AN ILLUSTRATION OF AUTOMATIC GENERATION OF ZONAL MODELS

S. Gagneau*, J.M. Nataf* and E. Wurtz**

* **Groupe Informatique et Systèmes Energétiques**
Ecole Nationale des Ponts et Chaussées
6 et 8 avenue Blaise Pascal, Cité Descartes
77 455 Marne-la-Vallée cedex 2, France

** **LEPTAB**
Université de La Rochelle
Avenue Marillac
17042 La Rochelle cedex 1, France

ABSTRACT

In this paper ongoing work using the computer algebra system Maple V and the object-based environment Spark is presented, that aim at the analytical approximate resolution of problems encountered in air flows in dwelling zones, using computer algebra and expert system techniques. The application domain is the building science. The current scope of the work at this stage is two dimensional and steady state. The work presented includes a package for domain decomposition based on expectations about the nature and qualitative behavior of the flow , themselves derived from the boundary conditions of the problem, first principles, and scaling arguments. A numerical engine with an object-based interface, called Spark, is used for solving hard numerical closure equations. The current state of the project as well as the experience of using Maple V and Spark for it is also described.

INTRODUCTION

The simulation of air flow in dwelling zones can be based on numerical simulations using computational fluid dynamics (CFD) codes, that solve the coupled energy and Navier-Stokes equations with a buoyancy term . Simple 2D problems in natural convection, like the celebrated "differentially heated cavity problem", can readily be handled [3] [11]. Variations in geometry are treated for example in [16] applications to real 3D dwelling zones. Possible addition of forced convection are for example presented in [5] possibly with obstacles [6], and work was even performed with CFD for large zones or full buildings [9] . These calculations usually require considerable computational resources. They also yield large and unwieldy outputs, that require postprocessing and visualizing before any attempt of interpretation.

An alternate approach is to consider each dwelling zone as an air node and perform global mass and energy balance on those. That traditional method is often used in global building simulation codes or regulations. It also can be used for interzone flow modelling as in [4]. It is simple but often too crude, and gives no information on the flow pattern inside a zone.

In the past years an intermediate approach has been proposed, the so-called "zonal model" approach [7] [19]. It requires a coarse partitionning of the domain under study, states balance equations on each (big) computational cell, and closes the model by semi-empirical power law equations linking the mass flow rate and the pressure difference. Resolution in a modular environment (Spark) proved instrumental in [19].

However, as pointed out for example in [15] these models fail to represent properly zones of directed momentum like plumes and jets. Also the partitioning is usually arbitrarily rectangular, thus in effect biasing the kind of flow pattern calculated. Last, the output, although qualitatively correct, is the result of purely numerical solution of nonlinear equations systems, and also require extensive postprocessing and visualizing before being applicable.

Therefore it appears tempting to try to model air flows in dwellings with a coarse but "smart" partitionning of the domain under study. The partitionning should be based upon real expected fluid qualitative behavior, and be derived as far as possible from boundary conditions, first principle and physical expertise. Each domain "part" should correspond to a well defined part of the flow, drawn from an "alphabet" of basic flows like boundary layer flows, vortices, cores, potential flows, etc. Thus output would already contain its own physical interpretation. And each domain would, to the extent possible, supply an analytical expression of the dependent variables sought for, namely the velocity and the temperature. This is the object of this paper. Use of Maple V is justified by its symbolic handling of equations, its solution capabilities, and its programming language allowing the design of a rule-based expert system. Use of Spark is justified by its modular features, that allow connecting models together.

GENERAL OVERVIEW

1 Scope

For the sake of simplicity, the problem handled here will be limited to the steady state and two-dimensional case.

The steady state limitation could appear to preclude any simulation of turbulence, pervasive in air flow simulations in buildings, and intrisically time-dependent. However one can overcome that limitation by dealing with time-averaged values.

The 2D limitation also severely diminishes the applicability of that approach, that therefore must be considered as a feasibility study -possibly usable for "long" geometries when one dimension can be ignored. The main reasons for sticking to the two-dimensional case are:

- There are much more known analytical solutions to the equations of fluid mechanics in 2D than in 3D;
- One can fill a connected subzone with a "zonal model", that consists essentially of state equations, balance equations, and boundary mass and heat transfer equations, based on a fractional power law between flux and pressure difference.
- One might notice that the 2D Navier-Stokes equations can be nicely handled via complex numbers, as shown by [14] -although no use will be made of this, due to the complexity of the resulting equations, and the fact that the problem at hand is more difficult since coupled to the energy equation via the buoyancy term.

2 Goal

The goal to attain is a program that would be able, given the geometry and the boundary conditions of a problem, to decompose the computational domain into a few large subzones, each of them corresponding to a particular kind of flow, with a possibly analytical, probably approximate expression of the velocities and temperatures available in each subzone.

3 Tools

The fulfillment of the aforementioned goal is certainly problematic, due to the difficulty of the problem handled. The tools that are required are:

- The physical expertise needed to derive from first principles or inspection salient features of the general flow
- Some geometry handling features for the domain decomposition
- Symbolic abilities to derive and manipulate analytical expressions, either exact or approximate. That includes resolution of the equations, but also scaling considerations at the onset of the problem resolution
- Some form of library of known flows and known methods for obtaining analytical results, thus inducing a rule-based "flow pattern matching" process
- Possibly some nonlinear solving capability to be able to determine intermediate unknowns (like inner temperatures or velocities) that cannot be obtained from procedural application of first principles, evaluation of boundary conditions and flow pattern matching, but are nevertheless needed to proceed smoothly in the domain decomposition and flow pattern matching process

The Maple V environment was chosen for this task, because of its generality, availability and good reviews [17]. In principle, any general Computer Algebra System like Macsyma [12], or Mathematica [18], could be used for the task. It could be argued that the superior pattern matching languages and capabilities of Macsyma or Mathematica might be better suited for the project at hand. However Maple V, with its overall reliability, efficiency and compactness, proved powerful enough.

In case of numerical equation systems beyond the ken of Maple V, the Spark [2] object-based simulation environment was used. It is a modular differential algebraic solver, that has an object-based interface to the user, thus is well fitted to the model sintering approach described above.

DOMAIN DECOMPOSITION

1 Motivation

The computational domain, as as been seen above, is to be decomposed into subdomains were some kind of known flow is assumed, based on physical expertise, first principles, scaling considerations, or other methods. The program will have to perform that decomposition, and therefore have some way of managing the subdomain creation and possible ulterior modification.

In 2D things are simpler than in 3D, but a fully general handling of all possible domain frontiers is beyong the scope and ability of the project. As a reasonable engineering approximation one assumed that all computational domains could be represented as curved polygonal lines. Plain polygons are likely to be the most common occurrence as well, at least as a first approximation: plumes and jets can be represented as trapezes or rectangles, subzones containing boundary layers and adjacent to walls can be represented as rectangles -even though most boundary layer solutions have a power-law shaped profile. A common and convenient approximation will be to have the domains bounded by streamlines -or potential lines in case of inlets and outlets. The coupling between adjacent zones is thus facilitated, at least for the dynamics. Heat transfer between regions remain of course to be calculated.

2 Maple V implementation

With these assumptions, walls and borders can easily be modeled as lists of lists of descriptor : the first descriptor being the coordinates of segment extremities, the following ones being a set of boundary conditions on the frontier, the third one, a string to distinguish wall from ficticious frontiers inside the domain and the fourth one, a frontier equation, which is optional. The need for a list of lists is due to the fact that walls and inside frontiers can be piece-defined regular curves. For example, a vertical wall of 2.5 meters hight is described as follow:

```
Wall:=[[0.,0.],[0.,2.5],
[[u(0,y)=0,v(0,y)=0],[T(0,y)=291]],'wall'];
```

Based on the surrounding conditions, actions are taken to generate domains where one hopefully knows the nature of the flow. The associated domains cover some part of the global problem domain. The uncovered parts are then "filled" with a zonal model. Note that the filler could be another model, for example an interpolating function satisfying the boundary conditions of the uncovered part.

We now present only one iteration of the whole process, which is embedded in a loop. Inputs are a list of segments which describe the geometry domain, dynamic and thermal boundary conditions. Outputs are local velocity and temperature in subzones.
First, the program identifies the engines ie. the driving forces in the domain. It has to detect, if possible, air inlets and air outlets, heat sources, warm or cold walls.
Then, the program determines the name of the model correponding to the engines: boundary layers are attached to walls when applicable, jets attached to air inlets, and plumes attached to heat sources. Qualitative rules specify as much as possible the model. For instance, in case of a jet, a rule distinguishes a wall jet from a free jet and another one specifies if it is a cold or a warm jet, other rules specify its position (horizontal or vertical), estimate its thickness and evaluate its penetration depth.
After that, the program defines the subzone where the detected elementary flow is available.
Finaly, the model is chosen in a library and specifically solved.
Once it is performed, the domains created are taken away from the empty domain that remains to be filled, new boundary conditions are generated for that new empty domain, based on the adjacent solutions just determined, and the same iteration is performed again, and so on until the whole initial domain has been filled with local solutions.
That code can of course be refined. Interactions between conflicting engines is not resolved. It can nevertheless already be put to use on real problems.

EXAMPLES

1 The differentially heated cavity problem

1.1 Analytical study

The differentially heated cavity problem is a pure natural convection problem, with a square enclosure of height H with adiabatic floor and ceiling, an isothermal left warm wall and an isothermal right cold wall. It has been extensively numerically studied, with benchmark solutions available like [3]. [10] gives some insight about the physics at work when one increases the Rayleigh number.

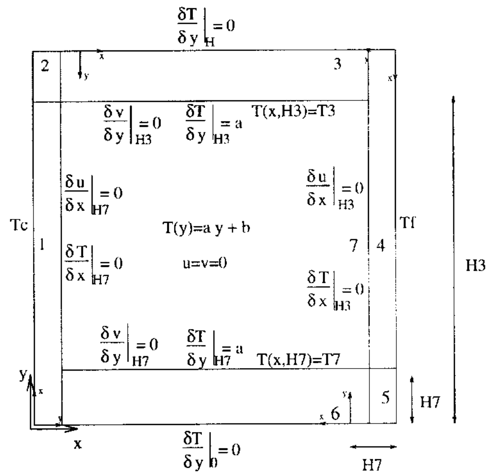
Getting a simplified analytical solution for that problem in the case of air ($Pr = 0.71$) can follow the following lines (fig. 1):

- Calculate the Rayleigh number Ra_H, based on geometry and boundary conditions
- From Ra_H, following [1], derive an order of magnitude for the boundary layer thickness, $(Pr^{-1/4}(H Ra_H{}^{-1/4}))$, the distance from wall to the velocity peak $(Pr^{1/4}(H Ra_H{}^{-1/4}))$, the thickness of the wall jet $(Pr^{1/4}(H Ra_H{}^{-1/4}))$, the velocity scale $(\frac{\alpha}{H}(Pr Ra_H)^{1/2})$ and the Nusselt number $Nu = \frac{hH}{k}$ $((Pr Ra_H)^{1/4})$.
- Compare those derived data with the geometrical data. In the following one will assume that boundary layer thicknesses obtained by the scaling method are small compared to all

characteristic lengths of the geometrical domain

- Derive from there the flow along the vertical jet walls, and assume it is approximately the jet mass flow along the ceiling and the floor
- Match the vertical walls flows with natural convection boundary layer flows, and the horizontal walls flows with polynomial velocity flows matching the boundary conditions, and of thickness initially the same as the one from the vertical flows
- Assume stagnant, purely conductive rectangular core zone

The above is actually quite a good representation of reality at relatively high Rayleigh numbers ($\simeq 10^8$), when the boundary layers are thin and the streamlines are parallel to the walls.



Zones 1 and 5 : Boundary layer model

Zone 9 : Stagnant core with linear thermical stratified model

Zones 3 and 7 : Velocity and temperature polynomial profiles

Zones 2, 4, 6, 8 : Mass flux and thermal flux conservation

Figure 1: Decomposition of the window problem

1.2 Detailed Equations

In vertical zones, we have the equations for the laminar boundary layer expressed in the local reference : x in the flow direction, y perpendicular to x directed to the cavity inside.
Continuity equation :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

Momentum and Energy equations in zone 1 (fig. 1):

$$\begin{aligned} u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} &= g\beta(T - T_\infty) + \nu(\frac{\partial^2 u}{\partial y^2}) \\ u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} &= a(\frac{\partial^2 T}{\partial y^2}) \end{aligned}$$

with boundary conditions :

$$[u(x,0) = 0, u(x,\delta(x)) = 0, v(x,0) = 0, \\ v(x,\delta(x)) = 0, D_2(u)(x,\delta(x)) = 0]$$

$$[T(x,0) = Tc, T(x,\delta(x)) = T_\infty, \\ D_2(T)(x,\delta(x)) = 0]$$

The solutions to the boundary layer problem have been obtained approximately, and not exactly, via so so-called Pohlhausen method, that is essentially a momentum and energy balance on the boundary layer itself. The exact velocity and temperature profiles are then not calculated, but assumed to be "reasonable", for example polynomial and satisfying the boundary conditions. That approach yields an ODE whose unknown is the boundary layer thickness δ depending on the longitudinal independent space coordinate x.
Implementation in Maple V of that method is straightforward. A function performing automatic determination of a lowest order polynomial satisfying the boundary conditions was written. These polynomials are then plugged into the integral equations, yielding ODE's in term of the velocity and thermal boundary layers. The resolution with Maple V's utility for solving ordinary differential equations, `dsolve`, was obtained with difficulties, however, due to the limitations of `dsolve`. But in this case, the resolution is completed.

In the horizontal zones, we have to resolve the following equational system :
Continuity equation :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

Momentum and Energy equations:

$$\begin{aligned} u\frac{\partial u}{\partial x} &= \nu(\frac{\partial^2 u}{\partial y^2}) \\ u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} &= a(\frac{\partial^2 T}{\partial y^2}) \end{aligned}$$

with the boundary conditions :

$$[u(x,0) = 0, u(x,\delta(x)) = 0, D_2(u)(x,\delta(x)) = 0, \\ \int_0^{\delta(x)} \rho u(x,y)dy = FM_{1->2}, \\ v(x,0) = 0, v(x,\delta(x)) = 0]$$

$$[D_2(T)(x,0) = 0, T(x,\delta(x) = T3, \\ D_2(T)(x,\delta(x)) = \frac{T7 - T3}{H7 - H3}]$$

$FM_{1->2}$ is the mass exit flux from the boundary layer zone to the corner zone. We suppose that this mass flux is totally transfered to the top horizontal zone. For the meaning of the others quantities, report to Figure 1.

1.3 Horizontal layer thickness determination

A still more refined model is to take the ceiling and floor jet thicknesses as unknowns, and use balance equations to determine them. To do this, one can consider the upper left and lower right corners of the cavity as places where singular head losses occur. A typical value for a corner is $C_f = 0.9$. Taking that into account, one can justify a 3-fold increase of the thickness of the horizontal zone as compared to the upstream vertical one, as follows:

$$C_f \frac{v^2}{2} = \frac{v^2}{2} - \frac{u^2}{2}$$

That equation means that the speed at the entrance of the downstream horizontal zone is $\sqrt{0.1}$ times the speed at the exit of the vertical upstream zone. Assuming conservation of the mass from one zone to the other (i.e. negligible loss or gain from the core) one obtains a 3.16 expansion of the zone thickness, from first principles, in good agreement with numerical experiments.

1.4 Model connection

Connection of the various zones is of course the tricky part. It is facilitated here by the fact that the core temperature variation is only one dimensional. Thus, vertical boundary layers are connected via the core average (i.e. mid height) temperature, that is taken to be their "ambiant medium" temperature. And horizontal zones (floor and ceiling layers) are connected to the core via the interface temperatures, that are constant because of vertical stratification. We thus end up with two temperature unknowns, the temperature at the contact between the top horizontal air zone and the core, and the temperature between the bottom horizontal air zone and the core.

"Closure" equations for those unknowns are readily obtained by energy balances on the top and bottom horizontal layers. They are linear and thus easily solved for.

1.5 Validation

The above simplified approach has been validated with a CFD code, on conditions for which laminar convection does take place (which supposes very small temperature differences: the hot wall temperature is 293 K, the cold wall temperature 292.9). The total zone height is 2.5m. The Rayleigh number is thus of the order of magnitude 2.10^8.

Resolution of the closure temperature equations leads to the values 292.984 K (top) and 292.9159 (bottom). From those values numerical evaluation of the analytical profiles in the layers and in the core can be carried out, yielding analytical expressions depending only on x and y, the space coordinates.

Validation can then be performed. The results in temperature in the median horizontal plane are virtually identical (cf figure 2), except for the numerically demonstrated temperature overshoot at the vertical boundary layer limits, that is not part of the analytical model.

In the median vertical plane, results are off by at most 30 % at the ceiling or the floor, and very close in the core zone (cf figures 4 and 5).

The boundary layer thicknesses are well predicted as well.

However, the analytical model overevaluates the velocity in the horizontal boundary layers (and thus in the vertical ones) by a factor 4 (cf figure 3). One reason might be the fact that turbulent effects are not taken into account. Lower speeds are to be expected once that is done. Another reason is the fact that we resort, close to the hot and cold walls, to a vertical natural convection boundary layer assumed to be in contact with an isothermal medium, while the real core is thermally stratified, thus limiting the buoyancy induced mass flow while the altitude increases on the wall.

Last, the heat flux density along the hot vertical wall was compared for the numerical and the analytical approach (cf figure 6). There is an analytical underevaluation by 46 % at the floor, and an overevaluation at the ceiling. At midheight the analytical model underevaluates the heat flux density by about 22 %. Those discrepancies are probably the combined effect of the nonstratified vertical boundary layer model used in the analytical approach, and the fact that turbulence is not taken into account. With a stratified boundary layer model as in [8] , one would expect an increase of the heat flux at the bottom of the hot wall, and a decrease of that flux at the top of the hot wall. Both these trends go in the right direction, and justify considering the introduction of the vertical boundary layer in contact with a thermally stratified medium, in order to replace the usual natural convection boundary layer in contact with an isothermal medium.

Furthermore the transition to turbulence at mid-height would increase the heat flux above the transition point. So one sees that improving the analytical model would surely improve the heat flux at the lower part of the hot wall. The effect on the upper part are unclear.

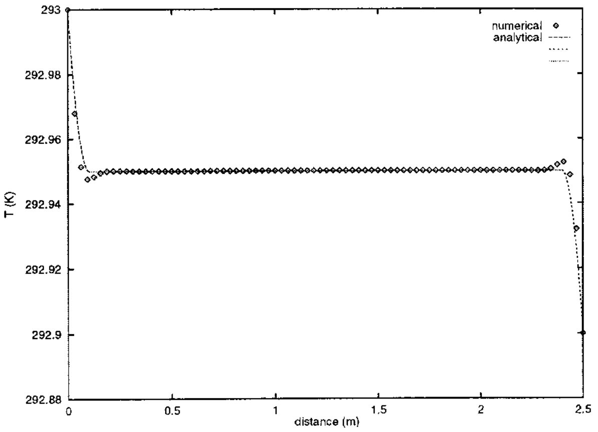


Figure 2: temperature profile in the median horizontal plane

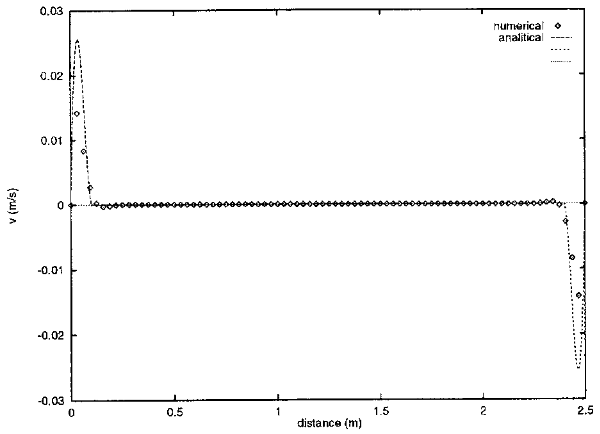


Figure 3: Speed profile in the median horizontal plane

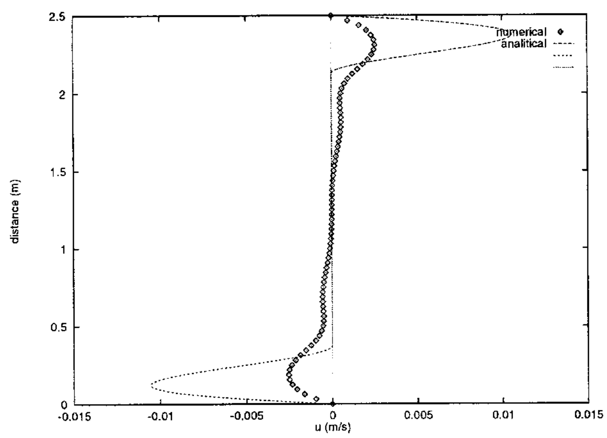


Figure 4: Speed profile in the median vertical plane

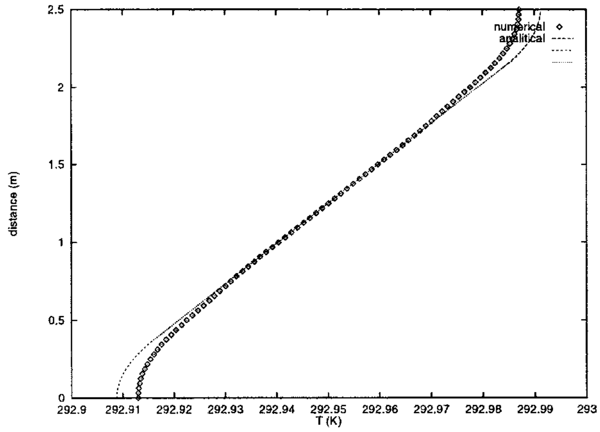


Figure 5: temperature profile in the median vertical plane

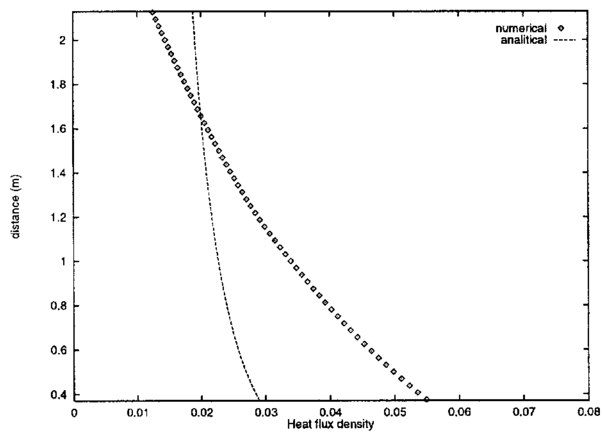


Figure 6: Heat flux density along the hot vertical wall

2 Introduction of zonal models

The core simulation can be refined by introduction of a partitionning of the core, using zonal models. Unfortunately, zonal models are based on fractional power laws between the pressure difference between zones and the mass flow. That introduces severely nonlinear equations in the overall simulation, and closure equations need to be solved for determining the boundary state variables (temperatures essentially) intervening both in the boundary layer behavior and in the zonal model behavior. The Maple V numerical solver, however, fails to solve the equations even for the simple four zone partitionning of the core.

One fixes that problem by solving numerically the closure equations generated by the expert system, then feeding them back for it to continue until the end.

2.1 Spark implementation

The numerical solver used for that purpose is Spark [2]. Spark has been extensively used for the resolution of zonal models, for example in [13] and [19]. Spark allows to consider equations subsystems as objects to be connected to other objects, which is specially suited to the partitionning approach presented in this paper.

The approach proposed is simply to connect 2D zonal models to the "closure equations" for the temperatures between core and hozizontal layers. We end up with a standard zonal problem, plus the closure equations, that are solved for simultaneously. The simplicity of this coupling is due to the fact that there is no heat transfer between the vertical layers and the core, thanks to the definition of the vertical layers, that include not only the vertical boundary layers, but also stagnant isothermal fluid, that on average exchanges no heat with the core. So, overall, the assumption is that the vertical zones do not exchange heat with the core, only the horizontal layers do so. That assumption is reasonably supported by numerical experiment or study of heat lines (see [1]). Full automation of that process would be in principle easy by invokation from Maple V of a shell running the numerical solver, and retrieving of the results via a pipe of intermediate files. That is not done yet.

2.2 Results

Several tries were performed with Spark. Essentially they are divided into two main categories: 3D models mimicking 2D, Real 2D models.
The 3D models are mimicking 2D phenomena by using simply having one layer of zones in the direction perpendicular to the 2D domain considered. Then provisions must be taken to ensure that there is no flow or energy escaping along that perpendicular dimension. Those models were successfully implemented in Spark and do verify 2D conservation laws, but only at the expense of having slightly permeable boundaries in the 2D domain representing the core, when the hypothesis is that the core does not exchange mass with the outside. They are therefore approximations. But this simulation shows a non negligeable convective heat flux versus the conductive heat flux through each side of the core. Then, the effect of the conductive heat flux i.e. the core thermical stratification, becomes negligeable.

The pure 2D models tried are themselves subdivided into two parts: the 2x2 simulation (with 4 zones piled up two above two) and the "stacked" simulations were zones are only piled up in the vertical direction.

The 2x2 simulation is hard to converge. One first had to converge a purely zonal 2x2 simulation, by guessing its iteration variables based on hydrostatic equilibrium: thoses starting values lead to convergence of an otherwise difficult problem. Then one couples the 4 zones with the horizontal top and bottom layers, and guess the iteration variables of that new problem based on the purely zonal 2x2 problem.

The "stacked" simulation, on the other hand, converges easily. However, the results lead to no flow between the zones, which could be expected since no circulation is possible in stacked zones, due to the impermeability of the side walls. That simulation then amounts to try to simulate conductive behavior in the core via a convective model, which is not appropriate. The limitation would actually fall if mass transfer was allowed between the core and the vertical boundary layers on each side of the core.

CONCLUSION

This work is still under way. At this stage a full package is available for similarity solutions of boundary layer equations, automatic generation of profiles fitting specified boundary conditions, and Pohlhausen's integral method. A package was also developed for partitionning the domain, provided the user specifies the frontier equations -so these frontiers are not yet automatically determined by use of expert rules. Last, a qualitative engine deciding what analytical solution to use is being developed.
The main difficulties of this work are:

- the good specification and the structuration of the problem by a set of elementary qualitative rules.
- to find the analytical models.
- the control of conflictual situations.
- the numerical resolution of closure equations.

Also, symbolic computing softwares are more or less reliable and must sometimes be helped.

Now work is going on the rule-based system that will actually perform the decisions to invoke such or such model as the need arises. A trivial implementation would be to systematically partition the studied domain into zonal models, thus recovering the results of [19].
We have to grow the expert rules set and the models library to improce the modelling results and to handle others configurations with jets and plumes, for example. In the same time, we would like to couple our code with a conduction model. Both will be linked in an iterative process like the Newton-Raphson method. The convergence will

be obtained when heat flux are equal on both side of the wall. Our aim is to test the relevance of our tool on real cases, 3D problems, for example, like a ventilated room with a window. In the future, we can imagine to develop the code for the resolution of instationary problems or pollutant transport problems.

References

[1] A. Bejan. *Convection Heat Transfer.* John Wiley and Sons, 1984.

[2] F. Buhl, E. Erdem, F. Winkelmann, and E. Sowell. Recent improvements in spark: Strong component decomposition, multivalued objects and graphical interface. In *Proceedings of Building Simulation '93, Adelaide, Australia.* IBPSA, 16-18 August 1993.

[3] G. De Vahl Davis. Natural convection of air in a square cavity: a bench mark numerical solution. *International journal for numerical methods in fluids*, 3:249–264, 1983.

[4] H. E. Feustel and Al. The comis infiltration model. In *Proceedings of Building Simulation 89*, Vancouver, June 23-24 1989. IBPSA.

[5] A. Gadgil. On convective heat transfer in building energy analysis. L.B.L. Report LBL-10900, Lawrence Berkeley Laboratory, May 1980.

[6] G. Gan, H. B. Awbi, and D. J. Croome. Simulation of air flow in naturally ventilated buildings. In *Proceedings of Building Simulation '91, Sophia-Antipolis, Nice, France.* IBPSA, 20-22 August 1991.

[7] C. Inard. *Contribution à l'étude du couplage thermique entre un émetteur de chauffage et un local.* PhD thesis, Institut National des Sciences Appliquées de Lyon, July 1988.

[8] A. K. Kulkarni, H. R. Jacobs, and J. J. Hwang. Similarity solution for natural convection flow over an isothermal vertical wall immersed in thermally stratified medium. *IJHMT*, 30(4):691–698, 1987.

[9] J. Lam, R. Yuen, and T. Tau. Improvements to user-friendliness of a computational fluid dynamics (cfd) code for simulation of air movement in buildings. In *Proceedings of Building Simulation '93, Adelaide, Australia.* IBPSA, 16-18 August 1993.

[10] N. C. Markatos and C. A. Pericleous. Laminar and turbulent natural convection in an enclosed cavity. In *Proceedings of The 21st National Heat Transfer Conference*, Seattle, Washington, July 24-28 1983.

[11] N. C. Markatos and C. A. Pericleous. Laminar and turbulent natural convection in an enclosed cavity. *IJHMT*, 27(5):755–772, 1984.

[12] MIT. *MACSYMA Reference Manual, version 10.* Cambridge, MA, 1983.

[13] J.M. Nataf and E. Wurtz. Un exemple d'approche orientée objet appliquée à la thermique du bâtiment: l'environnement spark. *RGT*, XXXIV(399):237–246, March 1995.

[14] K. B. Ranger. A complex variable integration technique for the two-dimensional navier-stokes equations. *Quarterly of Applied Mathematics*, XLIX(3):555–562, September 1991.

[15] E. A. Rodriguez, S. Alvarez, and I. Caceres. Prediction of indoor temperature and air flow patterns by means of simplified zonal models. In *Proceedings of ISES Solar World Congress, 1993, Budapest, Hungary*, volume 6, 1993.

[16] G. Shiralkar, A. Gadgil, and C. L. Tien. High rayleigh number convection in shallow enclosures with different end temperatures. *IJHMT*, 24(10):1621–1629, 1981.

[17] M. Wester. A review of cas mathematical capabilities. 1994.

[18] S. Wolfram. *Mathematica - A System for Doing Mathematics by Computer.* Addison-Wesley Publ. Co., Redwood City, CA, 1988.

[19] E. Wurtz. *Modélisation tridimensionnelle des transferts thermiques et aérauliques dans le bâtiment en environnement orienté objet.* PhD thesis, ENPC, France, December 20 1995.

NOMENCLATURE

a : air thermal diffusivity $[m^2s^{-1}]$
C_f : fitting loss cœfficient
g : acceleration due to gravity $[ms^{-2}]$
H : size of the square cavity $[m]$
Ra_H : Rayleigh number, $Ra_H = \frac{g\beta(T-T_\infty)H^3}{a\nu}$
u : velocity component (x direction) $[ms^{-1}]$
v : velocity component (y direction) $[ms^{-1}]$
T : temperature $[K]$
T_c : hot wall temperature $[K]$
T_f : cold wall temperature $[K]$
T_∞ : average temperature $[K]$, $T_\infty = (Tc+Tf)/2$
β : volumetric cœff. of thermal expension $[K^{-1}]$
δ : boundary layer thickness $[m]$
ν : molecular kinematic viscosity $[m^2s^{-1}]$
ρ : air density $[kgm^{-3}]$