

Parallel Numerical Solution for Temperature Prediction on Laser Glass Cutting using AGE Variances Method

Lizawati Mi Yusuf, Mohammed Shariff Bashir
Ghouse, Mohd Shahizan Othman and
Siti Mariyam Shamsuddin
Faculty of Computer Science and Information System
Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
{lizawati, shahizan, mariyam}@utm.my,
mshariff5@live.utm.my

Norma Alias

Ibnu Sina Institute for Fundamental Sciences Studies
Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
norma@ibnusina.utm.my

Abstract - This paper discusses on the mathematical formulation in simulating the temperature behavior on a laser glass cutting. The mathematical simulation is solved by using three numerical methods, Alternating Group Explicit Method with Brian and Douglas variant, and Gauss Seidel Red Black method. This simulation is then being run on a parallel environment using the master and workers methodology where 20 units of processors work simultaneously in solving and visualizing the data. Lastly, the results obtained from the simulation is compared by using the parallel performance measurement and also on a large sparse matrix size.

Keywords - Alternating Group Explicit; Gauss Seidel Red Black; Parallel Computing; Numerical Method

I. INTRODUCTION

Glass materials are now widely used in many sciences and engineering field such as optoelectronic, devices, bottles, micro fluidics and windshields [1-8]. The structure of the glass is brittle in such where the atomic structures are arranged closely and have strong force knitted between them [9-10]. Therefore, in order to break up these bonds created by these atomic structures, a high temperature is needed to separate the glass [8-10]. Furthermore, a high quality glass is a glass has a high refractive index, which enables light to penetrate through. This can be achieved by using the laser technique [1, 6, 11].

Past researchers that conducted simulation using the laser technology concluded that laser simulation is expensive, requires high security and expertise in monitoring and running it [1, 7, 10]. Due to this, the analytical method is used to substitute the simulation by using laser technology. A partial differential equation (PDE) is used to represent the problem of laser glass cutting. The PDE is then discretized and solved by using three numerical methods which are the alternating group explicit (AGE)

method using two variants Brian and Douglas, while Gauss Seidel Red Black method is used as standard comparisons. The simulation is run in a parallel environment, which can support a simulation that requires complexity and iterations [2, 12-13].

The objective of this paper is to compare the parallel performance between these three numerical methods, AGE Brian, AGE Douglas and GSRB. Besides that, simulation results on the size of matrix used is also being compared.

II. STATEMENT OF THE PROBLEM

The case study in this paper will concentrate three dimensional cylindrical coordinates where the problem formulation on the laser glass interaction is represented as follows,

$$\frac{1}{a} \frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial r^2} + \frac{1}{r} \frac{\partial U}{\partial r} + \frac{1}{r^2} \frac{\partial^2 U}{\partial \theta^2} + \frac{\partial^2 U}{\partial z^2} + \frac{q}{k} \quad (1)$$

with initial condition:

$$T(r, \theta, z, t)|_{t=0} = T_0 \quad (2)$$

with T_0 is initial temperature where temperature of glass cylinder before laser process and boundary conditions:

$$T(r, \theta, z, t)|_{s_1=0} = T_1 \text{ and } T(r, \theta, z, t)|_{s_2=0} = T_2 \quad (3)$$

where,

- T_1, T_2 : Temperature (K) of inner and outer surface of glass cylinder
- s_1, s_2 : Area of inner (m^2) and other surface (m^2) of glass cylinder
- r, θ, z : Radius (m), angle ($^\circ$), and length (m) of glass cylinder

- q : Rate of heat flow (W/m²)
 a : Density (Kg/m³) and heat capacity (J)
 K : Thermal conductivity (W)
 t : Time (s)

From (1), the equation need to be discretized to simplify it and this is done by using the Finite Difference Method (FDM). Then the equation will be solved by using three numerical methods, which are Alternating Group Explicit Method (AGE) that consists of two variants Douglas and Brian, while Gauss Seidel Red Black (GSRB) method is used as standard comparison purposes. The following section will be discussing on the numerical methods used to solve the PDE

III. METHODS OF SOLUTIONS

Numerical methods are used to solve the PDE equation that formulates the problem on laser glass cutting. The AGE Brian and AGE Douglas used matrix block to solve, and GSRB is used as standard comparison between those two methods. In this section, the GSRB method is discussed first followed by AGE Brian and AGE Douglas.

A. Gauss-Seidel Red Black (GSRB) method

The GSRB method normally used as a control scheme in solving three dimensional cylindrical glass interaction. The GSRB method uses two sub domains which are the odd sub domain, Ω^R and even subdomain, Ω^H . As mentioned, the (1) will be discretized using FDM while the GSRB method is used to solve the equation as follows,

$$u_{i,j,k}^{(p+1)} = \frac{\begin{bmatrix} -a_i u_{i-1,j,k}^{(p)} - h_i u_{i,j-1,k}^{(p)} - d u_{i,j,k-1}^{(p)} \\ -b_i u_{i+1,j,k}^{(p)} - g u_{i,j,k+1}^{(p)} + \bar{b} u_{i-1,j,k}^{[N]} \\ + \bar{d}_i u_{i,j,k}^{[N]} + \bar{h} u_{i,j-1,k}^{[N]} + \bar{d} u_{i,j,k-1}^{[N]} \\ + \bar{e} u_{i+1,j,k}^{[N]} + \bar{f} u_{i,j+1,k}^{[N]} + \bar{g} u_{i,j,k+1}^{[N]} + \frac{p}{k} \end{bmatrix}}{c_i} \quad (4)$$

where i, j and k will be represented as odd or even numbers depending on the type of subdomains.

B. AGE Brian Method

AGE Brian method uses Brian variant and acceleration parameter, r based on linear polar interpolation. The formula is given as below [14],

$$(G_1 + rI)u_{[xy]}^{(k+\frac{1}{7})} = (rI - G_2 - G_3 - G_4 - G_5 - G_6) u_{[xy]}^{(k)} + f, \quad (5)$$

$$(G_2 + rI)u_{[xy]}^{(k+\frac{2}{7})} = r u_{[xy]}^{(k+\frac{1}{7})} + G_2 u_{[xy]}^k \quad (6)$$

$$(G_3 + rI)u_{[xy]}^{(k+\frac{3}{7})} = r u_{[xy]}^{(k+\frac{2}{7})} + G_3 u_{[xy]}^k \quad (7)$$

$$(G_4 + rI)u_{[xy]}^{(k+\frac{4}{7})} = r u_{[xy]}^{(k+\frac{3}{7})} + G_4 u_{[xy]}^k \quad (8)$$

$$(G_5 + rI)u_{[xy]}^{(k+\frac{5}{7})} = r u_{[xy]}^{(k+\frac{4}{7})} + G_5 u_{[xy]}^k \quad (9)$$

$$(G_6 + rI)u_{[xy]}^{(k+\frac{6}{7})} = r u_{[xy]}^{(k+\frac{5}{7})} + G_6 u_{[xy]}^k \quad (10)$$

$$u_{[xy]}^{(k+1)} = u_{[xy]}^{(k)} + 2(u_{[xy]}^{(k+\frac{6}{7})} - u_{[xy]}^{(k)}) \quad (11)$$

Based on (5) to (11), the equations are being simplified into 2×2 matrix block form. The calculation is as follows,

$$u_{1[xy]}^{(p+\frac{1}{7})} = \bar{C}_{1,i}^{-1} (D_{1,i} u_{1[xy]}^{(p)} + E_{1,i} u_{2[xy]}^{(p)} + F_{1,i} u_{1[xy]}^{[N]} + H_{1,i} u_{2[xy]}^{[N]} + g_{1[xy]}) \quad (12)$$

$$u_{k[xy]}^{(p+\frac{1}{7})} = \bar{C}_{2,i}^{-1} (\bar{E}_{1,i} u_{k-1[xy]}^{(p)} + E_{1,i} u_{k+1[xy]}^{(p)} + D_{2,i} u_{k[xy]}^{(p)} + \bar{H}_{1,i} (u_{k-1[xy]}^{[N]} + H_{1,i} u_{k+1[xy]}^{[N]}) + F_{1,i} u_{k[xy]}^{[N]} + g_{k[xy]}) \quad (13)$$

$$u_{k[xy]}^{(p+\frac{1}{7})} = \bar{C}_{1,i}^{-1} (\bar{E}_{1,i} u_{k-1[xy]}^{(p)} + E_{1,i} u_{k+1[xy]}^{(p)} + D_{1,i} u_{k[xy]}^{(p)} + \bar{H}_{1,i} (u_{k-1[xy]}^{[N]} + H_{1,i} u_{k+1[xy]}^{[N]}) + F_{1,i} u_{k[xy]}^{[N]} + g_{k[xy]}) \quad (14)$$

$$u_{m[xy]}^{(p+\frac{1}{7})} = \bar{C}_{1,i}^{-1} (\bar{E}_{1,i} u_{m-1[xy]}^{(p)} + D_{1,i} + \bar{H}_{1,i} u_{m-1[xy]}^{[N]} + \bar{H}_{1,i} u_{m-1[xy]}^{[N]} + F_{1,i} u_{m[xy]}^{[N]} + g_{m[xy]}) \quad (15)$$

Based from (12) to (15), it can be simplified that every calculation can be represented as $(p+1)$,

$$u_{[xy]}^{(p+1)} = u_{[xy]}^{(p)} + 2(u_{[xy]}^{(p+\frac{6}{7})} - u_{[xy]}^{(p)}) \quad (16)$$

C. AGE Douglas Method

The AGE Douglas method uses the Douglas-Rachford formula that splits a matrix system of linear equations by using fractional scheme [15]. It follows the same steps as Brian variant. However, Douglas only consists of six groups of matrices and being simplified by using 2×2 block matrices results and the solutions are as follows,

$$u_{1[xy]}^{(p+\frac{1}{6})} = \bar{C}_1^{-1} (D_1 u_{1[xy]}^{(p)} + E_1 u_{2[xy]}^{(p)} + F_1 u_{1[xy]}^{[N]} + F_2 u_{2[xy]}^{[N]} + 2g_{1[xy]}) \quad (17)$$

$$u_{k[xy]}^{(p+\frac{1}{6})} = \bar{C}_2^{-1} (E_1 (u_{k-1[xy]}^{(p)} + u_{k+1[xy]}^{(p)}) + D_2 u_{k[xy]}^{(p)} + F_2 (u_{k-1[xy]}^{[N]} + u_{k+1[xy]}^{[N]}) + F_1 u_{k[xy]}^{[N]} + 2g_{k[xy]}) \quad (18)$$

$$u_{k[xy]}^{(p+\frac{1}{6})} = \bar{C}_1^{-1} (E_1 (u_{k-1[xy]}^{(p)} + u_{k+1[xy]}^{(p)}) + D_1 u_{k[xy]}^{(p)} + F_2 (u_{k-1[xy]}^{[N]} + u_{k+1[xy]}^{[N]}) + F_1 u_{k[xy]}^{[N]} + 2g_{k[xy]}) \quad (19)$$

$$u_{m[xy]}^{(p+\frac{1}{6})} = \bar{C}_1^{-1} \left(E_1 \left(u_{m-1[xy]}^{(p)} + D_1 u_{m[xy]}^{(p)} \right) + F_1 u_{m[xy]}^{[N]} + F_2 u_{m-1[xy]}^{[N]} + 2g_{m[xy]} \right) \quad (20)$$

From (17) to (20), the derivation of AGE Douglas can be represented as $(p + 1)$ as following,

$$u_{[x,z]}^{(p+1)} = (G_1 + rI)^{-1} [G_1 u_{[xz]}^{(p)} + ru_{[xz]}^{(p+\frac{5}{6})}] \quad (21)$$

IV. METHODOLOGY

The methodology used to run the simulation of laser glass cutting is the master and workers methodology. The master will be one unit of computer monitoring a number of computers, which will be the workers in simulating the mathematical simulation. The master has its own distinctive responsibilities based on Fig. 1, the initial boundary memory is set together with initiating workers and decompose data. Then, the master will send and distribute data between itself and the workers. The workers will receive these data and will start on the numerical simulation. The algorithm for the numerical simulation by using the AGE Brian is shown in Fig. 2. During the numerical calculation is conducted, a global communication is established between the processors. The purpose of this communication is to send the recent value to the neighboring processors by checking the position of each neighboring processors. This is done by following Fig. 3

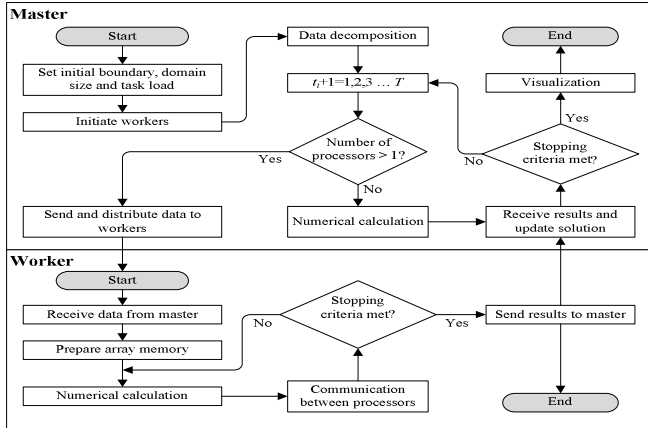


Figure 1. Master and worker flowchart methodology

Then, a local communication is established to ensure solution is accurate. The numerical calculation will undergo a loop until the stopping criteria is met which is the time step, t and local convergence, ϵ is obtained (refer Fig. 4).

When these two are met, the workers will collect and send the data to the master. Upon receiving, the master will conduct global convergence as the criteria before visualizing the data. The global convergence is run based on algorithm given in Fig. 5. If the global convergence is not met, then the simulation will start over in redistributing the data again to the workers. However, if the global convergence is met, then the solution obtained is ready for visualization.

V. NUMERICAL RESULTS AND DISCUSSIONS

The mathematical simulation is conducted on two matrix size which are $100 \times 100 \times 100$ (A) and $140 \times 140 \times 140$ (B). The simulation results are compared between four characteristics of parallel performance measurement. The four parallel performance characteristics are,

$$\text{Speed up:} \quad S_p = T_1/T_p \quad (22)$$

$$\text{Efficiency:} \quad C_p = S_p/p \quad (23)$$

$$\text{Effectiveness:} \quad F_p = S_p/T_p \quad (24)$$

$$\text{Temporal Performance:} \quad L_p = T_p^{-1} \quad (25)$$

```

Declare memory, set loop variables
{
  Receive data from master,  $p, \Delta t, \Delta x, \Delta y, \Delta z, r$ 
  Receive data from subdomain,  $(m \times m \times m)(p_i)$ 
  Count constant BRIAN,  $r_0 = r + (\frac{c}{6}), \text{delta} = (r_0 - a_0)$ ,
   $\bar{r} = r_0 - c$ 
  Do while time step,  $t$  not reach
  {
    Do while convergence not obtain
    {
      Use equation (12)-(15)
      Communicate between processors (Fig. 2)
      Calculate
       $u_{k[xy]}^{(p+\frac{2}{7})} = \bar{C}_2^{-1} \left[ \hat{C}_2 u_{k[xy]}^{(p)} + ru_{k[xy]}^{(p+\frac{1}{7})} \right], k = 1,3,5 \dots m$ 
       $u_{k[xy]}^{(p+\frac{2}{7})} = \bar{C}_1^{-1} \left[ \hat{C}_1 u_{k[xy]}^{(p)} + ru_{k[xy]}^{(p+\frac{1}{7})} \right], k = 2,4 \dots m-1$ 
      Calculate
       $u_{i[yz]}^{(p+\frac{3}{7})} = \bar{C}_1^{-1} \left[ \hat{C}_1 u_{i[yz]}^{(p)} + ru_{i[yz]}^{(p+\frac{2}{7})} \right], i = 1,3,5 \dots m$ 
       $u_{i[yz]}^{(p+\frac{3}{7})} = \bar{C}_2^{-1} \left[ \hat{C}_2 u_{i[yz]}^{(p)} + ru_{i[yz]}^{(p+\frac{2}{7})} \right], i = 2,4 \dots m-1$ 
      Calculate
       $u_{i[yz]}^{(p+\frac{4}{7})} = \bar{C}_2^{-1} \left[ \hat{C}_2 u_{i[yz]}^{(p)} + ru_{i[yz]}^{(p+\frac{3}{7})} \right], i = 1,3,5 \dots m$ 
       $u_{i[yz]}^{(p+\frac{4}{7})} = \bar{C}_1^{-1} \left[ \hat{C}_1 u_{i[yz]}^{(p)} + ru_{i[yz]}^{(p+\frac{3}{7})} \right], i = 2,4 \dots m-1$ 
      Calculate
       $u_{j[xz]}^{(p+\frac{5}{7})} = \bar{C}_1^{-1} \left[ \hat{C}_1 u_{j[xz]}^{(p)} + ru_{j[xz]}^{(p+\frac{4}{7})} \right], i = 1,3,5 \dots m$ 
       $u_{j[xz]}^{(p+\frac{5}{7})} = \bar{C}_2^{-1} \left[ \hat{C}_2 u_{j[xz]}^{(p)} + ru_{j[xz]}^{(p+\frac{4}{7})} \right], i = 2,4 \dots m-1$ 
      Calculate
       $u_{j[xz]}^{(p+\frac{6}{7})} = \bar{C}_2^{-1} \left[ \hat{C}_2 u_{j[xz]}^{(p)} + ru_{j[xz]}^{(p+\frac{5}{7})} \right], i = 1,3,5 \dots m$ 
       $u_{j[xz]}^{(p+\frac{6}{7})} = \bar{C}_1^{-1} \left[ \hat{C}_1 u_{j[xz]}^{(p)} + ru_{j[xz]}^{(p+\frac{5}{7})} \right], i = 2,4 \dots m-1$ 
      Calculate
       $u_{[xy]}^{(p+1)} = u_{[xy]}^{(p)} + 2 \left( u_{[xy]}^{(p+\frac{6}{7})} - u_{[xy]}^{(p)} \right)$ 
    }
  }
  Send data to master
  Pvm_exit
}

```

Figure 2. Numerical simulation algorithm for AGE-Brian

Figure 5. Algorithm for master global convergence test

The simulation is conducted on a number of homogenous computers, which comprises of 20 units of Intel Pentium processors. The software that is used to run the simulation is supported by Parallel Virtual Machine (PVM) and C programming. The results obtained are as follows.

Table 1 shows that as the number of processors increases the execution time decrease thus proving that as more processors is used in a parallel environment, the results can be obtained much more faster. This is due to more processors which are working simultaneously to solve the problem resulting a smaller value of execution time.

Table 1. Numerical results on execution time

No of processors	Brian		Douglas		GSRB	
	A	B	A	B	A	B
1	116.45	248.62	122.89	259.92	154.64	275.89
2	97.67	213.88	110.95	235.80	132.32	242.98
4	76.30	177.50	87.27	198.53	109.50	210.56
6	58.25	147.52	68.88	167.66	90.30	185.23
8	47.32	120.00	53.98	137.43	79.35	159.89
10	38.16	97.54	43.29	111.90	74.21	136.20
12	31.33	83.55	36.58	96.22	71.48	127.43
14	26.67	72.16	30.62	83.70	69.55	122.75
16	24.22	63.56	27.55	76.56	67.43	118.54
18	22.95	59.58	25.24	71.22	65.22	113.83
20	22.10	57.83	24.09	68.38	63.99	110.70

Legend: A – Matrix 100 × 100 × 100 B – Matrix 140 × 140 × 140

However, the results differ in the types of numerical methods used. Out of the three methods, AGE Brian shows the smallest value of execution time followed by AGE Douglas and GSRB. This proves that in simulating laser glass cutting, the AGE Brian method is the fastest in solving the problem followed by AGE Douglas and GSRB. This is because AGE Brian uses more matrix block points compared to other methods.

In the 140 x 140 x140 matrix, the trend is still the same where AGE Brian, AGE Douglas and GSRB execution time gets smaller as more processors are used. However, the execution time is quite slow compared to the 100 x 100 x 100 size of matrix due to more data points being used. When this is the case where more input is inserted, the simulation will therefore executed in a slower execution time but the simulation is still the fastest by using AGE Brian.

Table 2 compiles the results of the numerical results on speed up measurement. Speed up is a measurement based on the speed of one single die processors over the total number of processors used. Therefore, mathematically the speed of the simulation will increase when the total number of processors increases as well. This is supported by the results

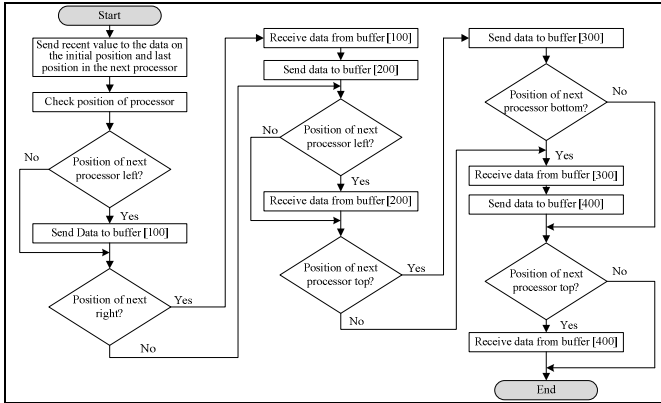


Figure 3. Flowchart on communication between processors

```

{
  Local convergence test,
   $e_{i,j,k}(p)^{(k)} \leftarrow |u_{i,j,k}^{(k+1)}(p) - u_{i,j,k}^k(p)|$ 
  If  $\max_k |e_{i,j,k}(p)^{(k)}| < \epsilon(p)$ 
  Then send  $\max_k |e_{i,j,k}(p)^{(k)}|$  to master
  {
    Change  $u_{i,j,k}(p) \leftarrow u_{i,j,k}^{(k+1)}(p)$ 
    Receive  $e_{i,j,k}(global)^{(k)}$  from master
    Loop++
  }
  Else
    Convergence obtain
}

```

Figure 4. Algorithm for workers local convergence test

```

Declare memory, looping variables
{
  Initiate workers processors,  $p_i \leq P$ 
  Send data  $\Delta t, \Delta x, \Delta y, \Delta z, r$  to all workers,  $p$ 
  {
    Do if time step,  $t$  not reach
    Time step,  $t < T$ 
    {
      Do if convergence not obtain
      Convergence maks  $|e_{i,j,k}^{(k)}| < \bar{\epsilon}$ 
      {
        Receive  $e_{i,j,k}^{(k)}(p)$  from workers
        Global convergence test maks  $|e_{i,j,k}^{(k)}| < \bar{\epsilon}$ 
        Send maks  $|e_{i,j,k}^{(k)}|$  to workers
        Loop++
      }
    }
    Receive data from workers
    Pvm_kill_processor
    Pvm_exit;
  }
}

```

obtained in Table 2 where the speed up measurement increases in both types of sizes of matrix.

However, there is a difference in the numerical methods used. Based on the results, AGE Brian shows the fastest speed up measurement followed by AGE Douglas and GSRB. This is because by having the fastest execution time, Brian able to maximize the speed of the processors. The results for 140 x 140 x 140 size of matrix shows that a much slower speed up is obtained. This is due to the reason that the execution time is slow and more data scale is inserted as input therefore, there is memory burden. However, when dealing with large size of data, the AGE Brian still proves to be the fastest compared to others.

Table 2. Numerical results on speed up measurement

No of processors	Brian		Douglas		GSRB	
	A	B	A	B	A	B
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	1.1922	1.1625	1.1076	1.1023	1.1687	1.1354
4	1.5262	1.4007	1.4081	1.3092	1.4123	1.3103
6	1.9991	1.6854	1.7840	1.5503	1.7125	1.4894
8	2.4608	2.0719	2.2765	1.8913	1.9489	1.7255
10	3.0515	2.5489	2.8386	2.3228	2.0838	2.0256
12	3.7167	2.9757	3.3593	2.7013	2.1634	2.1650
14	4.3661	3.4454	4.0132	3.1054	2.2235	2.2476
16	4.8078	3.9116	4.4604	3.3950	2.2934	2.3274
18	5.0739	4.1729	4.8687	3.6495	2.3711	2.4237
20	5.2690	4.2992	5.1011	3.8011	2.4167	2.4922

Legend: A – Matrix 100 x 100 x 100 B – Matrix 140 x 140 x 140

Table 3 provides the results on the efficiency on the three different numerical methods. The efficiency decreases as the simulations runs. This is due to the fact that 100% efficiency could not be obtained due to environmental factor such as heat and power dissipation. Despite that, the numerical method for Brian shows the highest efficiency value compared to the others. This is because Brian is able to neutralize the speed and memory during the simulation.

Table 3. Numerical results on efficiency performance

No of processors	Brian		Douglas		GSRB	
	A	B	A	B	A	B
1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	0.5961	1.1625	0.5538	1.1023	0.5843	1.1354
4	0.3815	1.4007	0.3520	1.3092	0.3531	1.3103
6	0.3332	1.6854	0.2973	1.5503	0.2854	1.4894
8	0.3076	2.0719	0.2846	1.8913	0.2436	1.7255
10	0.3051	2.5489	0.2837	2.3228	0.2082	2.0256
12	0.3097	2.9757	0.2799	2.7013	0.1803	2.1650
14	0.3119	3.4454	0.2867	3.1054	0.1588	2.2476
16	0.3005	3.9116	0.2788	3.3950	0.1433	2.3274
18	0.2819	4.1729	0.2703	3.6100	0.1317	2.4237
20	0.2635	4.2992	0.2551	3.8011	0.1208	2.4922

Legend: A – Matrix 100 x 100 x 100 B – Matrix 140 x 140 x 140

The case is still the same in a simulation using 140 x 140 x 140 matrix where Brian shows the best efficiency compared to AGE Douglas and GSRB, however the comparison between the size of matrix used will change the efficiency value. Even though AGE Brian proves to be the best method compare to the rest of method in both sizes of matrix, the efficiency decreases when more data points are being dealt in the simulation as more power is needed to run the simulation. Therefore a balanced between data points and the efficiency needed to be set so that it serves the objective for one particular simulation.

Table 4 shows the numerical results based on the effectiveness measurement. Again, the simulation is run on two types of matrix sizes. In both simulations, Brian proves to be best method in providing effectiveness during the simulation. This is followed by Douglas and GSRB method. However, the effectiveness decreases as a bigger matrix size is used. The effectiveness of the simulation decreases because the current speed obtained from the total processors is not effective enough to simulate a simulation which consist a large data points.

Alternative methods such as increasing the number of processors used or the power of each processor might provide a better effectiveness but an increase in cost may occur.

Table 4. Numerical results on effectiveness measurement

No of processors	Brian		Douglas		GSRB	
	A	B	A	B	A	B
1	0.0086	0.0040	0.0081	0.0038	0.0065	0.0036
2	0.0102	0.0047	0.0090	0.0042	0.0077	0.0041
4	0.0131	0.0056	0.0115	0.0050	0.0091	0.0047
6	0.0172	0.0068	0.0145	0.0060	0.0111	0.0052
8	0.0211	0.0083	0.0185	0.0073	0.0126	0.0063
10	0.0262	0.0102	0.0231	0.0089	0.0135	0.0073
12	0.0319	0.0120	0.0273	0.0104	0.0120	0.0078
14	0.0375	0.0139	0.0327	0.0119	0.0144	0.0081
16	0.0413	0.0157	0.0363	0.0130	0.0148	0.0084
18	0.0436	0.0168	0.0396	0.0140	0.0153	0.0088
20	0.0452	0.0173	0.0415	0.0146	0.0156	0.0090

Legend: A – Matrix 100 x 100 x 100 B – Matrix 140 x 140 x 140

Table 5 illustrates the results on numerical results on temporal performance measurement. Temporal performance measurement is a measurement that is conducted on the speed of a single die processor over microsecond. From the results, simulation using the Brian method proves to have the highest value of temporal performance as the simulation increases.

The same results also been obtained from the simulation running on a 140 x 140 x 140 matrix size where AGE Brian proves to be the better method compared to the other methods. However, due to larger size of matrix used the temporal performance in the 100 x 100 x 100 matrix proves to be better.

VI. CONCLUSION

In conclusion, the AGE Brian method proves to be the best method in both simulation sizes of matrix. However, a few criteria need to be observed before running on a larger size matrix because it will slower down the whole simulation and decrease the effectiveness and the efficiency of the simulation.

Table 5. Numerical results on temporal performance measurement

No of processors	Brian		Douglas		GSRB	
	A	B	A	B	A	B
1	0.0086	0.0040	0.0081	0.0038	0.0065	0.0036
2	0.0122	0.0054	0.0100	0.0047	0.0088	0.0047
4	0.0200	0.0079	0.0161	0.0066	0.0129	0.0062
6	0.0343	0.0114	0.0259	0.0092	0.0190	0.0080
8	0.0520	0.0173	0.0422	0.0138	0.0246	0.0108
10	0.0800	0.0261	0.0656	0.0208	0.0281	0.0149
12	0.1186	0.0356	0.0918	0.0281	0.0303	0.0170
14	0.1637	0.0477	0.1311	0.0371	0.0320	0.0183
16	0.1985	0.0615	0.1619	0.0443	0.0340	0.0196
18	0.2211	0.0700	0.1929	0.0512	0.0364	0.0213
20	0.2384	0.0743	0.2116	0.0556	0.0378	0.0225

Legend: A – Matrix $100 \times 100 \times 100$ B – Matrix $140 \times 140 \times 140$

ACKNOWLEDGEMENT

This material is based upon work supported by Short Term Grant, Universiti Teknologi Malaysia and Fundamental Research Grant Scheme (FRGS), Malaysian Ministry of Higher Education under Vote No. 77264 and 78598. Any opinions, findings, and conclusions or recommendations expressed in this material are those from the authors and do not necessarily reflect on the views of the Universiti Teknologi Malaysia and Malaysian Ministry of Higher Education.

REFERENCES

- [1] C.Hermans, "Laser Cutting of Glass", Proceedings of SPIE.. Mainz, Germany., pp. 219-226, 2002.
- [2] A. Tawai, and C. Panjapornpon, "Nonlinear Model-Based Control for Parabolic Partial Differential Equations Systems", Chiang Mai Journal Science, pp. 221-228, vol. 35, 2008.
- [3] P. Solana, P. Kapadia, J.M. Dowden, and P.J. Marsden, "An Analytical Model for the Laser Drilling of Metals with Absorption within the Vapour", Journal of Physics and Applied Physics, pp. 942-952, vol. 32, 1999.
- [4] W. Rheinboldt, "MANPAK: A Set of Algorithms for Computations on Implicity Defined Manifolds", Computer Science and Applied Mathematics, pp. 51-55, vol 5, 1992.
- [5] W.K. Giloi, "Parallel Programming Models and Their Interdependence with Parallel Architectures. Programming Models for Massively Parallel Computers", IEEE Proceeding. Berlin, Germany, pp. 2-11, 1993.
- [6] I. Kono, A. Nakanishi, S. Warisawa, and M. Mitsuishi "Non-Crack Laser Machining of Glass by using Absorbent Powder", American Society Precision Engineering Proceedings, Norfolk VA. pp. 1793-1797, 2005.
- [7] M.J. Weber, "Science and Technology of Laser Glass", Journal of Non Crystalline Solids, pp. 208-222, vol.123, 2003.

- [8] S.L. Lin, H.Y. Wang, and C.K. Chung "Simulation and Fabrication of Pyrex Glass Drilling using CO₂ Laser", Proceedings of the 2010th 5th IEEE International Conference on Nano/Micro Engineered and Molecular Systems. Xiamen, China, pp. 333-336. January 20-23, 2010
- [9] S.Sugihara, and K. Okazaki "Ceramic-Metal-Glass Bonding", Applications of Ferroelectronics IEEE 7th International Symposium, Fujisawa, Japan. pp. 429-43, 1990.
- [10] J. Jiao, and X Wang, "A Numerical Simulation of Machining Glass by Dual CO₂- Laser Beams". Optics and Laser Technology, Wuhan, China. pp. 297-301, 2007.
- [11] D. Kuo, D. Stan, and G. Rauch, "Laser Zone Texturing on Glass and Glass-Ceramic Substrates", IEEE Transactions on Magnetics, Fremont, California. pp. 944-949. 1997.
- [12] Z.H. Shen, S.Y. Zhang, J. Lu, and X.W. Ni, "Mathematical Modelling of Laser Induced Heating and Melting Solids." Optics and Laser Technology. pp. 533-537, vol. 33, 2000.
- [13] M. Papadrakakis, G. Stavroulakis, and A. Karatakis, "A New Era in Scientific Computing: Domain Decomposition methods in hybrid CPU-GPU Architectures", Computer Methods in Applied Mechanics and Engineering, pp. 1490-1508, vol. 200, 2011.
- [14] M.S. Sahimi, N.Alias, and E.Sundararajan, "The AGE algorithm for solving the Heat Equation in Three Space Dimensional and its Parallelization using PVM", Computational Science ICCS, pp. 918-927, vol.2073, 2001
- [15] F.Fairag, and M.S. Sahimi, "The Alternating Group Explicit (AGE) iterative method for solving Ladyzhenskaya-Type Model for Stationary Incompressible Vicious Flow", King Fahd University of Petroleum and Minearals, Saudi Arabia, pp. 323-343 vol. 61, 2008.
- [16] M. S. Sahimi, N. A. Mansor, N. M. Nor, N. M. Nusi, and N. Alias, "A high accuracy variant of the iterative alternating decomposition explicit method for solving the heat equation." International Journal of Simulation and Process Modelling, pp. 45-49, vol 2, 2006.