

## Numerical simulation of DIC drying process on matlab distributed computing server

Hafizah Farhah Saipan Saipol<sup>1</sup>, Norma Alias<sup>2</sup>

<sup>1</sup>Malaysia-Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Malaysia

<sup>2</sup>Institute for Industrial and Scientific Research, Universiti Teknologi Malaysia, Malaysia

### Article Info

#### Article history:

Received Feb 2, 2020

Revised Apr 4, 2020

Accepted Apr 18, 2020

#### Keywords:

Domain decomposition technique

Drying process

Matlab distributed computing server

Parallel algorithm

Sequential algorithm

### ABSTRACT

Instant controlled pressure drop, also known as DIC, is one of the drying techniques that has been used for texturing, extracting and drying various food products. Mathematical model has been used to explain the drying process, although most of the studies focused on the statistical regression model approach. Due to the limitations of regression model, which neglects the fundamental of dehydration process, this paper presents the development of mathematical models to detect, solve and visualize the three-dimensional (3D) heat and mass transfer in DIC drying process. Finite Difference Method (FDM) with central difference formula is used to discretize the mathematical models. A large sparse of system of linear equation (SLE), which represents the actual drying process, is solved by using some numerical methods, such as Jacobi (JB), Red Black Gauss Seidel (RBGS), Alternating Group Explicit with Douglas (AGED) and Brian (AGEB) methods. Based on the numerical results, high execution time and high computational complexity have been shown. In order to reduce the execution time and computational complexity, the parallel algorithm based on domain decomposition technique has been implemented on the MATLAB Distributed Computing Server (MDCS). The parallel algorithm of the numerical methods was evaluated and compared based on the parallel performance metrics: execution time, speed up, efficiency, effectiveness, temporal performance and granularity. From the parallel performance metrics, it was found that the PAGEB approach had better performance, followed by PAGED, PRBGS and PJB methods.

Copyright © 2020 Institute of Advanced Engineering and Science.  
All rights reserved.

### Corresponding Author:

Hafizah Farhah Saipan Saipol,  
Malaysia-Japan International Institute of Technology,  
Universiti Teknologi Malaysia Kuala Lumpur.  
Email: hafizah.farhah@utm.my

## 1. INTRODUCTION

Drying is one of the most common and important processes in food industry, where it involves simultaneous heat and mass transfer to remove moisture in food. It is a complex process because of the different product structures. The main purpose of drying is to reduce food weight, minimize biochemical reactions of degradation, and thus increase shelf life [1]. Various drying techniques were developed, such as solar, vacuum, freeze, osmotic, fluidized, and microwave drying [2]. Another drying technique is instant controlled pressure drop or DIC (Détente Instantanée Contrôlée) technique, which involves high pressure at a short heating time, followed by an instant pressure drop. The instant pressure drop results in volume expansion or texture modification [3]. The DIC technique is used for drying fruits and vegetables, cereal steaming and essential oils extraction [4-8].

Many research studies were done on mathematical models to solve, predict and visualize the drying process. The most widely used theoretical models are derived from Fick's 2nd law of diffusion. [9] used Fick's law by assuming that the drying kinetics is controlled by mass transfer of water within the material.

Many research studies had solved the mathematical model based on numerical approach due to the difficulties and complexities of the real drying mechanisms if they were solved based on analytical methods [10-12]. The numerical solution implementation is important to solve, predict and visualize the actual DIC drying process. Therefore, to solve and visualize the water content, temperature and pressure, based on time and dimension of region, a large sparse data of system of linear equations (SLE) is governed to approximate the mathematical model solution. The FDM scheme was chosen because the method formulates a set of discretized equations from the transport differential equations in differential manner [13-15]. Besides, this method is straightforward in determining the unknown values [16, 17]. Grid generation process in PDE exploits the mesh grid which involves the fine grained large sparse data by minimizing the interval size, increasing the model dimension and level of time steps. However, the increase in model dimension and matrix size comes with high execution time and high computational complexity costs.

To address this limitation, parallel algorithms approaches were considered. The parallel algorithms enable users to exploit their existing hardware to solve much larger problems at a minimal cost since many problems involve more than two variables and are even more complex to solve manually [18]. The implementation of the parallel algorithms based on the distributed parallel computing systems is used to carry out multiple tasks simultaneously with the aim to improve the performance and scalability of the software application [19]. High performance applications by using MDCS were widely used [20] and various research on parallel algorithms work has been performed to solve the drying applications [21, 22].

In the development of the parallel algorithms, a few important processes must be considered [23]. The parallelism approach is based on data dependencies. Different applications will utilize different aspects of communication. This research work involves tasks that share data through computations. The tasks are connected with each other through local and global communications. The local communication involves sending and receiving data between neighboring points in worker nodes and global communication requires communication with other tasks. The tasks must be equally distributed among workers to keep them busy throughout the computation process. Imbalance workload creates processors with less tasks which eventually become idle. In addition to load balancing, appropriate granularity is chosen to achieve better parallel performance [24]. The granularity that involves fine grain parallelism, where the size and execution time is too small, results in high communication overhead. On the contrary, if the granularity is too coarse, load imbalance may affect performance. However, the communication with the processor is less frequent due to large amount of computation. Therefore, the load and communication overhead must be in balance to achieve the best performance.

This paper is an improvement of the previous work [25] which was applied only on two-dimensional (2D) problems and involved a small sized problem. Since the mathematical model implementation was onto a complex system, larger problem size and higher dimensions are required. These will result in a higher computational complexity that may take a longer execution time when a single computer is used. Therefore, to speed up the computation and increase the efficiency of the numerical algorithm, the parallelization on MDCS is a very important process.

## 2. MATHEMATICAL MODEL

A theoretical 3D model of heat and mass transfer [25], which involves water content, temperature, pressure, time dependency, and dimension of region, are governed to describe the moisture movement and temperature behavior during food drying process by using DIC technique. A few assumptions were made:

- Changes in thickness, length and width are not considered.
- The main consideration is the change in moisture content and temperature in both time and space dimensions.
- The drying food material is in rectangular form.

The heat and mass transfer equation are given by,

$$\frac{\partial M}{\partial t} = v \left( \frac{\partial M}{\partial x} + \frac{\partial M}{\partial y} + \frac{\partial M}{\partial z} \right) + \frac{1}{v} \left( \frac{\partial P}{\partial x} + \frac{\partial P}{\partial y} + \frac{\partial P}{\partial z} \right) \quad (1)$$

$$\frac{\partial T}{\partial t} = -v \left( \frac{\partial T}{\partial x} + \frac{\partial T}{\partial y} + \frac{\partial T}{\partial z} \right) + \frac{v}{a} \left( \frac{\partial P}{\partial x} + \frac{\partial P}{\partial y} + \frac{\partial P}{\partial z} \right) \quad (2)$$

with initial and boundary conditions are given by

at  $t = 0; M(x, y, z, 0) = M_0, T(x, y, z, 0) = T_0,$  and  $P(x, y, z, 0) = P_0,$

$$\begin{aligned} M(0, y, z, t) &= M(1, y, z, t) = M(x, 0, z, t) = M(x, 1, z, t) = M(x, y, 0, t) = M(x, y, 1, t) = M_0, \\ T(0, y, z, t) &= T(1, y, z, t) = T(x, 0, z, t) = T(x, 1, z, t) = T(x, y, 0, t) = T(x, y, 1, t) = T_0 \\ \text{and } P(0, y, z, t) &= P(1, y, z, t) = P(x, 0, z, t) = P(x, 1, z, t) = P(x, y, 0, t) = P(x, y, 1, t) = P_0. \end{aligned} \quad (3)$$

Let  $M$  is the water content (g/mg dry matter),  $T$  is the temperature ( $^{\circ}C$ ),  $P$  is the steam pressure (Pa),  $t$  is the processing time (second),  $v$  is velocity vector (m/s) and  $\alpha = \rho C_p$ ,  $\rho$  is density ( $\text{kgm}^{-3}$ ) and  $C_p$  is the specific heat capacity ( $\text{m}^2\text{s}^{-2}\text{K}^{-1}$ ). The drying food is considered in a rectangular form with  $-L_x < x < L_x, -L_y < y < L_y$  and  $-L_z < z < L_z$ . The modelling will be discretized by using FDM with central difference formula. Given the  $\Delta t$  is the time step and  $\Delta x, \Delta y$  and  $\Delta z$  are the spatial step in  $x$ -,  $y$ - and  $z$ -axis, and the coordinates of the mesh points are  $x_i = i\Delta x$  where  $i = 1, 2, \dots, Ni + 1$ ,  $y_j = j\Delta y$  where  $j = 1, 2, \dots, Nj + 1$ ,  $z_k = k\Delta z$  where  $k = 1, 2, \dots, Nk + 1$ , and  $t_n = n\Delta t$  where  $n = 1, 2, \dots, Nn + 1$  with  $\Delta x = \frac{L_x}{(Ni + 1)}$ ,  $\Delta y = \frac{L_y}{(Nj + 1)}$ ,  $\Delta z = \frac{L_z}{(Nk + 1)}$  and  $\Delta t = \frac{T}{(Nt + 1)}$  the FDM becomes.

$$\begin{aligned} &\lambda\theta R_1 \left( M_{i+1,j,k}^{n+1} - M_{i-1,j,k}^{n+1} + M_{i,j+1,k}^{n+1} - M_{i,j-1,k}^{n+1} + M_{i,j,k+1}^{n+1} - M_{i,j,k-1}^{n+1} \right) + M_{i,j,k}^{n+1} \\ &= \lambda(1-\theta) \left[ R_1 \left( M_{i+1,j,k}^n - M_{i-1,j,k}^n + M_{i,j+1,k}^n - M_{i,j-1,k}^n + M_{i,j,k+1}^n - M_{i,j,k-1}^n \right) \right] \\ &\quad + S_1 \left( P_{i+1,j,k}^n - P_{i-1,j,k}^n + P_{i,j+1,k}^n - P_{i,j-1,k}^n + P_{i,j,k+1}^n - P_{i,j,k-1}^n \right) \\ &\quad + \lambda\theta S_1 \left( P_{i+1,j,k}^{n+1} - P_{i-1,j,k}^{n+1} + P_{i,j+1,k}^{n+1} - P_{i,j-1,k}^{n+1} + P_{i,j,k+1}^{n+1} - P_{i,j,k-1}^{n+1} \right) + M_{i,j,k}^n \end{aligned} \quad (4)$$

$$\begin{aligned} &\lambda\theta R_2 \left( T_{i+1,j,k}^{n+1} - T_{i-1,j,k}^{n+1} + T_{i,j+1,k}^{n+1} - T_{i,j-1,k}^{n+1} + T_{i,j,k+1}^{n+1} - T_{i,j,k-1}^{n+1} \right) + T_{i,j,k}^{n+1} \\ &= \lambda(1-\theta) \left[ R_2 \left( T_{i+1,j,k}^n - T_{i-1,j,k}^n + T_{i,j+1,k}^n - T_{i,j-1,k}^n + T_{i,j,k+1}^n - T_{i,j,k-1}^n \right) \right] \\ &\quad + S_2 \left( P_{i+1,j,k}^n - P_{i-1,j,k}^n + P_{i,j+1,k}^n - P_{i,j-1,k}^n + P_{i,j,k+1}^n - P_{i,j,k-1}^n \right) \\ \text{and } &+ \lambda\theta S_2 \left( P_{i+1,j,k}^{n+1} - P_{i-1,j,k}^{n+1} + P_{i,j+1,k}^{n+1} - P_{i,j-1,k}^{n+1} + P_{i,j,k+1}^{n+1} - P_{i,j,k-1}^{n+1} \right) + T_{i,j,k}^n \end{aligned} \quad (5)$$

where  $R_1 = v, S_1 = 1/v, R_2 = -v$  and  $S_2 = v/a$ . The scheme is explicit when  $\theta = 0$ , fully implicit when  $\theta = 1$  and Crank Nicolson when  $\theta = 1/2$ . This discretization (2) can be written in the form of  $Au=f$  where  $A$  is the coefficient matrix,  $u$  corresponds to vector of unknowns:  $M$  and  $T$ , and  $f$  is the right-hand side vector.

### 3. METHODOLOGY

This system of linear equation from the discretization is solved using some numerical methods: JB, RBGS, AGED and AGE methods. JB method is a simple and fundamental method; thus, it is used as a benchmark for other numerical methods. The method can be shown in discrete equation as,

$$u_i^{(n+1)} = \frac{1}{a_{ii}} \left[ f_i - \sum_{j=1}^{Ni} a_{ij} u_j^{(n)} \right], \quad i = 1, 2, \dots, Ni \quad (6)$$

The RBGS method contains two subdomains,  $\Omega^R$  and  $\Omega^B$ . The grid calculation at:

$$\Omega^R : u_i^{(n+1)} = \frac{1}{a_{ii}} \left[ f_i - \sum_{j>i} a_{ij} u_j^{(n)} - \sum_{j<i} a_{ij} u_j^{(n)} \right], \text{ for } i = 1, 3, 5, \dots, Ni - 1. \tag{7a}$$

$$\Omega^B : u_i^{(n+1)} = \frac{1}{a_{ii}} \left[ f_i - \sum_{j>i} a_{ij} u_j^{(n+1)} - \sum_{j<i} a_{ij} u_j^{(n+1)} \right], \text{ for } i = 2, 4, 6, \dots, Ni. \tag{7b}$$

The AGEB method employs fractional splitting and together with linear interpolation method. Matrix A which is a symmetric and positive definite matrix has the splitting strategy where  $A = G_1 + G_2 + G_3 + G_4 + G_5 + G_6$ . Given that the formula for AGEB method is as following:

$$(G_1 + rI) u_{[xy]}^{(N+\frac{1}{7})} = (rI - G_2 - G_3 - G_4 - G_5 - G_6) u_{[xy]}^{(N)} + f \tag{8a}$$

$$(G_2 + rI) u_{[xy]}^{(N+\frac{2}{7})} = G_2 u_{[xy]}^{(N)} + r u_{[xy]}^{(N+\frac{1}{7})} \tag{8b}$$

$$(G_3 + rI) u_{[xy]}^{(N+\frac{3}{7})} = G_3 u_{[xy]}^{(N)} + r u_{[xy]}^{(N+\frac{2}{7})} \tag{8c}$$

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

$$(G_5 + rI) u_{[xy]}^{(N+\frac{5}{7})} = G_5 u_{[xy]}^{(N)} + r u_{[xy]}^{(N+\frac{4}{7})} \tag{8e}$$

$$(G_6 + rI) u_{[xy]}^{(N+\frac{6}{7})} = G_6 u_{[xy]}^{(N)} + r u_{[xy]}^{(N+\frac{5}{7})} \tag{8f}$$

$$u_{[xy]}^{(N+1)} = u_{[xy]}^{(N)} + 2 \left( u_{[xy]}^{(N+\frac{6}{7})} - u_{[xy]}^{(N)} \right) \tag{8g}$$

**4. SEQUENTIAL AND PARALLEL ALGORITHM**

The 3D mathematical model for drying problem is solved straightforward by using numerical methods. The sequential algorithm runs on MATLAB 7.6.0 (R2008a) programming and supported by Intel@Core™ processor. The sequential algorithm is then extended to parallel algorithm. The parallel algorithm is based on the domain decomposition technique. The 3D domain problem was partitioned by column-wise distribution into several tasks which consists of a grid of subdomains. Each task was assigned to a number of workers. The parallel algorithm was implemented on MDCS client-workers environment with star topology on Beowulf cluster, which consists of a homogeneous computer cluster system containing 8 CPUs. Each CPU has a specification of Intel@ Core™ Duo processor under Red Hat Linux 2.6 operating system. All the computers were connected through Local Area Network (LAN). The communication involved sending data to and receiving from adjacent workers, and between client and workers. The sending and receiving data must be in the same dimensions in order to avoid a deadlock detection. The sequential and parallel algorithm of these numerical methods is illustrated in Figure 1.

The algorithm starts by defining the parameter values and initial and boundary conditions. The sequential algorithm computes directly through the sequence as shown in Figure 1(a). The sequential algorithm is designed to run on a single CPU. However, in order to be more cost effective by increasing speedup and efficiency of the programming, the computation is performed simultaneously on different workers as shown in Figure 1(b). The client sends the required workload data to the workers. The workers start the computation by calculating the values of  $M(i, j, k)$  and  $T(i, j, k)$  at the new time level for each worker.

Local communication involves when a set of data is required from the neighboring worker. For example, worker  $n$  requires data  $(i-1, j, k)$  from the closest neighbor; worker  $n-1$  and data  $(i+1, j, k)$  from worker  $n+1$ . Therefore, worker  $n$  requests from the neighbor to send data to obtain a new value of  $M(i, j, k)$  at  $(n+1)$  time. The data must be transferred between workers to proceed to the next computation. Once the local convergence criterion is met, the maximum error for each worker is sent to the client and compared with the global error. The worker will stop the computation once the global convergence criterion is met.

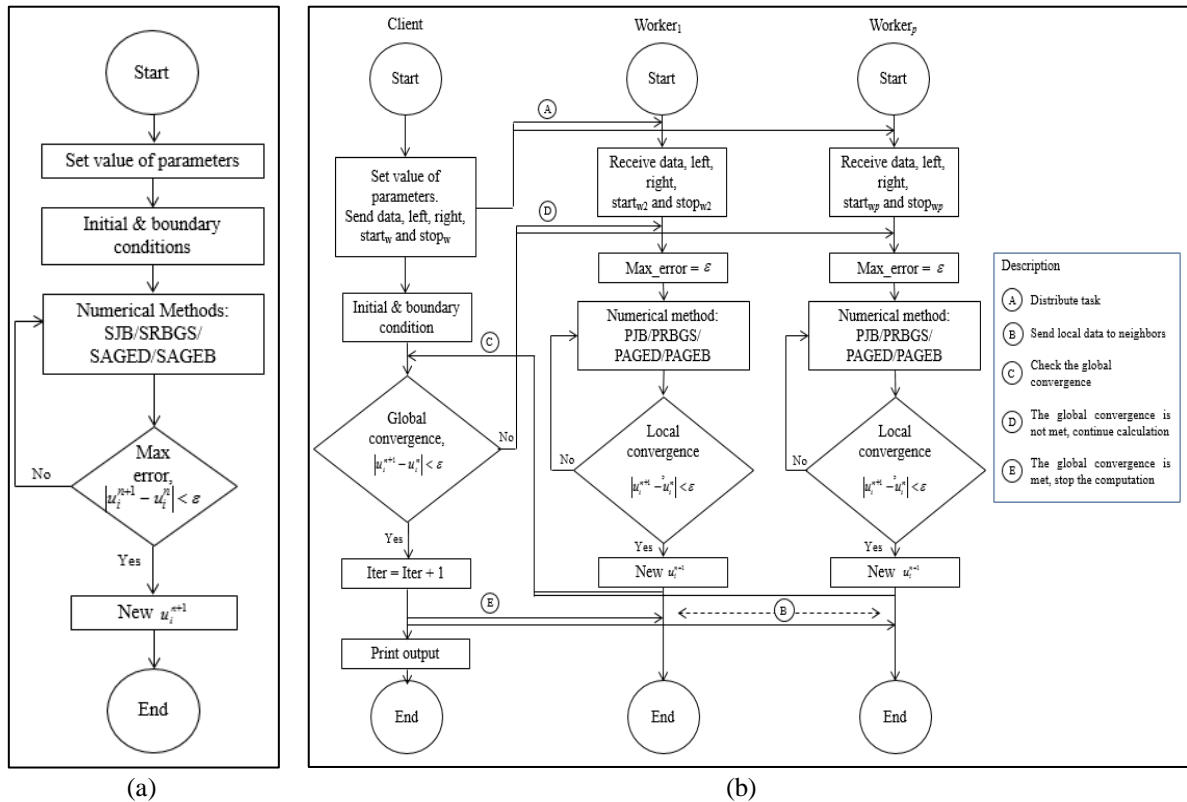


Figure 1. (a) Sequential and (b) parallel algorithm

**5. RESULTS AND DISCUSSION**

The analysis from the sequential algorithm was measured based on numerical results: execution time, number of iterations, maximum error and root mean square error (RMSE). The visualization of the heat and mass transfer for the 3D model by using MATLAB software and COMSOL Multiphysics software are presented in Figure 2.

Figure 2(a) visualizes the mass transfer inside the food with respect to  $x$ - axis,  $y$ - axis and  $z$ -axis. At different surfaces of  $k$ , the moisture content shows a decrement from the boundary to the inner surface of drying food. Figure 2(b) shows the internal food temperature with respect to  $x$ - axis,  $y$ - axis and  $z$ -axis. With the increasing surface of  $k$ , the temperature was found to be higher at the boundary as compared to the inner surface. The 3D model enables more details for different surfaces of  $k$  as compared to the lower dimension model which involved only one surface in visualizing the dehydration process. In order to make a clearer visualization of the dehydration process by using the DIC technique, the 3D model was simulated by using COMSOL Multiphysics software. Figure 2(c) shows the high temperature and high pressure that stimulate high moisture transfer at the boundary to the surrounding as compared to the inner surface. The numerical results for the 3D heat and mass equations are presented in Table 2 by using the value of parameters given in Table 1.

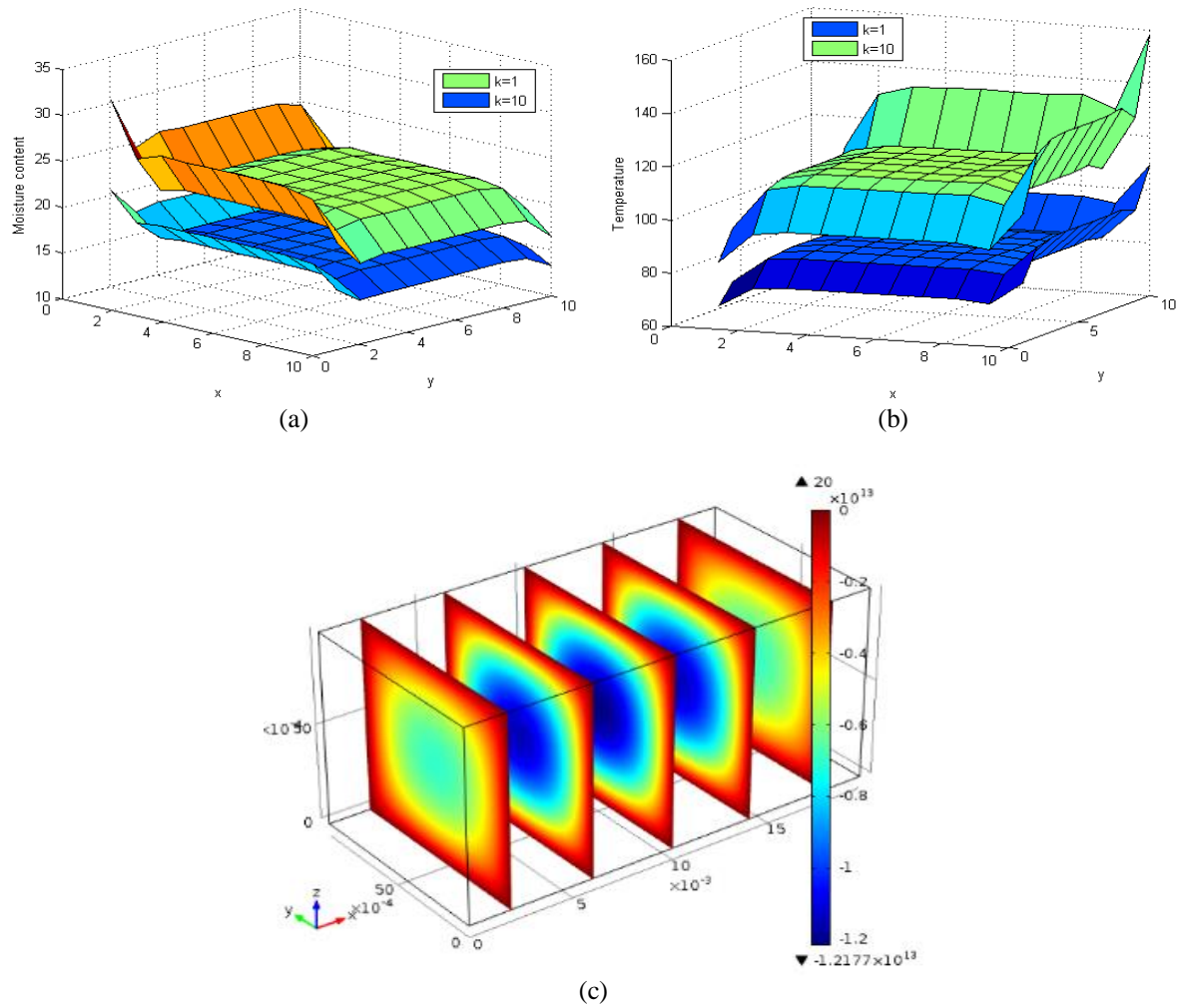


Figure 2. The visualization of (a) mass transfer, (b) heat transfer and (c) using Comsol Multiphysics software

Table 2 presents the numerical results of SAGEB, SAGED, SRBGS and SJB methods. SAGEB represents the AGEB method for sequential algorithm. The SJB is used as a benchmark to compare its numerical results with other numerical methods. The computational complexity is measured based on the amount of arithmetic operations, such as additions, subtractions, multiplication and divisions, required in the sequential algorithm. It is performed to estimate the complexity of the algorithm before it is run on a single or many computers. Different amount of computational complexity resulted in different execution times per number of iterations. Although the computational complexity of SAGEB and SAGED are higher than SRBGS and SJB, they took less time and converged quicker. This is the reason why the SAGEB method is 72% better than SJB in terms of execution time and iteration number. It is followed by SAGED method and SRBGS, which took a smaller number of iterations and execution time as compared to the SJB method. The convergence rate is measured based on the maximum error and RMSE. SAGEB has better convergence rate due to the value of maximum error and RMSE are the smallest as compared to other numerical methods.

Table 1. Value of parameters for heat and mass equations

Parameters	Symbols	Value	Parameters	Symbols	Value
Length (m)	$L$	0.005	Treatment time (s)	$Time$	8
Lambda	$\lambda$	0.05	Moisture content ( $kg/m^3$ )	$M$	20
Tolerance	$\varepsilon$	$10^{-10}$	Temperature ( $^{\circ}C$ )	$T$	100
Total grid	$N_i, N_j, N_k$	41	Pressure (kPa)	$P$	1.3
Spatial step	$\Delta x = \Delta y = \Delta z$	$1.22e-4$	Flow velocity m/s	$v$	3
Time step	$\Delta t$	$1.22e-5$	Density ( $kg/m^3$ )	$\rho$	0.946
Weight	$\theta$	1	Specific heat capacity ( $m^2 \cdot s^{-2} \cdot K^{-1}$ )	$C_p$	1.996

Table 2. The numerical results for 3D heat and mass equations

Eq.	Method	Comput. Complex	$r$	Exec Time (s)	No. of Iter.	Max Error	RMSE
Mass	SAGEB	$215(m-1)^3$	2	140.85	31	$5.06e-11$	$6.47e-12$
	%			72.6	72.1		
	SAGED	$233(m-1)^3$	1.06	152.34	33	$7.49e-11$	$4.19e-12$
	%			70.3	70.3		
	SRBGS	$14(m-1)^3$		281.56	56	$9.38e-11$	$1.93e-11$
	%			45.2	49.5		
Heat	SJB	$14(m-1)^3$		513.76	111	$9.38e-11$	$1.45e-11$
	SAGEB	$215(m-1)^3$	1.73	153.14	33	$4.97e-11$	$9.24e-12$
	%			73.5	73.2		
	SAGED	$233(m-1)^3$	1.06	137.17	35	$8.70e-11$	$5.04e-12$
	%			76.2	71.5		
	SRBGS	$14(m-1)^3$		333.30	62	$9.76e-11$	$2.15e-11$
%			42.3	49.6			
	SJB	$14(m-1)^3$		577.55	123	$9.76e-11$	$1.61e-11$

The performance of the sequential algorithms in terms of the execution time were improved by performing the parallel computing strategy. The parallel algorithms were measured based on their parallel performance metrics (PPM), such as speedup, efficiency, effectiveness, temporal performance and granularity [26] which is shown in Figure 3.

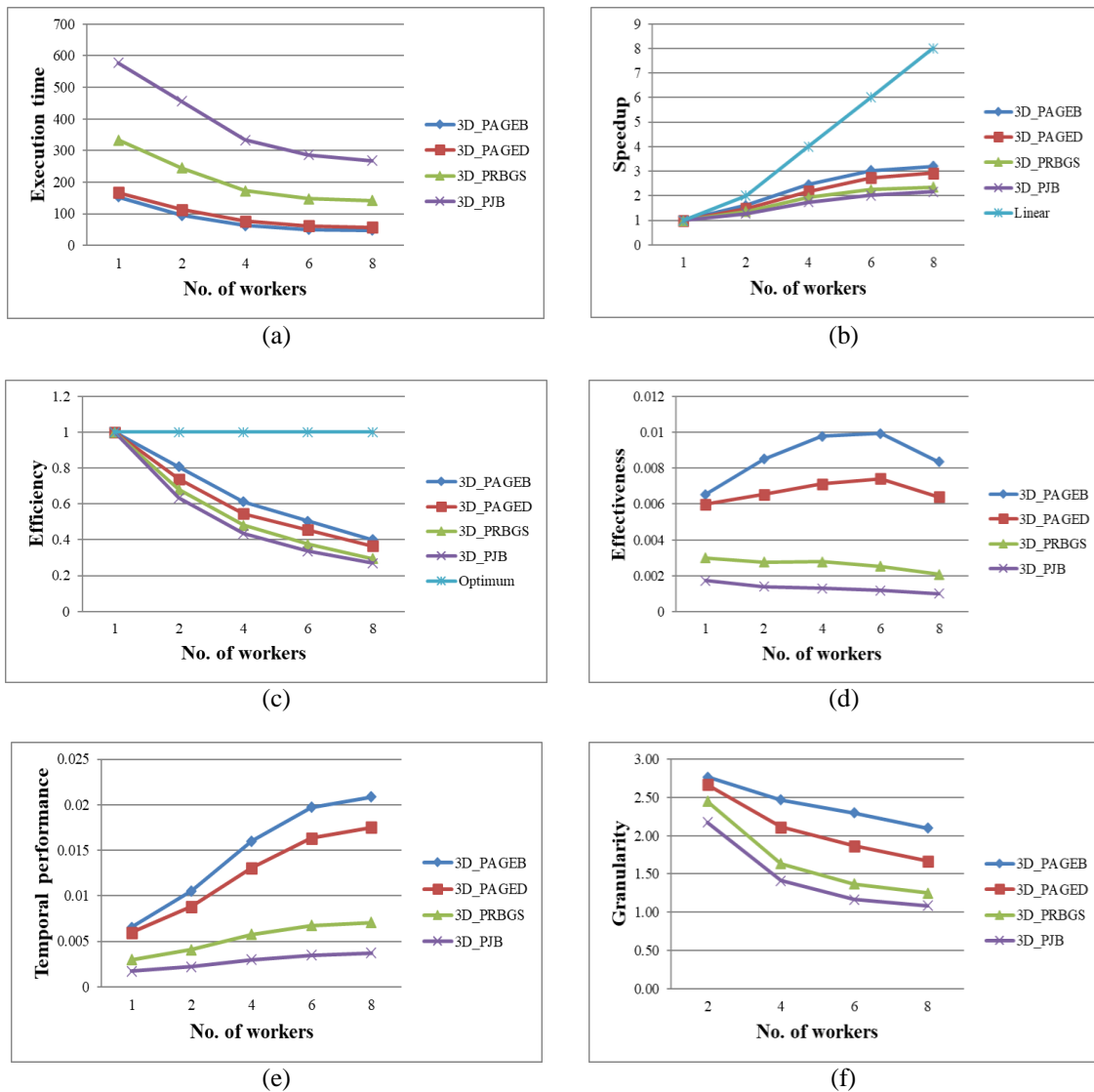


Figure 3. Parallel performance metrics

The execution time for each worker is illustrated in Figure 3(a) and it is found that as the number of workers increased, the execution time decreased. Based on the time execution, we can calculate how much faster the computation takes to complete when  $p$  workers are used as compared to one worker. Speedup is used to measure performance and to show the effect of any performance enhancement in parallel algorithms. Figure 3(b) shows that the speedup gradually increased over workers even though the execution time was decreased with the increasing number of workers. The PAGEB method had better speedup as compared to PAGED, PRBGS, and PJB methods.

Along with the increment in speedup, the efficiency decreased below its ideal level with the increasing number of workers. This was due to overhead factors, such as communication and idling. A process becomes idle because of lack of computation or data due to load imbalance, where each worker does not have equal problem size. From Figure 3(c), the efficiency for the PAGEB method gradually decreased as compared to PAGED, PRBGS and PJB methods.

Effectiveness shows the impact of the numerical methods based on cost and accuracy. This metric analyzed the efficiency over latency for  $p$  processors. According to the value obtained from the formula, it is possible to determine the suitable number of workers needed to run the parallel algorithms. Based on Figure 3(d), the effectiveness of the PAGEB method was the highest, followed by PAGED, PRBGS, and PJB methods. The effectiveness for PAGEB and PAGED methods increased with the increasing number of workers. However, the effectiveness decreased when the worker used was more than 6. It can be summarized that the optimum worker used for this parallel algorithm was less than 7 workers. Meanwhile, for PJB and PRBGS methods, the effectiveness was decreased with respect to the number of workers. Therefore, by comparing with the numerical methods, the highest value of effectiveness showed that the PAGEB method had high accuracy and low computation cost.

Another metric used to analyze the parallel performance is temporal performance. This metric is suitable for comparing the different algorithm performances to solve the same application. The highest value of temporal performance showed the best performance of parallel algorithms. Figure 3(e) shows PAGEB has the highest value followed by PAGED, PRBGS and PJB methods. The highest value of temporal performance showed that the algorithm for the PAGEB method was executed at the least time. Therefore, the PAGEB method was considered a better method as compared to other methods. The granularity analysis is presented in Figure 3(f). The granularity is decreased as the number of workers increased for each method. The PAGEB method had the largest granularity, followed by PAGED, PRBGS, and PJB methods. Large granularity value for the PAGEB method showed that the computational time was larger than the communication time.

## 6. CONCLUSION

The study described and visualized heat and mass transfer in food with respect to the  $x$ -axis,  $y$ -axis and  $z$ -axis during DIC drying process. Our results showed that the moisture content decreased from the boundary to the inner surface of drying food at different surfaces of  $k$ . On the other hand, the temperature was found to be higher at the boundary as compared to the inner surface with an increasing  $k$ . Based on these results, the 3D mathematical model had enabled more details for different surfaces of  $k$  as compared to the lower dimension model which involved only a surface in visualizing the dehydration process. Besides, based on the numerical results and parallel performance metrics, the numerical methods could solve the 3D heat and mass transfer equation that was sequentially executed and in parallel. The implementation of parallel algorithms on MDCS enhanced the execution time and improved the computation speedup and efficiency. Future research can improve the parallel algorithms for shared memory implementation because it does not involve message passing strategy among workers, which makes the implementation less expensive and more scalable.

## ACKNOWLEDGEMENTS

Authors would like to thank Malaysia-Japan International Institute of Technology, Ibnu Sina Institute, Universiti Teknologi Malaysia and UTMER research grant 18J22 for the financial support.

## REFERENCES

- [1] K. K. Hnin, *et al.*, "Emerging Food Drying Technologies with Energy-Saving Characteristics: A Review," *Drying Technology*, vol. 37, pp. 1465-1480, 2019.
- [2] V. R. Sagar and P. Suresh Kumar, "Recent Advances in Drying and Dehydration of Fruits and Vegetables: A Review," *Journal of Food Science and Technology*, vol. 47, pp. 15-26, 2010.



- [3] C. H. Hou, *et al.*, "Effects of Drying Methods and Raw Material Ratio on The Quality of Restructured Carrot-Potato Chips," *Modern Food Science and Technology*, vol. 33, pp. 140-147, 2017.
- [4] T. Allaf and K. Allaf, "Instant Controlled Pressure Drop (DIC) in Food Processing: From Fundamental to Industrial Applications," *Springer-Verlag*, New York, 2014.
- [5] B. Ben Amor and K. Allaf, "Impact of Texturing Using Instant Pressure Drop Treatment Prior To Solvent Extraction of Anthocyanins from Malaysian Roselle (*Hibiscus Sabdariffa*). *Food Chemistry*, vol. 115, pp. 820-825, 2009.
- [6] I. Pilatowski, *et al.* "The Instant Controlled Pressure Drop Process as A New Post-Harvesting Treatment of Paddy Rice: Impacts on Drying Kinetics and End Product Attributes. *Food Bioprocess Technology*, vol. 3, pp. 901-907, 2010.
- [7] I. Zarguili, *et al.*, "A Mathematical Model to Describe the Change in Moisture Distribution in Maize Starch During Hydrothermal Treatment," *International Journal of Food Science and Technology*, vol. 44, pp. 10-17, 2009.
- [8] X. Li, *et al.*, "Effect of The Moisture Equilibrium Process On The Expansion Behavior Of Instant Controlled Pressure Drop (DIC) Drying Of Dried Apple Cubes," *Journal of the Science of Food and Agriculture*, vol. 100, pp. 1635-1642, 2020.
- [9] M. A. Haddad, *et al.*, "Fruits & Vegetables Drying Combining Hot Air, DIC Technology and Microwaves," *International Journal of Food Engineering*, vol. 4, p. 9, 2008.
- [10] H. Ambarita and A. H. Nasution, "A Numerical Solution to Simultaneous Heat and Mass Transfer of Convective Drying of Food," *1st SEMIRATA-International Conference on Science and Technology, SEMIRATA-ICST 2018*, vol. 1116, art. no. 032002, 2008.
- [11] G. Arunsandeep and V. P. Chandramohan "Numerical Solution for Determining the Temperature and Moisture Distributions of Rectangular, Cylindrical, and Spherical Objects During Drying," *Journal of Engineering Physics and Thermophysics*, vol. 91, pp. 895-906, 2018.
- [12] A. M. Castro, *et al.*, "Mathematical Modelling of Convective Drying of Fruits: A Review," *Journal of Food Engineering*, vol. 223, pp. 152-167, 2018.
- [13] V. Mitrevski, *et al.* "Mathematical Modelling of Far-Infrared Vacuum Drying of Apple Slices," *Thermal Science*, vol. 23, pp. 12, 2009.
- [14] H. Wang, *et al.*, "Mathematical Model and Numerical Simulation of Biological Porous Medium During Hot Air Drying," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 30, pp. 325-333, 2014.
- [15] G. G. Botte *et al.*, "Comparison of Finite Difference and Control Volume Methods for Solving Differential Equations," *Computers and Chemical Engineering*, vol. 24, pp. 2633-2654, 2000.
- [16] F. P. Incopera and D. P. DeWitt, "Fundamentals of Heat and Mass Transfer," Wiley, New York, 1996.
- [17] S. Mazumder, "Comparative Assessment of the Finite Difference, Finite Element, and Finite Volume Methods for a Benchmark One-Dimensional Steady-State Heat Conduction Problem," *Journal of Heat Transfer*, vol. 139, art. no. 071301, 2017.
- [18] R. Rihartanto, *et al.*, "Performance of Parallel Computing in Bubble Sort Algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 7, pp. 861-866, 2017.
- [19] M. N. F. Jamaluddin, *et al.*, "Performance Comparison of Java Based Parallel Programming Models," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, pp. 1577-1583, 2019.
- [20] X. Guo, *et al.*, (2016) "HPC Matlab: A Framework for Fast Prototyping of Parallel Applications in Matlab," *Procedia Computer Science*, vol. 80, pp. 1461-1472, 2016.
- [21] N. Alias, *et al.* "Parallelization of Temperature Distribution Simulations for Semiconductor and Polymer Composite Material on Distributed Memory Architecture," *Lecture Notes Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5698, pp. 392-398, 2009.
- [22] P. Poonnoy, *et al.*, "Parallel Dynamic Artificial Neural Network for Temperature and Moisture Content Predictions in Microwave-Vacuum Dried Tomato Slices," *Chemical Product and Process Modeling*, vol. 2, pp. 23, 2007.
- [23] S. M. Borodo, *et al.*, "Big Data Platforms and Techniques," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 1, pp. 191-200, 2016.
- [24] J. Kwiatkowski and L. Olech, "Scalability Model Based on The Concept of Granularity. *Lecture Notes Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 9574, pp. 189-198, 2016.
- [25] N. Alias, *et al.*, "Chronology of DIC Technique Based on The Fundamental Mathematical Modeling and Dehydration Impact," *Journal of Food Science and Technology*, vol. 51, pp. 3647-3657, 2014.
- [26] N. Alias, *et al.*, "Parallel Computing of Numerical Schemes and Big Data Analytic for Solving Real Life Applications," *Jurnal Teknologi*, vol. 78, pp. 151-162, 2016.