

"O-ZONE": AUTOMATIC GENERATOR OF ZONAL MODEL DEDICATED TO DYNAMIC SIMULATIONS OF AIR FLOWS AND HEAT TRANSFERS IN A ROOM

Sihem Tasca-Guernouti¹, Marjorie Musy², and Gérard Hégron³

¹Centre d'Etudes Techniques de l'Équipement (CETE) de l'Ouest, MAN rue René Viviani, BP 46223, 44262 Nantes cedex 2, France

²Cerma-UMR CNRS 1563, ENSA de Nantes, 6 quai François Mitterrand, BP 16202, 44262 Nantes cedex 2, France

³LCPC Direction scientifique, Route de Bouaye, BP 4129, 44341 Bouguenais cedex, France

ABSTRACT

The aim of this paper is to present an automatic generator of zonal models that makes it possible to free the user from the choice of the specific flows models that have to be implemented in the zonal model and from the partitioning stage. The dynamic simulation tool called "O-Zone" is based on an advanced formulation of zonal models. It uses on a new way of sub-dividing the room that allows us to obtain a dynamic partitioning based on airflow patterns. The partitioning method chosen is the Octree method. It is a hierarchical representation of the space based on the successive and recursive subdivision of a cube in eight smaller cubes.

NOMENCLATURE

T_s	Inlet air temperature	[°C]
$q_{m,s}$	Supply air mass flow	[kg/s]
dm_{ij}	Mass flow rate (zone i to zone j)	[kg/s]
C_d	Empirical coefficient	[-]

INTRODUCTION

In order to design HVAC systems and to predict their performances in thermal comfort and indoor air quality terms, the estimation of airflow details and heat transfers within the building zones represents an important stake. Zonal models are dedicated to the prevision of the heterogeneity of the thermal and aerodynamic characteristics of an air space. They are based on dividing the considered room into a small number of sub-zones so that they imply moderate computing times. These models represent a good compromise between simplicity of models and quantity of data useful in order to contribute to the qualification of environments. If much progress has been made to automate the zonal models construction knowing the air space partitioning and a choice of the models to be used, these two tasks are still spring of the user whereas they require an important modelling expertise and experimentation in the building (experience of buildings' physics). In order to minimize the user's intervention, we developed an automatic generator of zonal models that requires a minimum user's expertise. This dynamic simulation tool called "O-Zone" is based on an advanced formulation of zonal models. It uses on a new way of sub-dividing the room that allows us to obtain a dynamic partitioning based on airflow patterns. The

selected method for partitioning is the Octree method. It is a hierarchical representation of the space based on the successive and recursive subdivision of a cube in eight smaller cubes. The first calculation, carried out for static configurations (Guernouti et al., 2004) shows that the partitioning method is well appropriated for this kind of airflow representation.

"O-ZONE": ZONAL MODEL GENERATOR

The designed generator "O-Zone" is constituted of five principal modules: an analysis tool, a database, a partitioning tool, a simulation tool and finally a visualization tool (Guernouti et al., 2003).

The analysis tool allows interpreting the boundary conditions and foreseeing the resulted flows. The most adapted analytical model is chosen automatically in the database to represent the expected airflow. Figure 1 presents the process of analysis.

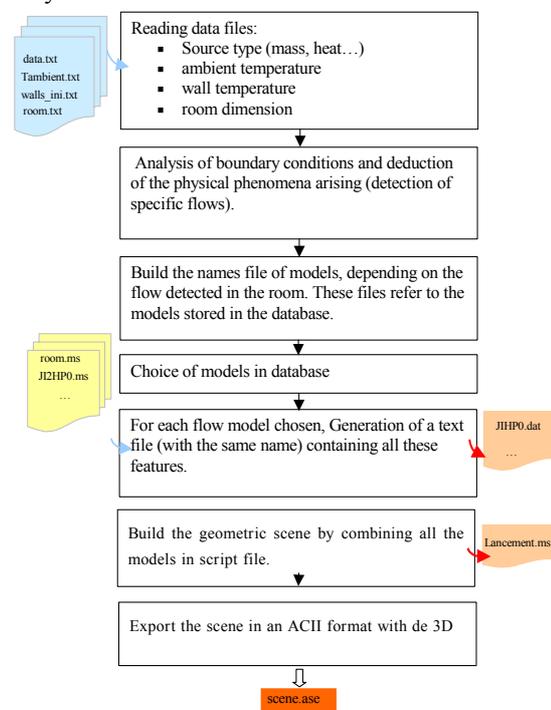


Figure 1: Analysis tool simplified flowchart

After the phase of analysis and choice of the models, a file containing the geometrical and physical characteristics of the models is obtained (Figure 2).

This is an “ASE” text file which is composed of successive structured lines. It’s consisting mainly of nodes (*NODE_NAME in figure 2). A node is a structure that represents an object in the “scene”. There are several different nodes that correspond to what may constitute a scene. A geometric object in this file consists of a series of plane surfaces associated with three nodes (triangle). The plane surfaces are oriented so that their normal is obtained by turning it clockwise. The vertex are first listed (*MESH_VERTEX_LIST in figure 2) followed by the list of the surfaces that constitute the object. This file is used to perform the partitioning of the studied volume.

```

)
*MATERIAL_REF 0
)
*GEOMOBJECT (
*NODE_NAME "JI2HPO"
*NODE_TM (
  *NODE_NAME "JI2HPO"
  *INHERIT_POS 0 0 0
  *INHERIT_ROT 0 0 0
  *INHERIT_SCL 0 0 0
  *TM_ROW0 1.0000 0.0000 0.0000
  *TM_ROW1 0.0000 -0.0000 1.0000
  *TM_ROW2 0.0000 -1.0000 -0.0000
  *TM_ROW3 -4.5000 0.0000 3.0000
  *TM_POS -4.5000 0.0000 3.0000
  *TM_ROTAXIS -1.0000 0.0000 0.0000
  *TM_ROTANGLE 1.5708
  *TM_SCALE 1.0000 1.0000 1.0000
  *TM_SCALEAXIS 0.0000 0.0000 0.0000
  *TM_SCALEAXISANG 0.0000
)
)
*MESH (
*TIMEVALUE 0
*MESH_NUMVERTEX 8
*MESH_NUMFACES 12
*MESH_VERTEX_LIST (
  *MESH_VERTEX 0 0.5620 4.5000 3.0000
  *MESH_VERTEX 1 -4.5000 4.5000 3.0000
  *MESH_VERTEX 2 -4.5000 4.5000 2.9984
  *MESH_VERTEX 3 0.5620 4.5000 2.1860
  *MESH_VERTEX 4 0.5620 -4.5000 3.0000
)
)

```

Figure 2: Example of file “scene.ase”

The partitioning tool is based on the octree method that allows having a dynamic partitioning, as we will see below. The terminal octants (cells of the studied volume) at the leaves of the tree represent either specific homogeneous zones containing a driving flow or ordinary homogeneous zones containing ambient air.

For our research, the scene consists of the room that represents the root and of the various geometric objects that represent the different specific flows in the room. Figure 3 presents the partitioning process. After having read the file scene.ase, the room (root) is divided into eight cubes (octants). The process of partitioning continues until the stopping criterion is reached. In our case, we have chosen two criteria. The first is the maximum number of plane surfaces (which constitute the object representing the flows) contained in an octant. The second is the minimum size of the octant. So, the partitioning continues until the number of plane surfaces in the current octant is less than the value “max_faces” or that its size is greater than or equal to “min_size” (these two values are defined by the user first). We set the value “max_faces” to 1. It means that if the number of surfaces in an octant is greater than 1, the process of partitioning goes on. The choice of the minimum size of octants “min_size” is let to the user. The

numbering of octants is recursive and is done according to the following pattern: the octants of the first level are numbering from 0 to 7. Then, the octants name of the second level is constructed by concatenating the name of the divided octant to a number between 0 to 7. This concatenation is repeated up to the last level of tree partitioning. For example, the octant [3_2_0] stems from three partitioning stages: In first stage, the octant [3] is one the six octants that results from the root partitioning (level 1). In second stage, octant [2] results from the partitioning of the octant [3] (level 2) and in third stage the octant [0] is the result of partitioning of the octant [2].

Once the partitioning done and the octal tree built, the next step is the analyze of the tree in order to make a list of the octants associated to their neighbours which will be necessary to write the exchange between cells. To do this, we established two kinds of tables: neighbourhood tables and its transposed. These tables make easy the search for neighbours. In practice, the neighbourhood table is, for each octant [i], the list of its same level neighbours [j] (Table 1) and also the lower level neighbours of each neighbour [j] in all orientations (Table 2). The transposed neighbourhood table represents the list of the same level neighbours of the octant [i] as well as the neighbours son of octant [i] in all orientations (Table 3).

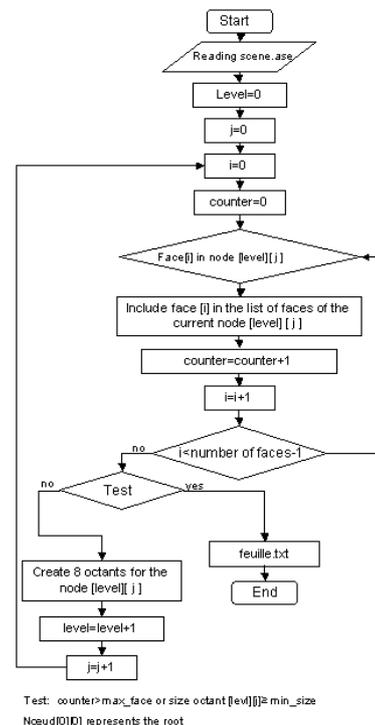


Figure 3: partitioning tool flowchart

By using the file “feuilles.txt” obtained after the partitioning and the neighbourhood tables, the octal tree is scanned from top to bottom (and vice versa).

The list of all neighbours is thus built and saved in the file "voisin.txt". This file contains, for each neighbour, its physical model (ambient air, specific flow), direction (up, down, etc.) and its coordinates.

Table 1: Same level neighbourhood table

Octant	0	1	2	3	4	5	6	7
Bottom		0		2		4		6
Top	1		3		5		7	
East	2	3			6	7		
West			0	1			4	5
North	4	5	6	7				
South					0	1	2	3

Table 2: Example of neighbourhood table: octant 0

Octant [0]	Orientation	Neighbour: same level	Neighbour: higher level
	Top	1	1_0
			1_2
			1_4
			1_6
	East	2	1_2
			1_3
			1_6
			1_7
	North	4	1_4
			1_5
			1_6
			1_7

Table 3: Example of transposed neighbourhood table: octant 0

Octant [0]	orientation	son	Neighbour: lower level
	Top	0_1	1
		0_3	
		0_5	
		0_7	
	East	0_2	2
		0_3	
		0_6	
		0_7	
	North	0_4	4
		0_6	
		0_5	
		0_7	

After the partitioning and the tree analysis phases, we have a set of volumes (cells) representing either ambient air or a specific flow. For each cell, a set of equations is written according to the physical phenomenon that it contains and the exchanges that occur with the other cells. To solve the set of equations obtained, we chose to implement our zonal model in the object-oriented simulation environment SPARK. It is a general simulation environment dedicated to the resolution of differential algebraic equations (Sowell et al., 2001). We used the existing

object libraries (Mora et al., 2003; Mendonça et al., 2005), after having modified them to meet the needs of our partitioning. We also have rewritten the automatic generator of zonal model. This one allows generating the problem specification file (*.pr) used by SPARK (Guernouti et al., 2004). The resolution process in SPARK is shown in figure 4. The file generated at the end of SPARK simulation contains results (octree.out). It is used:

- 1) to calculate the average ambient temperature in the room and to determine the new input data files *.Inp (used for the next time-step as long as the dynamic simulation is not completed).
- 2) by the visualization tool after the end of the dynamic simulation.

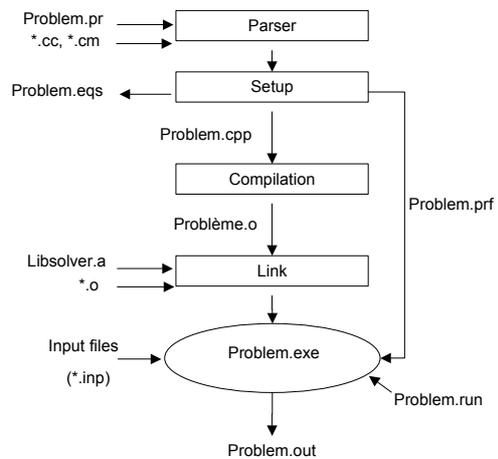


Figure 4: SPARK problem generation

Figure 11 presents the dynamic simulation process. So, after the phase of reading the global data file and the assumptions (initial ambient air temperature), the problem is built at the first iteration (named "i") and the first time-step (named "j") (it's not the same time-step than in SPARK simulation but include the SPARK simulation steps). The generation of the file "lancement_i_j.ms" is completed. This allows O-Zone to build a 3D scene (the room and the specific flows). Then, the partitioning tool uses the file «scene_i_j.ase» in order to perform partitioning. The resulting file "feuilles_i_j.txt" contains a list of all octants, their dimensions and their natures (ambient air or particular flows). Then, follows the phase of the tree analysis and research of each neighbouring cell. The results are stored in the file "voisins_i_j.txt". From these two files, the problem "octree_i_j.pr" is generated and then solved in SPARK. Results are stored in the "octree_i_j.out" file. The new assumptions (hyp_i_j) are then calculated (average ambient temperature of the cells) and compared to the previous assumptions. If the difference between the two temperatures is less than "ε", the time-step is incremented. If not, new assumptions are done and a new iteration begins. In both cases, the files "lancement.ms", "scene.ase" and "feuilles.txt" are generated. The current file "feuilles.txt" is then compared to the previous file.

Four scenarios are possible: 1) The tree has changed (“feuilles.txt” different) during the time-step. 2) The tree has not changed during the time-step. 3) The tree has changed to the new time-step. 4) The tree has not changed to the new time-step. If files are the same (case 2 and 4), the specification file problem is not generated. The new results are obtained by using the previous files “octree.pr” and “octree.run”. But, it is necessary to update the input data (*.inp) assigning them new values from previous file “octree.out”.

The dynamic simulation process described above will be discussed in more detail in the following applications.

APPLICATIONS

To validate the tool developed, we performed simulations in a steady-state configuration and in a dynamic one. In this paper, we present both of them in the case of a room with horizontal wall jet. The experimental data used to test the jet model are those acquired by CASTANET (Castanet, S., 1998) and used under the ARC CNRS / ECOTECH entitled "Ventilation and air quality in indoor climate". The dimensions of the studied cell are 3.1 m wide, 3.1 m deep and 2.5 m high (Figure 5). The ventilation of the room is ensured by an opening at the top of the West side and another at the bottom of the opposite side. Results from O-Zone are compared with experimental data.

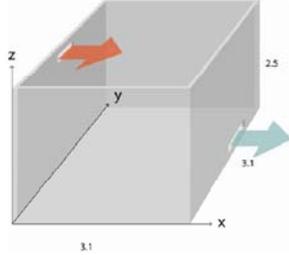


Figure 5: studied configuration

For the first simulation (steady case), tables 4 and 5 give the input values of the problem, which are the inlet air temperature, the inlet flow rate and the walls surface temperature.

Table 4: Boundary conditions: Inlet

Ts (°C)	qm_s (kg/s)
34	8.10 ⁻³

Table 5: Boundary conditions: surface temperature (°C)

Sout	North	East	West	Ceiling	Floor
21.5	21.7	19.6	21.6	22.5	21.2

Taking into account the conditions of air supply, a two-dimensional non-isothermal horizontal wall jet is chosen for this configuration.

The jet width $b_u(x)$, the longitudinal velocity profile $U(x)$ and the velocity amplitude $U_m(x)$ are given by:

$$b_u(x) = 2,36C_u x \quad (1.)$$

$$\frac{U(x)}{U_m(x)} = \exp \left[-0,937 \left(\frac{y}{C_u x} - 0,14 \right)^2 \right] \quad (2.)$$

$$\frac{U_m(x)}{U_0} = \frac{3,5}{\sqrt{x/h_0}} \quad (3.)$$

The decay of maximum temperature and the longitudinal temperature profile are given respectively by:

$$\frac{\Delta T_m}{\Delta T_0} = K_T \sqrt{\frac{h_0}{x}} \quad (4.)$$

$$\frac{\Delta T}{\Delta T_m} = \exp \left[-\ln 2 \left(\frac{y}{C_T x} \right)^2 \right] \quad (5.)$$

Where, x is the coordinate along the jet axis, y is the distance to wall; U_0 and h_0 are the inlet velocity and diffuser height respectively. K_v and K_T are empirical decay constants. C_u and C_T are empirical coefficients.

NOTTAGE (Zhang et al., 1993) recommends a value of 3,5 for K_v . The values for C_u , K_T , and C_T are respectively 0.074, 2.79 and 0.095 (Musy, M., 1999).

From these laws, we calculate the airflow rate $q_m(x)$ and the heat flux $\phi(x)$ in the jet vertical interfaces (portions of a jet section in which the wall distance ranges from y_1 to y_2) by:

$$q_m(x) = \int_{y_1}^{y_2} \rho U(x) dy \quad (6.)$$

$$\phi(x) = \int_{y_1}^{y_2} \rho C_p U \Delta T dy \quad (7.)$$

The airflow model between adjacent ordinary cells is based on a power-law formulation. The airflow rate dm_{ij} across the common surface between the cells i and j depends on the pressure difference between the two zones. We can represent the airflow rate by the expression:

$$dm_{ij} = C_d \rho_{i,j} w \left(\frac{2}{\rho_{i,j}} |\Delta P_{i,j}| \right)^{1/2} \varepsilon_{i,j} (\Delta P_{i,j}) dz \quad (8.)$$

Where:

$$\Delta P_{i,j} = (P_i - \rho_i g z_i) - (P_j - \rho_j g z_j), \quad (9.)$$

$\varepsilon_{i,j}$ is sign from $\Delta P_{i,j}$, And C_d is an empirical coefficient similar to a discharge coefficient, in common practice $C_d = 0,83$.

The studied room is subdivided into 36 cells like presented in Table 6.

In Figure 6b we present the prediction of airflow pattern and the temperature obtained by O-Zone (right) and the temperature predicted by the experimental data (left). We can see that the entrainment of the ambient air into the hot jet is well predicted.

Table 6: Distribution of the cells in the studied room

Level	Ordinary cells	Jet cells
1	4	0
2	24	8

Figure 7 shows the temperature profile predicted by O-Zone in the central vertical plane at $x=1.5\text{ m}$ and $x= 2.30\text{m}$. We can see that the value calculated for

the maximum temperature in the jet is quite similar to that obtained experimentally. However, in the occupied zone, the temperature difference between the calculated and measured values is about one degree.

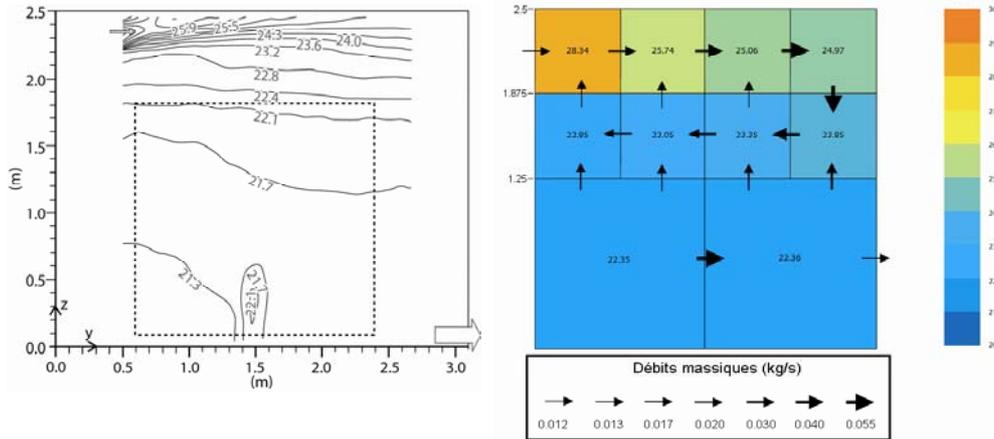


Figure 6: 6a(left): Room temperature predicted by experimental data (Castanet, 1998)
6b (right): Airflow pattern and room temperature predicted by O-Zone

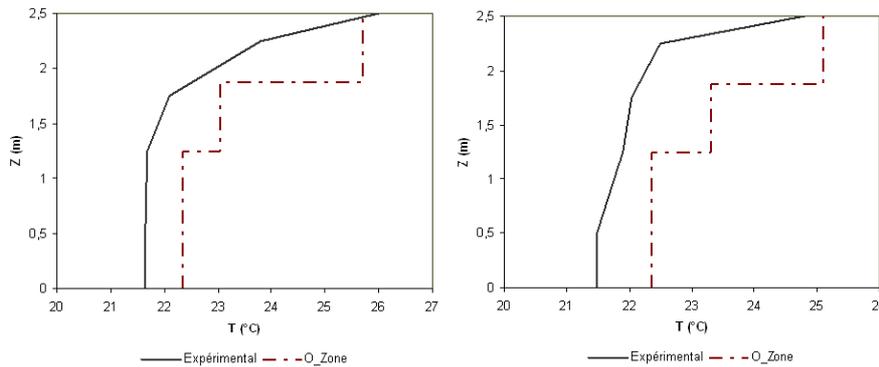


Figure 7: Temperature profiles predicted by O-Zone and experimental data in the central vertical plan at:
7a(left): $x= 1.50\text{m}$
7b (right): $x= 2.30\text{m}$

To show the evolution of the partitioning over the time, we conducted the second simulation (dynamic simulation) keeping the same configuration as the first time-step and changing the inlet conditions after. We chose to perform the simulation over a period of three hours with a major time-step of 1 hour (the time-step here is the time-step in which the simulation in SPARK will run, so that the time-step in SPARK are its divisions). It is an interval during which we consider that the conditions will not change. For example, it may correspond to the weather conditions. During this time, and to take into account the capacitive terms, the treatment of dynamic simulation in SPARK is used, with a calculation time-step of 18 seconds to ensure the

convergence of the dynamic system on the whole simulated period.

Table 7 gives the input values of the problem, which are the inlet air temperature and the inlet flow rate at each time-step. Table 5 gives the walls surface temperature.

Considering the inlet conditions of the first time-step (PDT1), the length of jet penetration is less than 1,6m. The room is divided into 22 cells as shown in Table 8. For both remaining time-step (PDT2 and PDT3), the room is divided into 36 cells as shown in Table 9. Figure 8 shows the evolution of partitioning at each time-step.

Table 7: Inlet conditions for dynamic simulation

Time-step	$T_s(^{\circ}\text{C})$	$q_{m,s}(\text{kg/s})$
PDT1	25	$0.458 \cdot 10^{-3}$
PDT2	25	0.017
PDT3	25	0.017

Table 8: Distribution of the cells (time-step: PDT 1)

Level	Ordinary cells	Jet cells
1	6	0
2	12	4

Figure 9 shows the evolution over the time of the average temperature in room and also the evolution of temperature in the jet cell whose center gravity coordinates are (1,1625; 1,1625; 2,1875).

Table 9: Distribution of the cells time-step: PDT 2 and PDT3)

Level	Ordinary cells	Jet cells
1	4	0
2	24	8

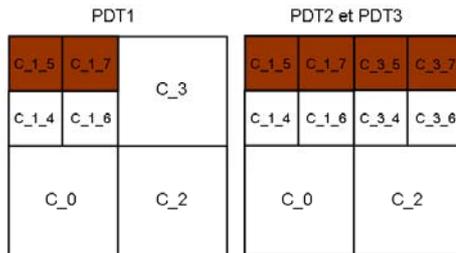


Figure 8: Partitioning evolution over the time

We can see that the mean air temperature in the room is evolving in a similar manner to the temperature in the jet cell (Cel_1_7). In both cases it increases between major time-step loads (each 1 hour). This increase is the change in boundary conditions (air mass flow supply). Moreover, for each major time step "i" we can observe the evolution of temperature after the achievement of the SPARK dynamic calculation with a 18 second time-step. This evolution follows an exponential curve.

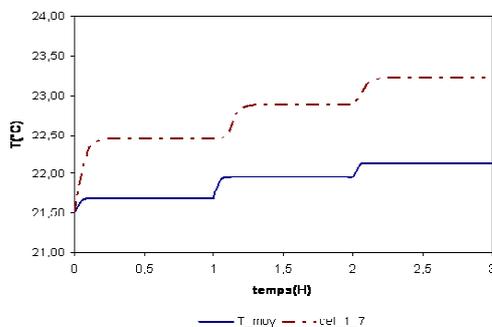


Figure 9: Temporal evolution of the average temperature in the room and the temperature in the jet cell cel_1_7

The convergence of the global dynamic system was obtained after one iteration for each major step loads. This number does not include iterations during the dynamic simulation in SPARK. It is designated by "j" in the flowchart presented in Figure 11. To be more clear we will detail the run process of the simulation by making the link between the results obtained and the flowchart in Figure 11.

The simulation begins at the time " t_0 " with a temperature equal to 21,5 °C as initial assumption (Hyp_0_0). We consider that the loads are constant between the major time-step. In the first major time-step "i" equal to 1 hour and the first iteration "j", an analysis of the boundary conditions is made, the models representing specific flows are selected in the database (for this application, the model retained is the two-dimensional hot wall jet) and the file "lancement_1_1" is generated. The scene (geometric representation of the room and specific flows in a 3D modeling software) is then built and the partitioning is achieved. At this stage of the simulation, the file containing the tree structure studied "feuille_1_1" is generated (Figure 11). Starting from this file, the problem "octree_1_1.pr" is built with the various files needed for simulation in SPARK (octree_1_1.run, *. inp). Taking into account the air heat capacity, the dynamic simulation is carried out in SPARK with a 18 second time-step and a maximum number of iterations of 600 (for each SPARK time-step). So, we get the file "octree_1_1.out" that stores the results of this simulation. The assumption (Hyp_1_1) representing the average temperature in the simulated room is then calculated and compared to the previous assumption. For this study, only one "major" iteration is necessary to check the test $((\text{Hyp}_{1_1}) - (\text{Hyp}_{0_0})) < \epsilon$, $\epsilon = 0,5^{\circ}\text{C}$. Then, the major time-step "i" is incremented, the new hypothesis (Hyp_1_1) is taken into account for the following calculations. So, the simulation runs as the same way as in the previous time-step until phase of the tree generation "feuille_2_1".

Indeed, for this application, the partitioning is not the same between the first "major" time-step and the second. The trees "feuille_1_1" and "feuille_2_1" are different. In this case, a new problem specification "octree_2_1.pr" is generated. The values of the old tree are forwarded to the new in order to perform the simulation in SPARK (Figure 11). This calculation represents the second step of the curves shown in Figure 11. As for the last time-step, one iteration is necessary for the convergence of the system. For the last "major" time-step, the simulation continues in the same manner as the previous ones.

Figure 10 shows the temperature profiles obtained by O-Zone in the central vertical plane at $x=1,50$ m and $x=2,50$ m from the supply opening. We can see two distinct zones at each time-step. We obtain an upper zone (altitude higher than 1,5m) warmer than the lower zone (occupancy zone). The figure 10 also enables us to highlight the evolution of the

partitioning over the time. Indeed, we can see that the temperature gradient is greater during the second and the third steps (where the jet is impacting) than during the first. The extinguishment of the hot jet occurs at the middle of the room.

DISCUSSION

On the whole, the results are promising. For both steady and dynamic simulations, the upper part of the room seems to be well characterized. The lower part is less well represented, as already noticed by other authors (Musy, 1999). It would be necessary to find other laws to represent the exchanges between the ordinary air cells.

Furthermore, it necessary to note, especially in the case of dynamic simulation, that the applications presented here are made to study the feasibility of the approach. So, we can't make conclusions about the results without numerical and experimental validation. The results presented are more a demonstration of the effective functioning of the tool developed "O-Zone" that a validation of the zonal model generated.

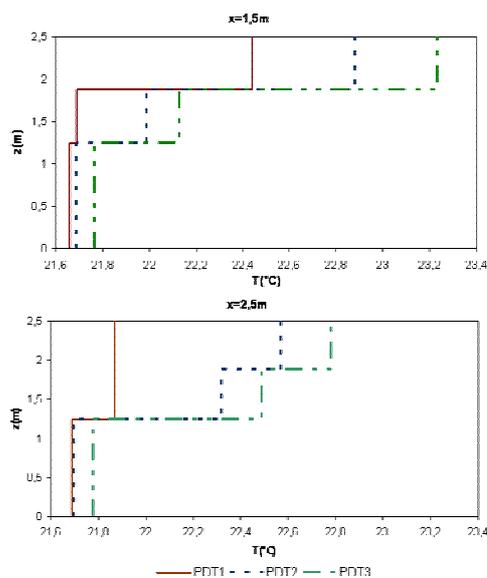


Figure 10: Temperature profiles in the central vertical plane at $x=1.5\text{ m}$ and $x=2.5\text{ m}$, predicted by O-Zone at each time-step

We can also note that during the dynamic simulation, the generated zonal model is revised only on the "major" time-step (out of SPARK simulation). But, during a dynamic simulation, the average temperature in the room changes, therefore, it may happen that the model generated is no longer valid (particularly in the case of simulation over a long period of time). Indeed, we can imagine, for example, that during the dynamic simulation, the difference between the average temperature in the room and the supply temperature is cancelled. In this case, the present jet is no longer a hot jet but an isothermal jet. So, the model chosen at the beginning of the simulation is no longer valid.

Therefore, it's necessary to revise the generated zonal model during a dynamic simulation in SPARK. This verification allows to ensure the convergence of the dynamic system to the physical solution of a problem. To overcome this difficulty, we propose to limit the period of dynamic simulation in SPARK and perform the calculation by steps. This solution will stop the calculation in SPARK every fifteen minutes for example, keeping the assumptions and the loads of the ongoing major step-time, and determine the relevance of the initial assumptions and the validity of the generated zonal.

CONCLUSION

The classical modeling approach focuses, in most cases, on the description of a problem, its mathematical representation and its numerical resolution. The majority of tools neglects the concept of understanding the model by a user. They don't include the personal specification of their potential users. In this context, we have developed an "intelligent" tool, which can generate "autonomous" zonal models. Indeed, our tool supports automatically all the steps of the dynamic simulation:

- the choice of models to represent the specific flows in the studied local,
- the partitioning of this one,
- the resolution of the set of equations results.

In this paper, we have presented two applications in order to validate the method of partitioning and the actual functioning of our tool. These applications have enabled us to demonstrate the ability of O-Zone to generate dynamic simulation with an adaptive partitioning to specific flows in the studied room. We have also applied the automatical partitioning change during a simulation. Of course, there is a lack of validation against experimental results but the results are coherent.

As mentioned above, none of these applications is performed to demonstrate the validity of the zonal models generated by O-Zone. It is a demonstration of the advantages of developing this new generation of "autonomous" zonal models based on an automatic and adaptive partitioning to generate simulations which are induced by boundary conditions changing over the time (such as solar insolation: sun patch problem). Thus, it is possible to validate for further work the representation of physical phenomena and models chosen by comparing the predictions with experimental data and numerical details.

REFERENCES

- Castanet, S , 1998. Contribution à l'étude de la ventilation et de la qualité de l'air intérieur des locaux, PHD thesis, University of Franche-Comté, France.

Guernouti, S., 2008, Méthode d'analyse du comportement thermo-aéraulique des bâtiments pour la construction de modèles zonaux adaptatifs, PHD thesis, University of Nantes, France.

Guernouti, S., Musy, Hégron G. 2003, Automatic generation of partitioning and modeling adapted to zonal method, IBPSA conference (p.427-434), Eindhoven, Netherlands.

Guernouti, S., Musy, M., Mora, L., Hégron G. 2004, Zonal model based on airflow partitioning, RoomVent International Conference, Portugal.

Mendonca, K. C.; Wurtz, E.; Inard, C., 2005, Predicting temperature and moisture distributions in conditioned spaces using the zonal approach, IBPSA Conference and Exhibition (p. 763-768.), Montreal, Canada.

Mora, L., Mendonça, K.C., Wurtz, E., Inard, C., 2003, SimSpark: an object-oriented environment to predict coupled heat and mass transfers in buildings, IBPSA Conference (p.903-910), Eindhoven, Netherlands.

Musy, M., 1999, Génération automatique de modèles zonaux pour l'étude du comportement thermo-aéraulique des bâtiments. PHD thesis, University of La Rochelle, France.

Sowell, E.F., Haves, P., 2001, Efficient solution strategies for building energy system simulation, Energy and buildings 33, 309- 317.

Zhang, J.S., Li Z.H., Zhivov, A.M., Christianson, L.L., 1993, Characteristics of diffuser air jets and airflow in occupied regions of mechanically ventilated rooms-a literature review, ASHRAE Transactions, Part1, Vol 99, p.1119-1127.

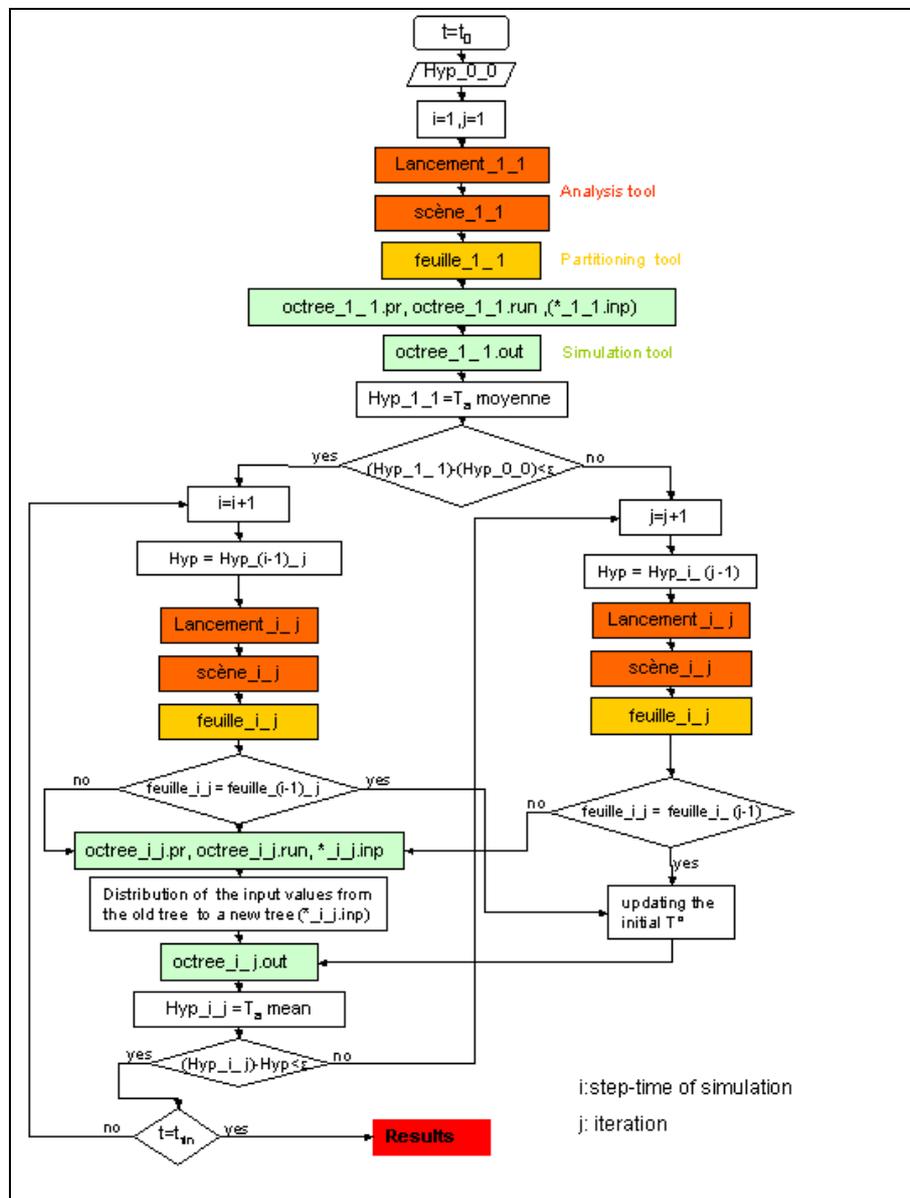


Figure 11: General flowchart of O-Zone dynamic simulation