

# An Efficient Acceleration of Solving Heat and Mass Transfer Equations with the First Kind Boundary Conditions in Capillary Porous Radially Composite Cylinder Using Programmable Graphics Hardware

Hira Narang, Fan Wu, Abdul Rafae Mohammed

Computer Science Department, Tuskegee University, Tuskegee, Alabama, USA

Email: hnarang@tuskegee.edu, fwu@tuskegee.edu, amohammed4505@tuskegee.edu

**How to cite this paper:** Narang, H., Wu, F. and Mohammed, A.R. (2019) An Efficient Acceleration of Solving Heat and Mass Transfer Equations with the First Kind Boundary Conditions in Capillary Porous Radially Composite Cylinder Using Programmable Graphics Hardware. *Journal of Computer and Communications*, 7, 267-281. <https://doi.org/10.4236/jcc.2019.77022>

**Received:** March 14, 2019

**Accepted:** July 28, 2019

**Published:** July 31, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

With the latest advances in computing technology, a huge amount of efforts have gone into simulation of a range of scientific phenomena in engineering fields. One such case is the simulation of heat and mass transfer in capillary porous media, which is becoming more and more necessary in analyzing a number of eventualities in science and engineering applications. However, this procedure of numerical solution of heat and mass transfer equations for capillary porous media is very time consuming. Therefore, this paper pursuit is at making use of one of the acceleration methods developed in the graphics community that exploits a graphical processing unit (GPU), which is applied to the numerical solutions of such heat and mass transfer equations. The nVidia Compute Unified Device Architecture (CUDA) programming model offers a correct approach of applying parallel computing to applications with graphical processing unit. This paper suggests a true improvement in the performance while solving the heat and mass transfer equations for capillary porous radially composite cylinder with the first type of boundary conditions. This heat and mass transfer simulation is carried out through the usage of CUDA platform on nVidia Quadro FX 4800 graphics card. Our experimental outcomes exhibit the drastic overall performance enhancement when GPU is used to illustrate heat and mass transfer simulation. GPU can considerably accelerate the performance with a maximum found speedup of more than

5-fold times. Therefore, the GPU is a good strategy to accelerate the heat and mass transfer simulation in porous media.

## Keywords

Numerical Solution, Heat and Mass Transfer, General Purpose Graphics Processing Unit (GPGPU), CUDA

---

## 1. Introduction

During the previous half of the century, many scientists and engineers working in Heat and Mass Transfer phenomena in porous media have devoted a fantastic amount of efforts in finding solutions each analytically/numerically, and experimentally. To exactly analyze physical transfer phenomena of heat and mass transfer such as heat conduction, convection, and radiation, the simulation of these phenomena is very important. A heat and mass transfer simulation is carried out through utilizing parallel computing resources to simulate such heat and mass transfer phenomena. With help from the computer, in the beginning the sequential solutions had been found, and later when more high-powered computer systems became available, faster solutions have been applied to heat and mass transfer problems. However, the heat and mass transfer simulation of coupled phenomena requires a whole lot of greater computing sources than the uncoupled simulations. Therefore, acceleration of this simulation is very vital to effectively analyze and understand a complex set of heat and mass transfer problems.

This paper makes use of the parallel computing power of GPUs to accelerate the heat and mass transfer simulation. GPUs are very proficient considering the hypothetical rates of floating-point operation [1]. Therefore, comparing with super-computer, GPU is a powerful co-processor on a regular PC which is ready to simulate a large-scale heat and mass transfer with much fewer resources. The GPU has various benefits over CPU architectures, such as being relatively parallel, computation-intensive workloads, being inclusive of higher bandwidth, higher floating-point throughput. The GPU can be an alluring choice to clusters or super-computer in high performance computing areas.

CUDA [2] by nVidia already proved its effort to boost each programming and memory models. CUDA is a new parallel, C-like programming language Application Program Interface (API), which bypasses the rendering Interface and avoids the difficulties from the usage of GPGPU. Parallel computations are expressed as general-purpose, C-like language kernels operating in parallel over all the factors in an application.

The rest of the paper is organized as follows: Section 2 quickly introduces some intently related work; Section 3 describes the primary facts on GPU and CUDA; Section 4 gives the mathematical model of heat and mass transfer and

numerical solutions to heat and mass transfer equations; Section 5 offers our experimental results; and Section 6 concludes this paper and suggests some possible future work directions.

## 2. Related Work

The simulation of heat and mass transfer has been a very important subject matter for many years. And there is loads of work related to this field, such as fluid and air flight simulation. We just refer to some most current work described to this subject here.

Soviet Union was once 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 [3]. Later England and India took the lead and made further contributions for analytical and numerical options 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 equations with the features of programmability of fragments on GPU, and in addition 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 approach to remedy the fluid problem. In the meantime, Goodnight *et al.* [13] used the multigrid method to resolve the boundary price 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 different researchers. Kim *et al.* [16] solved the crystal formation equations on GPU. Lefohn *et al.* [17] matched the level-set iso surface records into a dynamic sparse texture format. Another innovative usage was to pack the records of the subsequent energetic tiles into a vector message, which used to be used to control the vertices and texture coordinates wanted to send from CPU to GPU. To examine more purposes about general-purpose computations GPU, more data can be determined from here [18]. Recently, Narang, *et al.* [19] [20] [21], have extended the heat and mass transfer to porous solid, hollow and composite cylinders.

## 3. 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**.

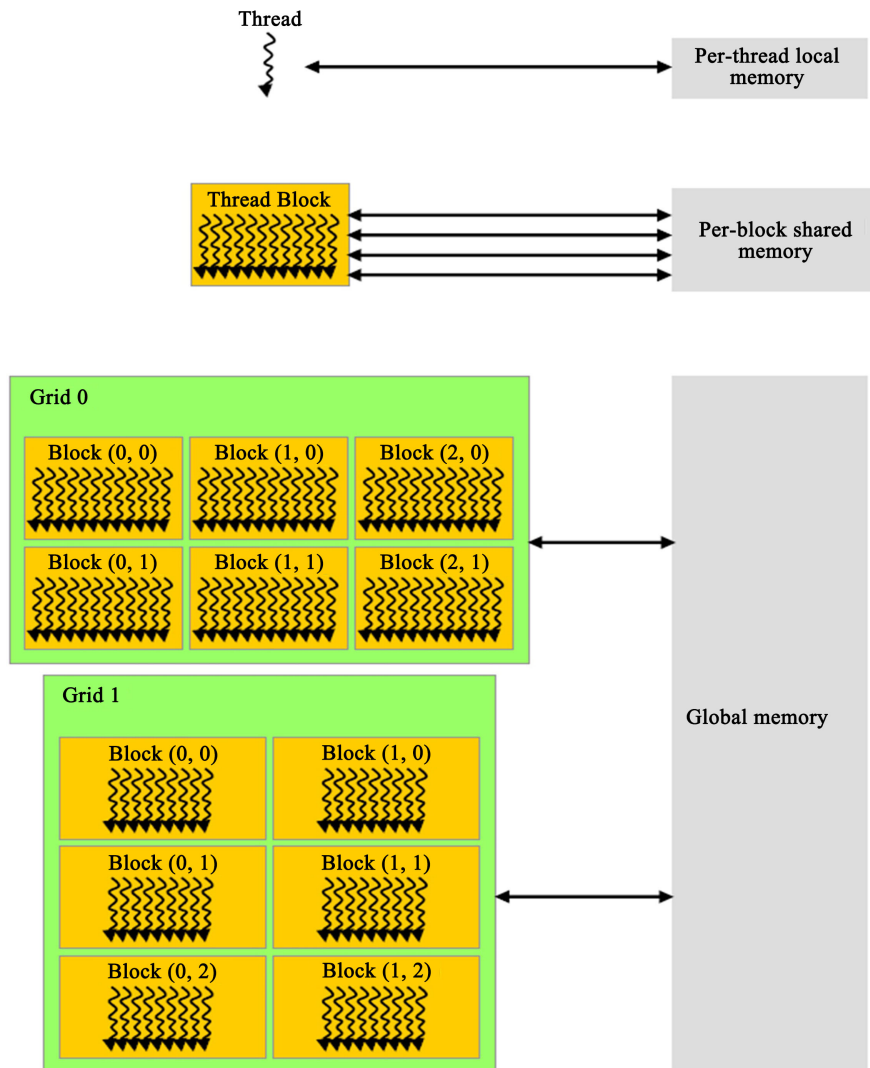


Figure 1. GPU memory architecture [2].

## 4. Mathematical Model and Numerical Solutions of Heat and Mass Transfer

### 4.1. Mathematical Model

Consider the Heat and Mass Transfer via a capillary porous radially composite cylinder with boundary conditions of the first kind. Let the  $z$ -axis be directed upward along the capillary porous radially composite cylinder and the  $r$ -axis radius of the capillary porous radially composite cylinder. Let  $u$  and  $v$  be the speed components along the  $z$ - and  $r$ -axes respectively. We have to write separate equations for each material as both will have special properties. Since we are concerned about analyzing the effect of conductivities of the 2 substances, we observe their behavior under the same initial and boundary conditions. So, the first equation will correspond to the first material ( $0 < r < a$ ) whereas the second equation correspond to the second material with specific heat and mass constants ( $a < r < 1$ ). Then the heat and mass transfer equations in the Boussinesq's ap-

proximation, are:

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

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

$0 < z < 2L$ ,  $a < r < b$ ,  $t > 0$ , where  $a = 0, b = 0.5$

for the experimental case here,

$2L$  is the length of the material

$r$  is the radius for capillary porous

Radially composite cylinder.

$$\frac{\partial T}{\partial t} = k_{11} \left( \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} \right) + k_{21} \left( \frac{\partial C}{\partial t} \right) \quad (1a)$$

$$\frac{\partial C}{\partial t} = k_{31} \left( \frac{\partial^2 C}{\partial r^2} + \frac{1}{r} \frac{\partial C}{\partial r} + \frac{\partial^2 C}{\partial z^2} \right) + k_{41} \left( \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} \right) \quad (2a)$$

$0 < z < 2L$ ,  $a < r < 1$ ,  $t > 0$ , where

$2L$  is the length of the second

Material,  $r$  is the radius for capillary

porous radially composite cylinder and  $t$  is the time

$a = 0.5$ , for this experimental case

for capillary porous composite cylinder.

Initial Conditions

$$\begin{aligned} T(r, z, 0) &= 0 \\ C(r, z, 0) &= 1 \end{aligned} \quad (3)$$

Boundary Conditions

$$\begin{aligned} T(r, 0, t) &= T_0 \\ C(r, 0, t) &= C_0 \end{aligned} \quad (4)$$

$$\begin{aligned} T(1, z, t) &= T_0 \\ C(1, z, t) &= C_0 \end{aligned} \quad (5)$$

$$\begin{aligned} T(r, 2L, t) &= T_{2L} = 0 \\ C(r, 2L, t) &= C_{2L} = 1 \end{aligned} \quad (6)$$

**Interface Conditions at  $r = a$ : Continuity of Temperatures and Concentrations as well as their fluxes in the two materials.**

Since the radially composite cylinder is assumed to be capillary porous,  $\mu_1$  is the velocity of the fluid,  $T_p$  the temperature of the fluid near the capillary porous radially composite cylinder,  $T_\infty$  the temperature of the fluid far away from the capillary porous radially composite cylinder,  $C_p$  the concentration near the capillary porous radially composite cylinder,  $C_{2L}$  the concentration far

end of the capillary porous radially composite cylinder,  $g$  the acceleration due to gravity,  $\beta$  the coefficient of volume expansion for heat transfer,  $\beta'$  the coefficient of volume expansion for concentration,  $\nu$  the kinematic viscosity,  $\sigma$  the scalar electrical conductivity,  $\omega$  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.

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

$$u = \frac{u_1}{V_0}, T = \frac{T_1 - T_\infty}{T_p - T_\infty}, C = \frac{C_1 - C_\infty}{C_p - C_\infty}, P_r = \frac{\nu}{k}, S_c = \frac{\nu}{D'}$$

$$M = \frac{\sigma B_0^2 \nu}{\rho V_0^2}, G_r = \frac{\nu g \beta (T_p - T_\infty)}{V_0^3}$$

$$G_m = \frac{\nu g \beta' (C_p - C_\infty)}{V_0^3}, \omega = \frac{4\nu \omega_1}{V_0^2}$$

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

$$\frac{\partial T}{\partial t} + \frac{\partial^2 T}{\partial r^2} - 4 \frac{\partial C}{\partial t} + \frac{1}{r} \frac{\partial T}{\partial r} = \frac{4}{P_r} \frac{\partial^2 T}{\partial z^2} \tag{8}$$

$$\frac{\partial C}{\partial t} + \frac{\partial^2 C}{\partial r^2} - 4 \frac{\partial T}{\partial t} + \frac{1}{r} \frac{\partial C}{\partial r} = \frac{4}{P_r} \frac{\partial^2 C}{\partial z^2} \tag{9}$$

### 4.2. 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 simulation is divided into a grid 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  $z = 0$  are given by  $T(0, t)$  and  $C(0, t)$  hence are known from the boundary conditions.

In **Figure 2**,  $\Delta z$  &  $\Delta r$  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. 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 composite cylinder is 1.0

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

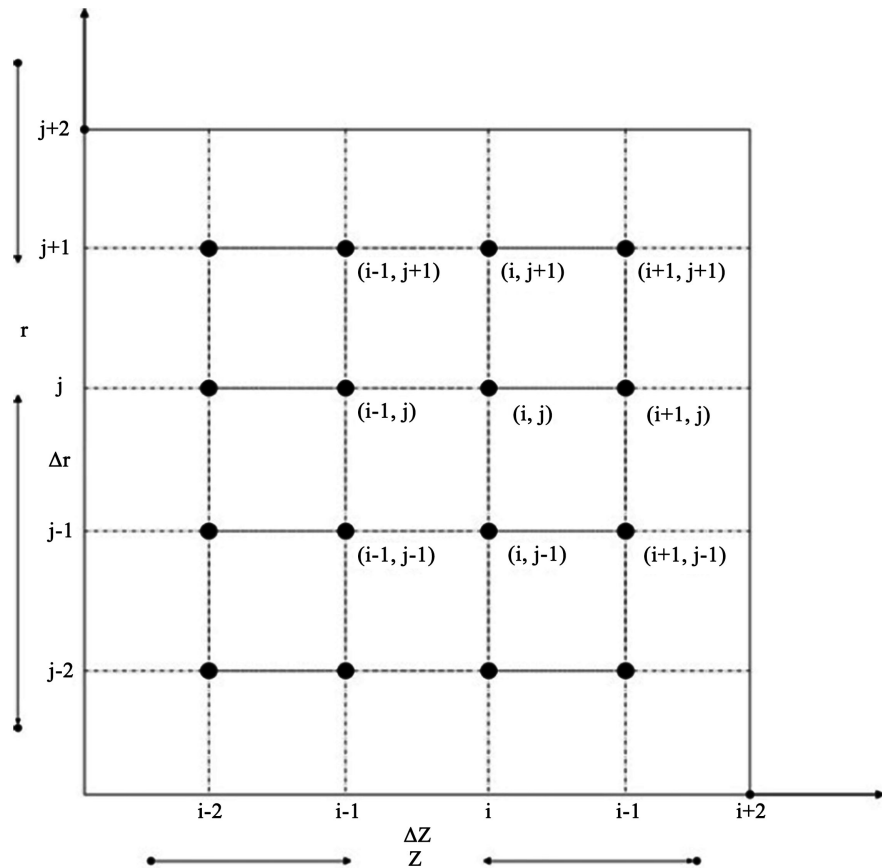


Figure 2. Finite difference grid for capillary porous composite cylinder.

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

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

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

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

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

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

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

The finite difference approximation of Equations (8) and (9) are obtained with substituting finite differences into Equations (8) and (9) and multiplying both

sides by  $\Delta t$  and after simplifying, we let  $\frac{\Delta t}{(\Delta z)^2} = r' = 1$  (method is always stable and convergent), under this condition the above equations can be written as:

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

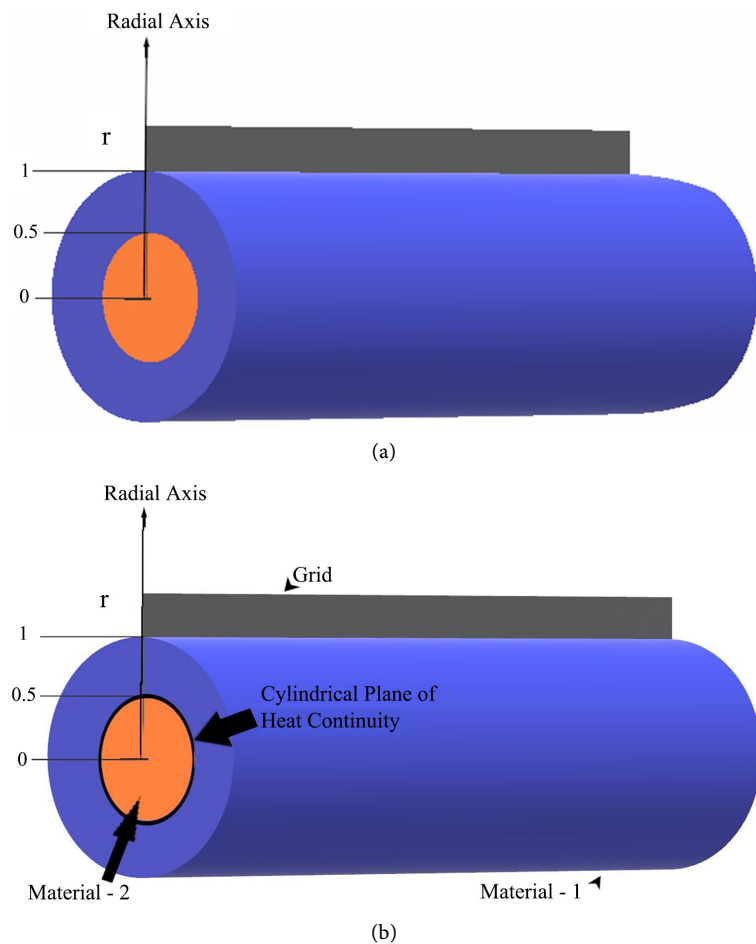
$$\frac{\partial T}{\partial t} = \frac{1}{2} \left( \frac{2U+V-2(2T_{i,j}+C_{i,j})}{(\Delta r)^2} + \frac{2U+V}{2r(\Delta r)} + \frac{2U+V-2(2T_{i,j}+C_{i,j})}{(\Delta z)^2} \right)$$

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

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

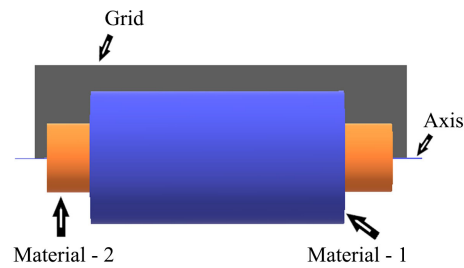
### 4.3. Grid Structure and Plane of Heat and Mass Continuity

The plane of heat continuity is the cylindrical plane along the horizontal axis of the solid **Figure 3(a)** and **Figure 3(b)** where the material properties change *i.e.* where the two different materials are joined or merged together. So, the grid position in the above radially composite cylinder is as depicted in **Figure 4**.



**Figure 3.** (a) Plane of heat and mass continuity in a composite solid cylinder; (b) Grid position in a composite solid cylinder.





**Figure 4.** Grid position in a composite solid cylinder.

At this point of interface, *i.e.*, the temperature at the last grid point along any radius of the first material is approximately equal to the temperature at the first grid point in the second material and similarly heat flux continuity is assumed. Similarly, the concentration at the last grid point along any radius of the first material is approximately equal to the concentration at the first grid point in the second material and similarly the concentration flux. So, the heat and mass continuity equations can be written as follows:

$$T_1(r, z_l, t) = T_2(r, z_0, t) \quad (10)$$

$$C_1(r, z_l, t) = C_2(r, z_0, t) \quad (11)$$

The flux continuities can be similarly described.

## 5. Experimental Results and Discussion

### 5.1. 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 4.00 GB. 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:	65,536 bytes
Total amount of shared memory per block:	16,384 bytes
Total number of registers available per block:	16,384
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:	57,509.6 (MB/s)

In the experiments, we considered solving heat and mass transfer differential equations in capillary porous radially composite cylinder with boundary condi-

tions of first kind 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 radially composite cylinder as heat and mass are transferred from one end of the capillary porous radially composite 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 radially composite cylinder. Also, there is a constant temperature and concentration  $N_0$  constantly working the surface of the capillary porous radially composite cylinder. The temperature at the other end of the capillary porous radially composite 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 radially composite cylinder where  $z = \infty$  is assumed to be negligible ( $\approx 0$ ). 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  $2L$  for the capillary porous radially composite cylinder and varied the number of nodal points  $N$  to be determined in the capillary porous radially composite cylinder. Since  $N$  is inversely proportional to the step size  $\Delta z$ , increasing  $N$  decreases  $\Delta z$  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 ran the tests for several different values of  $N$  and  $\Delta z$  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.

## 5.2. Experimental Results

The normalized temperature and concentration distributions at various points in the capillary porous radially composite cylinder are depicted in **Table 1** and **Table 2** respectively. We can immediately see that, at each point in the capillary porous radially composite cylinder, the CPU and GPU computed results are similar. In addition, the value of temperature is highest and the value of concentration is lowest at the point on the capillary porous radially composite cylinder

**Table 1.** Comparison of GPU and CPU results for capillary porous composite cylinder (concentration).

Z	CPU RESULTS	GPU RESULTS
1	0.05102320	0.048674520
2	0.16001332	0.179985630
3	0.32310478	0.340000563
4	0.40124587	0.444023145
5	0.50029010	0.530001245
6	0.56310245	0.551212230
7	0.64102563	0.600102345
8	0.67502143	0.655201462
9	0.74001265	0.779856321
10	0.84420135	0.859874563
11	1	1

**Table 2.** Comparison of GPU and CPU results for capillary porous composite cylinder (temperature).

Z	CPU RESULTS	GPU RESULTS
1	1	1
2	0.76698542	0.788596412
3	0.70998745	0.700120469
4	0.63332156	0.644210635
5	0.56998410	0.570084361
6	0.51112035	0.501584703
7	0.45010236	0.421364512
8	0.37897856	0.397511246
9	0.32010510	0.330014056
10	0.26213120	0.250708942
11	0.17010405	0.164425182

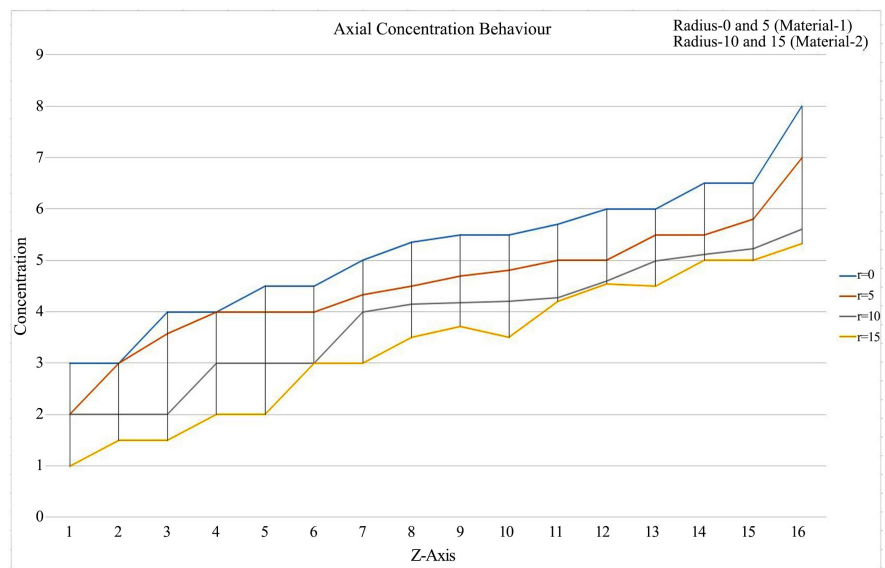
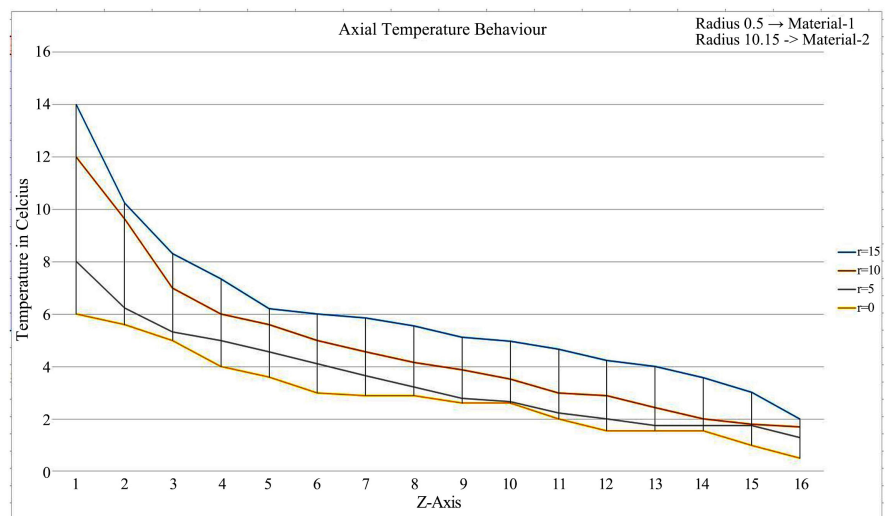
where the heat resource and mass resource are constantly applied. As we move away from this point, the values of the temperature decrease and concentration increase. At a point near the designated end of the capillary porous radially composite cylinder, the values of the temperature approach zero and concentration approach one.

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).

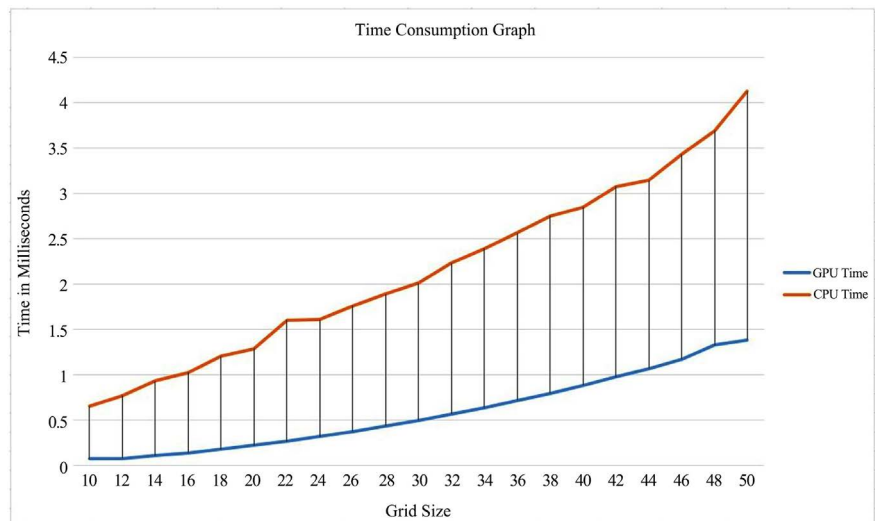
For the purpose of measuring the execution time, the same functions were implemented in both the device (GPU) and the host (CPU), to initialize the temperature and concentration and to compute the numerical solutions. In this case, we measured the processing time for different values of  $N$ . The graph in **Figure 5** depicts the performance of the GPU versus the CPU in terms of the processing time. We ran the test for  $N$  running from 10 to 550 with increments of 30 and generally, the GPU performed the calculations a lot faster than the CPU.

- 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.

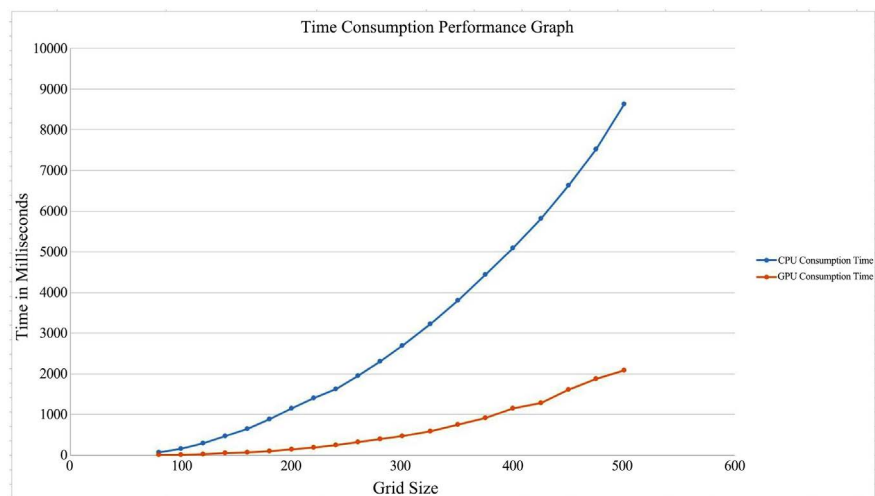
**Figure 6(a)** and **Figure 6(b)** show some of our experimental results for both capillary porous radially composite cylinder.



**Figure 5.** (a) Temperature distribution; (b) Concentration distribution.



(a)



(b)

**Figure 6.** (a) Performance of GPU and CPU implementations for capillary porous composite cylinder; (b) Performance of GPU and CPU implementations for capillary porous composite cylinder with incremental number of nodes.

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 radially composite cylinder show about 5 times speed-up

## 6. Conclusion and Future Work

We have presented our numerical approximations to the solution of the heat and mass transfer equation with the first kind of boundary and initial conditions for capillary porous radially composite 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 radially composite cylinder indicate that our GPU-based implementation shows a significant performance improvement over CPU-based implementation and the maximum observed speedups are about 10 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 geometries.

### Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

### References

- [1] Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E. Purcell, T.J. (2007) A Survey of General-Purpose Computation on Graphics Hardware. *Computer Graphics Forum*, **26**, 80-113. <https://doi.org/10.1111/j.1467-8659.2007.01012.x>
- [2] NVIDIA Corporation (2009) NVIDIA Programming Guide 2.3. <https://www.nvidia.com>
- [3] Luikov, A.V. (1966) Heat and Mass Transfer in Capillary Porous Bodies. Pergamon Press, Oxford, 233-303. <https://doi.org/10.1016/B978-1-4832-0065-1.50010-6>
- [4] Narang, H.N. and Nekkanti, R.K. (2001) 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*, Rhodes, Greece, 3-6 July 2001.
- [5] Narang, H.N. and Nekkanti, R.K. (2002) Wavelet-Based Solution of Boundary Value Problems Involving Hyperbolic Equations. *Proceedings from the IATED International Conference Signal Processing, Pattern Recognition & Applications*, Grete, Greece, 25-26 June 2002, 470-473.
- [6] Narang, H. and Nekkanti, R. (2001) Wavelet-Based Solutions to Problems Involving Parabolic Equations. *Proceedings of the IATED International Conference Signal Processing, Pattern Recognition & Applications*, Rhodes, Greece, 3-6 July 2001.
- [7] Narang, H.N. and Nekkanti, R.K. (2002) 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*, Skiathos, Skiathos Island, Greece, 25-28 September 2002.
- [8] Narang, H.N. and Nekkanti, R.K. (2003) Wavelet-Based Solution to Some Time-Dependent Two-Point Initial Boundary Value Problems with Non-Linear Non-Periodic Boundary Conditions. *International Conference on Scientific Com-*

- putation and Differential Equations, SCICADE 2003*, Trondheim, Norway, 30 June-4 July 2003.
- [9] Narang, H.N. and Nekkanti, R.K. (2004) 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*, Rhodes, Greece, 30 June-2 July 2003, 130-135.
- [10] Ambethkar, V. (2008) Numerical Solutions of Heat and Mass Transfer Effects of an Unsteady MHD Free Convective Flow Past an Infinite Vertical Plate with Constant Suction. *Journal of Naval Architecture and Marine Engineering*, **5**, 27-36.
- [11] Krüger, J. and Westermann, R. (2003) Linear Algebra Operators for GPU Implementation of Numerical Algorithms. *ACM Transactions on Graphics*, **22**, 908-916. <https://doi.org/10.1145/882262.882363>
- [12] Bolz, J., Farmer, I., Grinspun, E. and Schröder, P. (2003) Sparse Matrix Solvers on the GPU: Conjugate Gradients and Multigrid. *ACM Transactions on Graphics*, **22**, 917-924. <https://doi.org/10.1145/882262.882364>
- [13] Goodnight, N., Woolley, C., Luebke, D. and Humphreys, G. (2003) A Multigrid Solver for Boundary Value Problems Using Programmable Graphics Hardware. *Proceeding of Graphics Hardware*, San Diego, CA, 26-27 July 2003, 102-111.
- [14] Harris, M., Baxter, W., Scheuermann, T. and Lastra, A. (2003) Simulation of Cloud Dynamics on Graphics Hardware. *Proceeding of Graphics Hardware*, San Diego, CA, 26-27 July 2003, 92-101.
- [15] Harris, M. (2003) Real-Time Cloud Simulation and Rendering. Ph.D. Thesis, The University of North Carolina, Chapel Hill, NC.
- [16] Kim, T. and Lin, M. (2003) Visual Simulation of Ice Crystal Growth. *Proceedings of SIGGRAPH/Eurographics Symposium on Computer Animation*, San Diego, CA, 26-27 July 2003, 86-97.
- [17] Lefohn, A., Kniss, J., Hansen, C. and Whitaker, R. (2003) Interactive Deformation and Visualization of Level Set Surfaces Using Graphics Hardware. *IEEE Visualization*, Seattle, WA, 19-24 October 2003, 75-82.
- [18] GPGPU Website. <http://www.gpgpu.org>
- [19] Narang, H.N., Wu, F., Ogunniyan, A. and Mohammed, A.R. (2017) An Efficient Acceleration of Solving Heat and Mass Transfer Equations with Second Kind Boundary and Initial Conditions in Solid and Hollow Cylinder Using Programmable Graphics Hardware. *Journal of Computations & Modelling*, **7**, 29-50.
- [20] Narang, H.N., Wu, F. and Mohammed, A.R. (2017) An Efficient Solution of Heat and Mass Transfer Equations using Programmable General Purpose Processing Unit under Natural Boundary Conditions in Capillary Porous Solid and Hollow Cylinder. *International Advanced Research Journal in Science, Engineering and Technology*, **4**, 1-11.
- [21] Narang, H.N., Wu, F. and Mohammed, A.R. (2018) An Efficient Solution of Heat and Mass Transfer Equations using Programmable General Purpose Processing Unit under Natural Boundary Conditions in Capillary Porous Solid and Composite Cylinder, with Boundary Conditions of Second Kind. *International Journal of Computation and Communication*, **6**, 24-38.