

# **An NMF Based Model Library for Building Climate and Energy Simulation**

Mika Vuolle

HVAC-laboratory, Helsinki University of Technology

Axel Bring

Div. of Building Services Engineering, KTH, Stockholm

## **ABSTRACT**

During the academic year 95-96 an NMF based model library has been developed by the IDA group. The library contains detailed and simplified zone models, full finite difference and reduced RC-network wall models, different types of solar radiation calculation models, state-of-the-art window models, local climate control models and some supporting Fortran subroutines.

Among the most interesting models are the detailed convective - radiative zone with full non-linear radiative and convective heat transfer using a general view factor calculation algorithm and the RC-wall model. These models are described in some detail in the paper. The full library has been comparatively validated against the BRIS program. In its most basic version, the library is more or less a BRIS re-implementation in IDA. The execution times are compared for applications developed in a modular simulation environment vs stand-alone programs. The relative development times are estimated as well.

## **INTRODUCTION**

Simulation is an important tool for building research and design. A building, together with its installations, is a complicated system, and accurate predictions of its behavior require dynamic simulation.

Simplified tools have abounded for a long time and have been made more attractive with attractive user interfaces. More detailed tools, capable of accurate predictions, have also been available for several decades, but their use has been hampered by several factors: accurate predictions need large amounts of input data, thus, effective user interfaces require much more development effort; developed codes have been monolithic and difficult to adapt to changing and growing user needs.

Modular simulation environments (MSE:s) developed during the last decade have the potential to change this picture. Interfaces using familiar drag-and-drop techniques are available. The maintenance problem has

been alleviated by the modular approach combined with effective handling of model development, using e.g. the Neutral Model Format (NMF).

## **THE NEUTRAL MODEL FORMAT**

This format for description of building simulation models was first presented at the BS'89 in Vancouver. It is a program independent language for modelling of dynamical systems using differential-algebraic equations. NMF serves both as clear model documentation for human readers and as direct input for automatic machine translation [Sahlin 1996a].

Some key aspects of NMF are discussed when model examples are presented below.

ASHRAE has supported NMF and paid for development of translators from NMF to TRNSYS and HVACSIM+ [Grozman 1996]. Translators have also been developed for SPARK [Nataf 1995], IDA [Shapovalov 1997, Kolsaker 1994], ESACAP [Pelletret 1994], and MS1 [Lorenz 1994].

## **THE IDA SIMULATION ENVIRONMENT**

A modular simulation environment, IDA, has been developed by the Division of Building Services Engineering, KTH, and the Swedish Institute of Applied Mathematics (ITM) [Sahlin 1996b]. IDA is now being made available to the Swedish building industry via a consortium of some thirty companies, building designers, contractors, property owners, and equipment manufacturers. The consortium develops building design applications based on the IDA software during the period 1995-98. In support of this effort, an NMF-based model library has been developed at KTH during the academic year 1995-96.

## **SCOPE OF MODEL LIBRARY**

One aim of the consortium effort is to provide easy-to-use applications for early design estimates. Another aim is that applications should be based on detailed physical

models, so that complicated problems, involving interaction between building and systems, can be studied accurately. These diverging interests can be served by detailed model libraries, which can be used by interfaces with different focus and degree of detail.

The library presented in this paper was developed as a first major step towards a complete and detailed model library for studies of room climate and energy consumption. It contains all models required for detailed climate studies.

The library is used in a tool for early estimates of local cooling requirements; a tool with a very simple user interface, where some twenty parameters are given (see Figure 1). Remaining input parameters are defined by suitable default values, describing typical simulation objects in various categories, office, school, hospital, etc.

### CONTENT OF MODEL LIBRARY

The library contains models for the following components

- zone, simplified and detailed
- walls, RC-models and full finite difference models
- state-of-the-art window models
- solar radiation on windows and facades, several radiation models
- local climate control.

Descriptions of the RC-wall model and of the detailed convective - radiative zone model with full non-linear radiative and convective heat transfer are given below.

### RC-WALL MODEL

The most common way to model a wall is to discretize it in one dimension into a number of equidistant layers (nodes) by using some finite difference method. The number of nodes is selected as a compromise between execution time and required accuracy of the results. The same accuracy can be reached with fewer nodes with an RC-network, if the thermal resistances, the heat capacities and the construction of the RC-network are chosen properly [Akander 1996].

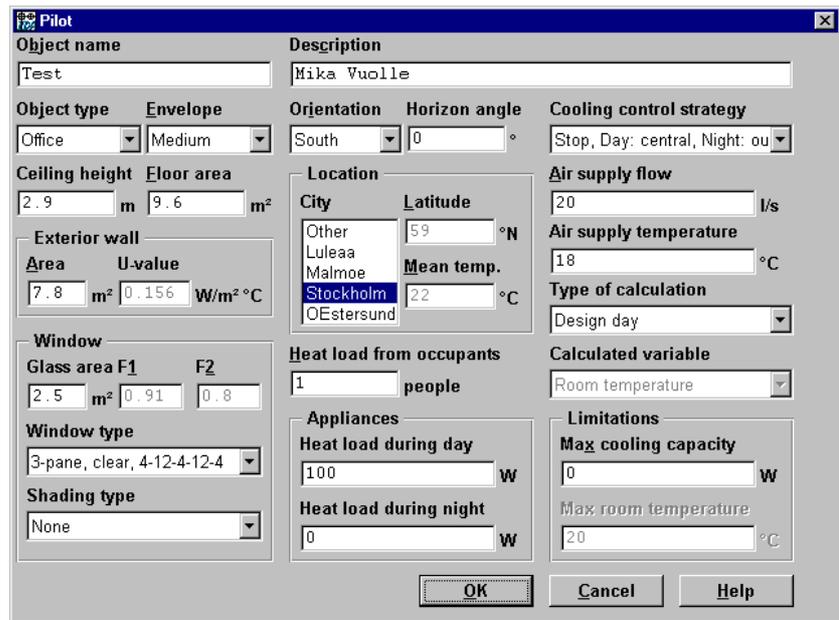


Figure 1. Interface of cooling application

In the project, a three-node RC model has been used, see Figure 2. A least-square optimization procedure determines the parameter values that minimize the deviations between the model and the analytical solution, observed for harmonic oscillations with periods of 1, 3, 6, 12, 24, 48 and 96 hours. The calculation of optimal RC parameters is based on a physical description of the wall, given in the PARAMETERS section of the NMF model. It is performed before a simulation starts, as described in the PARAMETER\_PROCESSING section, which is executed prior to the actual simulation.

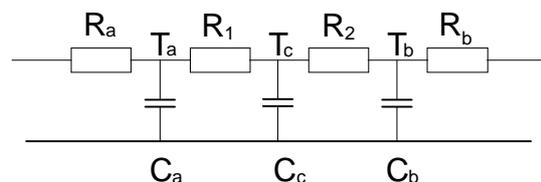


Figure 2. The RC-network used.

The NMF code for a three node RC-wall model is described in Figure 3. In the case of a light internal wall, where only two mass nodes are needed, the middle capacity will automatically be put to zero by the optimization procedure. The advantages of the RC-wall model are the decreased calculation time due to fewer nodes and the fact that the discretization accuracy is known. A disadvantage is the lack of physically meaningful temperatures inside the wall.

Most programming languages are procedural and describe a program flow where variables are assigned new values in a prescribed order. NMF, on the other hand, is basically a declarative language, i.e. relations between variables are defined by equations, as can be seen from the figure 3.

## CONTINUOUS\_MODEL Wall

### ABSTRACT

" Abstract omitted here"

### EQUATIONS

/\* Capacities - dynamic equations \*/

$$Ca * a * Ta' = (Qa + a/R1 * (Tc - Ta)) ;$$

$$Cb * a * Tb' = (Qb + a/R2 * (Tc - Tb)) ;$$

$$0 = \text{IF } Cc == 0 \text{ THEN } 0$$

$$\text{ELSE } -Cc * a * Tc'$$

END\_IF

$$+ a/R1 * (Ta - Tc) + a/R2 * (Tb - Tc);$$

/\* Resistances - algebraic equations \*/

$$Qa = a * (Tpa - Ta) / Ra ;$$

$$Qb = a * (Tpb - Tb) / Rb ;$$

### LINKS

/\* type name variables \*/

TQ Term\_a Tpa, POS\_IN Qa ;

TQ Term\_b Tpb, POS\_IN Qb ;

### VARIABLES

/\* type name role def min max description \*/

type	name	role	def	min	max	description
Temp	Tpa	OUT	0	ABS_ZERO	BIG	"Temp at term_a"
Temp	Tpb	OUT	0	ABS_ZERO	BIG	"Temp at temp_b"
Temp	Ta	OUT	0	ABS_ZERO	BIG	"Temp at capacity a"
Temp	Tb	OUT	0	ABS_ZERO	BIG	"Temp at capacity b"
Temp	Tc	OUT	0	ABS_ZERO	BIG	"Temp at capacity c"
HeatFlux	Qa	IN	0	-BIG	BIG	"Influx term_a"
HeatFlux	Qb	IN	0	-BIG	BIG	"Influx term_b"

### MODEL\_PARAMETERS

/\* type name role def min max description \*/

type	name	role	def	min	max	description
INT	n	SMP	1	1	7	"Number of wall layers"

### PARAMETERS

/\* type name role def min max description \*/

type	name	role	def	min	max	description
Area	a	S_P 10.	SMALL	BIG	BIG	"wall area"
Length	l[n]	S_P .1	SMALL	BIG	BIG	"layer length"
HeatCondL	lambda[n]	S_P 1.5	SMALL	BIG	BIG	"layer conductivity"
Density	rho[n]	S_P 2300.	SMALL	BIG	BIG	"layer density"
HeatCapM	specc[n]	S_P 880.	SMALL	BIG	BIG	"layer spec heat"

/\* derived parameters \*/

HeatCapA	Ca	C_P	1.	SMALL	BIG	"opt. active heat cap side a"
HeatCapA	Cb	C_P	1.	SMALL	BIG	"opt. heat cap side b"
HeatCapA	Cc	C_P	1.	SMALL	BIG	"opt. heat cap center"
HeatResA	R1	C_P	1.	SMALL	BIG	"resistance between nodes a,c"
HeatResA	R2	C_P	1.	SMALL	BIG	"resistance between nodes c,b"
HeatResA	Ra	C_P	1.	SMALL	BIG	"optimized resistance side a"
HeatResA	Rb	C_P	1.	SMALL	BIG	"optimized resistance side b"

### PARAMETER\_PROCESSING

call rcopt(n,[1],lambda[1],rho[1],specc[1],Ca,Cb,Cc,Ra,Rb,R1,R2);

### END\_MODEL

Figure 3. NMF code for RC wall.

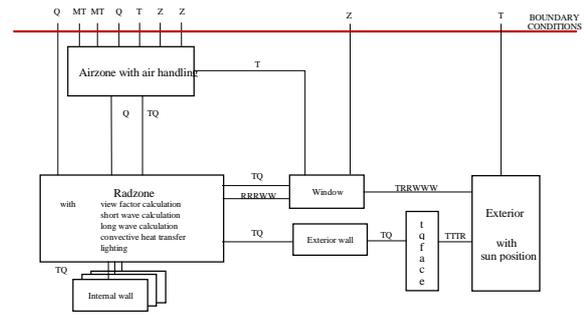


Figure 4. Component structure of zone system

## INTERFACES BETWEEN COMPONENTS

Components in a library are connected to build systems that can be simulated. For this to work, the models must have matching interfaces, described as LINKS in the NMF code. One very important issue when designing a library is thus the segmenting of the problem into components and the choice of link types. Figure 4 shows the component structure of a zone built from the current library, and indicates the link types used in the connections. The most common link type is named TQ, and contains one temperature and one heat flux. To enforce standardization, the types of links and variables are defined in a global list, common to all models in a library, and preferably also between libraries.

## CONVECTIVE - RADIATIVE ZONE

The zone model has been divided into two separate parts: a convective - radiative zone model and the so-called air zone model. The first treats short and long wave radiation and convection at the zone surfaces. The latter deals with ventilation, infiltration, convective cooling and ventilation control.

For a few simple geometries, view factors are available in analytic, tabular, or graphical form. For more complex geometries, numerical integration of a double integral has to be performed in one of several different ways. In the project, a projection method [Li 1992] is used to calculate accurate view factors for convex rooms, enclosed by flat surfaces of arbitrary shape. Non-convex rooms, where the views between some surfaces are partially or wholly blocked by other objects, are not yet encompassed. View factors are calculated in advance of the simulation, as described in the PARAMETER\_PROCESSING section of the pertinent NMF model.

## Long and short wave radiation

Calculation of radiative exchange within a diffuse-gray enclosure relies on two classical methods: the net radiation method or the absorption method. In the current library, the net radiation method is used to calculate internal radiation flows.

The internal radiation flows are absolute, i.e., expressed in W rather than W/m<sup>2</sup>. All view factors, material

properties and surface sizes have been incorporated in the NMF PARAMETER\_PROCESSING section into a matrix of total absorption factors. In the model equations, this matrix is multiplied by naked source terms (black body emission, short wave radiation from sun and lighting) to generate net energy flows to each surface.

## The Model Code

The central part of the radiative-convective zone model is shown below in figure 5. Variable declarations can be found in Appendix 1.

Zone surfaces can be walls, windows, or a lit portion of the ceiling and these surfaces require different treatment. For wall surfaces and windows there are matching links to surrounding components; these links are of different types. However, much of the convective and radiative exchange within the zone is independent of surface type. In order to achieve a common handling when feasible, surface temperatures appear in two guises: specialized TWall[nWall], TWind[nWind], TLite, and general T[nSurf]. The brackets specify vector sizes.

The first statement in the model describes how the general variables are fetched from the specialized ones, prior to calculation of heat transfer coefficients etc.. This statement also illustrates that NMF is not strictly equation based; local variables are calculated by assignments. These assignments could in principle be formulated as equations, but the current format is allowed for convenience and legibility. It also simplifies generation of effective code by translators.

### CONTINUOUS\_MODEL PZone

#### ABSTRACT

" documentation omitted here"

#### EQUATIONS

*/\* Collect surface temperatures, add incoming solar direct+diffuse, prepare film coefficients and emitted long wave radiation \*/*

```
FOR i = 1, nSurf
  T[i] := IF i <= nWall THEN TWall[i]
          ELSE_IF i <= nWall+nLite THEN TLite
          ELSE TWind[i - (nWall+nLite)] END_IF ;
  h[i] := IF LINEARIZE(1) THEN 1.
          ELSE U_film (TAir, T[i], slopeSurf[i]) END_IF ;
  M0[i] := BOLTZ * (T[i] - ABS_ZERO)**4 ;
END_FOR ;
FOR i = 1, nWall P2SurfSpec[i] := 0 END_FOR ;
```

*/\* Distribute light supply \*/*

```
PswLite := fLite2SW * PLite ;
PcvLite2Vent = - fLite2Vent * PLite ;
```

*/\* Handle upside of light. Prepare film coefficients and exchanged long wave radiation \*/*

```
TAirHi := TAir ;
TCeilOccl := T[mainSurfNo[nLite]] ;
hLite := U_film (TAirHi, TLite, 0.) ;
```

```
hOccl := U_film (TAirHi, TCeilOccl, 180.) ;
PlwLite2Occl := ALite * BOLTZ
* ( epsSurf[nWall+nLite] * (TLite - ABS_ZERO)**4
- epsSurf[mainSurfNo[nLite]] * (TCeilOccl - ABS_ZERO)**4 ) ;
```

*/\* Use film coefficients and long wave radiation to accumulate convection and radiation on surfaces \*/*

```
Plc2Occl := ALite * hOccl * (TAirHi - TCeilOccl) + PlwLite2Occl ;
PlcLiteUp := ALite * hLite * (TLite - TAirHi) + PlwLite2Occl ;
```

*/\* Prepare for average ceiling variables \*/*

```
P2SurfSpec[mainSurfNo[nLite]] := Plc2Occl ;
```

*/\* Heat balance for air \*/*

```
PcvLite2AirHi := PlcLiteUp - Plc2Occl ;
```

*/\* Heat balance for lights \*/*

```
fLiteAbs * PLite = PlcLiteDown + PlcLiteUp ;
```

*/\* Total contribution to zone \*/*

```
PLite2Zone = PLite - (PcvLite2Vent) ;
```

*/\*Direct solar is distributed isotropically \*/*

```
FOR j = 1, nWind PswWind2Zone[j] := PdrWind[j] + PdfWind[j] ;
END_FOR ;
```

*/\* Distribute emitted black body radiation to accumulate absorbed LW, and incoming SW from windows and lites to accumulate absorbed SW in surfaces \*/*

```
FOR i = 1, nSurf
  Plw2Surf[i] := SUM k = 1, nSurf psi_lw[k, i] * M0[k] END_SUM ;
  Psw2Surf[i] := psi_sw[i, nLite] * PswLite +
  SUM k = 1, nWind psi_sw[k, nLite + k] * PswWind2Zone[k]
  END_SUM ;
END_FOR ;
```

*/\* Get net heat flow from surfaces; add special heat flows for hidden surfaces \*/*

*/\* walls \*/*

```
FOR i = 1, nWall
  0 = PcdWall2Surf[i] + ASurf[i] * h[i] * (TAir - T[i]) + Plw2Surf[i]
  + Psw2Surf[i] + P2SurfSpec[i] END_FOR ;
```

*/\* light \*/*

```
0 = PlcLiteDown + ASurf[nWall+nLite]
* h[nWall+nLite] * (TAir - T[nWall+nLite])
+ Plw2Surf[nWall+nLite] + Psw2Surf[nWall+nLite] ;
```

*/\* windows \*/*

```
FOR i = 1, nWind
  0 = PlcWind2Zone[i] + ASurf[nWall+nLite+i] * h[nWall+nLite+i]
* (TAir - T[nWall+nLite+i]) + Plw2Surf[nWall+nLite+i] ;
END_FOR ;
```

*/\* Let re-circulated short wave leave into windows, currently, all as diffuse \*/*

```
FOR j = 1, nWind
  PswWind2Amb[j] = - Psw2Surf[nWall+nLite+j] ;
END_FOR ;
```

*/\* Accumulate convected heat to air \*/*

```
Pcv = SUM k = 1, nSurf ASurf[k] * h[k] * (T[k] - TAir) END_SUM
+ PcvLite2AirHi ;
```

*/\* Add up heat transfer thru walls \*/*

```

PcdWals2Surf = SUM k = 1, nWall PcdWall2Surf[k] END_SUM ;
/* Add up window heat transfers */
PswWdws = SUM i = 1, nWind PswWind2Zone[i] + PswWind2Amb[i]
END_SUM ;
PlcWdws2Zone = SUM i = 1, nWind PlcWind2Zone[i] END_SUM ;
LINKS
/* with air flow zone */
TQ      ZoneAir      TAir, POS_OUT Pcv ;
Q       Vented       POS_IN PcvLite2Vent ;
/* with windows */
FOR j = 1, nWind
TQ      WindTQ[j]    TWind[j], POS_IN PlcWind2Zone[j] ;
RRRWW  WindRad[j]   POS_IN PdfWind[j], POS_IN PdrWind[j],
POS_IN PswWind2Amb[j], Azimut[j], Elev[j]
END_FOR ;
/* with walls */
FOR I = 1, nWall
TQ      Wall[I]      TWall[I], POS_IN PcdWall2Surf[I]
END_FOR ;
/* for data input */
Q       LightPower   POS_IN PLite ;
NB! Model Variable and Parameter declarations are located in Appendix 2
PARAMETER_PROCESSING
nRad := nWind + nLite ;
CALL view_fac ( nSub,mainSurfNo[1],nSurf,nc,nv,nvert,ncsurf,nMain,
xyzVertice[1,1],verticesSurf[1,1], fi[1,1],ASurf[1]);
CALL lw_fac (nSurf,epsSurf[1], fi[1,1],ASurf[1],psi_lw[1,1]);
CALL sw_fac (nRad,nSurf,rhoSurf[1],fi[1,1], ASurf[1],psi_sw[1,1]);
ALite := ASurf[nWall+nLite] ;
fLiteAbs := 1 - fLite2SW - fLite2Vent ;
END_MODEL

```

Figure 5. NMF code for convective-radiative zone.

## VALIDATION

The models in the current library have been verified by comparisons with the BRIS program. BRIS is a heat balance program for room climate studies; it was developed in Sweden in the early sixties. The program is based on detailed physical relations, which are solved by an iterative finite-difference method without any linearizations. BRIS has been validated against measurements in a number of studies. After several extensions, it is still widely used and well trusted by the Swedish building industry. [Brown 1990]

Comparisons against BRIS have been made, both for individual models and for a test case containing a full set of models. The test case was a normal size office room with different kinds of heat loads, envelope constructions, and windows with and without shading devices. The only significant variations observed were related to handling

of solar radiation. The new library allows selection between three methods, derived from: ASHRAE fundamentals, Threlkeld [Threlkeld 1962], and Kondratjev [Kondratjev 1977]. These methods give significant differences in incoming radiation. Using Threlkeld's model, which is used in BRIS, the comparison with BRIS was entirely satisfactory.

## MSE PERFORMANCE

### Run Time Efficiency

The set of test cases was also used to make comparisons of speed and robustness between the two programs. Differences in time-stepping algorithms and in I/O-handling make exact comparisons difficult. BRIS uses a fixed timestep, which for the comparison was set to 0.5hr. The IDA application uses variable timesteps, but the tolerances were chosen to produce approximately the same number of timesteps. The comparisons showed an advantage for BRIS of a factor 2 - 4. A factor of this magnitude is to be expected; the flexibility of a general, modular tool has a price in overhead, whereas a special-purpose tool can tailor solution methods and streamline program structure to fit the restricted problem category. Current execution times for a 24 hour simulation using the IDA application are 5 s on a Pentium PC.

It could be noted that a full non-linear heat balancing program is one of the worst cases for the MSE development when runtime efficiency is considered. Most other building applications that will be developed in these environments have execution times that are insignificant compared to user handling times.

### Efficiency of Application Development

It is interesting to compare the development time required for an application based on a modular simulation environment with the time required to develop the same application as a stand-alone program. Such a comparison is possible in this case, because the main part of the new library is more or less a re-implementation of the existing BRIS program. The original BRIS program was implemented by Axel Bring, who was also the project leader for the NMF based effort.

Development times have been compared assuming that both projects were carried out on modern platforms. It has to be noticed that the NMF models have a wider scope. For the sake of simplicity, it can be assumed that the models are of comparable overall quality. Table 1 gives a breakdown of the estimated development times for the two cases.

The comparison shows that the development time using NMF and a modular simulation environment is about one third of the development time in the traditional environment.

Table 1. Times for main phases in the development

Phase	special purpose time in weeks	IDA time in weeks	comment
<i>Design program architecture</i>	4	2	NMF model architecture in the IDA case.
<i>Design and implement a suitable numerical method</i>	8	0	Method is already available in the IDA case
<i>Implement math models as formal code</i>	4	4	More information must be input in the NMF case, but the structure is given. In the special purpose case, the model code interacts with the solution procedure.
<i>Write I/O interfaces</i>	12	0	I/O already available in IDA case
<i>Testing and tuning of full implementation</i>	10	4	Less new code to be tested in IDA case. IDA tuning includes selection of suitable methods.
<i>Documentation of implemented models</i>	4	4	Written account of model equations.
<i>Development of user's manual</i>	4	1	IDA Solver file structure is already documented.
<b>Total time in man-weeks</b>	<b>46</b>	<b>15</b>	

## CONCLUSIONS

The work reported here shows that

- development times for MSE-generated applications are favorable compared to programs of special purpose
- runtime efficiency and robustness for MSE-generated applications are acceptable
- development of building energy and room climate applications, using NMF model libraries and a MSE, is a highly competitive alternative, when maintenance issues and compatibility between applications are taken into account.

The NMF library discussed is available in the directory <ftp://urd.ce.kth.se/pub/rp839/dzhh> and the ASHRAE translator as <ftp://urd.ce.kth.se/pub/rp839/nmfwin.exe>.

## REFERENCES

- Akander, J., "Effective Modelling of Energy Flow in Building Components", Licentiate Thesis, KTH, Stockholm, 1996
- Brown, G., "The BRIS Simulation Program for Thermal Design of Buildings and Their Services", Energy and Buildings, 14 385-400, 1990
- Grozman, P., P. Sahlin, "ASHRAE RP-839 NMF Translator - User's Guide", ASHRAE Inc. And Bris Data AB

Kolsaker, K., "NEUTRAN - A Translator of Models from NMF into IDA and SPARK", Proceedings to the BEPAC conference, BEP'94, York

Kondratjev, K. "Radiation Regime of Inclined Surfaces", WMO Technical Note 152, Geneva, 1977

Li, Y., "Simulation of Flow and Heat Transfer in Ventilated Rooms", Doctoral Dissertation KTH Stockholm Sweden, 1992

Lorenz, F., "Comments on the Neutral Model Format", presented to the TC 4.7 NMF Ad Hoc Subcommittee at the ASHRAE winter meeting 1994

Nataf, J.-M., "Translator from NMF to SPARK", Conference proc. Building Simulation'95, IBPSA, Madison, WI, USA, Aug. 1995

Pelletret, R., personal communication

Sahlin, P., "NMF Handbook, An Introduction to the Neutral Model Format, NMF Version 3.02", ASHRAE RP-839, report from Building Sciences, KTH, Stockholm Sweden, 1996a

Sahlin, P., "Modelling and Simulation Methods for Modular Continuous Systems in Buildings", Doctoral Dissertation KTH Stockholm Sweden, 1996b

Shapovalov, A., "IDA NMF translator, on-line help", 1997, <ftp://urd.ce.kth.se/pub/nmftransl>

Threldkeld, J., "Solar Irradiation on Surfaces on Clear Days", Ashrae Journal, 4 1962.

## APPENDIX 1

Declaration sections of PZone NMF-code.

### VARIABLES

```

/* type name role description*
/* Zone */
Temp Tair IN "zone temperature"
/* Surfaces */
HeatFlux Pcv OUT "convected flux to zone"
Radiation Plw2Surf[nSurf] LOC "l-w net to surface"
Radiation Psw2Surf[nSurf] LOC "s-w net to surface"
Temp T[nSurf] LOC "surface temp"
HeatCondA h[nSurf] LOC "film coeff"
RadiationA M0[nSurf] LOC "black body excitation"
/* Walls */
HeatFlux P2SurfSpec[nWall] LOC "out from hidden parts of wall
surfaces"
HeatFlux PcdWall2Surf[nWall] OUT "flow from wall"
HeatFlux PcdWals2Surf OUT "total flow from walls"
Temp TWall[nWall] IN "wall surface temp"
/* Windows */
Angle Azimut[nWind] IN "azimuth "
Angle Elev[nWind] IN "elevation"
Radiation PdfWind[nWind] IN "diffuse rad entering"
Radiation PdrWind[nWind] IN "direct rad entering"
Radiation PswWind2Zone[nWind] LOC "s-w entering thru window"
Radiation PswWind2Amb[nWind] OUT "s-w rad leaving, (all diffuse),
pos in "
HeatFlux PswWdws OUT "s-w net from windows"
HeatFlux PlcWind2Zone[nWind] OUT "heat flow from window"
HeatFlux PlcWdws2Zone OUT "total from from windows"
Temp TWind[nWind] IN "window surface temp"
/* Lite */
Heatflux PLite IN "supplied heat to light"
Heatflux PcvLite2Vent OUT "flux FROM light to ventilation,
always neg"
Radiation PswLite LOC "s-w from lit surface"
HeatFlux PcvLite2AirHi LOC "net convective flux from lite to
hi air"
Heatflux PlwLite2Occl LOC "lite to ceil radiation"
Heatflux Plc2Occl LOC "flux to occluded ceil"
Heatflux PLite2Zone OUT "total flow to zone"
HeatFlux PlcLiteDown OUT "down from lighted surf"
HeatFlux PlcLiteUp LOC "up from lighted surf"
Temp TAirHi LOC "ceiling air temp"
Temp TLite OUT "fixture surface temp"
Temp TCeilOccl LOC "temp on occluded ceil"
Factor hLite LOC "film coeff for fixture"
Factor hOccl LOC "film coeff occl'd ceil"

```

### MODEL\_PARAMETERS

```

/* type name role [def min max] description */
INT nLite SMP 1 1 BIGINT "number of lighted surfaces"
INT nMain SMP 1 1 BIGINT "number of main surfaces"
INT nSub SMP 1 1 BIGINT "number of subsurfaces"
INT nWall SMP 1 1 BIGINT "number of walls"
INT nWind SMP 0 0 BIGINT "number of windows"
INT nc SMP 3 3 BIGINT "number of coordinates/vert=3"
INT nv SMP 4 3 BIGINT "number of vertices/surface"
INT nvert SMP 8 8 BIGINT "number of vertices"
INT ncsurf SMP 12 9 BIGINT "number of coordinates * surface"
INT nRad CMP 1 1 BIGINT "number of SW-emitting surf"
INT nSurf SMP 2 2 BIGINT "number of all surfaces"

```

### PARAMETERS

```

/* Subsurfaces are ordered: light, windows */
/* type name role def min max description */
Factor fLite2SW S_P .5 0. 1. "fraction short wave"
Factor fLite2Vent S_P .5 0. 1. "fraction ventilated"
Factor epsSurf[nSurf] S_P 0.9 0. 1. "long wave
emittance"

```

```

Factor rhoSurf[nSurf] S_P 0.9 0. 1. "short wave
reflectance"
Angle slopeSurf[nSurf] S_P 90. 0. 180. "surface slope,
0=floor, 180=ceil"
Factor mainSurfNo[nSub] S_P 0 1 BIG "subsurface's main
surface number"
Length xyzVertice[nc,nVert] S_P 0 -BIG BIG "coordinates of
vertices"
Factor verticesSurf[nv,nSurf] S_P 1 1 BIG "list of vertices"
Factor fLiteAbs C_P .5 0. 1. "fraction absorbed"
Area ALite C_P "lit surface area"
Area ASurf[nSurf] C_P "surface area"
Factor fi[nSurf,nSurf] C_P "angle Factors"
Factor psi_lw[nSurf,nSurf] C_P "long wave absorption coeff"
Factor psi_sw[nSurf,nRad] C_P "short wave absorption coeff"

```