M2M&ROOFSOL: A TOOL FOR EVALUATING THE COOLING PERFORMANCE OF PASSIVE ROOF COMPONENTS USING THE MODEL SYNTHESIS METHOD

Gilles LEFEBVRE

École Nationale des Ponts et Chaussées 6&8 av. Blaise Pascal 77455 Marne la Vallée – France

gilles.lefebvre@enpc.fr

ABSTRACT

This communication presents a tool, **m2m&Roofsol**, which has been developed in the framework of a European project, ROOFSOL[1][11], dedicated to the study of passive cooling by roof components. This tool is based on an existing simulation tool, m2m, which allows detailed analysis and simulation of the thermal dynamic behaviour of large building envelopes, and model size reduction techniques[2][3].

Two main problems have been solved in this software extension. The first one is the coupling between an envelope model and a roof model respecting an existing **m2m** description strategy, the second one is the linearization of non linear models of the roof components.

These modifications of m2m are extremely useful, not only for modelling roof component, but as a general modelling solution for many building components other than simple conductive walls and windows. Due to the limited length of this communication, only the principles of the methods and techniques will be shown.

INTRODUCTION

Modelling specific building components in order to evaluate their efficiency and to convince the designers of including it in their projects requires that analysis and simulation tools are able to include it. This was one of the main objectives of ROOFSOL, a project dedicated to the study of passive cooling with roof solutions.

Linear phenomenological models have proved to be efficient and accurate enough when used to describe the behaviour of very complex building structures and geometry's. But specific components as the ones studied in ROOFSOL, which are based on "special" phenomena as evaporation-condensation of water or long wave radiation toward the sky, are difficult to elaborate directly as linear models.

In order to not develop an entire other new simulation tool, it has been decided to reuse a validated building envelope modelling tool that we are developing from several years, m2m. m2m allows detailed analysis and simulation of the thermal dynamic behaviour of large building envelopes and makes use of modal and model size reduction techniques [2][3]. m2m has been from the beginning thought and developed using modular and object concepts, fact that makes that it has been relatively easy to include new modelling facilities as the one required in ROOFSOL.

Two main problems had to be solved. The first one was to include the possibility to describe special roof components, to model it and to coupled it to the rest of the envelope. The other one was that it was relatively simple to get non linear models of the components, as they were available in ROOFSOL, but the modelling techniques used in m2m require that the resulting model be linear versus the state variables.

We implemented in m2m a model synthesis facility which allows to couple heterogeneous state space models, and a general linearization procedure which permits to deduce a linear model from a non linear one.

These new facilities have been used to model special cooling roof components. We will show the different modelling strategies which have been used, depending on the kind of components.

Many parts of **m2m** and **m2m&Roofsol** are the result of contributions of E. PALOMO (stochastic simulation, linearization process, meteorological data process, etc.).

M2M

M2m is a set of executable modules which perform a specific task. These modules can be run in a batch way and can be linked together through files. Some of the modules which are used in m2m&Roofsol are described here under. It is here enough to know that m2m builds a finite difference model based on the building envelope description, performing an automatic meshing of the walls. A configuration file indicates some modelling hypothesis and requirements (how are modelled the radiative and convective exchanges? which inputs variables are needed? fluxes in the air zones, fluxes at the air-wall interfaces? which outputs are needed? etc.). The resulting model is a multizone one, may contain heat sources in the air zones, in layers of the walls, at the interfaces between the wall and the air, the temperature may be imposed in any zone, the incident solar radiation may be absorbed by outside opaque surfaces, transmitted by transparent walls, and then distributed among the inside opaque surfaces following a repartition rule, etc. Very huge models may be modelled with m2m as it accepts models including several thousands nodes, and may take advantage of many size reduction techniques. M2m is freely available at [3] for dos and linux operating systems.

N°	Name	Action
0	m2m0.exe	Builds a finite difference model
1	m2m1.exe	Transforms a state space model in a modal one
6	m2m6.exe	Complete a config.3 or config.5 file by looking for correspondances between the input file columns and the model inputs name
9	m2m9.exe	Performs a deterministic simulation with a state space model
31	m2m31.exe	Splits on a building description in several finite difference model components and coupling description
33	m2m33.exe	Synthetises state space models in a unique one
99	m2m99.exe	m2m.99 performs deterministic simulations filtering the inputs/outputs
S0	m2mS0.bat	Builds a synthesised model of a usual building envelope
S1	m2mS1.bat	Builds a synthesised model of a building envelope coupled with a special roof
S2	m2mS2.bat	Builds a synthesised model of a building envelope coupled with a special roof which offers a supplementary flux source

Table 1: some m2m executable modules

MODEL SYNTHESIS

Model synthesis relies on a systemic description of complex physical objects. The approach is quite similar to the one used in Motorlab [4]except that the objective and the process are distinct. In Motorlab, the coupling is processed at simulation time, and may link heterogeneous models. In the model synthesis, all models are linear state space models and the result is a linear state space model.

The model synthesis is a systemic and systematic approach which allows to model very complex objects by taking advantage of modularity and reutilisability [5]. At every stage of the model synthesis process, the synthesised model may be reduced and the memory and storage requirements are lower than in a traditional approach.

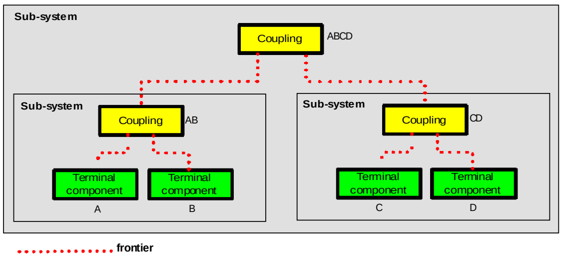


Figure 1: system structure and characteristic elements

In m2m, all models are modelled as state space models, which have the following form:

$$\frac{d\vec{X}}{dt} = \boldsymbol{A}\,\vec{X} + \boldsymbol{B}\,\vec{U}$$
$$\vec{Y} = \boldsymbol{C}\,\vec{X} + \boldsymbol{D}\,\vec{U}$$
$$\vec{Z} = \boldsymbol{E}\,\vec{X} + \boldsymbol{F}\,\vec{U}$$
$$\vec{T} = \boldsymbol{G}\,\vec{X} + \boldsymbol{H}\,\vec{U}$$

where $\vec{X}$, $\vec{Y}$, $\vec{Z}$ and $\vec{T}$ are respectively the state vector (which contains all the state variables), the output vectors (which contains all the outputs which could be required in the future use of the model), the observation vector (which contains all the variables one want to observ in the model) and the field vector (which may contain a discretised variable as a whole field temperature).

These models may have been the result of applying a finite difference method, a modal one, a reduction technique, or a previous synthesis process.
A system is described as a set of couplings between different models. A coupling is described as logical relation (which must have a physical meaning) between frontiers of models. A frontier is a set of connectors of a given model which correspond to the required variables for expressing the coupling relation. For example, a simple thermal contact will require frontiers which respectively contain a temperature and a thermal flux. Here is an example of a system as a set of three coupled models A, B and C.

```
%BEGIN HEADER
example
%END HEADER
%BEGIN MODEL
A.dif
B.dif
C.dif
%END MODEL
%BEGIN COUPLING
CONTACT|2|A.southwall_side1|1.|B.southwall_
Side1|200.00|FXD_south_inp|120.00|
CONTACT|2|A.window_side1|1.|C.window_side1|
20.000|FXD_south_inp|0.0000|
CONTACT|2|A.southwall_side2|1.|B.southwall_
side2|200.00
CONTACT|2|A.window_side2|1.|C.window_side2|
20.000
%END COUPLING
```

These couplings are translated under a mathematical form as equations which link the involved input and outputs of the coupled models:

$$\boldsymbol{K}_{y,i}\vec{Y}_{c,i} + \boldsymbol{K}_{c,i}\vec{U}_{c,i} + \boldsymbol{K}_{e,i}\vec{U}_{e,i} + \boldsymbol{K}_{n,i}\vec{U}_{n,i} = \vec{0}$$

The synthesis process is then a mathematical transformation which calculates the state space model of the whole system, and handles the algorithmic process in order to eliminate, to rename, unnecessary connectors and variables.

The file which describes a (sub)system as a set of coupled models can be written by hand. M2m produces automatically the required coupling files for building envelope synthesised models, and envelope and special roof synthesised models.

MODEL LINEARIZATION

The linearization process has been described in [5]. The results is a simple and systematic way of deriving a linear model from a non linear one for a given meteorological sequence.

MODELS GENERATION

Preliminaries

m2m&Roofsol is a set of three kinds of executable modules. The first set, which origin is m2m, are compiled "C" codes. The second one are dos or unix scripts. The third ones are command scripts for scilab [6]. All m2m&Roofsol modules must be used in "batch" mode: a module reads files, processes it, performs calculations, and finally write new files. The names of the files read or produced as the actual behaviour of the module (modelling hypothesis, etc.) must be indicated in a configuration file which default name is config.x when the module is m2m.x. For example, `m2m.0` looks for a file which name is `config.0` in the current directory where the module is run. It is possible to force the module to use another configuration file, or to indicate a different location by giving an argument to the executable module, thanks to the "`-f`" option. For example, `m2m.8 -f ../foo.cfg` forces `m2m.8` to the configuration file `8.cfg` which is located in the "father" directory of the current one. Many optional arguments can be specified. "-version" displays information identifying the exact code you are using, as the version number, where, when and by who the code has been compiled; when asking for information about a possible problem, this information is required.

When it has been possible, dos script files are provided in order to hide the low-level commands and to provide easy to use tools to the user.

Envelope model

The description of a building envelope is made in two specific files, and three library files. The first specific file contains the description of the structure and the composition of the envelope. The second specific file contains the description of the ventilation paths and flows. This second specific file is logically related to the first one as it uses zone names which are defined in the first file. Coherence verifications are done at run time. The three library files contain respectively the thermophysical characteristics of materials, the compositions of wall (which uses the names of the materials which are defined in the first library) and finally characteristics of glazing. The libraries may contain many elements and can be shared between different envelope models.

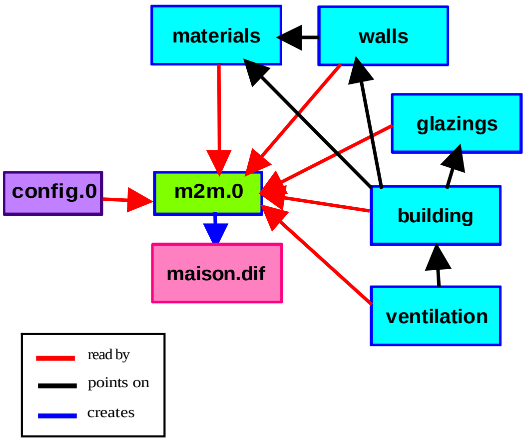


Figure 2 : creating a model of usual building envelope with m2m.0

The actual behaviour of the different m2m modules is determined by a configuration file which may contain the actual name of the input and output files, some parameters, etc. In order to simplify the model creation process, the execution can be run using a script called m2mS0.bat which refers to standard configuration files and avoids to write a `config.0` configuration file specific to the particular project which is studied. The script must be used as follows:

```
m2mS0 material wall glazing project
```

where :

- `material` is the base name of a material file called `material.mtr`
- `wall` is the base name of a wall composition file called `wall.cmp`
- `glazing` is the base name of a glazing file called `glazing.glz`
- `project` is the base name of the project ; the envelope description file called `project.dsc` and the ventilation file called `project.air` must exist.

Whole building model

Whatever kind of coupling having been described, the synthesis procedure remains the same, as the required frontiers and coupling information are automatically generated.

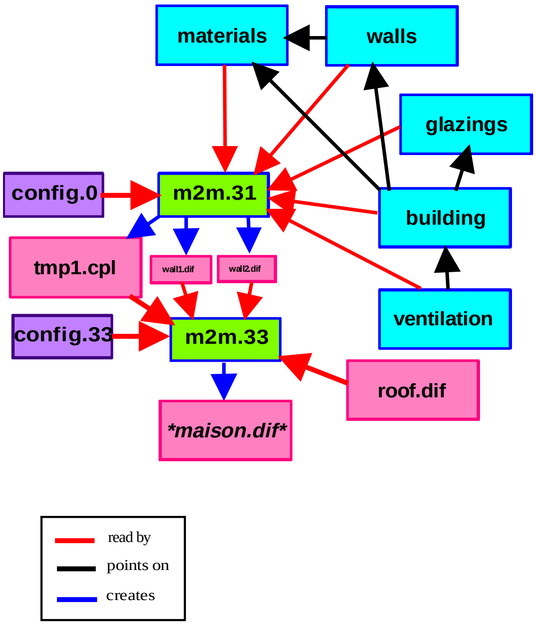


Figure 3: synthesis process

The following m2m modules are used :

- m2m.31 splits up a building description in as many as required *finite difference* model of parts (walls, windows, radiative and convective models, coupling models) of a building.
- m2m.33 then synthesises the envelope and the roof models

These processes are gathered in a single one, m2mS1[1]. This stage is called *synthesing* a building and roof description. If some special component (called for example *foo*) is specified in the building description file, a file called *foo*.dif is searched for by the synthesis process in the working current directory. Depending on the coupling file which is used, the coupling strategy is different. Different script command files are provided for every coupling strategy: m2mS1 couples a building envelope with a roof component through a contact connection; m2mS2 performs the same thing as m2mS1 except that it furthermore proceeds to the coupling of supplementary cooling fluxes with an internal virtual zone called “`cooledzone`”.

[1] When *x* is a letter, the corresponding executable in a dos/windows environment is a batch m2m*x*.bat file and must be run typing the command m2m*x*. In a *nix environment, the executable is a script m2m*x* file and is run by typing the command m2m*x*.

BUILDINGS WITH SPECIAL ROOF COMPONENT DESCRIPTION

A usual roof is modelled as a wall, or as a set of walls and zones. A building which integrates a special roof component is described exactly as a usual building for all parts but the special roof, which is described as a *special roof component.*

Special roof declaration

A *special component* as a roof is specified as a special block in the building description file. Roofs are specified in a piece of text limited by the reserved tags : `%BEGIN SPECIAL ... %END SPECIAL`, and using two lines per roof:

```
%BEGIN SPECIAL
|myrad|86.0|radiator|EXTERIEUR    |1E6|
|     |    |        |intermediate|1E6|
%END SPECIAL
```

The name indicated in the first column (first row) is the one of the particular component which will be searched for, and that which will have to be provided by another process (the linearization process). The second column (first row) contains the area of the roof. The third one (first row) contains one of the reserved keyword roof types which actually are those who are indicated in a configuration file, `roof.cmp`:

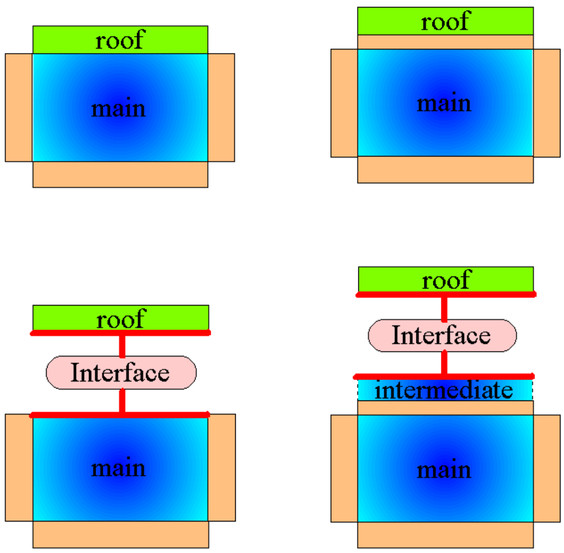
```
green_roof
radiator
roof_pond
diode_roof
evap_roof
```

The three first columns of the second row must be empty. The fourth column contains the name of the ambient connected to the respective side of the roof. The first line **must** contain the outside ambient name, whatever name it has. The fifth column contains the superficial heat exchange coefficient ; the one corresponding to the outside face (first line) **must** be equal to 1E6, which is a finite numerical transcription of an infinite value as the exchanges with the outside ambient are already taken into account inside the special component model .

It must be noticed that including a special roof component does not forbid to have a traditional roof at the same time on a part of the building. The respective areas and logical connections should then be adapted to the actual described configuration.

Description of the contact between an envelope and a special roof

Two different strategies can be used to incorporate a cooling roof component in a building model.



first coupling strategy

second coupling strategy

Figure 4: two coupling strategies

The first one considers that the roof component is the whole roof structure, including the ceiling part which is in contact with the main zone. This is the kind of models which is provided by the linearization process, and then the more general and standard way of actuating. The envelope which will be connected to the roof component must include the roof description as in the following example; the main (supposed!) cooled zone must have a prescribed temperature ("1" in fourth column of its description in the ZONE text block.

```
%BEGIN BUILDING
...
%BEGIN ZONE
------------------------------------------
-- ROOMS      | Volume | Capacite
--            |  (m3)  |  (J/K)
------------------------------------------
OUTSIDE        |1.0    |0.0    |1
main           |215.0  |0.0    |1
%END ZONE
...
%BEGIN ROOF
|myrad |86.0|radiator|OUTSIDE|1E6|    |
|      |    |main    |6.1    |   |    |
%END ROOF
%END BUILDING
```

The second strategy for including a roof component is to install it over an existing roof yet integrated in the envelope, generally a terrace.

This will be the case when one will want for example to couple directly a linear radiator model with an envelope without passing through the linearization procedure. In this case, as the present version of m2m does not include the possibility to prescribe a temperature anywhere but in zones, a trick must be used. The component itself and the terrace will be connected through a virtual zone called `intermediate` through infinite heat superficial exchange coefficient values in order to simulate a perfect contact. Creating the model of a special roof of this type coupled with an envelope is then performed using the following command:

`m2mS1 material composition glazing project`

```
%BEGIN BUILDING
...
%BEGIN ZONE
------------------------------------------
-- ROOMS         | Volume| Cap.  |Imposed
------------------------------------------
OUTSIDE          |1.0     |0.0    |1
Main             |215.0   |0.0    |
Intermediate     |1.0     |0.0    |
%END ZONE
%BEGIN WALL
------------------------------------------
-- WALLS|AREA |Compos.  |  ROOM        |HGl|
------------------------------------------
...
|ceiling|86.0 |C         |intermediate|1E6.0
|       |     |          |main        |6.1
...
%END WALL
...
%BEGIN ROOF
|myrad   |86.0 |radiator |OUTSIDE     |1E6
|        |     |         |intermediate|1E6
%END ROOF
%END BUILDING
```

Components with independent output cooling flux

Some components have another way of cooling than the only contact with an envelope by their bottom side. They may offer another output cooling flux which may be even the main way of cooling for some of them.

It is the case, for example, of a cooling water-based [9], or an air-based radiators or an evaporative cooling collector.

The problem that rises with these components is that these output cooling fluxes can be and generally should be controlled because the output flux can be sometimes a cooling flux, and sometimes a heating flux.

It is the case for example of a radiator which actuates as a cooling component at night time, but maybe as a heating component at day time.

Two possibilities have then been studied to couple this other output flux to the building. The first one is based on a permanent coupling without controlling the flux. The second one connects the output only when the component actuates as a cooling one.

Non controlled independent cooling flux

The permanent coupling is modelled by integrating a virtual supplementary zone in the envelope which must be called `cooledzone` (this is the reserved word that the automatically generated coupling file will use) and must be connected to the main zone to cool by an infinite air flow (a numerical infinite, of course!). When generating the envelope model with `m2m.31`, a coupling file called `tmp4.cpl` is generated and include the description of the coupling `FRT_cooledzone` with `FRT_cooling`. This coupling assumes that the flux coming from the component is added to the main zone air of the building with an efficiency equal to 1.

```
…
---------------------------------------------
-- ROOMS      | Volume | Capacite
--            |  (m3)  |  (J/K)
---------------------------------------------
OUTSIDE        |1.0    |0.0    |1
main           |215.0  |0.0    |
intermediate   |1.0    |0.0    |
cooledzone     |1.0    |0.0    |
%END ZONE
```

Table 2

```
%BEGIN VENTILATION
------------------------------------
--AMB DEPART | AMB ARRIVEE|  Debit |
--           |            | (m3/s) |
------------------------------------
|main          |OUTSIDE       |0.033  |
|OUTSIDE       |main          |0.033  |
|main          |cooledzone    |1E6    |
|cooledzone    |main          |1E6    |
%END VENTILATION
```

Table 3

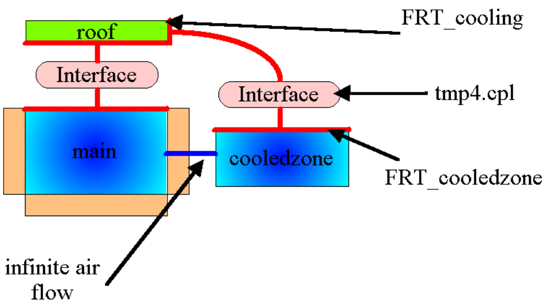


Figure 5: permanent connection of an independant cooling output

With the same logic, the flux can be injected at any place in the building envelope, in any zone, or in the middle of a wall by inserting a virtual zone as shown in the second strategy in Creating the model of a special roof of this type coupled with an envelope is then very simple. It is sufficient to type the following command:

```
m2mS2 material composition glazing project
```

Controlled independent cooling flux

If we want to couple the output flux of the component only when cooling, a filtering process must be adopted.

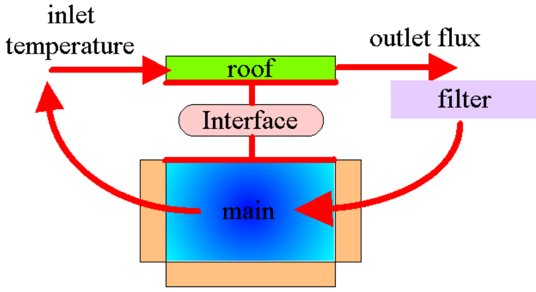


Figure 6: principal of filtering the component output flux

As m2m does not include any explicit control possibility, the control must be simulated outside the simulation itself. An iterative approach is adopted. A first simulation of the envelope and the component coupled only by their contact interface is performed, assuming, for the component, that the inlet temperature is for example 25°C and that there is no incoming flux in the zone to be cooled. A simulation is performed and, among the results, is the evolution of the flux that the component may offer. This output is filtered in order to keep only the values corresponding to a cooling for the building. Filtering is performed thanks to a specific m2m utility, `m2m.99`. Then a new simulation is performed, but using as the component inlet temperature, the previously calculated inside main zone temperature, and connecting the filtered cooling flux to the input corresponding to fluxes which are eventually added to the main zone. This process is then iteratively repeated until convergence. We practically observed that two iterations are sufficient to have a good evaluation of the results and is not really time consuming. Creating the model of a special roof of this type coupled with an envelope is then very simple. It is sufficient to type the following command:

```
m2mS1 material composition glazing project
```

and the coupling of the external supplementary cooling will be handle during the simulation, as explained before.

SIMULATIONS

Three kinds of simulation tools for building thermal analysis should have been available in m2m&Roofsol (see Figure 7):

- The conventional one (deterministic simulation tool), which performs time integration of the building model. The meteorological inputs required must be supplied as time series. The building behaviour is described as the time evolution of the model outputs. This simulation tool is recommended for detailed analysis.
- The harmonic simulation tool, which allows quick estimations of the heat/cooling building requirements. The meteorological input variables are now represented by means of

Fourier series, with a reduced number of harmonics. (not yet implemented)

- The stochastic simulation tool, which is specially oriented for performing quick analysis of thermal comfort [8]. The climate, as well as the thermal building behaviour, is now represented by a reduced set of statistical regularities (e.g. means values, standard deviations, autocorrelations and cross-correlations, probability curves).

Only the first and last kinds of simulation have been implemented.

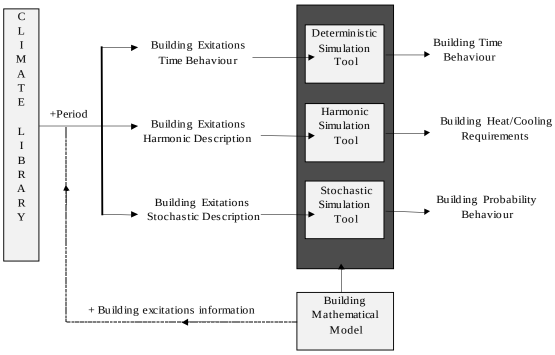


Figure 7: m2m&Roofsol simulation tools.

Adapted meteorological data files.

The meteorological data files required by the different simulation tools must be created before simulations. This can be done automatically from the m2m&Roofsol climate library using specific tools or manually.

FXD_global	Global solar irradiance on an horizontal plane (w.m^{-2})
FXD_diffus	Diffuse solar irradiance on an horizontal plane (w.m^{-2})
FXD_direct	Direct solar irradiance (w.m^{-2})
TMP_EXTERIEUR	Outdoor dry bulb temperature (°C)
HMR_EXTERIEUR	Outdoor relative humidity (%)
VEW_viento	Wind velocity (m.s^{-1})
TMP_sky	Sky temperature (°C)
DPV_EXTERIEUR	Outdoor air vapour depression (Pa)

Table 4: contents of the m2m&Roofsol meteorological files

m2m&Roofsol is supplied with a small climate library. For each one of the localities available, the library includes 12 binary files (one by month) containing a Typical Meteorological Year (representative hourly time series), 12 binary files providing an harmonic description of the data (Fourier coefficients for the most important harmonics); and 12 binary files where the stochastic description of the locality climate is included (some statistical regularities of the time series).

Adapting climate files from the M2m&Roofsol library to a particular analysis means:

- Selecting the files corresponding to the locality and the period desired.
- Adding new meteorological variables, mainly the solar fluxes on the different building facades.
- Deleting the meteorological variables which are not involved in the building analysis.

The code performing this task is **ClmSim**, a SCILAB script.

POST-PROCESSING

All deterministic simulators produce data files that can be read by gnuplot, a free grapher which can be found in every good ftp site. A copy of gnuplot for windows95 is included in the m2m&Roofosl distribution. A deterministic simulation can potentially produce four result files, one for the outputs, one for the observation variables, one for the whole field, and one for the state variables evolution.

A SHORT SUMMARY OF A MODELLING SEQUENCE

A first envelope model

In order to begin to use the m2m package, the easiest way is to use the data contained in the `data` directory. To do that, create a work directory, for example `tmp`, and copy in it the contents of the `data` directory:

```
mkdir tmp
cd tmp
cp .../M2m/data/* .
```

It is now possible to run the `m2m` modules. We begin to build the finite difference model of the building envelope, which has been described thanks to the files `material.mtr`, `composition.cmp`, `glazing.glz`, `maison.dsc and maison.air` (it is a simple monozone building with a unique facade which receives short wave radiation).

The simplest way for running the model creation process is to type:

```
m2mS0 material composition glazing house
```

If this command produces the following message: "m2m.0: Command not found", you should modify your environment variable `$PATH` in order to be able to use the m2m executable modules. If all is OK, you should see that some indications are displayed (because we specified that we wanted a "verbose" execution with the `-v` argument), than some messages indicating the progress of the calculations. The calculations are rapidly finished.

In order to have a better control on the created model, it is possible to use the configuration faile provided and to modify it. After modifying the `config.0` file, we now can run `m2m.0` :

```
m2m.0 -v -f config.1
```

Depending on the choices you made in the configuration file, some files are created.

Describing a particular envelope and creating the model

In order to get a building envelope model, the user should use the following process. First, the user should verify, create or modify the libraries :

- If there is no existing material library, first create one ; if some required materials are missing, add it to the existing library.
- If there is no wall composition library, create one ; if some required compositions are missing, add it to the existing library.
- If there is no glazing library, create one ; if some required glazing are missing, add it to the existing library.

The envelope can then be described thanks to two different files : a description of the structure of the envelope, and a description of the ventilation of the envelope :

- If there is no existing adequate envelope description, modify an existing one or create one from scratch
- If there is no existing adequate ventilation description, modify an existing one or create one from scratch.

Then create the model by running m2mS0 as explained before.

Connecting a roof component model with an envelope model

Copy the previous component model in the directory where is located the building description files. Include the special roof component in your building description and include the connections with the rest of the structure. Take care to place an infinite heat exchange coefficient on the component side connected with the outside ambient. Depending on the connection between the component and the envelope (contact with a solid support or not), the interior heat exchange coefficient should be a finite number or an infinite one. Then run:

```
m2mS1 material composition glazing house
```

Idem *plus external supplementary cooling flux without any control*

Proceed as in the previous case. Add a special zone called "cooledzone" in the description file (*.dsc) and connect it to the zone you want to cool by an "infinite" flow rate in the ventilation file (*.air). Then run:

```
m2mS2 material composition glazing house
```

CONCLUSION

All the mechanisms presented in this communication are implemented, but using m2m&Roofsol as is requires access to many low level information which should be hidden behind a Graphical User Interface. In order to be effectively used by designers, the libraries should be filled up with more examples, and the roof component library should be completed. Furthermore, the GUI should include some "intelligent" agent for helping the designer to choose the right modelling process adapted to his problem.

REFERENCES

[1] ROOFSOL, European project, 1995-1998, final report to be published, 1999.

[2] LEFEBVRE G., "Modal-based simulation of the thermal behavior of a building : the m2m software", Energy & Buildings 25, 1997, p. 19-30.

[3] http://www.enpc.fr/gise/m2m/m2m.html

[4] LEFEBVRE G., PALOMO E., IZQUIERDO M.M., "Reproducing thermal coupling between components in a generic environment like Matlab", Int. IBPSA Conf., PRAGUES, 1997.

[5] LEFEBVRE G, "Substructured modelling of linear thermal systems: the model synthesis", GISE/ENPC, to be published.

[6] E. PALOMO, "Roof components models simplification via statistical linearisation and model reduction techniques", Energy & Buildings, 1998.

[7] INRIA-ENPC, Scilab, ftp://ftp.inria.fr

[8] E. PALOMO, G. LEFEBVRE, "Stochastic simulations versus deterministic ones: advantges and drawbacks", ", Building Simulation'95, *Fourth Int. IBPSA Conf.*, MADISON, August 14-16, 1995.

[9] E. ERELL, Y. ETZION, "Radiative cooling with flat-plate solar collectors", CDAUP, to be published.

[10] E. PALOMO, "Analysis of the green roofs cooling potential in buildings", Energy & Buildings, 1998.

[11] G. LEFEBVRE, E. PALOMO, "m2m&Roofsol: user and reference manual", work Roofsol document, GISE/ENPC, oct. 1998, 33 p.