

THE IMPROVEMENT OF THE COMPUTATIONAL PERFORMANCE OF THE ZONAL MODEL POMA USING PARALLEL TECHNIQUES

¹Yao Yu, ²Ahmed Cherif Megri, ³Kenneth M. Flurchick and ⁴Dukka Bahadur K.C.

^{1,3,4}Department of Computational Science and Engineering,

²Civil, Architectural and Environmental Engineering, CERT Research Center,
North Carolina A and T State University, Greensboro, North Carolina, USA

Received 2013-04-12; Revised 2013-07-16; Accepted 2014-02-15

ABSTRACT

The zonal modeling approach is a new simplified computational method used to predict temperature distribution, energy in multi-zone building and indoor airflow thermal behaviors of building. Although this approach is known to use less computer resource than CFD models, the computational time is still an issue especially when buildings are characterized by complicated geometry and indoor layout of furnishings. Therefore, using a new computing technique to the current zonal models in order to reduce the computational time is a promising way to further improve the model performance and promote the wide application of zonal models. Parallel computing techniques provide a way to accomplish these purposes. Unlike the serial computations that are commonly used in the current zonal models, these parallel techniques decompose the serial program into several discrete instructions which can be executed simultaneously on different processors/threads. As a result, the computational time of the parallelized program can be significantly reduced, compared to that of the traditional serial program. In this article, a parallel computing technique, Open Multi-Processing (OpenMP), is used into the zonal model, Pressurized zOnal Model with the Air diffuser (POMA), in order to improve the model computational performance, including the reduction of computational time and the investigation of the model scalability.

Keywords: Zonal Model, POMA, OpenMP, Computational Time, Computing Parallelization

1. INTRODUCTION

A zonal model is a numerical modeling approach especially developed for indoor environment building application, such as the investigation of indoor airflow movement and temperature distribution, the estimation of building energy demand and so on (Megri and Haghghat, 2007). Unlike the single or multi-zone modeling approaches where a uniform temperature distribution in a room is assumed, zonal models geometrically subdivide a room into several zones where simplified mass and energy conservation equations are utilized along with the power law model (Haghghat *et al.*, 2001; Yu, 2012). As a result, detailed

information can be obtained by using zonal models, such as the indoor temperature distribution. Although the Computational Fluid Dynamics (CFD) models may provide the required detailed results as well, the complicated model structure of CFD techniques make these models difficult to be used effectively into situations involving a number of rooms over long periods of time (Clarke *et al.*, 1995), in consideration of the computational time. As a matter of fact, the zonal modeling approach is an intermediate numerical method between single/multi-zone and CFD models. It combines the simplicity of single and multi-zone models with the comprehensiveness of CFD models and becomes a better substitute to predict detailed thermal and flow behaviors

Corresponding Author: Yao Yu, Department of Computational Science and Engineering, North Carolina A and T State University, Greensboro, North Carolina, USA

in buildings (Megri *et al.*, 2005; Jiru and Haghghat, 2004; Megri and Yu, 2010; Yu and Megri, 2011).

Although the zonal modeling approach is known to use less computer resource than CFD models, the computational time is still an issue, compared to single or multi-zone models, especially when buildings are characterized by a large number of rooms with complicated geometry and indoor layout of furnishings. Therefore, it is not feasible for the zonal modeling approach to be used in this kind of buildings, because of the huge computational work and time, which limits the wide applications of the zonal models into building environment. One way to solve this problem is to improve the computational performance by using advanced computing techniques, such as parallel computing techniques.

In the traditional serial computers, instructions are executed sequentially. The Central Processing Unit (CPU) executes a program instruction after instruction until to the end. In this execution mode, one instruction has to wait for the completion of the previous one. Therefore, the computational time is dependent on how fast data can be moved through the CPU. Unfortunately, based on the current development of computer techniques, the processing time is still significant if a large number of data are involved. Parallel computing techniques provide a solution for the large-number-data problem. In the parallel techniques, a large number of data can be decomposed into the ones with smaller size. Each of them can be performed or executed simultaneously on different processors/threads. The work of the decomposition can be accomplished by Message Passing Interface (MPI) (Pacheco, 1997), Open Multi-Processing (OpenMP, 2013). Consequently, the parallel technique is not only able to improve the computational performance and reduce the execution time, but also to allow the completion of a work that is impossible to perform in a serial computer within a reasonable time period.

In this study, the zonal model, Pressurized zOnal Model with the Air diffuser (POMA) (Haghghat *et al.*, 2001), was developed using C language and then executed on a High Performance Computer (HPC) with multiple threads of control. The parallel technique, OpenMP, was used in this program for the data decomposition. The feasibility of using the parallel technique into the zonal model POMA was investigated. The improvement of performance in terms of computational time was demonstrated. Additionally, the investigation of the model scalability was accomplished.

2. DESCRIPTION OF THE ZONAL MODEL

The zonal model POMA (Haghghat *et al.*, 2001) is based on the fundamental conservation equations (both mass and energy balances) and the power law model to predict temperature and air flow distributions within a room.

The mass and energy balances equations are described below.

Mass balance equation:

$$\frac{dM_{ij}}{dt} = \sum \dot{m}_{ij} + \dot{m}_{source} + \dot{m}_{sink} \text{ [kg / s]} \quad (1)$$

Where:

- M_{ij} (kg) = The fluid mass of zone ij
- \dot{m}_{ij} (kg / s) = The rate of mass from surrounding zone to zone ij
- \dot{m}_{source} (kg / s) = The rate of mass supplied as a source in the zone, i.e., the air flow from a diffuser
- \dot{m}_{sink} (kg / s) = The rate of a mass removed from the zone

Energy balance equation:

$$\frac{dQ_{ij}}{dt} = \sum \dot{q}_{ij} + \dot{q}_{source} + \dot{q}_{sink} \text{ [w]} \quad (2)$$

Where:

- Q_{ij} (J) = The heat energy in zone ij
- \dot{q}_{ij} (W) = The rate of heat energy from surrounding zones to zone ij
- \dot{q}_{source} (W) = The rate of heat energy supplied by a source in the zone
- \dot{q}_{sink} (W) = The rate of heat energy removed from the zone

In Equation 1, \dot{m}_{ij} is defined based on $\dot{m}_{ij} = \rho k A (\Delta P)^n$, where ρ (kg/m³) is the fluid density; A (m²) is the cross-sectional area between two zones; ΔP (Pa) is the pressure difference between two zones; k (m s⁻¹ Pa⁻ⁿ) is the flow coefficient, which is assumed to be 0.83 for all zones; n is called flow exponent, which is 0.5. This above equation to define \dot{m}_{ij} is called the Power Law Model (PLM), which is used to describe the mass flow rate from one zone to another.

In POMA, pressures in each zone are assumed to be horizontally homogenous and vertically affected only by the gravity. Thus, to determine the ΔP in the PLM,

pressure differentials between any two zones have to be computed differently.

The horizontal pressure difference ΔP (**Fig. 1**) used in the PLM can be defined as Equation 3 and 4:

$$\Delta P = P_1 - P_2 = (P_{ref1} - \rho_1gz) - (P_{ref2} - \rho_2gz) = \Delta P_{ref} - \Delta\rho gz = \Delta\rho g(z_n - z) \quad (3)$$

Where:

- ΔP_{ref1} (Pa) = The reference pressure at the bottom of zone 1
- ΔP_{ref2} (Pa) = The reference pressure at the bottom of zone 2
- P_1 (Pa) = The pressure in zone 1 at certain height level z
- P_2 (Pa) = The pressure in zone 2 at certain height level z
- z (m) = The height level from the bottom of a zone
- ρ_1 (kg/m³) = The density of the fluid in zone 1
- ρ_2 (kg/m³) = The density of the fluid in zone 2
- ΔP (Pa) = The pressure difference at the same height level
- ΔP_{ref} (Pa) = The reference pressure difference which equals to $\Delta P_{ref1} - \Delta P_{ref2}$
- $\Delta\rho$ (kg/m³) = The fluid density difference between zone 1 and zone 2, which equals to $\rho_1 - \rho_2$
- g (9.81 m/s²) = The gravitational acceleration
- z_n (m) = The height of neutral plane at which height, the pressure difference between two zones is equal to zero, which is defined as:

$$z_n = \frac{\Delta P_{ref}}{\Delta\rho g} \quad (4)$$

And H (m) is the height of the zone (**Fig. 1**).

Therefore, the total horizontal mass flow rate \dot{m} [kg/s] can be determined as follows in Equation 5:

$$\dot{m} = \dot{m}_{0-z_n} + \dot{m}_{z_n-H} = \int_0^{z_n} kA\rho|\Delta P|^n dz + \int_{z_n}^H kA\rho|\Delta P|^n dz \quad (5)$$

Where:

- k, A and ΔP = The same as those in the PLM
- \dot{m}_{0-z_n} (kg/s) = The mass flow rate from height zero to z_n
- \dot{m}_{z_n-H} (kg/s) = The mass flow rate from height z_n to H
- ρ (kg/m³) = The fluid density of the zone from which the flow comes and can be calculated using the ideal gas law

The vertical pressure difference ΔP (**Fig. 2**) used in the PLM can be defined as:

$$\Delta P = P_{ref2} - P_{ref1} - \rho_2gH \quad (6)$$

Where:

- P_{ref1} (Pa) = The reference pressure at the bottom of zone 1
- P_{ref2} (Pa) = The reference pressure at the bottom of zone 2
- ΔP (Pa) = The pressure difference at the cross-sectional boundary between two zones
- ρ_2 (kg/m³) = The density of the fluid in zone 2
- H (m) = The height of the zone
- g (9.81 m/s²) = The gravitational acceleration

Thus, the total vertical mass flow rate \dot{m} [kg/s] can be calculated from Equation 6 and the PLM directly.

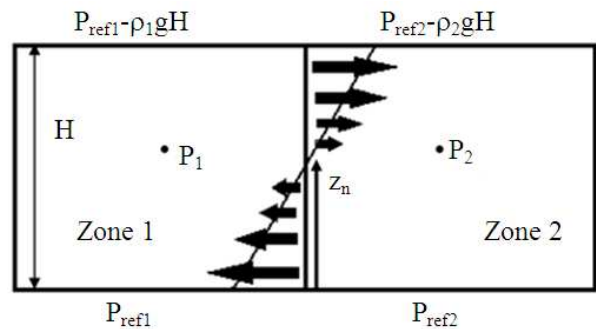


Fig. 1. Two horizontal zone divisions

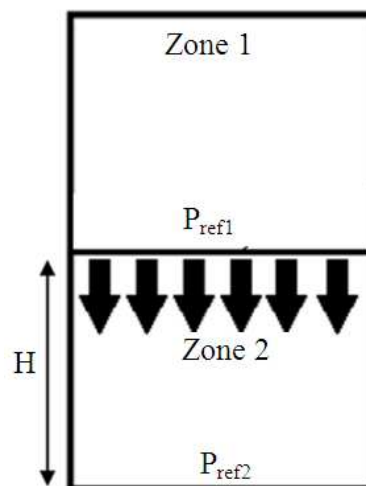


Fig. 2. Two vertical zone divisions

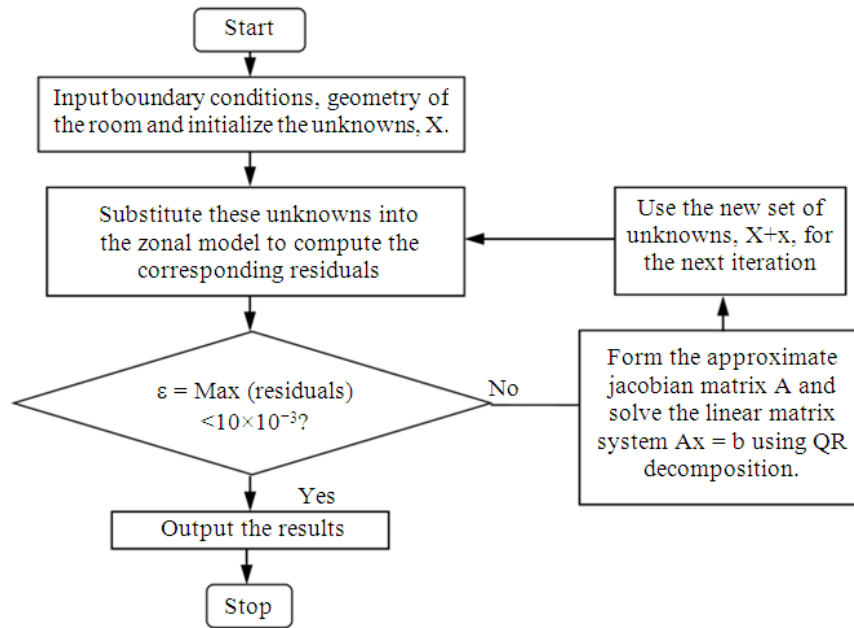


Fig. 3. Structure of the zonal model

In Equation 2, \dot{q}_{ij} is defined as, $\dot{q}_{ij} = \dot{m}_{ij} cp \Delta T$, where cp (J/(kg. K)) is the specific heat; ΔT (°C) is the temperature difference between two zones; \dot{m}_{ij} (kg/s) is the rate of mass from surrounding zone to zone ij and can be computed using PLM.

In POMA, a room is geometrically subdivided into n zones. The total number of unknowns (X) is $2n$, i.e., temperatures and pressures of these n zones. There is the same number of independent equations as the unknowns based on mass and energy balance equations of all the zones. The multidimensional secant method, Broyden's method (Press *et al.*, 2007), is used to solve this set of nonlinear conservation equations (Equation 1 and 2). This method needs reasonable initial trial guesses (temperatures and pressures of all the zones) and then calculates the final results using an iteration process. In this method, these unknowns will be initialized at the beginning and then substituted into these corresponding balance equations to compute the residuals. After that, a matrix A representing the approximate Jacobian will be formed based on these unknowns and corresponding residuals. The linear matrix system $Ax = b$ then can be solved by using the QR decomposition method and the results, x , represent the correction values for these unknowns. Therefore, a new set of unknowns, $X + x$, will be used for the next

iteration. The program converges when the Max residual ϵ of all the zones is less than a very small value, as defined by the user (let say 1.0×10^{-3}) i.e., $\epsilon = \text{Max (residuals)} < 1.0 \times 10^{-3}$. The flow chart in **Fig. 3** shows the structure of this zonal model.

3. PARALLEL COMPUTING TECHNIQUE AND OPENMP

Parallel computing techniques have been developed and employed for many years. These computation techniques are able to work on the problems that are impossible with traditional computers. The parallel techniques partition a large problem or task into multiple smaller ones and then distribute these smaller problems or tasks into different computing units, i.e., processors/threads. These smaller problems or tasks can be finished simultaneously on these computing units, which significantly improves the computational performance compared to the traditional serial computing techniques, in which the program for a large problem or task is executed sequentially on a single computing unit. One of the popular parallel computing techniques is known as OpenMP (2013).

As an Application Programming Interface (API), OpenMP accomplishes the mapping of computations onto different processing units characterized with shared memory architectures.

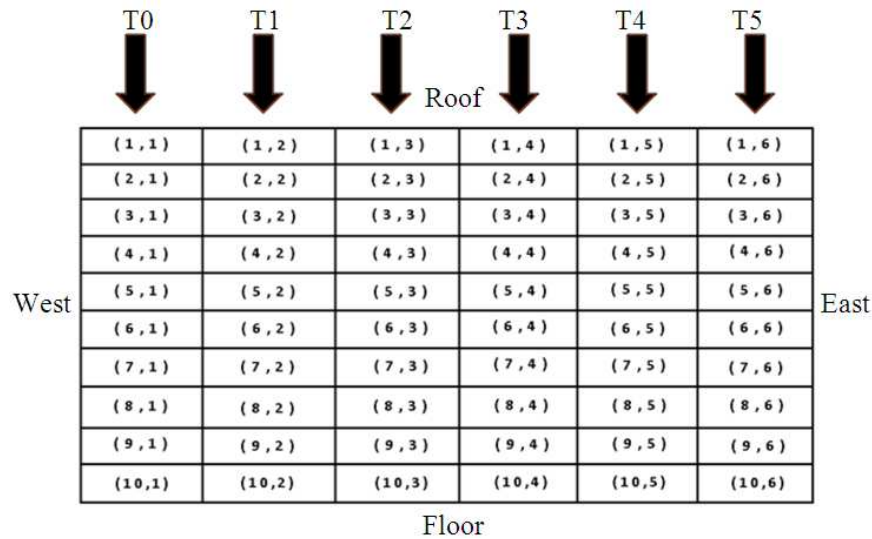


Fig. 4. The potential to use openMP in the zonal model

In OpenMP, the different parallelized works are executed through independent instruction streams, called threads, which can communicate each other by accessing the data located in the shared memory space. An OpenMP command, *#pragma omp parallel*, will indicate the beginning of the parallelized environment. The advantages of OpenMP include (OIMC, 2011):

- It is easy to program and debug
- The serial code usually does not need large modifications
- The OpenMP code is easy to understand and maybe easily maintained

Additionally, OpenMP is easy to be implemented on loops (OIMC, 2011). Therefore, in this article, the parallel computing technique, OpenMP, was used to the program of the zonal model POMA in the C language platform. This zonal model program consists of 18 functions which would be called multiple times. Also, in these functions, plenty of “for” loops are used to perform the zone-by-zone computations related to the mass or energy conservations (Equation 1 and 2). **Figure 4** shows the feasibility of the data/task decomposition. As shown in **Fig. 4**, a room is subdivided into 60 zones for illustration. Thus, instead of performing the calculation zone after zone, each column can be executed simultaneously by different threads from T0 to T5. Theoretically, the computational efficiency or speedup can be 6 times greater than before, if 6 threads are used

in the parallel computations. Nevertheless, the actual increase would be less than 6 times because of the additional communication overheads. Hence, it is suitable for OpenMP to be used in this program, especially in these “for” loops to accomplish the parallelization without significant modification to the original zonal model program.

4. RESULT AND DISCUSSION

In the zonal model, a room with the size 3.1×3.1×2.5 m is considered, as shown in **Fig. 5**. The boundary conditions (the room interior surface temperatures) are shown in **Table 1**.

As shown in **Table 1**, temperature stratifications were created by using a relatively low temperature on the west surface in order to investigate the prediction ability of the zonal model in the cooling application.

Both of the serial and parallelized programs were executed on the High Performance Computer (HPC), called Hermes, with multiple threads of control. Different problem sizes, i.e., the different numbers of zone divisions were used, including 60 zones (6×1×10), 80 zones (8×1×10), 120 zones (10×1×12) and 192 zones (12×1×16). Since the temperature boundary conditions of the south and north surfaces are identical (**Table 1**), only one zone was applied in the south-north direction (y direction in **Fig. 5**). Additionally, besides the serial program that just uses one thread, various numbers (2, 6 and 10) of threads were used in the

parallelized programs in order to investigate the model scalability. The result of the zonal model with 120 zone divisions in terms of temperature distribution is displayed in **Fig. 6**. The **Fig. 7** shows the actual room air temperature distribution obtained from the experiment that was carried out in the MINIBAT test cell (Inard and Buty, 1991), which is located in CETHIL (Centre de Thermique de l'INSA de Lyon) with the dimensions of L (3.1m)×W(3.1m)×H(2.5 m) and is specially designed to measure temperature distributions in a controlled environment. Comparing **Fig. 6** with **Fig. 7**, a good agreement is observed between the results of POMA and the experimentation due to the quite similar patterns of these two temperature distributions, which demonstrates the validity of the programmed zonal model. Further validation of POMA program has been done previously (Haghighat *et al.*, 2001). In fact, the execution time is dependent not only on the problem size, but also on the initial guesses. A good initial guesses (close to the roots of the balance equations) contribute to fast convergence. In this simulation, the initial guesses far from the roots were used, which therefore caused a slow convergence for the serial program, especially when the problem sizes are relatively large, such as 120 and 192 zones. Additionally, due to the uncertainty of

the allocation of the resource on the HPC, the measured execution time varies from one run to another. Therefore, for each situation (different zone divisions or various numbers of threads), the program has been executed several times and the average time has been calculated and recorded.

The **Fig. 8 and 9** show the execution times (represented using log value) against the various numbers of threads and zone divisions, respectively. As shown in the **Fig. 8** when the parallel technique OpenMP is used, the execution time reduces significantly, especially from one thread (serial) to two threads. Also, the different curves corresponding to the different problem sizes from 60 to 192 zones behave very similarly in terms of computational time. The **Fig. 9** also demonstrates similar results. In this figure, when the number of zone divisions increases, the execution time rises accordingly; and when the number of threads are changed from 1 to 10, the execution time decreases. As a matter of fact, these two figures imply the scalability of this algorithm (model), i.e., when the problem size (zone division) increases, the execution time can be reduced by increasing the number of threads used in the parallelized program without losing the efficiency.

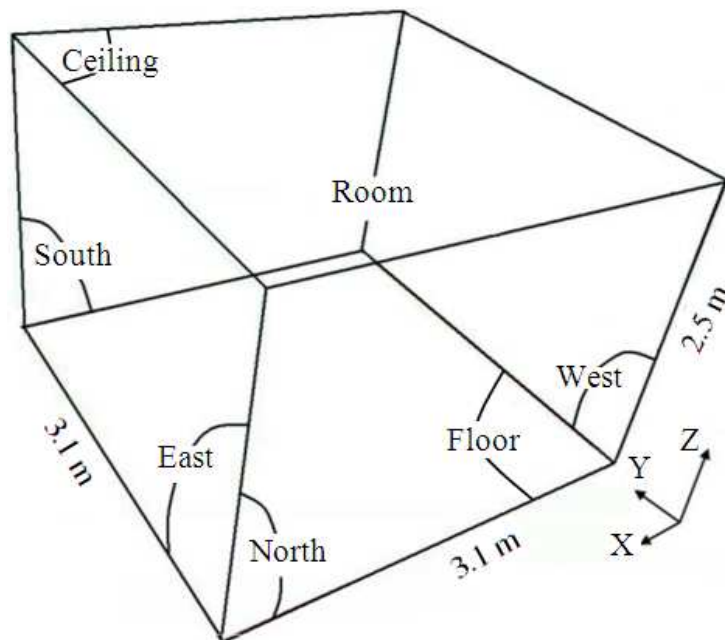


Fig. 5. Simulated room

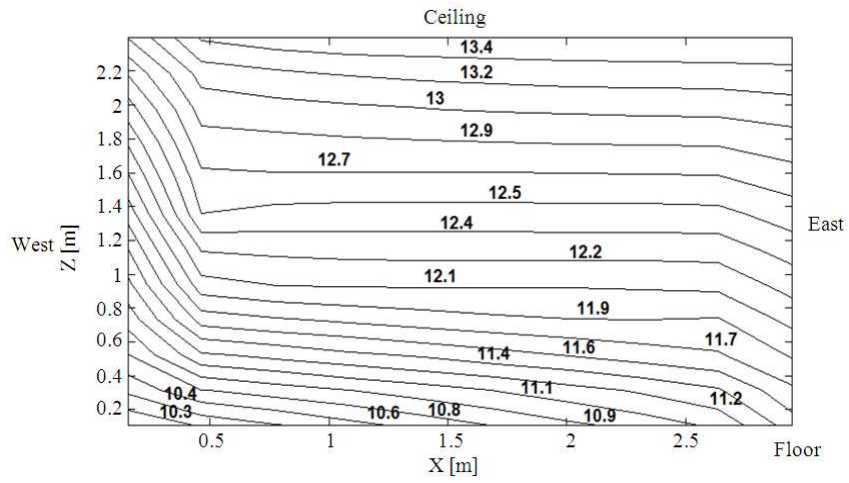


Fig. 6. Simulated room temperature distribution [°C]

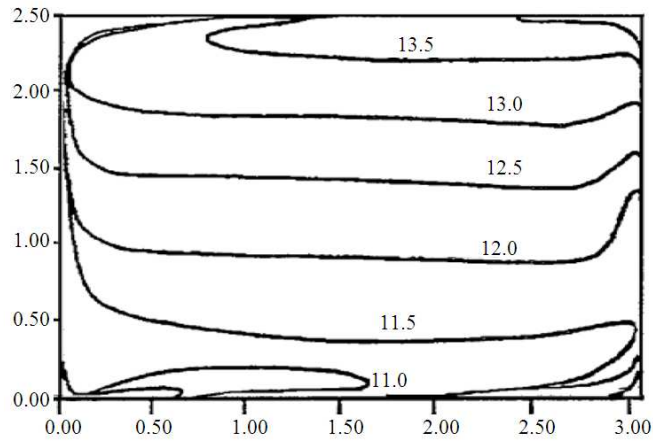


Fig. 7. Actual room temperature distribution [°C] (Haghighat *et al.*, 2001)

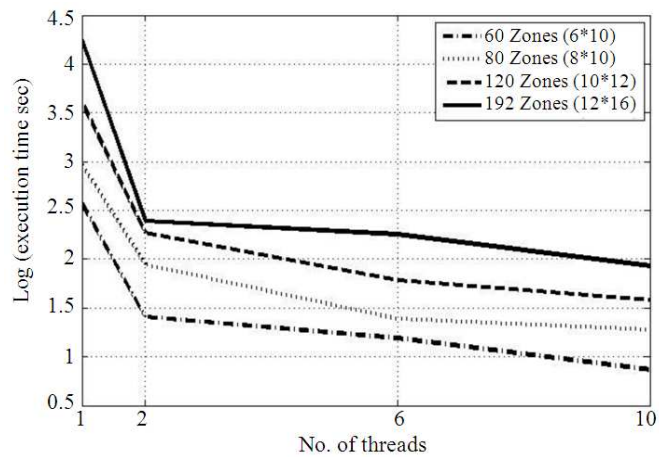


Fig. 8. Execution times with various numbers of threads

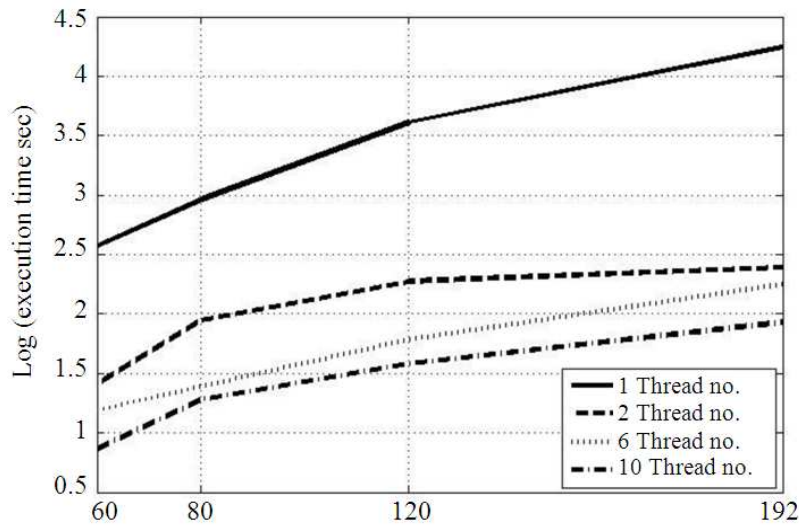


Fig. 9. Execution times with various numbers of zone divisions

Table 1. Boundary conditions

	West [°C]	East [°C]	South [°C]	North [°C]	Ceiling [°C]	Floor [°C]
Temperature	6.0	13.9	14.1	14.1	13.5	11.8

5. CONCLUSION

The zonal model POMA was developed in C language and then executed on a HPC with multiple threads of control. This programed model was validated by comparing its result with the experiment data obtained from the MINIBAT test cell. The parallel technique, OpenMP, was successfully used in this program by decomposing the data/task into a number of pieces that were executed simultaneously on different threads. The results demonstrate not only the significant reduction of the computational time but also the scalability of this model/program.

In fact, this parallel technique provides a way to maximize the usage of available computer resources. After the verification of the scalability of the zonal model, this parallel technique may be used to more complicated situations, for example, when buildings are characterized by a large number of rooms with complicated geometry and indoor layout of furnishings. With the assistance of the parallel technique, the simulation computational time for these complicated cases may be effectively reduced, which contributes to the wide application of the zonal model.

6. REFERENCES

Clarke, J.A., J.L.M. Hensen and C.O.R. Negrao, 1995. Predicting indoor airflow by combining CFD and thermal simulation. Proceedings of the 16th AIVC Conference on Implementing Results of Ventilation Research, (RVR' 95), Palm Springs, CA, pp: 145-54.

Haghighat, F., Y. Lin and A.C. Megri, 2001. Development and validation of a zonal model-POMA. *Build. Environ.*, 36: 1039-1047. DOI: 10.1016/S0360-1323(00)00073-1

Inard, C. and D. Buty, 1991. Simulation of thermal coupling between a radiator and a room with zonal models. Proceedings of the Building Simulation, (PBS' 91), Nice, France, pp: 113-117.

Jiru, T.E. and F. Haghighat, 2004. Zonal models for indoor air flow-a critical review. *Int. J. Ventilat.*, 3: 119-129. DOI: 10.5555/ijov.2004.3.2.119

Megri, A.C. and F. Haghighat, 2007. Zonal modeling for simulating indoor environment of buildings: Review, recent developments and applications. *HVAC&R Res.*, 13: 887-905. DOI: 10.1080/10789669.2007.10391461

- Megri, A.C. and Y. Yu, 2010. New calibrated zonal model for temperature, pressure and airflow predictions. Proceedings of the 10th REHVA world Congress Sustainable Energy Use in Buildings, May 9-12, Antalya, Turkey.
- Megri, A.C., M. Snyder and M. Musy, 2005. Building zonal thermal and airflow-a review. *Int. J. Ventilat.*, 4: 177-188. DOI: 10.5555/ijov.2005.4.2.177
- OpenMP, 2013. OpenMP application program interface. OpenMP Architecture Review Board.
- OIMC, 2011. Overview of Intro to MPI class. Dartmouth College.
- Pacheco, P.S., 1997. *Parallel Programming with MPI*. 1st Edn., Morgan Kaufmann, San Francisco, ISBN-10: 1558603395, pp: 418.
- Press, W.H., S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, 2007. *Numerical Recipes, the Art of Scientific Computing*. 3rd Edn., Cambridge University Press, Cambridge, UK, ISBN-10: 0521880688, pp: 1235.
- Yu, Y. and A.C. Megri, 2011. Development and application of a new dynamic zonal model (POMA+). Proceedings of the 12th International Conference on Air Distribution in Rooms, Jun. 19-22, Trondheim, Norway.
- Yu, Y., 2012. Zonal modeling approach and its applications in buildings. MSc., Thesis, Department of Civil and Architectural Engineering, University of Wyoming, Laramie, Wyoming, USA.