedings, Gainesville, FL 2004.
[10] Vishwavidyalaya Ambethkar. Numerical Solutions of Heat and Mass Transfer Effects of an Unsteady MHD Free Conective Flow Past an Iffinite Vertical Plate With Constant Suction. Journal of Naval Architecture and Marine Engineering, pages 28-36, June, 2008.
[11] Jens Krüger and Rüdiger Westermann. Linear Algebra Operators for GPU Implementation of Numerical Algorithms. ACM Transactions on Graphics (Proceedings of SIGGRAPH), pages908-916, July 2003.
[12] Jeff Bolz, Ian Farmer, Eitan Grinspun and Peter Schröoder. Sparse Matrix Solvers on the GPU: Conjugate Gradients and Multigrid. ACM Transactions on Graphics (Proceedings of SIGGRAPH), pages917-924, July 2003.
[13] Nolan Goodnight, Cliff Woolley, David Luebke and Greg Humphreys. A Multigrid Solver for Boundary Value Problems Using Programmable Graphics Hardware. In Proceeding of Graphics Hardware, pages 102-111, July 2003.
[14] Mark Harris, William Baxter, Thorsten Scheuermann and Anselmo Lastra. Simulation of Cloud Dynamics on Graphics Hardware. In Proceedings of Graphics Hardware, pages 92-101, July 2003.
[15] Mark Harris. Real-Time Cloud Simulation and Rendering. PhD thesis, 2003.
[16] Theodore Kim and Ming Lin. Visual Simulation of Ice Crystal Growth. In Proceedings of SIGGRAPH/Eurographics Symposium on Computer Amination, pages 86-97, July 2003.
[17] Aaron Lefohn, Joe Kniss, Charles Hansen and Ross Whitaker. Interactive Deformation and Visualization of Level Set Surfaces Using Graphics Hardware. In IEEE Visualization, pages 75-82, 2003.
[18] GPGPU website. http://www.gpgpu.org.
[19] N.C. Markatos, D.G. Tatchell, M. Cross, N. Rhodes. Numerical Simulation of Fluid Flow and Heat/Mass Transfer Processes. Springer-Verlag, 1986.