

## **ESPMrt, a new module for the ESP-system**

*Guido Sars, Cor Pernot, Martin de Wit*

University of Technology Eindhoven  
Institute of Applied Physics TNO-TH

April 1988

-88-PASSYS-MVD-WD-2

### *ABSTRACT*

A new module for the ESP-system is described: ESPMrt. It is based on the existing module ESPvwf. ESPMrt offers the possibility to calculate the mean radiant temperature ( $T_{mrt}$ ) as a function of position in space. It also incorporates the calculation of viewfactors for use in conjunction with ESPsim. With respect to ESPvwf the numerical solution process is improved without costs of extra calculation time. If the geometry allows so, the calculation of the viewfactors is done in an analytical way. This decreases the calculation time dramatically. Tests with the new model are described as well as an example run for the first time user.

November 3, 1999

## Table of Contents

1	Introduction .....	1
2	Description of the program ESPmrt .....	2
2.1	Viewfactors .....	2
2.2	Tmrt and VRT .....	4
3	Description of the subroutines that have been changed or added .....	5
4	Tests for the ESPmrt program .....	14
4.1	Viewfactors .....	14
4.1.1	Tests with improved numerical viewfactor calculation..	15
4.1.2	Tests with the analytical calculation of viewfactors..	16
4.2	TMRT .....	18
4.2.1	Tests with improved numerical viewfactor calculation..	19
4.2.2	Tests with the analytical calculation of viewfactors..	20
5	Remarks and suggestions for future developments .....	21
6	Example run of ESPmrt .....	25
7	APPENDIX .....	49
7.1	Viewfactors .....	49
7.1.1	Viewfactors for the change in patch division .....	49
7.1.2	Viewfactors for the tests with the analytical calculation .....	52
7.2	TMRT .....	55
7.2.1	Tmrt calculation; change in patch division .....	55
7.2.2	Tmrt; tests with the analytical viewfactor calculation .....	58
7.3	The input Geometry files .....	60
7.4	The input TMRT files .....	63

## **ESPMrt, a new module for the ESP-system**

*Guido Sars, Cor Pernot, Martin de Wit*

University of Technology Eindhoven  
Institute of Applied Physics TNO-TH

April 1988  
-88-PASSYS-MVD-WD-2

### **1. Introduction**

With the module ESPMrt, it is possible to calculate the mean radiant temperature ( $T_{mrt}$ ) in a room at a certain position. It is a scalar quantity and a function of position only. The definition of  $T_{mrt}$  is: 'The temperature of a uniform enclosure with which a small black sphere at the test point would have the same radiation exchange as it does with the real environment'.

The  $T_{mrt}$  is an important parameter when doing thermal comfort studies. In normal indoor conditions a person loses roughly equal quantities of heat by convection to the air and by radiation to the surrounding surfaces. The radiation heat losses becomes noticeable when sitting or standing near cold or warm surfaces. Thermofysiological models, used in ESP as well, do need information about  $T_{mrt}$ . The default procedure in ESPout is to calculate the  $T_{mrt}$  with an area weighted method. This may lead to great errors. The method presented here gives a good estimation of the  $T_{mrt}$ .

Another parameter that can be calculated with ESPMrt is the vector radiant temperature (VRT). This is a vector quantity and gives information about the radiation asymmetry in a room. In the literature, criteria can be found in relation to radiation asymmetry. In the calculation method used here the sphere at the test point is replaced by a small cube. Furthermore it is assumed that the surrounding surfaces are black and that the intervening medium does not scatter or absorb radiation.

The module ESPMrt is based on ESPvwf. In chapter 2 a brief description of the program and the theoretical background is given. In chapter 3 information is given about the subroutines that have been changed or added. In chapter 4 test runs are described. Comparisons of numerical and analytical calculated viewfactors and  $T_{mrt}$  values are made. Remarks and suggestions for future developments are made in chapter 5. For those who intend to work with the new ESPMrt module, a test run is described in chapter 6.

## 2. Description of the program ESPmrt

Use menu pick:

HELP           for explanation of menu  
FINISH        to QUIT.

The mrt program has two main possibilities:

- 1 Calculate viewfactors in a geometry. The same as ESPvwf but with slightly different results because of the changed routines.
- 2 Calculate the mean radiant temperature ( $T_{mrt}$ ) and the Vector Radiant Temperature (VRT) whether for a rectangular Body of specified dimensions or for a number of test points somewhere inside the geometry.

### 2.1. Viewfactors

The program ESPmrt has been made with as basis the program ESPvwf. This program calculates viewfactors for the surfaces of a geometry that may be used in conjunction with ESPsim. ESPmrt offers the same possibility.

Use menu pick:

VIEWFC       for input and calculation  
pARAM        for accuracy  
OUTPUT       to output geometry viewfactors to user defined file.

The procedure is briefly as follows.

A visibility matrix is determined. The elements of this matrix can have the values -1, 1, 2, 3.

If element  $i,j$  equals -1 than the surfaces  $i$  and  $j$  cannot "see" each other. For instance two windows that are in the same wall.

Element  $i,j$  equal 1 means  $i$  and  $j$  see each other completely (e.g. opposite walls).

Element  $i,j$  equal 2 means that there is another surface which partly shields  $i$  from  $j$ .

Element  $i,j$  equal 3 means  $i$  and  $j$  are perpendicular planes and  $i$  can see  $j$  only partly

After this is done, a rectangular check is carried out. It will not only check if surfaces are rectangular but also if the other surfaces that can be seen, are situated in such a way that they will lend themselves for analytical viewfactor calculation.

Then each surface is substituted by the smallest rectangle that can contain this surface. This rectangle is subdivided into a number of equal subrectangles. This number is represented by the variable GFACT. There are  $GFACT \times GFACT$  sub-rectangles. Then all rectangles are considered one by one. If the rectangle is on a surface that will "allow" analytical calculations then the viewfactors are calculated analytically. If this is not the case then the "bubble method" is followed. At the middle of each subrectangle a half sphere (bubble) is placed which is "looking" to the negative side of the surface. That is opposite to the normal of the plane. The normal is defined, according to routine plneqn.f :

THE PLANE EQUATION IS IN THE FORM;

$$A*X + B*Y + C*Z = D$$

WHERE THE VECTOR (A,B,C) IS THE UNIT NORMAL VECTOR TO THE PLANE. THE CONVENTION IS ADOPTED THAT THIS NORMAL WILL BE IN THE DIRECTION GIVEN BY THE RIGHT HAND ROTATION RULE REGARDLESS OF THE HANDEDNESS OF THE AXES DEFINING THE POLYGON VERTICES. THIS MEANS THAT THIS HANDEDNESS MUST BE INPUT TO THE ROUTINE. D IS THE SMALLEST DISTANCE (I.E. NORMAL DISTANCE) FROM THE ORIGIN TO THE PLANE. THE DIRECTION FOR POSITIVE MEASUREMENT OF THIS DISTANCE IS THE DIRECTION FROM THE ORIGIN GIVEN BY THE UNIT NORMAL VECTOR FOR THE PLANE.

THE ABOVE CONVENTION MEANS THAT FOR SOLID BODIES WHOSE FACES ARE DESCRIBED IN ANTI-CLOCKWISE FASHION (AS VIEWED FROM OUTSIDE THE BODY) THE UNIT NORMALS FOUND BY THIS ROUTINE WILL POINT OUTWARDS FROM THE BODY.

Each bubble is divided into a number of rings and each ring is divided into a number of "patches". The patches must have equal surface area to yield equal solid angles. The number of rings is determined by the variable NPATCH. There are exactly NPATCH/4 rings and about NPATCH\*NPATCH/4 patches on a bubble. Through the middle of each patch and the middle of the subrectangle, a line is generated. The intersection from this line with all surfaces is calculated if the visibility is not -1. The intersection which is closest to the bubble will be taken as the intersectionpoint and the surface it is on, will be "seen" by the bubble.

For further information about viewfactors one is advised to look in the ESPvwf manual.

## 2.2. Tmrt and VRT

Use menu pick:

```

INPUT      for input of geometry and temperatures.
PARAM      for accuracy.
gCHECK     to check input of BODY-dimensions and place,
           together with the VIEWER program.
tMRT       for calculation of viewfactors, Tmrt and VRT.
rEADFC     for reading viewfactors from file cube.fac
           that have been calculated in a previous run,
           also calculation of Tmrt and VRT.
oUTPUT     for output of Tmrt and VRT values to a user
           specified file.

```

First one must enter the temperatures of the surrounding surfaces and for the:

CUBE(S): the position(s)

BODY : position and dimensions.

The body or cube is treated as part of the GEN geometry and so bubbles are placed on the negative sides of it. The negative side of body or cube-face is pointing outwards from it. Viewfactors are calculated in the same way as with the VIEWFC-option but the surfaces of the surrounding geometry are not considered. So we only get the viewfactors  $F(i,j)$  for  $i$  is a cube-surface and  $j$  is a geometry-surface. Then the mrt-value for each cube-face is calculated. This is done by summing over  $j$ , the product of viewfactor  $F(i,j)$  and the absolute temperature of surface  $j$ , to the power 4.

```

viewfactors from cube to geometry :      F(I,J)
Temperature of geometry-surface j :      TEMP(j)

```

$$T_{mrt}(i) = \left( \sum_{j=1}^4 F(i,j) \cdot TEMP(j)^4 \right)^{1/4}$$

The VRT values are the differences of the plane radiant temperature values  $T(i)$ , of opposite cubefaces:

```

VRT(x)=T(4)-T(2)
VRT(y)=T(1)-T(3)
VRT(z)=T(6)-T(5)

```

Cubefaces 4,1,6 have the larger x,y or z coordinates.

### 3. Description of the subroutines that have been changed or added.

The changes made to the program ESPvwf to make ESPmrt. The Files that are used for ESPmrt :

filename	old	changed	added
COMMON.DIM		X	
PLYLIB.PAR	X		
MRTLIB.DIM			X
adcube.f			X
analytic.f			X
bound.f	X		
bub.f		X	
bubble.f		X	
checkrec.f			X
delcube.f		X	
encpol.f	X		
espmrt.f			X
eyemat.f	X		
filein.f		X	
getpol.f	X		
grid.f		X	
inside.f	X		
matpol.f	X		
mrtinput.f			X
mzrecc.f	X		
obstrt.f		X	
orttrn.f	X		
parallel.f			X
perpendicular.f			X
plne.f	X		
plneqn.f	X		
putpol.f	X		
readfac.f			X
sarea.f		X	
sparea.f	X		
surinv.f	X		
survis.f	X		
temperature.f			X
vtest.f	X		
writecube.f			X
writefac.f			X

A part of the files are exactly the same as those of ESPvwf. The routines that are changed or added are described below.

-----  
 \* COMMON.DIM (changed)

We have set some parameters different to allow for somewhat larger geometries. We have added MPATCH and NPATCH and gtype and a few comment lines. the parameter changes are:

```

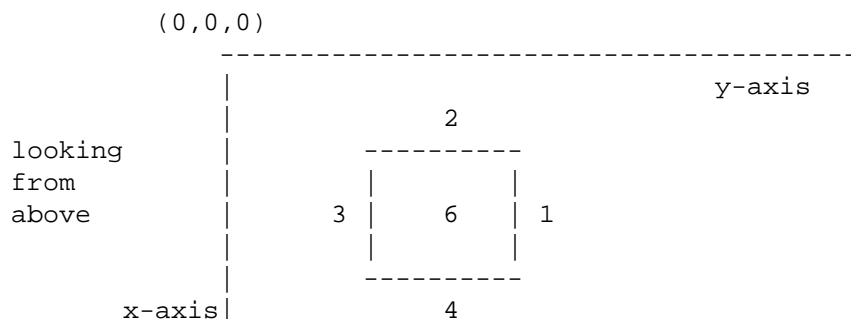
PARAMETER (MSUR=70, MGC=800,MPATCH=200)
PARAMETER (MTV=90, MS=70, MV=20, MG=5, MD=3)
COMMON /BUBXYZ2/ XBB(11000), YBB(11000), ZBB(11000)
  
```

-----  
 \* MRTLIB.DIM (added)

Commonblock variables for tmrt calculations

-----  
 \* adcube.f (added)

This routine adds a rectangular body (cube) of specified dimensions and a specified place to a GEN type geometry. The surfaces of the cube are defined with an opposite orientation to the other geometry-surfaces. The geometry surfaces have vertices in anti-clockwise order looking from the outside. The cube-faces have clock-wise order looking from the outside. The cube-faces are added to the end of the input geometry-file so for a geometry with 24 vertices and 17 surfaces the cube will have vertices 25 to 32 and surfaces 18 to 23. Cube surface 1 and 3 are parallel to the y=0 plane with surface 1 having the largest y-value. Cube surface 2 and 4 are parallel to the x=0 plane with surface 4 having the largest x-value. Cube surfaces 5 and 6 are in horizontal planes with 6 being the highest.



If the input geometry has any doors or windows which were not given as surfaces in the input-file, then they are discarded for the tmrt-calculations. They are not considered in viewfactor routines. A door or window should be input as a surface in the GEN- GEOMETRY to be incorporated in the calculations. After the new geometry with cube is made, it will be written to the data-file POLYGON.DIM which is only used as an extra calculation- memory. This is done in exactly the same way as in the routine filein.f

The geometry with cube will also be written to a file called cube.geo. This is done in the main program with the routine writecube.f



-----  
\* analytic.f                    (added)

This routine calculates the viewfactor from one surface to another analytically. In a way, the two surfaces are placed in a new coordinate system as pointed out in the figure below. In this new coordinate system, the values of the parameters  $a_1$ ,  $a_2$  .....  $d_1$ ,  $d_2$  can be calculated. After that it is possible to calculate the viewfactor analytically. In the text below is indicated in more detail how things are worked out in the subroutine.

In general terms:

First the vertices and the normal-vectors of the two surfaces are read in. Then both surfaces are rotated in standard position, that is: normal to the first surface along + z-axis and the edges along x or y-axis. If the surfaces are perpendicular then the normal to the second surface will be along + x-axis.

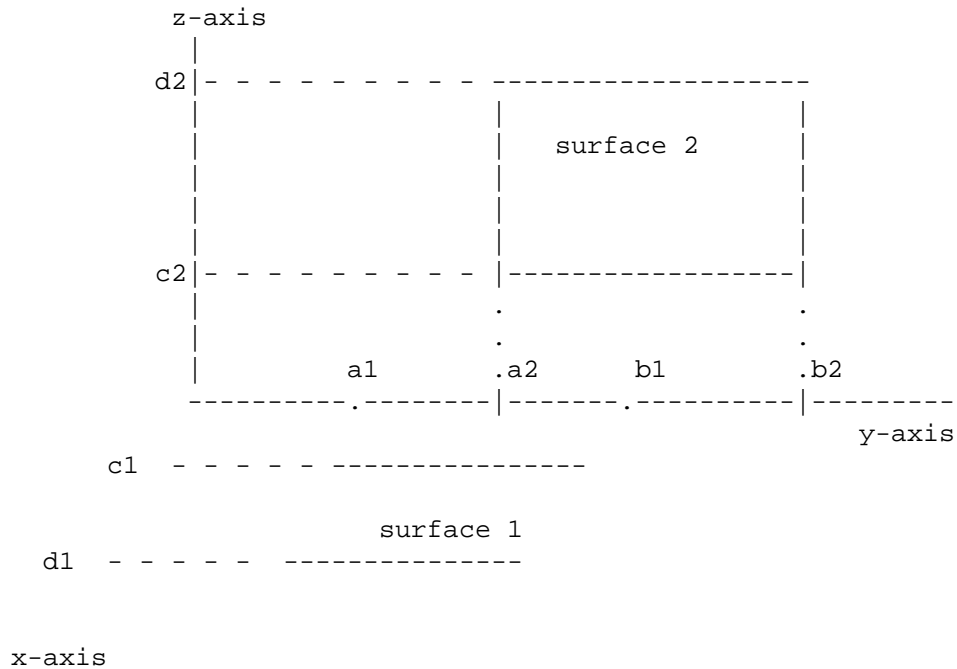
In more detail, this is what the subroutine does:

First it rotates round z-axis making the y-component of the normal to surface 1 zero. Then it rotates round y-axis aligning the normal with the z-axis. Then it rotates in x,y plane until the edges are in right direction. If the surfaces are parallel, it aligns the first edge of 1 with the + x axis. If the surfaces are perpendicular, it aligns the normal to 2 with the + x axis. Now surface 1 is translated to  $z=0$  plane.

If the surfaces are perpendicular translate surface 2 to  $x=0$  plane. No negative values are allowed for  $paralf$  and  $perpf$ . So translate both surfaces so that the smallest y-value is 0.0.

Then it determines the input values for analytical functions. In the main program  $ff(i,j)$  is divided by  $Pi*gfact*gfact$ . This is compensated in this routine for analytical values.

Below the standard position of two perpendicular surfaces is sketched. For parallel surfaces the  $a,b,c$  and  $d$ -values are similar for surface 1 and two. Here an extra variable  $g$  is needed for the difference in z-coordinate between 1 and 2. It is important that all values are positive.




---

\* bub.f (changed)

This routine determines the points on a bubble through which one "shoots". Originally there was another routine in ESPvwf with this name. This routine divided one bubble into 10 rings of equal height, and each ring in 40 patches. This resulted in patches of equal area but of quite different form.

The division of the surface of the bubble into patches has been changed. The new routine has a variable number of rings. This is determined by "NPATCH". There are NPATCH/4 rings. We choose the rings to have equal angle rather than equal height. To ensure that the solid angle per patch is the same, the number of patches on a ring will vary.

Two consequences are of importance to this change:

- 1 The patches are more or less of the same form.
- 2 The distribution of the patches over the surface of the bubble is more uniform. In the old situation, 'rays' with small angles with the normal to the surface were a minority. In the new situation however, there are 'rays' with smaller angles with the normal to the surface, decreasing the 'blind angle' in the forward direction.

Although this procedure introduces small rounding-errors, it gives better results than the old routine "bub.f", as will be shown in chapter 4.1. The calculation of the patches will be clear from the comments in the listing of the routine.

-----  
 \* bubble.f (changed)

For tmrt calculations it is sufficient to take only cube-gridpoints. Therefore the variable strtgrid is given the number of the first gridpoint. This way the other gridpoints will be skipped in the calculations. If viewfactors are calculated only, (no tmrt) then the value of strtgrid is 1.

In this routine is checked if the viewfactors can be calculated analytically with the information from routine checkrec.f and it will be calculated analytically if possible. If not possible the viewfactors will be calculated with the bubble method as in ESPvwf.

----- \*  
 checkrec.f (added)

This routine checks if the geometry-viewfactors can be calculated analytically. This information is stored in the matrix ARECT. The elements arect(i,j) contains information about the geometrical relation between the surfaces i and j. The diagonal-elements arect(i,i) contain the information about the total row of elements. The possible values of those elements are listed below.

arect(i,j) with i not equal j:

- will be 2 if the surfaces i and j are perpendicular and totally visible to each other, with perpendicular or parallel edges.
- will be 1 if the surfaces i and j are parallel and totally visible to each other, with perpendicular or parallel edges.
- will be 0 if surfaces are not visible
- will be -1 if parallel and partly visible (shielded)
- will be -2 if perpendicular and partly visible
- will be -3 if neither perpendicular nor parallel
- will be -9 if surface j is visible but not rectangular
- will be -10 if surface i not rectangular

arect(i,i)

- will be -20 if surface cannot see any other surface
- will be -19 if somewhere on the row is a -9
- will be -13 if somewhere on row is -3 (and no -9)
- will be -11 if somewhere on row is -1 (no -9 or -3)
- will be -12 if somewhere on row is -2 (no -9,-3,-1)  
 (can also be computed analytically)
- will be 0 if viewfactors ff(i,j) can be calculated analytically.

First it is checked if surfaces i and j are rectangular. This is done by calculating the direction-vectors for the edges of one surface. The inner product is taken of two subsequent vectors. We trace around the surface from vertex to vertex. If the surface is rectangular the inner product should be 4 times 0 (indicating right angles) and all other

times equal to 1. If this is not the case the surface is not rectangular.

Then the inner product of the normal-vectors is taken. There are 4 different situations.

inner product = 1

Parallel and same orientation of surfaces.

inner product = -1

Parallel and opposite orientation of surfaces.

inner product = 0

Perpendicular surfaces.

inner product = another value

Angle between surfaces not 0 and not 90 degrees

Now it is checked if in the first three cases, the edge of the first surface is parallel or perpendicular to the edge of the second surface. If the first edge is parallel or perpendicular and the surface is rectangular then all edges are perpendicular or parallel.

After the elements `arect(i,j)` are given a value, the elements `arect(i,i)` are calculated.

In this routine the inner product values are checked with the logical function `equal(x1,x2)`. This function is true if `x1` and `x2` are equal within 0.00001 and it is false otherwise. This function is used to allow for small machine inaccuracies.

-----  
\* `delcube.f` (added)

This routine deletes the cube that has been added to the geometry with `adcube.f`. It does this by subtracting 6 from the number of surfaces (`nsur`) and subtracting 8 from the number of vertices (`nver`). The other cube-data are not deleted but will be overwritten in the next call of `adcube.f`. The file `cube.geo` will contain the geometry with the cube that was last added to the file.

-----  
\* `espmrt.f` (added)

This is the MAIN program. It has been derived from `<espvwf.f>`. A description is already given in chapter 1.

-----  
\* `filein.f` (changed)

This routine is slightly changed. The manner in which the geometry-file is opened is done with the routine `rpopen` of ESP instead of the standard `OPEN` of fortran. We have also introduced an extra variable `gtype` which contains the type of geometry (GEN REG REC). This is used to check if the geometry for `tmrt`-calculations is GEN type in the main program. We have also changed the error-messages somewhat.

-----

\* grid.f (changed)

This routine is a slightly changed version of <grid.f>. We have included the mrt-common block. We introduced an initialisation loop because this routine is called more than once. And for tmrt-calculations we calculate the first gridpoint that is located on the cube.

-----  
\* mrtinput.f (added)

This routine will get the input needed for TMRT-calculations.

It asks for an input-filename. If the file exists, and matches the geometry-input file, it is read. Then a menu appears with 4 options.

INPUT : You can give the values of all variables and write them to the given TMRT file.  
For every variable there will be a brief explanation  
CHANGE : You can change some of the given temperature values and write to given TMRT file.  
With every change, all temperature values are displayed.  
-END : Return to main menu of main program. The input values will be written to the given file and can be read in the next run.  
HELP : Gives an explanation of the first three options.

-----  
\* obstrt.f (changed)

This routine contained a flaw. The routine gets as input two points. Through these points a line or "ray" is taken. The line which goes through the two points of the bubble is checked for intersection with all surfaces that do not have a visibility -1. Next the intersection point which is closest to the bubble is taken as the final intersectionpoint, and the surface on which this point lies is seen by the bubble. However this intersection-point can lie behind the bubble. One would expect that this cannot happen because the visibility would be -1, but a surface which has visibility 3, can lie partly before and partly behind the bubble. Therefore only intersectionpoints that lie before the bubble should be taken into account. This is realised in the routine by limiting the value of variable "T" to positive values only. This guarantees that only intersection-points in front of the bubble are considered.

```
* parallel.f      (added)
```

Calculates the viewfactor for two parallel surfaces, given the right values of a1,b1,c1,d1,a2,b2,c2,d2,g. These values are explained in the description of routine analytic.f

\* perpendicular.f (added)

Calculates the viewfactor for two perpendicular surfaces, given the right values of a1,b1,c1,d1,a2,b2,c2,d2. These values are explained in the description of routine analytic.f

```
* readfac.f (added)
```

This routine can read the viewfactors for CUBE or BODY calculations from the file cube.fac. These values will be written to this file by routine writefac.f. The file cube.fac is compared with the tmrt-inputfile.

```
*   sarea.f          (changed)
```

This function determines the surface-area of a Polygon in 3 dimensions. The polygon is divided into triangles which are half parallelogramms or diamonds (see picture on next page). The vectors from the first to the second point, and from the first to the third point of the polygon form the first triangle. The surface area of a diamond is equal to the length of the vector-product of the vectors that make up the diamond. By summing all vector-products, the diamonds outside are counted negative. The length of the resulting vector-product (divided by 2) gives the surface-area of the surface.

Normalised surfaces are rotated so as to have no z-component. The vector products of the corresponding diamonds have only a z-component. Therefore it is sufficient for normalised surfaces to take the z-component of the vectorproducts for the surface area.

```
* temperature.f      (added)
```

This routine computes Mean Radiant Temperature (TMRT) It should be given as input:

### Cube dimensions

viewfactors from cube to geometry  $F(I,J)$ 

Temperatures of all geometry surfaces TEMP(j)

$$\text{The model used is } T_{\text{mrt}}(i) = \left( \sum_{j=1}^4 F(i,j) * \text{TEMP}(j) \right)^{1/4}$$

Theoretically, the sum over a row in the viewfactormatrix should be 1.00. Sometimes the program gives values that do not follow this rule. Therefore the values of one row are scaled, in order to ensure that the

sum is equal to one. If in one row all viewfactors are zero an error message will be given. For every surface of the cube the radiant temperature is calculated and stored in array  $t(k)$ .

The mrt value is estimated as the mean radiant temperature of all cube surfaces, where the temperatures are weighed with their relative surface-area. The Vector Radiant Temperature is calculated. This is the "temperature-vector" that results when subtracting the plane radiant temperature of one cube-face from the plane radiant temperature of the face that has the opposite orientation.

-----  
\* writecube.f (added)

This routine writes the geometry that is composed by the routine `adcube.f`, to the file `cube.geo`. This file can be used as input for the VIEWER program.

It is also used to check the input. Then it will write the input-geometry, to which all tmrt-cubes are added at once, to file `check.geo` (main-program-option `gCHECK` ). This file can also be used for the VIEWER program.

-----  
\* writefac.f (added)

Writes the viewfactors for the cube to file `cube.fac`. The latter can be used for later calculations by reading it with the routine `readfac.f` (main-program-option `readFC` ).

-----

#### 4. Tests for the ESPmrt program

The tests of the ESPmrt program are split in two. In the first part (chapter 4.1) the viewfactor prediction is treated and in the second one (chapter 4.2) the Tmrt calculation.

Both parts are split in two again. In the first one is indicated what the effect is of only changing the patch division on the bubble. In the second part the analytical solution of the viewfactors is tested.

So this chapter is set up as follows:

- 4.1 Viewfactor calculation.
  - 4.1.1 Tests with improved numerical viewfactor calculation.
  - 4.1.2 Tests with the analytical calculation of viewfactors.
- 4.2 Tmrt calculation.
  - 4.2.1 Tests with improved numerical viewfactor calculation.
  - 4.2.2 Tests with the analytical calculation of viewfactors.

##### 4.1. Viewfactors

Comparison of ESPmrt, ESPvwf and analytical results

The input geometry is taken from the report for the PASSYS Validation Subgroup of M. Bean and A. Pinney of 27-11-1987 "VALIDATION OF VIEW FACTOR CALCULATIONS IN ESP"

The geometry is given in their "test3" and is listed below.

GEN

16,	12,	0.000
0.0000,	0.0000,	0.0000,
2.7680,	0.0000,	0.0000,
2.7680,	2.5000,	0.0000,
2.7680,	5.0000,	0.0000,
0.0000,	5.0000,	0.0000,
0.0000,	2.5000,	0.0000,
0.0000,	0.0000,	2.7500,
2.7680,	0.0000,	2.7500,
2.7680,	2.5000,	2.7500,
2.7680,	5.0000,	2.7500,
0.0000,	5.0000,	2.7500,
0.0000,	2.5000,	2.7500,
1.9180,	5.0000,	0.7550,
0.8480,	5.0000,	0.7550,
0.8480,	5.0000,	1.8050,
1.9180,	5.0000,	1.8050,
4, 4, 5, 11, 10,		
4, 5, 6, 12, 11,		
4, 1, 7, 12, 6,		
4, 1, 2, 8, 7,		
4, 2, 3, 9, 8,		
4, 3, 4, 10, 9,		
4, 6, 5, 4, 3,		
4, 1, 6, 3, 2,		
4, 9, 10, 11, 12,		
4, 7, 8, 9, 12,		



```
4, 13, 14, 15, 16,  
4, 16, 15, 14, 13,  
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 12, 0, 0
```

Note : surface 12 is the same as 11 but with the vertex-order reversed. This was done to avoid error-messages in the program "VIEWER". This surface cannot be seen from any other surface so all viewfactors are zero.

#### 4.1.1. Tests with improved numerical viewfactor calculation.

The values of the viewfactors for all surfaces are given in appendix 7.1.1. We make a distinction between three viewfactor values.

The vwf values are generated with:

F77 version of ESPvwf 1.1a December 1986

The mrt values are generated with the program we are documenting here.

The ana values are calculated analytical values and are taken from the above mentioned PASSYS report.

For vwf and mrt the surface grid-division, represented by 'GFACT', was 2 (which means 4 bubbles on each surface). The important difference between the programs is the bubble-division into patches. This is done in the routine 'bub.f'. In the vwf version the bubble is divided in rings of equal surface area, and each ring is divided in the same number of patches. In the mrt version the rings are not of equal area but of equal angle. This means that each ring has a different number of patches. HOWEVER we have ensured that each patch represents an equal solid angle on the bubble. The advantage of this method is that the patches have more or less the same form whereas in vwf the patches can differ much in form depending on their place on the bubble. Patches of the same form ensure a more regular distribution over the bubble. See also the remarks on this subject in chapter 3.

To compare the results of the viewfactors we use the 'residue'. This is the sum of the squared differences over one row.

ffv(i,j): viewfactor from surface i to j, calculated with ESPvwf  
ffm(i,j): viewfactor from surface i to j, calculated with ESPmrt  
ffa(i,j): viewfactor from surface i to j, calculated analytically

```

residue of vwf = {sum over j}(ffa(i,j)-ffv(i,j))**2
residue of mrt = {sum over j}(ffa(i,j)-ffm(i,j))**2

```

The total residue is the sum over i of all residue's. For the test used here yields:

1 Residue of vwf:	0.00012	Residue of mrt:	0.00027
2 Residue of vwf:	0.00078	Residue of mrt:	0.00040
3 Residue of vwf:	0.00079	Residue of mrt:	0.00018
4 Residue of vwf:	0.00033	Residue of mrt:	0.00018
5 Residue of vwf:	0.00079	Residue of mrt:	0.00024
6 Residue of vwf:	0.00078	Residue of mrt:	0.00033
7 Residue of vwf:	0.00031	Residue of mrt:	0.00020
8 Residue of vwf:	0.00027	Residue of mrt:	0.00016
9 Residue of vwf:	0.00028	Residue of mrt:	0.00014
10 Residue of vwf:	0.00027	Residue of mrt:	0.00014
11 Residue of vwf:	0.00029	Residue of mrt:	0.00012

```

Total residue of vwf: 0.00502
Total residue of mrt: 0.00234

```

It turns out that the total residue of mrt is twice as small as that of vwf. In appendix 6.1 this can be seen in more detail.

#### 4.1.2. Tests with the analytical calculation of viewfactors

The values of the viewfactors for all surfaces are given in appendix 7.1.2.

The ANA values are generated with the program ESPmrt that we have changed to include analytical calculations of viewfactors.

The mrt values are generated numerically with ESPmrt by using the "Bubble-method".

The ana values are calculated analytical values and are taken from the above mentioned PASSYS report.

To compare the viewfactors we use again the residue as defined in paragraph 4.1.1. This is the sum of the squared differences over one row.

```

ffa(i,j): viewfactor from surface i to j, calculated analytically
           with ESPmrt.
ffm(i,j): viewfactor from surface i to j, calculated with ESPmrt.
ffa(i,j): viewfactor from surface i to j, calculated analytically
           in the PASSYS report.

```

```

residue of ANA = {sum over j}(ffa(i,j)-ffA(i,j))**2
residue of mrt = {sum over j}(ffa(i,j)-ffm(i,j))**2

```

The total residue is the sum over i of all residue's.

1 Residue of ANA:	0.00003	Residue of mrt:	0.00027
2 Residue of ANA:	0.00088	Residue of mrt:	0.00040
3 Residue of ANA:	0.00005	Residue of mrt:	0.00018
4 Residue of ANA:	0.00016	Residue of mrt:	0.00018
5 Residue of ANA:	0.00005	Residue of mrt:	0.00024
6 Residue of ANA:	0.00088	Residue of mrt:	0.00033
7 Residue of ANA:	0.00104	Residue of mrt:	0.00020
8 Residue of ANA:	0.00005	Residue of mrt:	0.00016
9 Residue of ANA:	0.00076	Residue of mrt:	0.00014
10 Residue of ANA:	0.00006	Residue of mrt:	0.00014
11 Residue of ANA:	0.00000	Residue of mrt:	0.00012

Total residue of ANA: 0.00398

Total residue of mrt: 0.00234

We see that the differences between "ANA" and "ana" are rather large. The residues of the numerical results are usually smaller. This is the result of the fact that surface 11 overlaps surface 1. The analytical procedure of ANA does not recognise this and therefore the viewfactors for surface 1 to all others, and the viewfactors from all others to surface 1 will not be correct. We can prove that this is indeed so by subtracting the viewfactors from all others to 11 from the viewfactors from all others to 1 :

$$FFANA(i,1) := FF(i,1) - FF(i,11) \quad i=2,10$$

The first row of viewfactors can not be corrected this way and will continue to be incorrect. The other viewfactors will be correct upto rounding errors. Below we see that this is indeed the case.

1 Residue of ANA:	0.00003	Residue of mrt:	0.00027
2 Residue of ANA:	0.00000	Residue of mrt:	0.00040
3 Residue of ANA:	0.00000	Residue of mrt:	0.00018
4 Residue of ANA:	0.00000	Residue of mrt:	0.00018
5 Residue of ANA:	0.00000	Residue of mrt:	0.00024
6 Residue of ANA:	0.00000	Residue of mrt:	0.00033
7 Residue of ANA:	0.00000	Residue of mrt:	0.00020
8 Residue of ANA:	0.00000	Residue of mrt:	0.00016
9 Residue of ANA:	0.00000	Residue of mrt:	0.00014
10 Residue of ANA:	0.00000	Residue of mrt:	0.00014
11 Residue of ANA:	0.00000	Residue of mrt:	0.00012

Total residue of ANA: 0.00003

Total residue of mrt: 0.00234

The value of the residue for ANA is entirely due to the faults in the viewfactors for surface 1. We conclude that if we would take away surface 11 so that there are no more overlapping surfaces, the analytical results are correct.

A last remark concerns the computing time. With the analytical solution the calculation is performed much faster then in the numerical case.

#### 4.2. Tmrt calculation.

Comparison of the ESPmrt temperature-values with analytical results.

The program that we have used to calculate the analytical solutions of the mean radiant temperature is KLI/TMRT. A program developed at the FAGO group in Eindhoven. KLI/TMRT is only applicable for simple, rectangular geometries. This program can put many small cubes at certain points in the geometry, or one larger cube, and estimate the Tmrt value for cube-face i according to the formula:

$$TMRT(i) = \left\{ \sum_{j=1}^4 F(i,j) T(j) \right\}$$

where  $F(i,j)$  is the viewfactor from cube-face i to surface j and  $T(j)$  is the absolute temperature of surface j.

The results of KLI/TMRT are indicated with "ana".

The geometry in which we calculated the tmrt-values is rectangular. We have done the test for a geometry with several small cubes (CUBE) and for a geometry with one larger body (BODY)

There are more ways in which one can describe the geometry with ESPimp.

- \* For viewfactor calculations it is important that all surfaces that contribute to radiation are given explicitly. Therefore we choose the GEN-type of geometry.
- \* One can first describe the main surfaces of the geometry (like walls etc) and than place smaller surfaces (like windows, doors) in those surfaces. This is done in mrt-geometry 1. This means that the windows overlap the parts of the wall that they are on.
- \* One can also make sure that there is no overlap of surfaces and that all surfaces are rectangular. This means that some extra surfaces will have to be defined. This is done in the mrt-geometry 2.

Both geometries are listed in appendix 7.3.

CUBE: the tmrt-values are calculated at nine points inside the geometry. On all 8 corners, staying 0.5 meter off all walls, and in the middle of the room.

BODY: The tmrt for a rectangular body is calculated in the middle of the room.

The temperatures of the surfaces are given in appendix 7.4 in the form of an input-file for the mrt-program. We have given "windows" a temperature of 1 C, "doors" are 6 C, walls and floor are 18 C, the ceiling is 20 C and there is one surface with 30 C.

#### 4.2.1. Tests with respect to a change in patch division.

The test with several cubes  
TMRT-file : cubet1

Each surface of the cubes had one bubble (GFACT=1) and the bubbles are divided into 25 rings. (See also description of routine bub.f)

##### Conclusions:

The 1mrt-geometry gives some accurate results, and some results that are not so accurate (lines are denoted with a \* in appendix 7.2.1).  
The 2mrt-geometry always gives fairly accurate results.

The difference can be explained by the overlap that occurs in the geometry 1mrt. If we examine the viewfactors from the cubefaces to the windows we see that these are sometimes too small and that the wall in which the window is situated has a too large viewfactor. This can be explained as follows. There are beams "fired" from the bubble to the surrounding geometry. The point where the beam hits a surface is calculated. Then the point which is closest to the bubble will be taken as the final "hit-point" and the surfaces in which this point lies will be "seen" by the bubble. In the case of the overlapping surfaces, the surfaces will both be hit and the points will be, theoretically, at equal distance. Due to machine inaccuracies however one of both surfaces will be chosen and this will sometimes be the wrong one. One can circumvent this problem by placing the window a small distance off the wall but there is another reason why it is better to make sure that all geometry surfaces are rectangular without overlapping. This reason has already been pointed at by Pinney and Bean in their report of 27-11-1987, "VALIDATION OF VIEW FACTOR CALCULATIONS IN ESP". They remark that when a surface is not rectangular, some of the bubbles that are initially placed on them can be disregarded in the calculations, yielding viewfactors that are incorrect. This problem will not arise with tmrt-calculations because the bubbles will be placed only on cube faces, and not on geometry faces. However if one wants to know the viewfactors from geometry surfaces to geometry surfaces it can be important, so it is good practice to have rectangular surfaces.

We also conclude that the calculated Tmrt values are fairly accurate. However accuracy can be further increased by increasing the bubble division (rather than the grid division). This division is represented by variable npatch.

With the above mentioned geometries we have performed another test. This time we placed a body of 0.2 x 0.4 x 1.8 meters cube in the middle of both geometries. (at point (1.25,2.50,1.00))

Now we have for each body-face 4 bubbles, and for each bubble 12 rings. The increase in bubbles is done because the cubefaces no longer have negligible dimensions compared with the geometry dimensions. The decrease of bubble-rings was done to reduce calculation time.

The result is not quite as accurate as the one described above. Although the results are certainly acceptable. This can be ascribed

mainly to the fact that the cube no longer has negligible surface area and that the bubbles have less rings than with the cube calculations. Greater accuracy can be obtained by taking more bubbles per surface and more rings per body. This is shown in appendix 7.2.1. However this increased accuracy also increases the computation-time by roughly a factor 8.

It is surprising that both geometries give the same results for tmrt and vrt values. However the great discrepancies between the two geometries with the cube-calculations, only showed up when a cube was quite close to a window. Here the body is relatively far from the windows and they play a minor role

#### **4.2.2. Tests with the analytical calculation of viewfactors.**

We also compared the results of Tmrt calculations with the numerical and analytical results of ESPmrt and with other analytical results. This is done with exactly the same as in paragraph 4.2.1. The results are printed in paragraph 7.2.2.

The results of KLI/TMRT are indicated with "ana". The analytical results of ESPmrt are indicated with "ANA". The numerical results of ESPmrt are indicated with "2mrt".

In both cases that we used ESPmrt we used as input the 2mrt-geometry. This geometry had no overlapping surfaces.

No great differences in Tmrt values can be seen. The VRT values look better in the analytical case (ANA) but here also, no dramatic difference between ANA and 2mrt VRT-values occur.

A last remark concerns the computing time. With the analytical solution the calculation is performed much faster than in the numerical case.

## 5. Remarks and suggestions for future developments.

### 5.1. Remarks

- \* The change made in the routine obstprt.f should be made in the program ESPvwf too, if this program is used for calculating viewfactors. Else it could give wrong results.
- \* In the original program ESPvwf, the routine "sparea" was called (in the routine "filein") but with the wrong arguments. This routine is said to calculate the surface area of a polygon. We corrected the argument of the routine-call. However we didn't manage to get the right results from this routine at first. We then changed the routine "sarea.f" (which is called in sparea.f) as is described in chapter 3.
- \* In the original program ESPvwf after the viewfactors are calculated they are printed. Also a matrix is printed that is said to contain "exchange- factors". These exchange-factors are calculated from :  

$$EX(I,J) = FF(I,J) * GAREA(I)$$

The  $FF(I,J)$  are the viewfactors. The  $GAREA(I)$  are first called in "sparea" but this routine didn't function correctly (see remark above). Later the  $GAREA(I)$  are made equal to the surface area of one gridelement (in routine GRID). We believe that in neither case the  $EX(I,J)$  represent the correct exchange-factors so we omitted the calculation of them in our program ESPmrt (by placing comment "C"-s on these lines).
- \* Overlapping of surfaces should be avoided in order to avoid errors in the calculation of the viewfactors. This goes for ESPvwf and ESPmrt alike. A test could be included (as a preprocessor) to detect these situations.
- \* In a rectangular geometry it is possible to calculate the viewfactors completely analytical if all the surfaces are input as rectangles. See also the suggestion below.

### 5.2. Suggestions for future developments.

- \* In the example run in chapter 5 we see that for surfaces that are not rectangular (8 and 9) the viewfactors can be wrong. This expresses itself in the fact that the sum is not equal to 1.00. This is because all the surfaces are substituted by rectangles (see description of routine bubble.f) and on this rectangle the bubbles are placed. If a bubble does not fall within the original surface it is later discarded. One could compensate for this error by counting the bubbles, or, which is exactly the same, scale up the viewfactors by dividing by the sum over the row.  

The situation becomes worse when all bubbles fall outside the original surface. For an extreme example see surface 1 of geometry "cubical1" below. In this case a way should be found to move some bubbles to the original surface.
- \* The visibility of surfaces is not always calculated properly. See the geometry "cubical2" below. Sometimes the view from one surface to another is estimated as obstructed (ivis=2) while there is no obstruction. This is because all the surfaces are replaced by rectangles. If these rectangles have obstructed views to each others it is concluded that the view of the original surfaces is obstructed. In the example below this is clearly not the case.

A possible future activity would be to improve the calculation of the visibility.

- \* In viewfactor-calculations, on every surface (or on the rectangle it is substituted by) the same number of bubbles is placed. Likewise for small and very large surfaces. This could mean that for large surfaces the errors due to the numerical method are larger then for small surfaces where perhaps one bubble would suffice. It could be profitable to calculate the number of bubbles on a surface taking into account the surface-area, placing more bubbles on large surfaces. With the corrected routine "sarea.f" this can be done easilly.
- \* Another future development would be to couple ESPout with ESPmrt. In this way a direct transfer of surface temperatures to ESPmrt would be possible. On the other hand within ESPout it would be possible to predict thermal comfort on realistic calculations of Tmrt.



cubical1 :

GEN

12,	7,	0.000							
0.0000,	0.0000,	0.0000,							
0.1000,	0.0000,	0.0000,							
2.9000,	0.0000,	0.0000,							
3.0000,	0.0000,	0.0000,							
3.0000,	3.0000,	0.0000,							
0.0000,	3.0000,	0.0000,							
0.1000,	0.0000,	2.9000,							
2.9000,	0.0000,	2.9000,							
0.0000,	0.0000,	3.0000,							
3.0000,	0.0000,	3.0000,							
3.0000,	3.0000,	3.0000,							
0.0000,	3.0000,	3.0000,							
8,	1,	2,	7,	8,	3,	4,	10,	9,	
4,	2,	3,	8,	7,					
4,	4,	5,	11,	10,					
4,	5,	6,	12,	11,					
4,	6,	1,	9,	12,					
6,	6,	5,	4,	3,	2,	1,			
4,	9,	10,	11,	12,					
0,	0,	0,	0,	0,	0,	0,			
0,	0,	0,	0,	0,	0,	0,			
1,	7,	0,	0						

cubical2 :

GEN

	8,		6,		0.000
0.0000,		0.0000,		0.5000,	
3.0000,		0.0000,		0.5000,	
3.0000,		3.0000,		0.0000,	
0.0000,		3.0000,		0.0000,	
0.0000,		0.0000,		3.0000,	
3.0000,		0.0000,		3.0000,	
3.0000,		3.0000,		3.0000,	
0.0000,		3.0000,		3.0000,	
4,	1,	2,	6,	5,	
4,	2,	3,	7,	6,	
4,	3,	4,	8,	7,	
4,	4,	1,	5,	8,	
4,	5,	6,	7,	8,	
4,	4,	3,	2,	1,	
0,	0,	0,	0,	0,	0,
0,	0,	0,	0,	0,	0,
1,	6,	0,	0		

## 6. Example run of ESPmrt

We start with a geometry file which is made with the module ESPimp. It contains information about the vertices and the surfaces of the geometry we are considering. For tmrt-calculations the geometry has to be of GEN-type.

The name of the file for this example run is : 'geo'

This geometry consists of 26 vertices and 15 surfaces. For a more elaborate description of geometry files see the ESP-manual.

For reasons of clarity, in the next example run a great number of menu-picks is chosen to give the first-time-user a good idea of the different possibilities of the program.

In the picture below a lay out of the test geometry is given, seen from above. This picture is generated with VIEWER. On the next page a listing of the file 'geo' is printed.

Note that we have made sure that the window (surface 4) would not overlap with the wall it is on, and that we have not taken 2, 3 and 6 together. This is to ensure that all surfaces are rectangular.

However we break this rule with surfaces 8 and 9 deliberately. We will see that this ruins the analytical calculations for most surfaces, and also that the viewfactors from surface 9 (and 8) are not correct. The sum over a row is not equal to 1.00 because some bubbles fall off the surfaces.

GEN

```

      26,      15,      0.000
0.0000, 0.0000, 0.0000,
4.0000, 0.0000, 0.0000,
4.0000, 5.0000, 0.0000,
0.0000, 5.0000, 0.0000,
0.0000, 0.0000, 2.7500,
4.0000, 0.0000, 2.7500,
4.0000, 5.0000, 2.7500,
0.0000, 5.0000, 2.7500,
4.0000, 1.0000, 0.0000,
4.0000, 2.0000, 0.0000,
4.0000, 1.0000, 1.2500,
4.0000, 2.0000, 1.2500,
4.0000, 1.0000, 2.2500,
4.0000, 2.0000, 2.2500,
4.0000, 1.0000, 2.7500,
4.0000, 2.0000, 2.7500,
0.0000, 3.5000, 0.0000,
0.0000, 5.0000, 0.7500,
0.0000, 3.5000, 2.0000,
0.0000, 3.5000, 2.7500,
0.0000, 2.5000, 0.0000,
0.0000, 1.0000, 0.0000,
0.0000, 2.5000, 2.2500,
0.0000, 1.0000, 2.2500,
0.0000, 2.5000, 2.7500,
0.0000, 1.0000, 2.7500,
4,  1,  2,  6,  5,
6,  2,  9, 11, 13, 15,  6,
4,  9, 10, 12, 11,
4, 11, 12, 14, 13,
4, 13, 14, 16, 15,
6, 10,  3,  7, 16, 14, 12,
5,  3,  4, 18,  8,  7,
4,  4, 17, 19, 18,
4, 18, 19, 20,  8,
6, 17, 21, 23, 25, 20, 19,
4, 21, 22, 24, 23,
4, 23, 24, 26, 25,
5, 22,  1,  5, 26, 24,
9,  1, 22, 21, 17,  4,  3, 10,  9,  2,
9,  6, 15, 16,  7,  8, 20, 25, 26,  5,
0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
1, 15,  0,  0

```

We run the program by typing its name, preceded by the proper directory-path if necessary. For clarity the input which is given by the user is followed by \*\*\*. The rest of the text is the output of the program.

- - - -

mrt \*\*\*

TERMINAL TYPE (press return for options) ?>  
-1 \*\*\*

-----  
ESPmrt: Mean Radiant Temperature program

F77 version; 1.1a of March 1988  
-----

ESPmrt:     iNPUT  
             pARAM  
             vIEWFC  
             tMRT  
             oUTPUT  
             rEADFC  
             gCHECK  
             fINISH  
             hELP

ESPmrt:??>  
h \*\*\*

     iNPUT     : Reads input from geometry-file made with  
              ESPIMP. (only GEN-type for TMRT).  
              Also asks temperatures and position of cube.  
              or reads input from TMRT-file.  
     pARAM     : Change number of bubbles on a surface,  
              Or number of patches on bubble  
     vIEWFC    : Calculate viewfactors for all surfaces  
              of the input geometry.  
     tMRT      : Place a rectangular body of known dimensions  
              inside the geometry and compute viewfactors  
              from this body to the geometry only.  
              Also compute Mean Radiant Temperature (TMRT)  
              or compute TMRT on several points of geometry  
              by placing small cubes on those points.  
     rEADFC    : Read viewfactors from < cube.fac > and  
              calculate TMRT for this case.  
     oUTPUT    : Write viewfactors (vIEWFC option only)  
              or write temperatures (TMRT only) to a  
              chosen file.  
     gCHECK    : Writes TMRT-geometry to file <check.geo>.  
     fINISH    : End program.

ESPmrt:??>  
i \*\*\*

Geometry file name ?

```
geo ***
  TMRT input filename ?
geo.temp ***
```

-----

```
input :    input
        change
        -END
        help
```

```
input :?>
h ***
```

```
input    : Give the values of all variables and write
           them to the given TMRT file.
change    : Change some of the given temperature values
           and write to given TMRT file.
-END      : Return to main menu
```

-----

```
input :    input
        change
        -END
        help
```

```
input :?>
c ***
No input data yet, choose input first
input :?>
i ***
```

-----

There are 15 surfaces in this geometry.

Give the temperature for each surface [Celsius]

```
Temperature for surface 1 is ? 18 ***
Temperature for surface 2 is ? 18 ***
Temperature for surface 3 is ? 18 ***
Temperature for surface 4 is ? 1  ***
Temperature for surface 5 is ? 18 ***
```

```
Temperature for surface 6 is ? 18 ***
Temperature for surface 7 is ? 18 ***
Temperature for surface 8 is ? 17 ***
Temperature for surface 9 is ? 19 ***
Temperature for surface 10 is ? 18 ***
Temperature for surface 11 is ? 6 ***
Temperature for surface 12 is ? 1 ***
Temperature for surface 13 is ? 18 ***
Temperature for surface 14 is ? 16 ***
Temperature for surface 15 is ? 20 ***
```

-----

Do you want to calculate the Mean Radiant Temperature  
for one body in the geometry, or do you want to calculate  
it for a number of points inside the geometry ?

cHOICE:     bODY  
             cUBES

cHOICE: ?>

c \*\*\*

How many cubes are there in the geometry ?

2 \*\*\*

Give the X,Y,Z-coordinates of cube    1

2 \*\*\*

1.5 \*\*\*

1.5 \*\*\*

Give the X,Y,Z-coordinates of cube    2

2 \*\*\*

3.5 \*\*\*

1.5 \*\*\*

-----

iNPUT :     iNPUT  
             cHANGE  
             -END  
             hELP

iNPUT : ?>

\_ \*\*\*

Grid division is set to    1

Patch division set to    25

Choose pARAM-option to change

---

```
ESPmrt:  iNPUT
          pARAM
          vIEWFC
          tMRT
          oUTPUT
          rEADFC
          gCHECK
          fINISH
          hELP
```

```
ESPmrt:??>
```

```
g ***
```

```
  2 cubes added to geometry and written in file <check.geo >
```

---

```
ESPmrt:  iNPUT
          pARAM
          vIEWFC
          tMRT
          oUTPUT
          rEADFC
          gCHECK
          fINISH
          hELP
```

```
ESPmrt:??>
```

```
t ***
```

---

```
Cube    1 added to geometry and written in file <cube.geo    >
```

---

```
The viewfactors that will be calculated  analytically
are from the surface(s) :  18  19
```

```
Computation commences
```

```
1/4 complete
```

```
1/2 complete
```



3/4 complete  
Almost complete

-----  
View factor information  
From Cube nr. 1 to geometry

0.0000, 0.0000, 0.0041, 0.0048, 0.0015, 0.1313, 0.2183, 0.0284,  
0.0248, 0.0543, 0.0281, 0.0054, 0.0000, 0.2322, 0.2677,  
0.1402, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0275, 0.0194,  
0.0182, 0.0807, 0.1945, 0.0332, 0.1149, 0.1683, 0.2038,  
0.5914, 0.0532, 0.0038, 0.0043, 0.0015, 0.0000, 0.0000, 0.0000,  
0.0000, 0.0000, 0.0081, 0.0015, 0.0531, 0.1239, 0.1586,  
0.1389, 0.1140, 0.0679, 0.0721, 0.0249, 0.1824, 0.0285, 0.0000,  
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.1675, 0.2034,  
0.1220, 0.0200, 0.0259, 0.0012, 0.0000, 0.0321, 0.0225, 0.0064,  
0.0003, 0.0137, 0.0391, 0.0000, 0.0198, 0.6975, 0.0000,  
0.1009, 0.0151, 0.0000, 0.0106, 0.0121, 0.0246, 0.0159, 0.0001,  
0.0051, 0.0107, 0.0141, 0.0172, 0.0145, 0.0000, 0.7599,

Calculationpoint [m] : 2.00 1.50 1.50  
TMRT-value [C] : 17.14  
VRT values(x,y and z) [C] : 1.62 -0.29 2.81

-----  
Cube 2 added to geometry and written in file <cube.geo >

-----  
The viewfactors that will be calculated analytically  
are from the surface(s) : 18 19

Computation commences

1/4 complete  
1/2 complete  
3/4 complete  
Almost complete

-----  
View factor information  
From Cube nr. 2 to geometry

0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0632, 0.5928, 0.0264,

0.0368, 0.0000, 0.0000, 0.0000, 0.0000, 0.1220, 0.1594,  
0.0275, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.1414, 0.1001,  
0.0944, 0.1499, 0.0785, 0.0159, 0.0210, 0.1680, 0.2039,  
0.2182, 0.0298, 0.0206, 0.0193, 0.0078, 0.0628, 0.0000, 0.0000,  
0.0000, 0.0341, 0.0639, 0.0122, 0.0298, 0.2339, 0.2689,  
0.0285, 0.0192, 0.0212, 0.0198, 0.0082, 0.3929, 0.1390, 0.0000,  
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.1675, 0.2034,  
0.0225, 0.0038, 0.0081, 0.0002, 0.0000, 0.0671, 0.1220, 0.0316,  
0.0016, 0.0258, 0.0168, 0.0000, 0.0036, 0.6975, 0.0000,  
0.0159, 0.0029, 0.0000, 0.0025, 0.0042, 0.0528, 0.1009, 0.0007,  
0.0249, 0.0202, 0.0054, 0.0077, 0.0027, 0.0000, 0.7599,

Calculationpoint	[m]	:	2.00	3.50	1.50
TMRT-value	[C]	:	17.56		
VRT values(x,y and z)	[C]	:	0.83	1.23	2.95

-----

ESPmrt:     iNPUT  
              pARAM  
              vIEWFC  
              tMRT  
              oUTPUT  
              rEADFC  
              gCHECK  
              fINISH  
              hELP

ESPmrt:??>  
O \*\*\*

-----

Temperature output file name ? (enter - to return to main menu)

tmrt1 \*\*\*

-----

ESPmrt:     iNPUT  
              pARAM  
              vIEWFC  
              tMRT  
              oUTPUT  
              rEADFC  
              gCHECK  
              fINISH  
              hELP

```

ESPmrt:??>
i ***
  Geometry file name ?
geo ***
  TMRT input filename ?
geo.temp ***

```

```

-----

iINPUT :      iINPUT
          cCHANGE
          -END
          hHELP

```

```

iINPUT :??>
c ***
Surface   1 Temperature   18.0000 [ C ]
Surface   2 Temperature   18.0000 [ C ]
Surface   3 Temperature   18.0000 [ C ]
Surface   4 Temperature    1.0000 [ C ]
Surface   5 Temperature   18.0000 [ C ]
Surface   6 Temperature   18.0000 [ C ]
Surface   7 Temperature   18.0000 [ C ]
Surface   8 Temperature   17.0000 [ C ]
Surface   9 Temperature   19.0000 [ C ]
Surface  10 Temperature   18.0000 [ C ]
Surface  11 Temperature    6.0000 [ C ]
Surface  12 Temperature    1.0000 [ C ]
Surface  13 Temperature   18.0000 [ C ]
Surface  14 Temperature   16.0000 [ C ]
Surface  15 Temperature   20.0000 [ C ]

```

```

Change temperature of surface nr. ? (0 = stop)
4 ***
  Temperature of surface  4 in degrees Celsius ?
8 ***

```

```

Surface   1 Temperature   18.0000 [ C ]
Surface   2 Temperature   18.0000 [ C ]
Surface   3 Temperature   18.0000 [ C ]
Surface   4 Temperature    8.0000 [ C ]
Surface   5 Temperature   18.0000 [ C ]
Surface   6 Temperature   18.0000 [ C ]
Surface   7 Temperature   18.0000 [ C ]
Surface   8 Temperature   17.0000 [ C ]
Surface   9 Temperature   19.0000 [ C ]
Surface  10 Temperature   18.0000 [ C ]
Surface  11 Temperature    6.0000 [ C ]
Surface  12 Temperature    1.0000 [ C ]
Surface  13 Temperature   18.0000 [ C ]
Surface  14 Temperature   16.0000 [ C ]
Surface  15 Temperature   20.0000 [ C ]

```

Change temperature of surface nr. ? (0 = stop)  
0 \*\*\*

---

iNPUT :     iNPUT  
          cHANGE  
          -END  
          hELP

iNPUT :?>  
\_ \*\*\*  
Grid division is set to    1  
Patch division set to     25  
  
Choose pARAM-option to change

---

ESpmrt:     iNPUT  
          pARAM  
          vIEWFC  
          tMRT  
          oUTPUT  
          rEADFC  
          gCHECK  
          fINISH  
          hELP

ESpmrt:??>  
r \*\*\*

---

Calculationpoint	[m]	:	2.00	1.50	1.50
TMRT-value	[C]	:	17.24		
VRT values(x,y and z)	[C]	:	2.06	-0.29	2.86
Calculationpoint	[m]	:	2.00	3.50	1.50
TMRT-value	[C]	:	17.61		
VRT values(x,y and z)	[C]	:	0.95	1.11	2.96

---

ESpmrt:     iNPUT

pARAM  
vIEWFC  
tMRT  
oUTPUT  
rEADFC  
gCHECK  
fINISH  
hELP

ESPmrt:??>  
f \*\*\*  
No output yet !

Finish ? (y/n)  
n \*\*\*

-----

ESPmrt:     iNPUT  
             pARAM  
             vIEWFC  
             tMRT  
             oUTPUT  
             rEADFC  
             gCHECK  
             fINISH  
             hELP

ESPmrt:??>  
o \*\*\*

-----

Temperature output file name ? (enter - to return to main menu)

tmrt2 \*\*\*

-----

ESPmrt:     iNPUT  
             pARAM  
             vIEWFC  
             tMRT  
             oUTPUT  
             rEADFC  
             gCHECK  
             fINISH  
             hELP

```
ESPmrt:??>
p ***
```

Variable gfact represents the grid subdivision.  
Every surface is divided in gfact\*\*2 sub-surfaces  
At the center of each sub-surface a bubble is placed  
Current value of grid subdivision , gfact = 1

```
Do you want to change it ?
y ***
```

```
Input new value ?
4 ***
```

Variable np represents the patch subdivision.  
Every bubble is divided into np rings  
Current value of patch subdivision np = 25

```
Do you want to change it ?
y ***
```

```
Input new value ?
(max. set to 50)
10 ***
```

---

```
ESPmrt:      iINPUT
              pPARAM
              vIEWFC
              tMRT
              oUTPUT
              rEADFC
              gCHECK
              fINISH
              hELP
```

```
ESPmrt:??>
v ***
Griddivision = 4 Patchdivision = 10
Geometry file name ?
geo ***
```

---

Surface-by surface visibility matrix

-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1
1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1
1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1
1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1
1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	-1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	1	1
1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	1	1
1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	1	1
1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	1	1
1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	1	1
1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	-1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1

---

The viewfactors that will be calculated analytically  
are from the surface(s) : 10 11 12 13

Computation commences

1/4 complete  
1/2 complete  
3/4 complete  
Almost complete

---

## View factor information

```

0.0000, 0.0820, 0.0207, 0.0183, 0.0069, 0.0521, 0.1104, 0.0091,
0.0086, 0.0194, 0.0530, 0.0108, 0.0805, 0.2641, 0.2641,
0.3461, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0417, 0.0127,
0.0101, 0.0273, 0.0499, 0.0123, 0.0514, 0.2243, 0.2243,
0.1803, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0628, 0.0210,
0.0152, 0.0379, 0.0582, 0.0111, 0.0420, 0.3837, 0.1880,
0.1965, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0663, 0.0196,
0.0169, 0.0398, 0.0557, 0.0129, 0.0456, 0.2251, 0.3217,
0.1571, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0597, 0.0147,
0.0187, 0.0358, 0.0485, 0.0190, 0.0403, 0.1558, 0.4506,
0.0707, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.2039, 0.0343,
0.0352, 0.0450, 0.0451, 0.0094, 0.0210, 0.2678, 0.2678,
0.1104, 0.0124, 0.0069, 0.0052, 0.0021, 0.1549, 0.0000, 0.0522,
0.0556, 0.0382, 0.0191, 0.0039, 0.0112, 0.2641, 0.2641,
0.0320, 0.0100, 0.0074, 0.0063, 0.0027, 0.0886, 0.1914, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.2248, 0.1245,
0.0285, 0.0082, 0.0065, 0.0059, 0.0030, 0.0865, 0.2229, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.1119, 0.2141,
0.0811, 0.0258, 0.0168, 0.0140, 0.0064, 0.1287, 0.1376, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.2948, 0.2948,
0.1654, 0.0401, 0.0219, 0.0174, 0.0075, 0.1027, 0.0737, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.3182, 0.2532,
0.1420, 0.0374, 0.0162, 0.0177, 0.0095, 0.0960, 0.0679, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.1605, 0.4530,
0.3416, 0.0470, 0.0191, 0.0159, 0.0072, 0.0652, 0.0398, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.2321, 0.2321,
0.1425, 0.0333, 0.0233, 0.0116, 0.0037, 0.1127, 0.1425, 0.0325,
0.0173, 0.0410, 0.0524, 0.0052, 0.0331, 0.0000, 0.3489,
0.1425, 0.0333, 0.0120, 0.0176, 0.0090, 0.1127, 0.1425, 0.0191,
0.0307, 0.0410, 0.0446, 0.0131, 0.0331, 0.3489, 0.0000,

```

The sum over one row in the viewfactor matrix should be equal to 1.00.

A deviation larger then 0.01 was encountered

The largest deviation was : 0.3124 in row 9

```

-----

ESPmrt:  iNPUT
          pARAM
          vIEWFC
          tMRT
          oUTPUT
          rEADFC
          gCHECK
          fINISH
          hELP

```

ESPmrt:??>

○ \*\*\*



-----  
View factor file name ? (enter - to return to main menu)

geo.view \*\*\*  
-----

ESPMrt:     iNPUT  
             pARAM  
             vIEWFC  
             tMRT  
             oUTPUT  
             rEADFC  
             gCHECK  
             fINISH  
             hELP

ESPMrt:??>  
f \*\*\*

=====  
Here the program ends. The program has made a few files and saved them  
under the working directory. They are :

check.geo	Default-name
cube.fac	Default-name
cube.geo	Default-name
geo.temp	User chosen-name
geo.view	User chosen-name
tmrt1	User chosen-name
tmrt2	User chosen-name

These files are listed below.

-----  
 In check.geo the geometry with the two cubes is represented. This  
 file was made with the gCHECK-menu-pick.

GEN

42,	27,	0.000
0.0000,	0.0000,	0.0000,
4.0000,	0.0000,	0.0000,
4.0000,	5.0000,	0.0000,
0.0000,	5.0000,	0.0000,
0.0000,	0.0000,	2.7500,
4.0000,	0.0000,	2.7500,
4.0000,	5.0000,	2.7500,
0.0000,	5.0000,	2.7500,
4.0000,	1.0000,	0.0000,
4.0000,	2.0000,	0.0000,
4.0000,	1.0000,	1.2500,
4.0000,	2.0000,	1.2500,
4.0000,	1.0000,	2.2500,
4.0000,	2.0000,	2.2500,
4.0000,	1.0000,	2.7500,
4.0000,	2.0000,	2.7500,
0.0000,	3.5000,	0.0000,
0.0000,	5.0000,	0.7500,
0.0000,	3.5000,	2.0000,
0.0000,	3.5000,	2.7500,
0.0000,	2.5000,	0.0000,
0.0000,	1.0000,	0.0000,
0.0000,	2.5000,	2.2500,
0.0000,	1.0000,	2.2500,
0.0000,	2.5000,	2.7500,
0.0000,	1.0000,	2.7500,
1.9900,	1.4900,	1.4900,
1.9900,	1.4900,	1.5100,
1.9900,	1.5100,	1.4900,
1.9900,	1.5100,	1.5100,
2.0100,	1.4900,	1.4900,
2.0100,	1.4900,	1.5100,
2.0100,	1.5100,	1.4900,
2.0100,	1.5100,	1.5100,
1.9900,	3.4900,	1.4900,
1.9900,	3.4900,	1.5100,
1.9900,	3.5100,	1.4900,
1.9900,	3.5100,	1.5100,
2.0100,	3.4900,	1.4900,
2.0100,	3.4900,	1.5100,
2.0100,	3.5100,	1.4900,
2.0100,	3.5100,	1.5100,
4,	1,	2,
6,	2,	9,
4,	9,	10,
4,	11,	12,
4,	13,	14,
6,	10,	3,
		7,
		16,
		14,
		12,

5, 3, 4, 18, 8, 7,  
4, 4, 17, 19, 18,  
4, 18, 19, 20, 8,  
6, 17, 21, 23, 25, 20, 19,  
4, 21, 22, 24, 23,  
4, 23, 24, 26, 25,  
5, 22, 1, 5, 26, 24,  
9, 1, 22, 21, 17, 4, 3, 10, 9, 2,  
9, 6, 15, 16, 7, 8, 20, 25, 26, 5,  
4, 29, 33, 34, 30,  
4, 27, 29, 30, 28,  
4, 27, 28, 32, 31,  
4, 31, 32, 34, 33,  
4, 27, 31, 33, 29,  
4, 28, 30, 34, 32,  
4, 37, 41, 42, 38,  
4, 35, 37, 38, 36,  
4, 35, 36, 40, 39,  
4, 39, 40, 42, 41,  
4, 35, 39, 41, 37,  
4, 36, 38, 42, 40,  
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
0, 0, 0, 0

-----  
 In cube.fac the calculated viewfactors from the cubefaces to the geometry-surfaces are stored. These can later be read with the READFC-menu-pick.

Number cubes = 2  
 Cube index = 1  
 Coordinates = 2.0000 1.5000 1.5000

Dimensions: 0.0200 0.0200 0.0200

Viewfactors from cube surfaces (vertical 1 to 6)  
 to geometry surfaces (horizontal)

0.0000	0.0000	0.0041	0.0048	0.0015	0.1313	0.2183
0.0284	0.0248	0.0543	0.0281	0.0054	0.0000	0.2322
0.2677						
0.1402	0.0000	0.0000	0.0000	0.0000	0.0000	0.0275
0.0194	0.0182	0.0807	0.1945	0.0332	0.1149	0.1683
0.2038						
0.5914	0.0532	0.0038	0.0043	0.0015	0.0000	0.0000
0.0000	0.0000	0.0000	0.0081	0.0015	0.0531	0.1239
0.1586						
0.1389	0.1140	0.0679	0.0721	0.0249	0.1824	0.0285
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1675
0.2034						
0.1220	0.0200	0.0259	0.0012	0.0000	0.0321	0.0225
0.0064	0.0003	0.0137	0.0391	0.0000	0.0198	0.6975
0.0000						
0.1009	0.0151	0.0000	0.0106	0.0121	0.0246	0.0159
0.0001	0.0051	0.0107	0.0141	0.0172	0.0145	0.0000
0.7599						

Number cubes = 2  
 Cube index = 2  
 Coordinates = 2.0000 3.5000 1.5000

Dimensions: 0.0200 0.0200 0.0200

Viewfactors from cube surfaces (vertical 1 to 6)

to geometry surfaces (horizontal)

0.0000	0.0000	0.0000	0.0000	0.0000	0.0632	0.5928
0.0264	0.0368	0.0000	0.0000	0.0000	0.0000	0.1220
0.1594						
0.0275	0.0000	0.0000	0.0000	0.0000	0.0000	0.1414
0.1001	0.0944	0.1499	0.0785	0.0159	0.0210	0.1680
0.2039						
0.2182	0.0298	0.0206	0.0193	0.0078	0.0628	0.0000
0.0000	0.0000	0.0341	0.0639	0.0122	0.0298	0.2339
0.2689						
0.0285	0.0192	0.0212	0.0198	0.0082	0.3929	0.1390
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1675
0.2034						
0.0225	0.0038	0.0081	0.0002	0.0000	0.0671	0.1220
0.0316	0.0016	0.0258	0.0168	0.0000	0.0036	0.6975
0.0000						
0.0159	0.0029	0.0000	0.0025	0.0042	0.0528	0.1009
0.0007	0.0249	0.0202	0.0054	0.0077	0.0027	0.0000
0.7599						

-----  
 In cube.geo the geometry with the cube that is used in the current  
 viewfactor-calculation is stored. If more cubes are used then this file  
 will be overwritten and after the run will contain the geometry with the  
 last cube.

GEN

34,	21,	0.000			
0.0000,	0.0000,	0.0000,			
4.0000,	0.0000,	0.0000,			
4.0000,	5.0000,	0.0000,			
0.0000,	5.0000,	0.0000,			
0.0000,	0.0000,	2.7500,			
4.0000,	0.0000,	2.7500,			
4.0000,	5.0000,	2.7500,			
0.0000,	5.0000,	2.7500,			
4.0000,	1.0000,	0.0000,			
4.0000,	2.0000,	0.0000,			
4.0000,	1.0000,	1.2500,			
4.0000,	2.0000,	1.2500,			
4.0000,	1.0000,	2.2500,			
4.0000,	2.0000,	2.2500,			
4.0000,	1.0000,	2.7500,			
4.0000,	2.0000,	2.7500,			
0.0000,	3.5000,	0.0000,			
0.0000,	5.0000,	0.7500,			
0.0000,	3.5000,	2.0000,			
0.0000,	3.5000,	2.7500,			
0.0000,	2.5000,	0.0000,			
0.0000,	1.0000,	0.0000,			
0.0000,	2.5000,	2.2500,			
0.0000,	1.0000,	2.2500,			
0.0000,	2.5000,	2.7500,			
0.0000,	1.0000,	2.7500,			
1.9900,	3.4900,	1.4900,			
1.9900,	3.4900,	1.5100,			
1.9900,	3.5100,	1.4900,			
1.9900,	3.5100,	1.5100,			
2.0100,	3.4900,	1.4900,			
2.0100,	3.4900,	1.5100,			
2.0100,	3.5100,	1.4900,			
2.0100,	3.5100,	1.5100,			
4,	1,	2,	6,	5,	
6,	2,	9,	11,	13,	15,
4,	9,	10,	12,	11,	
4,	11,	12,	14,	13,	
4,	13,	14,	16,	15,	
6,	10,	3,	7,	16,	14,
5,	3,	4,	18,	8,	7,
4,	4,	17,	19,	18,	
4,	18,	19,	20,	8,	
6,	17,	21,	23,	25,	20,
4,	21,	22,	24,	23,	
4,	23,	24,	26,	25,	

```

5,    22,    1,    5,    26,    24,
9,     1,   22,   21,   17,    4,    3,   10,    9,    2,
9,     6,   15,   16,    7,    8,   20,   25,   26,    5,
4,   29,   33,   34,   30,
4,   27,   29,   30,   28,
4,   27,   28,   32,   31,
4,   31,   32,   34,   33,
4,   27,   31,   33,   29,
4,   28,   30,   34,   32,
0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
0,    0,    0,    0,    0,    0,
0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
0,    0,    0,    0,    0,    0,
0,    0,    0,    0

```

-----  
 In geo.temp the tmrt input-values that the user has given, are stored. These are the temperatures of all geometry-surfaces and the information about the cubes or body. In the next run these values can be read from this file.

```

15      Nsurf
18.0000    temperature surface :    1
18.0000    temperature surface :    2
18.0000    temperature surface :    3
 8.0000    temperature surface :    4
18.0000    temperature surface :    5
18.0000    temperature surface :    6
18.0000    temperature surface :    7
17.0000    temperature surface :    8
19.0000    temperature surface :    9
18.0000    temperature surface :   10
 6.0000    temperature surface :   11
 1.0000    temperature surface :   12
18.0000    temperature surface :   13
16.0000    temperature surface :   14
20.0000    temperature surface :   15
      CUBE
0.0200    0.0200    0.0200    Dimensions cube/body
      2    Number of cubes
2.0000    1.5000    1.5000    Coordinates of cube    1
2.0000    3.5000    1.5000    Coordinates of cube    2

```

-----  
 In geo.view the viewfactors from the geometry surfaces to other geometry surfaces are stored. They have been calculated with the VIEWFC-menu-pick and stored with the OUTPUT-menu-pick. (see next page)



15						
0.0000	0.0820	0.0207	0.0183	0.0069	0.0521	0.1104
0.0091	0.0086	0.0194	0.0530	0.0108	0.0805	0.2641
0.2641						
0.3461	0.0000	0.0000	0.0000	0.0000	0.0000	0.0417
0.0127	0.0101	0.0273	0.0499	0.0123	0.0514	0.2243
0.2243						
0.1803	0.0000	0.0000	0.0000	0.0000	0.0000	0.0628
0.0210	0.0152	0.0379	0.0582	0.0111	0.0420	0.3837
0.1880						
0.1965	0.0000	0.0000	0.0000	0.0000	0.0000	0.0663
0.0196	0.0169	0.0398	0.0557	0.0129	0.0456	0.2251
0.3217						
0.1571	0.0000	0.0000	0.0000	0.0000	0.0000	0.0597
0.0147	0.0187	0.0358	0.0485	0.0190	0.0403	0.1558
0.4506						
0.0707	0.0000	0.0000	0.0000	0.0000	0.0000	0.2039
0.0343	0.0352	0.0450	0.0451	0.0094	0.0210	0.2678
0.2678						
0.1104	0.0124	0.0069	0.0052	0.0021	0.1549	0.0000
0.0522	0.0556	0.0382	0.0191	0.0039	0.0112	0.2641
0.2641						
0.0320	0.0100	0.0074	0.0063	0.0027	0.0886	0.1914
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2248
0.1245						
0.0285	0.0082	0.0065	0.0059	0.0030	0.0865	0.2229
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1119
0.2141						
0.0811	0.0258	0.0168	0.0140	0.0064	0.1287	0.1376
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2948
0.2948						
0.1654	0.0401	0.0219	0.0174	0.0075	0.1027	0.0737
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3182
0.2532						
0.1420	0.0374	0.0162	0.0177	0.0095	0.0960	0.0679
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1605
0.4530						
0.3416	0.0470	0.0191	0.0159	0.0072	0.0652	0.0398
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2321
0.2321						
0.1425	0.0333	0.0233	0.0116	0.0037	0.1127	0.1425
0.0325	0.0173	0.0410	0.0524	0.0052	0.0331	0.0000
0.3489						
0.1425	0.0333	0.0120	0.0176	0.0090	0.1127	0.1425
0.0191	0.0307	0.0410	0.0446	0.0131	0.0331	0.3489
0.0000						

-----  
 In tmrt1 the TMRT and VRT values are stored that have been calculated  
 with the tmRT-menu-pick. They are stored with the oUTPUT-menu-pick.

Calculationpoint	[m]	:	2.00	1.50	1.50
TMRT-value	[C]	:	17.14		
VRT (x,y and z) values	[C]	:	1.62	-0.29	2.81
Calculationpoint	[m]	:	2.00	3.50	1.50
TMRT-value	[C]	:	17.56		
VRT (x,y and z) values	[C]	:	0.83	1.23	2.95

-----  
 In tmrt2 the TMRT and VRT values are stored that have been calculated  
 with the rEADFC-menu-pick and the changed surface-temperature of surface  
 4 They are stored with the oUTPUT-menu-pick.

Calculationpoint	[m]	:	2.00	1.50	1.50
TMRT-value	[C]	:	17.24		
VRT (x,y and z) values	[C]	:	2.06	-0.29	2.86
Calculationpoint	[m]	:	2.00	3.50	1.50
TMRT-value	[C]	:	17.61		
VRT (x,y and z) values	[C]	:	0.95	1.11	2.96

## 7. APPENDIX

## 7.1. VIEWFACTORS

## 7.1.1. Viewfactors for the change in patch division.

	1	2	3	4	5	6	7
vwf	0.0000	0.1989	0.0321	0.0760	0.0321	0.1989	0.1989
1 mrt	0.0000	0.1937	0.0383	0.0923	0.0345	0.1927	0.1888
ana	0.0000	0.1945	0.0348	0.0802	0.0348	0.1945	0.1933
	8	9	10	11			
vwf	0.0321	0.1989	0.0321	0.0000			
1 mrt	0.0355	0.1888	0.0355	0.0000			
ana	0.0352	0.1982	0.0347	0.0000			
	1	2	3	4	5	6	7
vwf	0.1803	0.0000	0.0000	0.0399	0.0903	0.1663	0.2074
2 mrt	0.1714	0.0000	0.0000	0.0433	0.0799	0.1971	0.1924
ana	0.1836	0.0000	0.0000	0.0402	0.0876	0.1857	0.1938
	8	9	10	11			
vwf	0.0397	0.2074	0.0397	0.0292			
2 mrt	0.0436	0.1924	0.0436	0.0365			
ana	0.0428	0.1938	0.0428	0.0297			
	1	2	3	4	5	6	7
vwf	0.0326	0.0000	0.0000	0.2095	0.1663	0.0903	0.0397
3 mrt	0.0356	0.0000	0.0000	0.2104	0.1971	0.0821	0.0422
ana	0.0328	0.0000	0.0000	0.2133	0.1857	0.0876	0.0428
	8	9	10	11			
vwf	0.2074	0.0397	0.2074	0.0073			
3 mrt	0.1924	0.0422	0.1924	0.0059			
ana	0.1938	0.0428	0.1938	0.0074			
	1	2	3	4	5	6	7
vwf	0.0736	0.0321	0.1989	0.0000	0.1989	0.0321	0.0321
4 mrt	0.0791	0.0345	0.1927	0.0000	0.1937	0.0383	0.0355
ana	0.0684	0.0363	0.1927	0.0000	0.1927	0.0363	0.0364
	8	9	10	11			
vwf	0.1989	0.0321	0.1989	0.0024			
4 mrt	0.1888	0.0355	0.1888	0.0132			
ana	0.1940	0.0364	0.1940	0.0128			

	1	2	3	4	5	6	7
vwf	0.0326	0.0903	0.1663	0.2095	0.0000	0.0000	0.0397
5 mrt	0.0368	0.0799	0.1971	0.2078	0.0000	0.0000	0.0436
ana	0.0328	0.0876	0.1857	0.2133	0.0000	0.0000	0.0428
	8	9	10	11			
vwf	0.2074	0.0397	0.2074	0.0073			
5 mrt	0.1924	0.0436	0.1924	0.0065			
ana	0.1938	0.0428	0.1938	0.0074			
	1	2	3	4	5	6	7
vwf	0.1803	0.1663	0.0903	0.0399	0.0000	0.0000	0.2074
6 mrt	0.1733	0.1971	0.0821	0.0414	0.0000	0.0000	0.1924
ana	0.1836	0.1857	0.0876	0.0402	0.0000	0.0000	0.1938
	8	9	10	11			
vwf	0.0397	0.2074	0.0397	0.0292			
6 mrt	0.0422	0.1924	0.0422	0.0371			
ana	0.0428	0.1938	0.0428	0.0297			
	1	2	3	4	5	6	7
vwf	0.1745	0.2008	0.0388	0.0399	0.0388	0.2008	0.0000
7 mrt	0.1768	0.1859	0.0464	0.0426	0.0421	0.1887	0.0000
ana	0.1812	0.1925	0.0425	0.0401	0.0425	0.1925	0.0000
	8	9	10	11			
vwf	0.0000	0.1795	0.0921	0.0350			
7 mrt	0.0000	0.1945	0.0845	0.0387			
ana	0.0000	0.1882	0.0883	0.0322			
	1	2	3	4	5	6	7
vwf	0.0316	0.0388	0.2008	0.2095	0.2008	0.0388	0.0000
8 mrt	0.0374	0.0464	0.1859	0.2155	0.1887	0.0421	0.0000
ana	0.0330	0.0425	0.1925	0.2134	0.1925	0.0425	0.0000
	8	9	10	11			
vwf	0.0000	0.0921	0.1795	0.0083			
8 mrt	0.0000	0.0845	0.1945	0.0051			
ana	0.0000	0.0883	0.1882	0.0071			

	1	2	3	4	5	6	7
vwf	0.1811	0.2008	0.0388	0.0399	0.0388	0.2008	0.1795
9 mrt	0.1862	0.1859	0.0464	0.0426	0.0421	0.1887	0.1945
ana	0.1858	0.1925	0.0425	0.0401	0.0425	0.1925	0.1882
	8	9	10	11			
vwf	0.0921	0.0000	0.0000	0.0284			
9 mrt	0.0845	0.0000	0.0000	0.0293			
ana	0.0883	0.0000	0.0000	0.0276			
	1	2	3	4	5	6	7
vwf	0.0329	0.0388	0.2008	0.2095	0.2008	0.0388	0.0921
10mrt	0.0341	0.0464	0.1859	0.2155	0.1887	0.0421	0.0845
ana	0.0325	0.0425	0.1925	0.2134	0.1925	0.0425	0.0883
	8	9	10	11			
vwf	0.1795	0.0000	0.0000	0.0070			
10mrt	0.1945	0.0000	0.0000	0.0085			
ana	0.1882	0.0000	0.0000	0.0076			
	1	2	3	4	5	6	7
vwf	0.0000	0.1813	0.0450	0.0831	0.0450	0.1812	0.1855
11mrt	0.0000	0.1759	0.0474	0.0891	0.0486	0.1775	0.1955
ana	0.0000	0.1819	0.0452	0.0869	0.0453	0.1816	0.1982
	8	9	10	11			
vwf	0.0526	0.1751	0.0511	0.0000			
11mrt	0.0499	0.1703	0.0459	0.0000			
ana	0.0439	0.1702	0.0468	0.0000			

**7.1.2. Viewfactors for the tests with the analytical calculation.**

	1	2	3	4	5	6	7
ANA	0.0000	0.1927	0.0363	0.0812	0.0363	0.1927	0.1940
1 mrt	0.0000	0.1937	0.0383	0.0923	0.0345	0.1927	0.1888
ana	0.0000	0.1945	0.0348	0.0802	0.0348	0.1945	0.1933
	8	9	10	11			
ANA	0.0364	0.1940	0.0364	0.0000			
1 mrt	0.0355	0.1888	0.0355	0.0000			
ana	0.0352	0.1982	0.0347	0.0000			
	1	2	3	4	5	6	7
ANA	0.2133	0.0000	0.0000	0.0402	0.0876	0.1857	0.1938
2 mrt	0.1714	0.0000	0.0000	0.0433	0.0799	0.1971	0.1924
ana	0.1836	0.0000	0.0000	0.0402	0.0876	0.1857	0.1938
	8	9	10	11			
ANA	0.0428	0.1938	0.0428	0.0297			
2 mrt	0.0436	0.1924	0.0436	0.0365			
ana	0.0428	0.1938	0.0428	0.0297			
	1	2	3	4	5	6	7
ANA	0.0402	0.0000	0.0000	0.2133	0.1857	0.0876	0.0428
3 mrt	0.0356	0.0000	0.0000	0.2104	0.1971	0.0821	0.0422
ana	0.0328	0.0000	0.0000	0.2133	0.1857	0.0876	0.0428
	8	9	10	11			
ANA	0.1938	0.0428	0.1938	0.0074			
3 mrt	0.1924	0.0422	0.1924	0.0059			
ana	0.1938	0.0428	0.1938	0.0074			
	1	2	3	4	5	6	7
ANA	0.0812	0.0363	0.1927	0.0000	0.1927	0.0363	0.0364
4 mrt	0.0791	0.0345	0.1927	0.0000	0.1937	0.0383	0.0355
ana	0.0684	0.0363	0.1927	0.0000	0.1927	0.0363	0.0364
	8	9	10	11			
ANA	0.1940	0.0364	0.1940	0.0128			
4 mrt	0.1888	0.0355	0.1888	0.0132			
ana	0.1940	0.0364	0.1940	0.0128			

		1	2	3	4	5	6	7
	ANA	0.0402	0.0876	0.1857	0.2133	0.0000	0.0000	0.0428
5	mrt	0.0368	0.0799	0.1971	0.2078	0.0000	0.0000	0.0436
	ana	0.0328	0.0876	0.1857	0.2133	0.0000	0.0000	0.0428
		8	9	10	11			
	ANA	0.1938	0.0428	0.1938	0.0074			
5	mrt	0.1924	0.0436	0.1924	0.0065			
	ana	0.1938	0.0428	0.1938	0.0074			
		1	2	3	4	5	6	7
	ANA	0.2133	0.1857	0.0876	0.0402	0.0000	0.0000	0.1938
6	mrt	0.1733	0.1971	0.0821	0.0414	0.0000	0.0000	0.1924
	ana	0.1836	0.1857	0.0876	0.0402	0.0000	0.0000	0.1938
		8	9	10	11			
	ANA	0.0428	0.1938	0.0428	0.0297			
6	mrt	0.0422	0.1924	0.0422	0.0371			
	ana	0.0428	0.1938	0.0428	0.0297			
		1	2	3	4	5	6	7
	ANA	0.2134	0.1925	0.0425	0.0401	0.0425	0.1925	0.0000
7	mrt	0.1768	0.1859	0.0464	0.0426	0.0421	0.1887	0.0000
	ana	0.1812	0.1925	0.0425	0.0401	0.0425	0.1925	0.0000
		8	9	10	11			
	ANA	0.0000	0.1882	0.0883	0.0321			
7	mrt	0.0000	0.1945	0.0845	0.0387			
	ana	0.0000	0.1882	0.0883	0.0322			
		1	2	3	4	5	6	7
	ANA	0.0401	0.0425	0.1925	0.2134	0.1925	0.0425	0.0000
8	mrt	0.0374	0.0464	0.1859	0.2155	0.1887	0.0421	0.0000
	ana	0.0330	0.0425	0.1925	0.2134	0.1925	0.0425	0.0000
		8	9	10	11			
	ANA	0.0000	0.0883	0.1882	0.0071			
8	mrt	0.0000	0.0845	0.1945	0.0051			
	ana	0.0000	0.0883	0.1882	0.0071			

	1	2	3	4	5	6	7
ANA	0.2134	0.1925	0.0425	0.0401	0.0425	0.1925	0.1882
9 mrt	0.1862	0.1859	0.0464	0.0426	0.0421	0.1887	0.1945
ana	0.1858	0.1925	0.0425	0.0401	0.0425	0.1925	0.1882
	8	9	10	11			
ANA	0.0883	0.0000	0.0000	0.0277			
9 mrt	0.0845	0.0000	0.0000	0.0293			
ana	0.0883	0.0000	0.0000	0.0276			
	1	2	3	4	5	6	7
ANA	0.0401	0.0425	0.1925	0.2134	0.1925	0.0425	0.0883
10mrt	0.0341	0.0464	0.1859	0.2155	0.1887	0.0421	0.0845
ana	0.0325	0.0425	0.1925	0.2134	0.1925	0.0425	0.0883
	8	9	10	11			
ANA	0.1882	0.0000	0.0000	0.0076			
10mrt	0.1945	0.0000	0.0000	0.0085			
ana	0.1882	0.0000	0.0000	0.0076			
	1	2	3	4	5	6	7
ANA	0.0000	0.1819	0.0453	0.0869	0.0453	0.1816	0.1980
11mrt	0.0000	0.1759	0.0474	0.0891	0.0486	0.1775	0.1955
ana	0.0000	0.1819	0.0452	0.0869	0.0453	0.1816	0.1982
	8	9	10	11			
ANA	0.0439	0.1705	0.0468	0.0000			
11mrt	0.0499	0.1703	0.0459	0.0000			
ana	0.0439	0.1702	0.0468	0.0000			



## 7.2. TMRT

### 7.2.1. Tmrt calculation; change in patch division.

CUBE

Calculationpoint	[m]	:	0.50	0.50	0.50	
ana TMRT-value	[C]	:	15.61			
1mrt TMRT-value	[C]	:	15.58			
2mrt TMRT-value	[C]	:	15.58			
ana VRT (x,y and z)	[C]	:	-1.25	5.24	-2.75	
1mrt VRT (x,y and z)	[C]	:	-1.31	5.14	-2.84	
2mrt VRT (x,y and z) values	[C]	:	-1.31	5.15	-2.84	
Calculationpoint	[m]	:	2.00	0.50	0.50	
ana TMRT-value	[C]	:	18.76			
1mrt TMRT-value	[C]	:	18.66			
2mrt TMRT-value	[C]	:	18.66			
ana VRT (x,y and z)	[C]	:	12.67	5.63	-0.80	
1mrt VRT (x,y and z)	[C]	:	12.72	5.64	-0.84	
2mrt VRT (x,y and z) values	[C]	:	12.72	5.65	-0.82	
Calculationpoint	[m]	:	2.00	4.50	0.50	
ana TMRT-value	[C]	:	15.68			
1mrt TMRT-value	[C]	:	15.67			
2mrt TMRT-value	[C]	:	15.68			
ana VRT (x,y and z)	[C]	:	-2.78	2.37	-2.30	
1mrt VRT (x,y and z)	[C]	:	-2.77	2.33	-2.34	
2mrt VRT (x,y and z) values	[C]	:	-2.78	2.36	-2.31	
Calculationpoint	[m]	:	0.50	4.50	0.50	
ana TMRT-value	[C]	:	16.80			
1mrt TMRT-value	[C]	:	16.79			
2mrt TMRT-value	[C]	:	16.80			
ana VRT (x,y and z)	[C]	:	-1.05	0.27	-2.56	
1mrt VRT (x,y and z)	[C]	:	-1.09	0.25	-2.58	
2mrt VRT (x,y and z) values	[C]	:	-1.12	0.25	-2.53	
Calculationpoint	[m]	:	0.50	0.50	2.00	
ana TMRT-value	[C]	:	15.69			
1mrt TMRT-value	[C]	:	16.00			*
2mrt TMRT-value	[C]	:	15.65			
ana VRT (x,y and z)	[C]	:	0.94	0.68	5.36	
1mrt VRT (x,y and z)	[C]	:	-1.16	0.62	5.42	*
2mrt VRT (x,y and z) values	[C]	:	1.04	0.68	5.42	

Calculationpoint	[m]	:	2.00	0.50	2.00	
ana TMRT-value	[C]	:	14.42			
1mrt TMRT-value	[C]	:	14.36			
2mrt TMRT-value	[C]	:	14.38			
ana VRT (x,y and z)	[C]	:	-5.84	0.76	1.76	
1mrt VRT (x,y and z)	[C]	:	-5.81	0.67	1.75	*
2mrt VRT (x,y and z) values	[C]	:	-5.90	0.75	1.72	
Calculationpoint	[m]	:	2.00	4.50	2.00	
ana TMRT-value	[C]	:	15.81			
1mrt TMRT-value	[C]	:	16.13			*
2mrt TMRT-value	[C]	:	15.80			
ana VRT (x,y and z)	[C]	:	0.73	-1.23	5.71	
1mrt VRT (x,y and z)	[C]	:	0.79	0.91	5.75	*
2mrt VRT (x,y and z) values	[C]	:	0.66	-1.27	5.70	
Calculationpoint	[m]	:	0.50	4.50	2.00	
ana TMRT-value	[C]	:	15.62			
1mrt TMRT-value	[C]	:	16.28			*
2mrt TMRT-value	[C]	:	15.60			
ana VRT (x,y and z)	[C]	:	0.25	-0.97	5.93	
1mrt VRT (x,y and z)	[C]	:	-1.86	1.20	5.97	*
2mrt VRT (x,y and z) values	[C]	:	0.29	-1.01	5.90	
Calculationpoint	[m]	:	1.25	2.50	1.25	
ana TMRT-value	[C]	:	16.81			
1mrt TMRT-value	[C]	:	16.80			
2mrt TMRT-value	[C]	:	16.80			
ana VRT (x,y and z)	[C]	:	3.37	0.00	-0.84	
1mrt VRT (x,y and z)	[C]	:	3.34	-0.04	-0.80	
2mrt VRT (x,y and z) values	[C]	:	3.33	-0.05	-0.79	

## BODY

-----

4 bubbles per surface

12 rings per bubble

Calculationpoint	[m]	:	1.25	2.50	1.00
ana TMRT-value	[C]	:	16.84		
1mrt TMRT-value	[C]	:	16.78		
2mrt TMRT-value	[C]	:	16.78		
ana VRT (x,y and z) values	[C]	:	3.24	-0.34	1.39
1mrt VRT (x,y and z) values	[C]	:	3.31	-0.28	1.38
2mrt VRT (x,y and z) values	[C]	:	3.31	-0.28	1.38

16 bubbles per surface

25 rings per bubble

Calculationpoint	[m]	:	1.25	2.50	1.00
ana TMRT-value	[C]	:	16.84		
1mrt TMRT-value	[C]	:	16.83		
2mrt TMRT-value	[C]	:	16.83		
ana VRT (x,y and z) values	[C]	:	3.24	-0.34	1.39
1mrt VRT (x,y and z) values	[C]	:	3.27	-0.35	1.39
2mrt VRT (x,y and z) values	[C]	:	3.27	-0.35	1.39

### 7.2.2. Tmrt ; tests with the analytical viewfactor calculation.

CUBE

Calculationpoint	[m]	:	0.50	0.50	0.50
ana TMRT-value	[C]	:	15.61		
ANA TMRT-value	[C]	:	15.59		
2mrt TMRT-value	[C]	:	15.58		
ana VRT (x,y and z)	[C]	:	-1.25	5.24	-2.75
ANA VRT (x,y and z) values	[C]	:	-1.23	5.23	-2.75
2mrt VRT (x,y and z) values	[C]	:	-1.31	5.15	-2.84
Calculationpoint	[m]	:	2.00	0.50	0.50
ana TMRT-value	[C]	:	18.76		
ANA TMRT-value	[C]	:	18.68		
2mrt TMRT-value	[C]	:	18.66		
ana VRT (x,y and z)	[C]	:	12.67	5.63	-0.80
ANA VRT (x,y and z) values	[C]	:	12.63	5.63	-0.78
2mrt VRT (x,y and z) values	[C]	:	12.72	5.65	-0.82
Calculationpoint	[m]	:	2.00	4.50	0.50
ana TMRT-value	[C]	:	15.68		
ANA TMRT-value	[C]	:	15.68		
2mrt TMRT-value	[C]	:	15.68		
ana VRT (x,y and z)	[C]	:	-2.78	2.37	-2.30
ANA VRT (x,y and z) values	[C]	:	-2.75	2.36	-2.26
2mrt VRT (x,y and z) values	[C]	:	-2.78	2.36	-2.31
Calculationpoint	[m]	:	0.50	4.50	0.50
ana TMRT-value	[C]	:	16.80		
ANA TMRT-value	[C]	:	16.80		
2mrt TMRT-value	[C]	:	16.80		
ana VRT (x,y and z)	[C]	:	-1.05	0.27	-2.56
ANA VRT (x,y and z) values	[C]	:	-1.06	0.26	-2.56
2mrt VRT (x,y and z) values	[C]	:	-1.12	0.25	-2.53
Calculationpoint	[m]	:	0.50	0.50	2.00
ana TMRT-value	[C]	:	15.69		
ANA TMRT-value	[C]	:	15.67		
2mrt TMRT-value	[C]	:	15.65		
ana VRT (x,y and z)	[C]	:	0.94	0.68	5.36
ANA VRT (x,y and z) values	[C]	:	0.92	0.67	5.36
2mrt VRT (x,y and z) values	[C]	:	1.04	0.68	5.42
Calculationpoint	[m]	:	2.00	0.50	2.00
ana TMRT-value	[C]	:	14.42		
ANA TMRT-value	[C]	:	14.38		
2mrt TMRT-value	[C]	:	14.38		
ana VRT (x,y and z)	[C]	:	-5.84	0.76	1.76
ANA VRT (x,y and z) values	[C]	:	-5.81	0.75	1.77
2mrt VRT (x,y and z) values	[C]	:	-5.90	0.75	1.72

Calculationpoint	[m]	:	2.00	4.50	2.00
ana TMRT-value	[C]	:	15.81		
ANA TMRT-value	[C]	:	15.79		
2mrt TMRT-value	[C]	:	15.80		
ana VRT (x,y and z)	[C]	:	0.73	-1.23	5.71
ANA VRT (x,y and z) values	[C]	:	0.74	-1.24	5.70
2mrt VRT (x,y and z) values	[C]	:	0.66	-1.27	5.70

Calculationpoint	[m]	:	0.50	4.50	2.00
ana TMRT-value	[C]	:	15.62		
ANA TMRT-value	[C]	:	15.60		
2mrt TMRT-value	[C]	:	15.60		
ana VRT (x,y and z)	[C]	:	0.25	-0.97	5.93
ANA VRT (x,y and z) values	[C]	:	0.25	-0.95	5.92
2mrt VRT (x,y and z) values	[C]	:	0.29	-1.01	5.90

Calculationpoint	[m]	:	1.25	2.50	1.25
ana TMRT-value	[C]	:	16.81		
ANA TMRT-value	[C]	:	16.79		
2mrt TMRT-value	[C]	:	16.80		
ana VRT (x,y and z)	[C]	:	3.37	0.00	-0.84
ANA VRT (x,y and z) values	[C]	:	3.37	0.00	-0.85
2mrt VRT (x,y and z) values	[C]	:	3.33	-0.05	-0.79

## BODY

-----

4 bubbles per surface  
12 rings per bubble

Calculationpoint	[m]	:	1.25	2.50	1.00
ana TMRT-value	[C]	:	16.84		
ANA TMRT-value	[C]	:	16.83		
2mrt TMRT-value	[C]	:	16.78		
ana VRT (x,y and z) values	[C]	:	3.24	-0.34	1.39
ANA VRT (x,y and z) values	[C]	:	3.23	-0.34	1.40
2mrt VRT (x,y and z) values	[C]	:	3.31	-0.28	1.38

16 bubbles per surface  
25 rings per bubble

Calculationpoint	[m]	:	1.25	2.50	1.00
ana TMRT-value	[C]	:	16.84		
ANA TMRT-value	[C]	:	16.83		
2mrt TMRT-value	[C]	:	16.83		
ana VRT (x,y and z) values	[C]	:	3.24	-0.34	1.39
ANA VRT (x,y and z) values	[C]	:	3.23	-0.34	1.40
2mrt VRT (x,y and z) values	[C]	:	3.27	-0.35	1.39

### 7.3. The input Geometry files

lmrt geometry

GEN

```

      34,      17,      0.000
0.0000, 0.0000, 0.0000,
2.5000, 0.0000, 0.0000,
2.5000, 5.0000, 0.0000,
0.0000, 5.0000, 0.0000,
0.0000, 0.0000, 2.5000,
2.5000, 0.0000, 2.5000,
2.5000, 5.0000, 2.5000,
0.0000, 5.0000, 2.5000,
0.5000, 0.0000, 0.0000,
1.0000, 0.0000, 0.0000,
2.0000, 0.0000, 0.0000,
2.0000, 0.0000, 2.0000,
1.0000, 0.0000, 2.0000,
0.5000, 0.0000, 2.0000,
0.0000, 0.5000, 1.0000,
0.0000, 2.5000, 1.0000,
0.0000, 2.5000, 2.0000,
0.0000, 0.5000, 2.0000,
0.0000, 3.5000, 1.0000,
0.0000, 4.5000, 1.0000,
0.0000, 4.5000, 2.0000,
0.0000, 3.5000, 2.0000,
0.5000, 5.0000, 1.0000,
2.0000, 5.0000, 1.0000,
2.0000, 5.0000, 2.0000,
0.5000, 5.0000, 2.0000,
2.5000, 4.5000, 0.0000,
2.5000, 3.5000, 0.0000,
2.5000, 3.5000, 2.0000,
2.5000, 4.5000, 2.0000,
2.5000, 2.5000, 0.0000,
2.5000, 2.5000, 1.5000,
2.5000, 2.5000, 2.5000,
2.5000, 0.0000, 1.5000,
4, 3, 4, 8, 7,
4, 4, 1, 5, 8,
10, 1, 9, 14, 13, 12, 11, 2, 34, 6, 5,
9, 31, 28, 29, 30, 27, 3, 7, 33, 32,
10, 4, 3, 27, 28, 31, 2, 11, 10, 9, 1,
5, 5, 6, 33, 7, 8,
4, 26, 25, 24, 23,
4, 18, 17, 16, 15,
4, 9, 10, 13, 14,
4, 28, 27, 30, 29,
4, 22, 21, 20, 19,
4, 10, 11, 12, 13,
4, 2, 31, 32, 34,
4, 34, 32, 33, 6,
4, 15, 16, 17, 18,
4, 19, 20, 21, 22,

```

```

4, 23, 24, 25, 26,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 9, 0, 0

```

-----

2mrt geometry

GEN

```

      44,      25,      0.000
0.0000, 0.0000, 0.0000,
2.5000, 0.0000, 0.0000,
2.5000, 5.0000, 0.0000,
0.0000, 5.0000, 0.0000,
0.0000, 0.0000, 2.5000,
2.5000, 0.0000, 2.5000,
2.5000, 5.0000, 2.5000,
0.0000, 5.0000, 2.5000,
0.5000, 0.0000, 0.0000,
1.0000, 0.0000, 0.0000,
2.0000, 0.0000, 0.0000,
2.0000, 0.0000, 2.0000,
1.0000, 0.0000, 2.0000,
0.5000, 0.0000, 2.0000,
2.0000, 0.0000, 2.5000,
0.5000, 0.0000, 2.5000,
0.0000, 0.0000, 1.0000,
0.0000, 0.5000, 1.0000,
0.0000, 2.5000, 1.0000,
0.0000, 3.5000, 1.0000,
0.0000, 4.5000, 1.0000,
0.0000, 5.0000, 1.0000,
0.0000, 0.0000, 2.0000,
0.0000, 0.5000, 2.0000,
0.0000, 2.5000, 2.0000,
0.0000, 3.5000, 2.0000,
0.0000, 4.5000, 2.0000,
0.0000, 5.0000, 2.0000,
0.5000, 5.0000, 1.0000,
2.0000, 5.0000, 1.0000,
2.5000, 5.0000, 1.0000,
0.5000, 5.0000, 2.0000,
2.0000, 5.0000, 2.0000,
2.5000, 5.0000, 2.0000,
2.5000, 4.5000, 0.0000,
2.5000, 3.5000, 0.0000,
2.5000, 2.5000, 0.0000,
2.5000, 2.5000, 1.5000,
2.5000, 0.0000, 1.5000,
2.5000, 4.5000, 2.0000,

```

[illegible]

-----



**7.4. The input TMRT files**

1mrt : cubet1

```

17      Nsurf
18.0000      temperature surface :    1
18.0000      temperature surface :    2
18.0000      temperature surface :    3
18.0000      temperature surface :    4
18.0000      temperature surface :    5
20.0000      temperature surface :    6
  1.0000      temperature surface :    7
  1.0000      temperature surface :    8
  1.0000      temperature surface :    9
  6.0000      temperature surface :   10
  1.0000      temperature surface :   11
  6.0000      temperature surface :   12
30.0000      temperature surface :   13
  1.0000      temperature surface :   14
  1.0000      temperature surface :   15
  1.0000      temperature surface :   16
  1.0000      temperature surface :   17
      CUBE
0.0200      0.0200      0.0200      Dimensions cube/body
  9      Number of cubes
0.5000      0.5000      0.5000      Coordinates of cube    1
2.0000      0.5000      0.5000      Coordinates of cube    2
2.0000      4.5000      0.5000      Coordinates of cube    3
0.5000      4.5000      0.5000      Coordinates of cube    4
0.5000      0.5000      2.0000      Coordinates of cube    5
2.0000      0.5000      2.0000      Coordinates of cube    6
2.0000      4.5000      2.0000      Coordinates of cube    7
0.5000      4.5000      2.0000      Coordinates of cube    8
1.2500      2.5000      1.2500      Coordinates of cube    9

```

-----  
2mrt : cubet1

```

25      Nsurf
18.0000      temperature surface :    1
  1.0000      temperature surface :    2
  6.0000      temperature surface :    3
18.0000      temperature surface :    4
18.0000      temperature surface :    5
18.0000      temperature surface :    6
18.0000      temperature surface :    7
  1.0000      temperature surface :    8
18.0000      temperature surface :    9
  1.0000      temperature surface :   10
18.0000      temperature surface :   11
18.0000      temperature surface :   12
18.0000      temperature surface :   13
18.0000      temperature surface :   14

```

1.0000	temperature surface :	15	
18.0000	temperature surface :	16	
18.0000	temperature surface :	17	
18.0000	temperature surface :	18	
6.0000	temperature surface :	19	
18.0000	temperature surface :	20	
18.0000	temperature surface :	21	
30.0000	temperature surface :	22	
1.0000	temperature surface :	23	
20.0000	temperature surface :	24	
18.0000	temperature surface :	25	
CUBE			
0.0200	0.0200	0.0200	Dimensions cube/body
9	Number of cubes		
0.5000	0.5000	0.5000	Coordinates of cube 1
2.0000	0.5000	0.5000	Coordinates of cube 2
2.0000	4.5000	0.5000	Coordinates of cube 3
0.5000	4.5000	0.5000	Coordinates of cube 4
0.5000	0.5000	2.0000	Coordinates of cube 5
2.0000	0.5000	2.0000	Coordinates of cube 6
2.0000	4.5000	2.0000	Coordinates of cube 7
0.5000	4.5000	2.0000	Coordinates of cube 8
1.2500	2.5000	1.2500	Coordinates of cube 9

-----  
1mrt :bodyt1

17	Nsurf		
18.0000	temperature surface :	1	
18.0000	temperature surface :	2	
18.0000	temperature surface :	3	
18.0000	temperature surface :	4	
18.0000	temperature surface :	5	
20.0000	temperature surface :	6	
1.0000	temperature surface :	7	
1.0000	temperature surface :	8	
1.0000	temperature surface :	9	
6.0000	temperature surface :	10	
1.0000	temperature surface :	11	
6.0000	temperature surface :	12	
30.0000	temperature surface :	13	
1.0000	temperature surface :	14	
1.0000	temperature surface :	15	
1.0000	temperature surface :	16	
1.0000	temperature surface :	17	
BODY			
0.2000	0.4000	1.8000	Dimensions cube/body
1	Number of cubes		
1.2500	2.5000	1.0000	Coordinates of cube 1

-----  
2mrt: bodyt1

25 Nsurf

18.0000	temperature surface :	1	
1.0000	temperature surface :	2	
6.0000	temperature surface :	3	
18.0000	temperature surface :	4	
18.0000	temperature surface :	5	
18.0000	temperature surface :	6	
18.0000	temperature surface :	7	
1.0000	temperature surface :	8	
18.0000	temperature surface :	9	
1.0000	temperature surface :	10	
18.0000	temperature surface :	11	
18.0000	temperature surface :	12	
18.0000	temperature surface :	13	
18.0000	temperature surface :	14	
1.0000	temperature surface :	15	
18.0000	temperature surface :	16	
18.0000	temperature surface :	17	
18.0000	temperature surface :	18	
6.0000	temperature surface :	19	
18.0000	temperature surface :	20	
18.0000	temperature surface :	21	
30.0000	temperature surface :	22	
1.0000	temperature surface :	23	
20.0000	temperature surface :	24	
18.0000	temperature surface :	25	
BODY			
0.2000	0.4000	1.8000	Dimensions cube/body
1	Number of cubes		
1.2500	2.5000	1.0000	Coordinates of cube 1