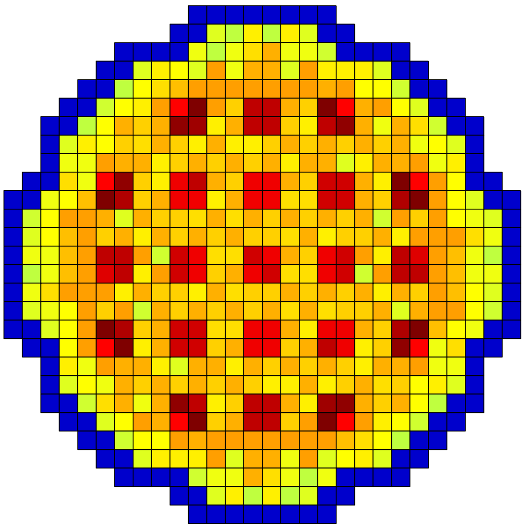
RESEARCH REPORT

VTT-R-01256-09



Computation package of nodal reactor physics constants for TRAB-3D

Authors:	Karin Rantamäki
Confidentiality:	Public

Report's title Computation package of nodal reactor physics constants for TRAB-3D	
Customer, contact person, address H. Räty, VTT	Order reference
Project name Vaikutusalojen luonti ja sovitus TRAB-3Dlle/TRICOT/TOPAS	Project number/Short name 32271 / 23857 / 23790
Author(s) Karin Rantamäki	Pages 57/
Keywords SIMULATE-3, TRAB-3D, nodal parameters	Report identification code VTT-R-01256-09

Summary

The transient analyses at VTT are done by TRAB-3D. However, the cross sections and other reactor physics constants need to be obtained from outside the code. They can be calculated by e.g. SIMULATE-3.

A script package to facilitate the use of cross sections generated by SIMULATE-3 in TRAB-3D has been written. It contains a number of scripts that are collected under the umbrella of SimtoTrab.pl, which is the main program of the package. The package takes care of running SIMULATE-3 and provides TRAB-3D with the relevant data. For each power level 5 files are produced: two files containing the material constants, two files containing the Xe and Sm concentrations and a file that contains some neutronics related input cards for TRAB-3D.

This report describes the scripts that have been written for this package as well as the changes made to the SIMULATE-3 code. It also gives the instructions to set up the package based on data received from TVO. Finally, a detailed description of the input files is provided.

Confidentiality	Public	
Espoo 9.3.2009 Written by Karin Rantamäki Karin Rantamäki, research scientist	Reviewed by [illegible] Markku Anttila senior research scientist	Accepted by [illegible] Timo Vanttola technology manager
VTT's contact address		
Distribution (customer and VTT) VTT Hanna Räty, Anitta Hämäläinen STUK Riku Mattila TVO Mikael Solala		

Preface

This work was done at VTT as part of the transient analysis work. It was partly funded by the National Research Programme SAFIR through it two subtasks TOPAS and TRICOT.

I wish to thank the reactor physics group of TVO, especially M. Paajanen and R. Höglund, for their patience and help during the beginning of this work. The discussions and advice of M. Paajanen during the course of this work are much appreciated. I am indebted to the operation department of TVO and especially, Simo Verta, for providing me the start-up sequence as an input file for SIMULATE-3. This work would not have been possible without the help of the two groups of TVO.

Thanks also to my colleagues at VTT who have answered my questions on reactor physics, TRAB-3D and HEXBU.

Espoo 9.3.2009

Karin Rantamäki

Karin Rantamäki

Contents

Preface ... 3
1 Introduction ... 6
2 Description of the script package ... 7
2.1 SimtoTrab.pl ... 7
2.2 sim_XS_extract.pl ... 8
2.3 sim_DEN_extract.pl ... 10
2.4 sim_KIN_extract.pl ... 10
2.5 sim_FUEmass.pl ... 11
2.6 sim2trab-kortti.pl ... 11
2.7 Trab3DNeutr.pl ... 13
3 Setting up the work ... 17
3.1 Files needed for initialisation ... 17
3.2 Input files for cross section generation ... 18
4 Summary ... 20
References ... 21
Attachment A Scripts ... 22
Attachment B Input files ... 46

1 Introduction

At VTT transient analyses are widely performed by TRAB-3D [1]. However, it does not produce cross sections internally but needs to get them from outside. On the other hand a widely used software that is able to create the cross sections is the Advanced Three-Dimensional Two-Group Reactor Analysis Code SIMULATE-3 [2].

In this work, a script package has been written to facilitate the use of cross sections and other reactor physics parameters generated by SIMULATE-3 in TRAB-3D. The package contains number of scripts that are collected together and run through SimtoTrab.pl, which is the main program of the package. The other scripts are mainly written to extract the desired data from the SIMULATE-3 output files. One script is used to write the reactor physics material constants into a file in the HEXBU-3D [3,4] format. In addition, a script was written that writes part of the TRAB-3D input file.

Some work is needed to set the system up once the SIMULATE-3 files are obtained from the power company. However, after that this script package makes it easy to calculate the needed parameters. Moreover, the produced files are directly in a format the can be read by TRAB-3D. Thus the need of hand work is considerably reduced.

This report describes the script package for computing the cross sections written at VTT. The package contains a number of scripts that are used to calculate the cross sections and to write them into files used as input for TRAB-3D. These scripts are described in Section 2. Section 3 explains the initialisation of the package including the files that should be obtained from the utility, in this case TVO, and the input files needed to run the package.

2 Description of the script package

This section describes the scripts that are used to get the input files for TRAB-3D [1]. The package contains a number of scripts that are used to run the SIMULATE-3 code [2] or to extract data from the output files. These scripts are listed in Attachment A. SIMULATE-3 is used to obtain the cross sections. Input files and initialisation procedure for the package will be described in the next section. Modifications made to SIMULATE-3 are described in Reference [5].

2.1 SimtoTrab.pl

The main program of the package is SimtoTrab.pl, which can be seen in Table 1. It takes care of running SIMULATE-3 and of extracting data from the output files to separate files. No input file is needed but an optional flag '-k' may be given to the main script. When this flag is used old files and directories are removed.

First, a few variables for the names of input files of various cases are defined. These variables $base, @Pcases, and @Dcases have to be defined separately for each application. The array @Rcases will be as given for all transient calculations done with TRAB-3D. It defines the two control rod cases: an unrodded (aro or all rods out) and a rodded one (ari or all rods in). The variable $base contains the base for the name. It can be arbitrarily chosen but the names of the input files must begin with this. The power fractions at which the calculations are to be done are given in the array @Pcases. The power fraction will be attached to the base name with a dash when creating the input file name. The various coefficient cases are given in the array @Dcases. The first item is the base case. The content of the array elements is attached to the input file name directly so the elements for the actual coefficient cases must contain the dash, if the coefficient case is separated by a dash in the input file name. Thus the input file name for a 60% power case with all rods withdrawn is for the base case trans-60-aro.inp. For the first actual coefficient case with temperature increased by 50 K the name would be trans-60-aro–T+50.inp.

Next, a directory ToTrab3D is created. All the files created by this package for the use in TRAB-3D are moved or copied to this directory so that they can be found in one place. Before moving to the actual computation, the old files and directories are removed if the optional flag for this is given.

For each power case a directory is created named according to the content of the array @Pcases preceded by a P, i.e. P60 for the 60% power case. A file fue.def is copied to the directory. This file is used to define whether a position with (x, y) indices (i, j) in a plane contains a fuel bundle. The appropriate input files for all the subcases are also linked to the directory. The SIMULATE-3 run is performed for both control rod cases and all the coefficient cases. Next, the parameters needed to write the input files for TRAB-3D are extracted from the SIMULATE-3 output file into their own files. The scripts written for this purpose will be described in sections 2.2 to 2.4.

After the coefficient cases the uranium mass is calculated for both the rodded and the unrodded cases. The script sim_FUEmass.pl written for this purpose will be described in section 2.5. Then all the data exists for writing the material constants for the cross sections. The script sim2trab-kortti.pl described in section 2.6 writes the data in HEXBU-3D format [3,4].

The two files are then copied to the directory ToTrab3D as well as the files containing the Xenon and Samarium concentrations. In this work, the concentrations are assumed to be constant for all subcases performed with the same power. Therefore, only one file per power case is copied for each nucleus. Finally, the script Trab3DNeutr.pl is run that writes part of the input file needed for TRAB-3D and the file is moved to the directory ToTrab3D.

Table 1: Main programme SimtoTrab.pl

```
#! /usr/local/bin/perl
use strict;
use warnings;
#use Links;
#my $path=$Links::inputpath;
my $fl;

if ($#ARGV <0) {$fl="";}
else {$fl=$ARGV[0];}        # SIMULATElle annettava flagi esim -k

my $base="trans";
my @Pcases=("60","80","100");
my @Rcases=("aro","ari");
my @Dcases=("","-T+50","-T-50","-30n","-60n");

`mkdir ToTrab3D`;
if ($fl=~ /k/) {            # jos annettu flagi k, poistetaan vanhat tiedostot
    `rm -f ToTrab3D/*`;     # hakemistosta ToTrab3D
    `rm -f P*/*`;           # tehohakemistoista P*
    `rmdir P*`;             # poistetaan myös tehohakemistot
}
foreach my $Pcase (@Pcases)
{
    `mkdir P$Pcase`;
    chdir "P$Pcase";
    system("pwd");
    `cp ../fue.def .`;
    foreach my $Rcase (@Rcases) {
        foreach my $dcase (@Dcases) {
            `ln -s  ../$base-$Pcase-$Rcase$dcase.inp .`;
            system("sim3_kmr $fl $base-$Pcase-$Rcase$dcase.inp");
            system("sim_XS_extract.pl $base-$Pcase-$Rcase$dcase.inp MAC TOT");
            system("sim_XS_extract.pl $base-$Pcase-$Rcase$dcase.inp DFS TOT");
            system("sim_DEN_extract.pl $base-$Pcase-$Rcase$dcase.inp 3DEN TOT");
            system("sim_DEN_extract.pl $base-$Pcase-$Rcase$dcase.inp 3TFU TOT");
            system("sim_DEN_extract.pl $base-$Pcase-$Rcase$dcase.inp 3SAM TRB");
            system("sim_DEN_extract.pl $base-$Pcase-$Rcase$dcase.inp 3XEN TRB");
            system("sim_DEN_extract.pl $base-$Pcase-$Rcase$dcase.inp 3EXP TOT");
            system("sim_KIN_extract.pl $base-$Pcase-$Rcase$dcase.inp");
#           print "\n";
        } # end foreach Dcase
        system("sim_FUEmass.pl $base-$Pcase-$Rcase.inp");
        system("sim2trab-kortti.pl $base-$Pcase-$Rcase.inp");
    } # end foreach Rcase (aro,ari)
    `cp *.ntr ../ToTrab3D`;
    `cp $base-$Pcase-$Rcases[0]_3SAM.trb ../ToTrab3D/$base-$Pcase-SAM.trb`;
    `cp $base-$Pcase-$Rcases[0]_3XEN.trb ../ToTrab3D/$base-$Pcase-XEN.trb`;
    chdir "../";
    system("Trab3DNeutr.pl $base-$Pcase.inp");
    `mv *.ntr ToTrab3D`;
    system("pwd");
} # end foreach Pcase
```

2.2 sim_XS_extract.pl

This script is used to extract the cross section data out of the SIMULATE-3 output file. It takes three input parameters. The first one is the SIMULATE-3 input file, which is used to define the output file to be read. Next, is given the parameter that is to be extracted, i.e. MAC for the cross sections or DFS for discontinuity factors. The names are directly the ones used in SIMULATE-3. The third parameter defines the output format of the script: ‘TOT’ writes all the data into one file in the format K, J, I, XS[K][J][I], where I, J, K are the x, y, z indices, and ‘MAP’ writes the data as maps into separate files for each node K in the z-direction. The indexing of the plane is shown in Figure 1.

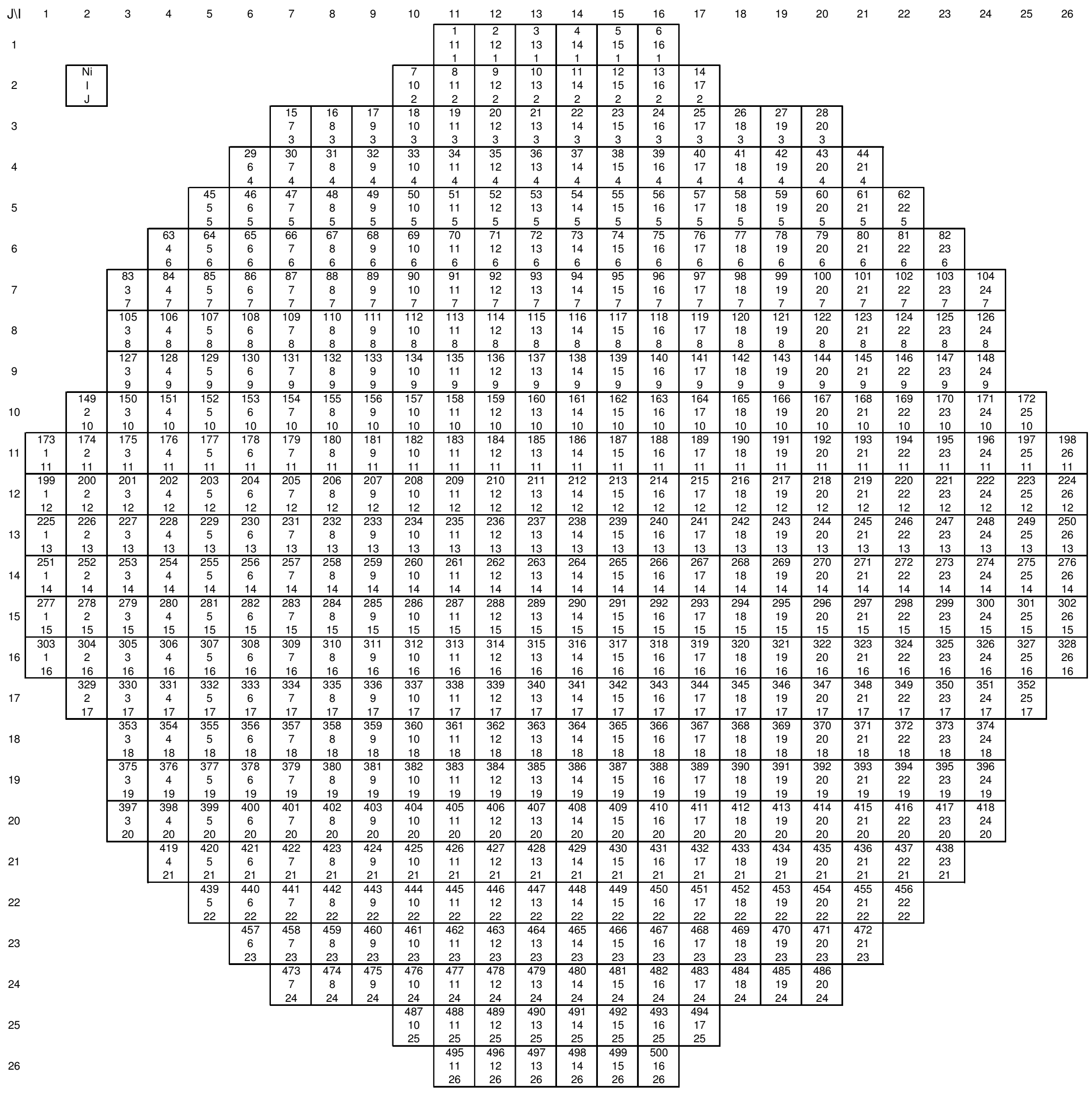


Figure 1: The indexing of the core plane. Both the running index of each assembly Ni, and the indices I and J are shown. I is running from left to right and J from top to bottom in this plane.

First, the file fue.def is read in order to get the variable @rfue which contains the information whether a given position (I,J,K) is fuelled or not. This data is needed later when the cross sections are written to the output file.

Next, the location is searched for where the data to be extracted starts in the SIMULATE-3 output file. The multiplication factor is extracted from the title row if it exists. Otherwise it is set to one. Also the K index is obtained from the title row. The I-indices are read from the following row.

Then, the data itself can be read. The first item on the row is always the J-coordinate. Since the data in the SIMULATE-3 output file is in map type format the left and right side data has to be handled separately. The data is read in a 'foreach' loop running from 1 to NJ, which has to be defined at the beginning of the file for each application.

Once the data is read for this level the name of the cross section or discontinuity factor is stored into the variable @XSname and its value is stored into the hash variable %XStot with the key being the cross section name. Since the output is written for each K level if the option 'MAP' is used, the writing is performed here while reading the file.

With the output option 'TOT', all the data is written to one single file. Therefore, it can be done separately at the end of the file. If the location contains a fuel bundle, the cross section value is written otherwise the script writes a zero to this location. The script also writes to standard output the content of the array variable @XSname. Thus, the user can check e.g. in which order the data is written since all the values are written after each other into columns.

2.3 sim_DEN_extract.pl

The script sim_DEN_extract.pl is analogous to the previous one but it is used to extract the state parameters. As the previous one, this script takes three input parameters: the SIMULATE-3 input file, the state parameter to be read and the output format. The state parameter can be any of the 3D parameters listed in the SIMULATE-3 manual for the PRI.STA card. The output options are the same as for sim_XS_extract.pl, but there is an additional option 'TRB'. This is used for Xenon and Samarium concentrations to write them to files that can be read directly by TRAB-3D.

Since the data in this case is in 3D format and not as maps, it can be read completely before writing it out. There are two different options for the SIMULATE-3 data. The parameters starting with a '3' are written by SIMULATE-3 on an assembly basis and the variables starting with 'N' on a nodal basis and contain the reflector area. There are small differences in the SIMULATE-3 output of these two cases and therefore the reading of the I and J indices has to be treated separately for the two cases. However, the actual data is in a similar format. For each line, the first item is the K index. Then the actual data is read. Again the left half is treated first and only after that the right half of the reactor is read.

Once the data is read, the printing part of the script starts. There are three actual parts for the writing of the data and an additional one for the parameter NTFU. The last one is used to create the file fue.def containing the indices of the fuelled nodes. There are only slight differences in the writing of the output files of the various cases. The main difference is in the naming of the files, which is done according to the output option. The output filename contains the variable name, which is added to the end of the SIMULATE-3 input filename before the extension. Moreover, in the case of 'MAP' an output file is created for each K value. The difference between 'TOT' and 'TRB' is basically in the order in which the data is written.

2.4 sim_KIN_extract.pl

The third analogous script is sim_KIN_extract.pl that is used to extract the 3D kinetic constants from the SIMULATE-3 output file *.k3d. These parameters cover βs and λs for each six group of the delayed neutrons as well as 1/v for the fast and thermal neutrons. Since this script is written for such a specific use it only needs the SIMULATE-3 input file as an input parameter.

As in sim_XS_extract.pl, first the file fue.def is read. Then the actual data is read. Each data line starts with the indices K, J and I. The constants have been defined in variable @XSname and data is stored only for the fuelled nodes. The parameters are stored in a hash variable %XStot. Then as many parameters are read as there are on the line. The number of groups is stored in a variable @IGMa for each parameter. The variable @IGMi gives the item where to start, i.e. for β and λ it is 1 but for 1/v it should be 3 since there are two other parameters on the line before the ones that are actually needed.

Then finally the data can be written to the output file. The name of the constant is attached to the filename based on the SIMULATE-3 input filename. For unfuelled locations, only zeros are written, as before. The maximum indices for I, J, K are hard coded and should be changed if the dimensions are different. However, one should note that K=0 and K=26 (in this case) are reflector zones as well as I, J = 1 and I, J = 28.

2.5 sim_FUEmass.pl

The script sim_FUEmass.pl is used to produce the axial fuel mass for each node. The mass in g/cm is written to the file *.fue, which is based on the SIMULATE-3 input filename given as input parameter to the script. The computation is based on SIMULATE-3 output files *.out and *.sum as well as a file fuel.inc that is linked to the SIMULATE-3 calculation. This file contains the definitions of the fuel and is obtained from TVO. Note that this file is highly sensitive to the application and the path name where it is to be found needs to be updated each time.

First, the assembly pitch, node height and core height are read from the file *.out. Next the fuelled segments are read. For each fuel segment the segment number is stored as well as the mass (g/cm^3) of the segment. The mass of the segment in the desired unit g/cm is obtained by multiplying it by the square of the assembly pitch and stored in the variable @segmass for each segment.

The fuel design is read from the file fuel.inc from cards called FUE.ZON. After skipping the line containing the radial reflector, the actual fuel bundles can be read. The fuel type is first read. After skipping a few parameters on the line the fuel design is obtained. For each segment in the bundle, the starting point is read and converted into nodal index. For nodes with just one segment, the mass can be stored directly from the segment mass. However, for the nodes that are at the boundary and have two segments, the mass is defined as linear weighting of the masses of the two segments. Thus the mass for each fuel type and node is stored in the variable @massa.

Once we have the fuel masses defined, the locations of the various fuel types are needed. This is obtained from the file *.sum from the card FUE.TYP. The first number on each line is the J coordinate and then come the fuel types. The reflector is denoted by ‘1’ and the unfuelled positions by ‘0’.

Now, the data can be written into the output file. The filename is the same as the input file with the extension ‘.inp’ changed into ‘.fue’. Note again that the maximum values for the indices are hard coded.

2.6 sim2trab-kortti.pl

The script sim2trab-kortti.pl is called only for the power and control rod cases, just like the script for the fuel mass. The purpose of this script is to write the material constants in

HEXBU-3D format. The input parameter is the SIMULATE-3 input file as for all the other scripts and the constants are written to a file with the extension 'ntr'.

The coefficient calculations have to be defined at the beginning of the script. The array @Dcases contains the density coefficient cases and the array @Tcases the same for the temperature. Three other arrays are also defined at the beginning of the script. These are the array @params that contains the parameters that are needed in writing the cards. The parameter names are those used in SIMULATE-3. The array @XSname contains the names of the cross sections in the order that SIMULATE-3 treats them and @DFname contains those of the discontinuity factors.

Within the variable definition also minimum and maximum indices have to be defined. Note, that at this stage the set is defined as to comprise the upper half of the reactor plane, i.e. minimum values $Kmin=1, $Jmin=1 and $Imin=1, and maximum values $Kmax=25, $Jmax=13 and $Imax=26. If the whole reactor is desired then these values have to be changed.

To start with, the job title is read from the output file using the subroutine HaeOtsikko. The title is taken from what is given in the TIT.PRO card in the SIMULATE-3 run. The nominal power, coolant temperature and density are obtained using the subroutine FluidParams. The subroutine reads the output file and looks for 'Thermal Power', 'Core Loading' and 'Saturated fluid'. The nominal power is given by Thermal Power/Core mass, and the density and temperature are read directly from the output file.

Then, the base case parameters for each parameter defined in the array @params are read from the appropriate files *.tot. These files were obtained by the scripts described in sections 2.2 to 2.4. To facilitate the reading of these files and to make the script easier to read, a subroutine LueTaulu was written in the package Apu. The subroutines written for this package are listed in Attachment A.7:. This package must be included in the script. The script returns an array with indices (K,J,I) and a possible group index if needed.

Next, the parameters are read for coefficient calculations. The temperature coefficients are treated first and then the density coefficients. In both cases the parameter itself is obtained first and then the cross section. Again the package Apu::LueTaulu is used to read the arrays.

For the FITCO and NOMCOM cards a few parameters need to be defined. $NXE and $NSM are the numbers of xenon and samarium parameters. In this case they are both set to zero. Subsequently, the elements for the NCOEF-matrix are given. The seven elements per row are for cross sections D_1, Σ_{a1}, $\nu\Sigma_{f1}$, Σ_{12}, D_2, Σ_{a2}, and $\nu\Sigma_{f2}$. Here the subscripts 1 and 2 denote the fast and thermal group, a stands for absorption and f for fission. The rows, or in this case the variables, represent the values for Doppler effect i.e. T_{fuel}, boron density, combined coolant density and temperature effect, coolant density, coolant temperature and buckling effect. $KSM and $KXE indicate the xenon and samarium in the nominal cross sections. They are set to zero as to zero concentration. $CB is the boron concentration, $alfa is the volume fraction of water outside shroud, $modc indicates the type of coolant fitting. Zero means that the fitting corresponds to water inside shroud. $nhet is the number of discontinuity factors, which in this case is eight. Note, that these values, both for the FITCO and the NOMCOM cards, may need some adjustment depending on the application.

Now all the data exists so that the material constant cards can be written. The cards are written for each node in the z-direction, which also defines the first numbers in the material number. The material number is defined as K×1000+2n, where K is the index of the node in z-direction and n is the running number of the fuel bundle, see Figure 1. The ordering of the fuel bundles is from left to right and from top to down in a plane. Moreover, the rodded cases have an even material number and the unrodded ones have an odd one.

For all the other cards but CASE and MATER, which only have a few parameters, a subroutine is written to take care of the writing. All other subroutines but the one for the FITCO card get the output file handle and the three indices as parameter. FITCO only gets the file handle.

The subroutines basically just write the data out. However, to print out the DIFCON card, the order of the cross sections has to be adjusted, since SIMULATE-3 and HEXBU-3D treat them in a different order. The order is set in the variables @XSout1 and @XSout2. Note, that since SIMULATE-3 does not provide the energy per fission, i.e. κ in SIMULATE-3 language or ε in HEXBU language, a trick is needed. In this case, $\nu\Sigma_f$ is written twice so that it is used instead of Σ_f. In addition, instead of κ or ε, κ/ν is written. The modifications to handle this change needs to be incorporated in TRAB-3D. The ordering of the discontinuity factors is the same in SIMULATE-3 and TRAB-3D. Finally also the kinetic parameters are written to the card.

The subroutine printDOPLER writes the DOPLER card for the fuel temperature coefficients. The coefficients are calculated with the help of an additional subroutine kerroin of Apu package, see Attachment A.7:. The subroutine kerroin just calculates the coefficients c_1 and c_2 of a pair of second order polynomials $\Delta y= c_1\ \Delta x + c_2\ (\Delta x)^2$, when the Δys and Δxs are given. For the Doppler coefficients the Δx is calculated from the square root of the temperature as

$$\Delta x^{1,2} = \sqrt{T_f^{1,2}} - \sqrt{T_{nom}}\ , \qquad (1)$$

where T_f is the fuel temperature, the superscripts 1 and 2 denote the coefficient calculations, and the subscript nom denotes the nominal or base case. Correspondingly,

$$\Delta y^{1,2} = \Sigma_g^{1,2} - \Sigma_{g,nom}\ , \qquad (2)$$

where the superscripts 1 and 2, and the subscript nom are as previously, and the subscript g indicates which cross section is dealt with.

The subroutine printMODER is similar to printDOPLER but writes the MODER card for the coolant temperature coefficients. The fitting coefficients are calculated again with the help of the Apu::kerroin subroutine. Now, Δx is calculated directly from the density ρ as

$$\Delta x^{1,2} = \rho^{1,2} - \rho_{nom}\ . \qquad (3)$$

Δy is calculated as in Equation (2) apart for the diffusion coefficients D. They are fitted as inverse values as

$$\Delta y^{1,2} = \frac{1}{D^{1,2}} - \frac{1}{D_{nom}}\ , \qquad (4)$$

where again the superscripts 1 and 2 denote the coefficient calculations and the subscript nom is the nominal or base case. The superscripts 1 and 2 in Equations (1) to (4) should not be confused with the subscripts in the list of the cross sections, where they denote the fast and thermal neutron group.

2.7 Trab3DNeutr.pl

Trab3DNeutr.pl is used to write some neutronics cards used in TRAB-3D input. A summary of these cards is given in Table 2. As for the other scripts the input parameter is the

SIMULATE-3 input file. The basis for the name and the power case are extracted from the input name and used to define the output filename. The extension of the output file is '.ntr'.

Table 2: Input cards written by Trab3DNeutr.pl

OPERAT
DELAYE
LAYOUT
POSITN
RODPOS
RODCON
BOUNDR
XSDATA
ASSTYP
END4
LOAD
END5

First some parameters need to be defined. $Npin is the number of fuel assemblies in the whole reactor, $sym defines the symmetry to be used: for a whole reactor $sym=1 and for half a reactor $sym=2. Next the number of nodes in all the three directions is given. The two variable @nshift and @nass define the reactor layout. @nshift gives the starting point of the assemblies on each row, one value per row. The first value has to be zero. For the subsequent rows the value is given with respect to the previous row such that negative values are to the left and positive values to the right. @nass gives the number of assemblies on each row. The control rod groups and their locations are defined in a file given in the variable $CRDmap.

After defining the output file and reading the fuel temperatures, the control rod positions are given. For each control rod group a position has to be given. The numbers given are directly from the SIMULATE-3 input used to set up the application, such that 100 is fully withdrawn from the reactor. The data is found from the control rod maps on the CRD.POS cards. All the numbers have to be given by hand. It might be possible to write a script to take care of this process. In this version, the positions are given for the three power cases, but they have to be updated for each application. At the end of this block a control rod inventory is computed and written to standard out.

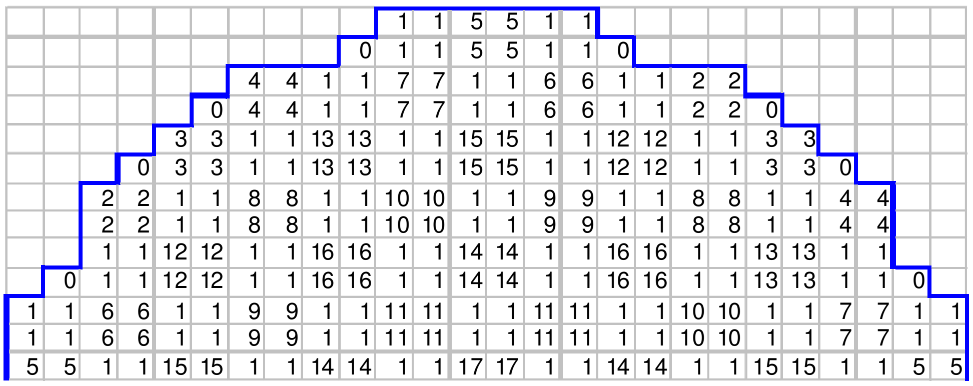
Next, the core height is read from the SIMULATE-3 output file. Then the OPERAT card is written. This card contains the insertion of each control rod group. The script also converts the rod positions from SIMULATE-3 coordinate system to metres.

The time constant of the delayed neutrons are read from the SIMULATE-3 output file for the base case with all rods withdrawn. They are found under the card KIN.0-D. The six values are then written to the DELAYE card in the output file.

The LAYOUT card contains various symmetry related variables that have to be adjusted according to the application. NSYM is the symmetry of the reactor (0=full, 1=upper half, 2 =1/4), NLAY is the location of the symmetry axis (0=between the assemblies) and NBCON is the type of the symmetry boundary. Zero is periodic and should be used also for rotational reactor symmetry. After these three parameters the position and the number of assemblies are written for each row.

The POSITN card contains the assembly positions or the numbering of their location. In this version a running number for each fuelled location is used. The numbers run from left to right and from top to bottom, see Figure 1. Thus, for a reactor with half symmetry the numbers run from 1 to 250.

The following two cards contain control rod data. The RODPOS card contains the control rod map that is read from the file in the variable $CRDmap. It is also shown in Figure 2 for this application. The RODCON card contains three variables. NCRA is the control rod type: 0 for BWR-type blades and -1 for finger type rod. HLROD is the height of the control rod, which in this case is equal to the core height. The last parameter to be written is MRROD which is the increment in material numbers from the unrodded to the rodded case. In this work, it should be one.



										1	1	5	5	1	1										
									0	1	1	5	5	1	1	0									
						4	4	1	1	7	7	1	1	6	6	1	1	2	2						
					0	4	4	1	1	7	7	1	1	6	6	1	1	2	2	0					
				3	3	1	1	13	13	1	1	15	15	1	1	12	12	1	1	3	3				
			0	3	3	1	1	13	13	1	1	15	15	1	1	12	12	1	1	3	3	0			
		2	2	1	1	8	8	1	1	10	10	1	1	9	9	1	1	8	8	1	1	4	4		
		2	2	1	1	8	8	1	1	10	10	1	1	9	9	1	1	8	8	1	1	4	4		
		1	1	12	12	1	1	16	16	1	1	14	14	1	1	16	16	1	1	13	13	1	1		
	0	1	1	12	12	1	1	16	16	1	1	14	14	1	1	16	16	1	1	13	13	1	1	0	
1	1	6	6	1	1	9	9	1	1	11	11	1	1	11	11	1	1	10	10	1	1	7	7	1	1
1	1	6	6	1	1	9	9	1	1	11	11	1	1	11	11	1	1	10	10	1	1	7	7	1	1
5	5	1	1	15	15	1	1	14	14	1	1	17	17	1	1	14	14	1	1	15	15	1	1	5	5

Figure 2: The control rod groups used in this work. Each fuel assembly is given a number as to which control rod group it sees, 0 means that this assembly is not facing any control rod. The rods that are completely out of the reactor already from the beginning of the cycle are grouped to group number 1. The core boundary is denoted by the thick blue line.

The BOUNDR card defines a virtual boundary in the reflector area. The top and bottom reflectors are given first, and then the boundary on the layout map. The top reflector is typically given the value one and the bottom reflector the value two. The boundary on the layout map is typically defined so that the corner element has the value 4 and one with a straight boundary has the value 3. This is illustrated in Figure 3.

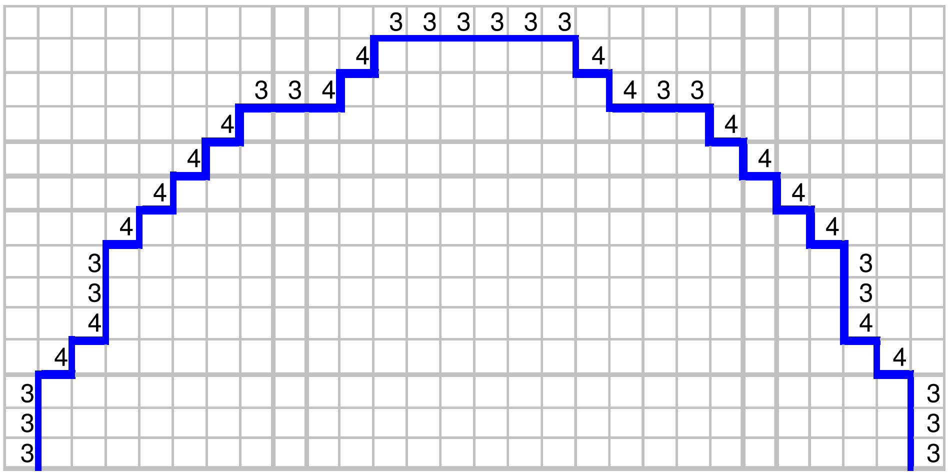


Figure 3: Numbering of the virtual boundary of BOUNDR card. The thick blue line denotes the boundary between the fuelled zone and the reflector. Only half a core is shown.

The first and the last row have to be treated separately. In addition, the rows where the boundary has several values, like the third row in Figure 3 that has the values '3 3 4', need some more attention. Currently, this block works for half-core symmetry but it has not been written or tested for a full core. The treatment of the lower part of the core needs to be added if the script is to be used for full-core symmetry.

Before writing the assembly data, the XSDATA card needs to be written. This card initiates the reading of the material constants. In this work, only one parameter NXDAT=0 is given on this card, which means that for the other parameters default values are used for all materials.

The ASSTYP card relates the material numbers to the fuel assembly numbers. First is given the assembly type number and then the material numbers for that assembly. The assembly number is a running number. The number runs from left to right and from top to bottom on the plain. The material numbers are given for all the nodes running from top to down in the axial direction. An END4 card ends this block in the input file.

Finally the LOAD card is written. It is needed to load the assemblies to their positions. The card is written for each assembly. The parameters on the card are the assembly number, the location number, as on the POSITN card, and then the assembly type number, as on the ASSTYP card. In this work, all these three numbers are the same. The possibility is included to write also the burnup on the card. Since this is not needed in this work, that part is commented out. The burnup would be read from the file '*3EXP.tot'.

3 Setting up the work

Setting up the script package needs close co-operation with the power company. This chapter describes the work needed to set up the system for TVO's units 1 or 2. In principle, it is possible to set up the system also for other reactors following the procedure described here. However, in this report we concentrate on the work prepared for this particular application.

3.1 Files needed for initialisation

In order to be able to use the previously described package, a certain amount of preparatory work is needed. This is only needed once for each application. It involves the setting up of the SIMULATE-3 restart file from which the reactor state is read for each power case that is needed.

TVO provided all the files needed to set up the system. To facilitate discussions and the use of the files, it is recommended to preserve the file layout used at TVO. The files are summarised in Table 3. The files included the end-of-cycle restart files for the previous two cycles. In this work these cycles are the cycles 28 and 29, which were in the directories /t1/c28 and /t1/c29. For the present cycle, c30, some more data was provided. The base directory is /t1/c30 under which the other directories are. First, we need to deal with the ini –directory. It contains the fuel.inc file that was mentioned already previously and the general.inc which contains some general calculation definitions for the SIMULATE-3 runs. The fuel.inc needs a library file cd-c30.lib, which is in the /t1/cd/ directory. The ini directory also contains the file margin.inp, which is a SIMULATE-3 input file to create a library file margin.lib. This file contains various margins and limits used by SIMULATE-3 and needs to be run through SIMULATE-3 first. A directory 'res' needs to be made under the c30 directory before running SIMULATE-3 with the second input file source-refu.inp. This file contains all the information needed to create a beginning-of-cycle restart file that is written to the res directory. The file fuel-weight.inc from the directory /t1/sim/ is included into the source-refu run so it should be checked that the file exists. A number of various files are created but the interesting one is called dist-boc.res. The procedure to set up the package is summarised in Table 4.

Table 3: SIMULATE-3 files received from TVO and needed to set up the package. Note, that sim-dep.inp or sim-108-simo.inp are optional and depend on the case. In addition they need to be adjusted for the application

/cms/t1/cd/cd-c30.lib		/cms/t1/c30/ini/fuel.inc
/cms/t1/sim/fuel-weight.inc		/cms/t1/c30/ini/general.inc
/cms/t1/c28/res/dist-eoc.res		/cms/t1/c30/ini/margin.inp
/cms/t1/c29/ res/dist-eoc.res		/cms/t1/c30/ini/source-refu.inp
/cms/t1/c30/bur/sim-dep.in	or	sim-108-simo.inp

Table 4: The procedure to set up the package after copying the files to appropriate locations.

1	In the directory /t1/c30/ini/, run SIMULATE-3 on margin.inp
2	Create directory res under the directory /t1/c30/
3	Run SIMULATE-3 on source-refu.inp in the /t1/c30/ini/ directory to create the BOC restart file of the current cycle c30.
4	Run the depletion calculation on a file that contains the control rod maps and the desired power levels. Make sure to write the power levels to the restart file. Check the name and path of the new restart file.

Once the ini-directory is dealt with and the library files and BOC restart file are done, it is time to move to the burnup calculations. They are all based on the dist-boc.res restart file. For the burnup calculation a file is again needed from the power company. This is because the control rod maps are needed for various power levels during the start-up phase. The input file can be either an actual one, which is a more complicated file, or a ‘planning’ file which is used while planning the start up. Especially if there are some unwanted transients during the start up that might interfere with the results, a planning file is perhaps better. In any case, this input file has to contain the power levels that are desired for the application. Moreover, the file needs to be set up so that it writes the reactor state to the restart file for the desired power levels. Also, the restart file name and path should be given as to suit the computer system, where it is run. The file used in this work for this purpose is listed in Attachment B.1.

3.2 Input files for cross section generation

Once the restart file with the desired state points is set up, the actual computation can start. First, however, the input files for the SIMULATE-3 runs have to be written. An example of an input file is given in Table 5 and some more are listed in Attachment B. In this package, ten input files for each power level is needed. For both the rodded (ari) and the unrodded (aro) case five inputs are needed: the base case, and two coefficient cases for both fuel temperature and moderator density. The input files of this package are listed in Table 6.

Table 5: The input file trans-100.aro.inp for the base case of 100% power and all rods out.

```
'COM' **** OL1 C30 pumpputransientti ***
'COM' -----------------------------------------------------|
'COM' Tiedosto, jolla lasketaan vaikutusaloja OL1 C30 aikana |
'COM' tapahtuneelle pumpputransienttilaskulle.             |
'COM' Tämä on perustapaus 100% teholla,                    |
'COM' kaikki säätösauvat ulkona                            |
'COM' -----------------------------------------------------|

'RES' '/nfs/h201/a/users/prokmr/Toimeksiannot/STUK/OL1-pumpputransientti/base/Tehonnosto.res' 69.0000/

'PRI.LIS' 'ON'/                       * Tämä pitää olla, jotta PRIMAC kirjoittaa
'PRI.STA' 50*' '/                     '3RPF'

'TIT.PRO' 'OL1 C30 pumpputransientti'/
'TIT.CAS' 'Perus ARO' /
'STA'/

'SAV.BAS','DEN','TFU','RPF'/          * Talleta tila
'SAV.LOK','HYD','FPD','TFU','XEN'/
'STA'/

'DEP.FPD', 0/                         * käytä edellisiä Xe,I,PM ja Sm-pitoisuuksia
'USE.BAS'     'DEN', 1.0, 0.0,
              'TFU', 1.0, 0.0
              'RPF', 1.0, 0.0 /       * pidä TFU, DEN ja RPF  vakiona
'CRD.ARO'/

'PRI.STA' '3DEN','3TFU','3XEN','3SAM','3EXP'/
'PRI.MAC' 'FULL' /                    * Vaikutusalat
'PRI.DFS' 'FULL'/                     * Epäjatkuvuustekijät
'KIN.EDT' 'ON' '1-D'/                 * Kirjoittaa kineettisen datan .kin-fileen

'STA'/

'END'/
```

Each input file has to start with reading the reactor state from the restart file. The power level is defined as the cycle exposure, which is the second parameter of the RES card. Note that the value has to match exactly the one in the restart file. The PRI-cards control the print out. PRI.LIS has to be set on in order for the cross sections to be written into the output file. This option also turns on a lot of other printing. The PRI.STA card is cleared with 50*' ', i.e. 50 spaces. The TIT-cards only write the given titles into the output. Then the STA card starts the calculation.

Table 6: Input files for SIMULATE-3 to generate cross sections for TRAB-3D. The input files listed in Attachment B are denoted by blue.

	60% Power	80% Power	100% Power
unrodded	trans-60-aro.inp	trans-80-aro.inp	trans-100-aro.inp
	trans-60-aro-T-50.inp	trans-80-aro-T-50.inp	trans-100-aro-T-50.inp
	trans-60-aro-T+50.inp	trans-80-aro-T+50.inp	trans-100-aro-T+50.inp
	trans-60-aro-30n.inp	trans-80-aro-30n.inp	trans-100-aro-30n.inp
	trans-60-aro-60n.inp	trans-80-aro-60n.inp	trans-100-aro-60n.inp
rodded	trans-60-ari.inp	trans-80-ari.inp	trans-100-ari.inp
	trans-60-ari-T-50.inp	trans-80-ari-T-50.inp	trans-100-ari-T-50.inp
	trans-60-ari-T+50.inp	trans-80-ari-T+50.inp	trans-100-ari-T+50.inp
	trans-60-ari-30n.inp	trans-80-ari-30n.inp	trans-100-ari-30n.inp
	trans-60-ari-60n.inp	trans-80-ari-60n.inp	trans-100-ari-60n.inp

The next block is used to freeze the reactor state. The SAV.BAS card is used to save for later use the nodal thermal-hydraulic conditions, i.e. the fuel temperature, moderator density and the relative power fraction. The SAV.LOK card is as well used for saving the nodal parameters, in this case the moderator density and temperature, the fuel temperature, and the Pm, Sm, I and Xe concentrations. These were just preparation of the computation which starts after the STA card.

First, the fission product concentrations that were saved above are loaded with the DEP.FPD card. Then the reactor state is loaded using the USE.BAS card. In addition to the cycle exposure on the RES card here are the major changes between the various coefficient cases. The settings for the USE.BAS cards for the various coefficient calculations are given in Table 7. The control rod position is controlled by the CRD card. In the rodded cases the CRD.ARI is used to insert all the rods and in the unrodded cases the CRD.ARO is used to withdraw all the rods. Note, that these cards may not be documented in the older SIMULATE-3 manuals even though they work in the older code versions.

Table 7: Settings for the USE.BAS card in the various coefficient calculations.

Base	T+50K	T-50K	30% density	60% density
'DEN', 1.0, 0.0, 'TFU', 1.0, 0.0 'RPF', 1.0, 0.0	'DEN', 1.0, 0.0, 'TFU', 1.0, 50.0 'RPF', 1.0, 0.0	'DEN', 1.0, 0.0, 'TFU', 1.0, -50.0 'RPF', 1.0, 0.0	'DEN', 0.30, 0.0, 'TFU', 1.0, 0.0 'RPF', 1.0, 0.0	'DEN', 0.60, 0.0, 'TFU', 1.0, 0.0 'RPF', 1.0, 0.0

Before the end of the file there are the cards that control the output data. The PRI.STA card is used to write the desired reactor state to the output file. The 3D exposure giving the burnup is in a sense excess as it is not used in this work. The PRI.MAC card is used to obtain the macroscopic cross sections. With the option 'FULL' all the cross sections are written to the file. The same option is also available for the PRI.DFS card that writes the discontinuity factors to the output file. In addition there is the KIN.EDT card that calculates and writes the kinetic data to the *.kin output file. With the SIMULATE-3 version modified for this work [5] this card also write the 3D kinetic parameters to the output file *.k3d.

4 Summary

A script package to facilitate the use of cross sections and other reactor physics parameters generated by SIMULATE-3 in TRAB-3D has been written. It contains a number of scripts that are collected under the umbrella of SimtoTrab.pl, which is the main program of the package. This script takes care of running SIMULATE-3 to obtain the cross sections and reactor state parameters for all the cases needed for transient calculations. Other scripts were written to extract the relevant data from the SIMULATE-3 output files. The script sim2trab-kortti.pl writes the reactor physics material constants into file in HEXBU-3D format.

In addition to the scripts handling the SIMULATE-3 related work, an additional script Trab3DNeutr.pl was written. This script is used to write some neutronics cards for TRAB-3D input. This script is also called by SimtoTrab.pl as all the other scripts.

In order to obtain all the parameters needed in the transient calculation some modifications to the SIMULATE-3 code were needed. Especially the 3D kinetic parameters were only obtained with these modifications. However, the modifications were kept to minimum and some effort was put to respect the SIMULATE-3 conventions. These changes are documented Reference [5].

Input files were created for three power levels and their coefficient calculations. For each power level, two temperature and two density perturbations were made to the base calculations. All the calculations were made with all rods both inserted and withdrawn. Consequently, ten calculations were made for each power level. The data was collected into three files per power level that are used as input files for TRAB-3D: two contain the material constant for the rodded and the unrodded cases and one contains the neutronics cards. In addition files are written that contain the Xenon and Samarium concentration in a format readable by TRAB-3D.

Unfortunately, neither the energy release per fission, ε or κ, nor the neutrons released per fission, ν, can be obtained from SIMULATE-3. Therefore, some modifications had to be made also to TRAB-3D to overcome this shortage. These modifications are, however, not reported here.

References

1. H. Räty, “User's manual for reactor dynamics codes TRAB-3D and HEXTRAN”, Research Report No VTT-R-04724-07, Espoo 22.5.2007.
2. Studsvik Scandpower, Inc., “SIMULATE-3, Advanced Three-Dimensional Two-Group Reactor Analysis Code, SIMULATE-3 User’s Manual”, Studsvik Scandpower Report SSP-01/414 Rev 3, (2003).
3. E. Kaloinen, “HEXBU-3D/MOD5, a New Version of the HEXBU-3D Code”, VTT Energy, Technical report RFD 26/92 (1992).
4. E. Kaloinen, “Input Instructions of HEXBU-3D/MOD5&6”, VTT Processes, Project report PRO1/P1027/05 (2005).
5. K. Rantamäki, “Modifications to SIMULATE-3” Research Report No VTT-R-01255-09, Espoo 24.2.2009 (Confidential)

Attachment A: Scripts

A.1: sim_XS_extract.pl

```
#! /usr/local/bin/perl
#
#--------------------------------------------------------------------#
#                                                                    #
# Tama skripti kasittelee SIMULATE -ohjelman output-filen ja hakee   #
# siita vaikutusalakartat, jotka se kirjoittaa kullakin tekijälle    #
# ja korkeudelle omaan tiedostoon *.xsm.                             #
#                                                                    #
# Input-parametri (h=help):                                          #
#           SIMULATE-input file *.inp                                #
#                                                                    #
#                                                                    #
# (c) Karin Rantamaki 25.11.2008                                     #
#                                                                    #
#                                                                    #
#--------------------------------------------------------------------#

use strict;
use warnings;

my $Input = $ARGV[0];
if($#ARGV<0 || $Input =~ 'h'){
    print "Anna SIMULATEn inputfile (.inp), luettava parametri ja tulostusmuoto.\n";
    print "  *** MAC vaikutusaloille  tai \n";
    print "  *** DFS epäjatkuvuustekijöille\n";
    print "  *** TOT (kaikki yhteen tiedostoon), MAP kartat\n";
    exit;
}
elsif($#ARGV<1){
    if($Input =~ 'h'){
       print "Anna inputfile, luettava parametri ja tulostusmuoto\n";
       print "  *** MAC vaikutusaloille  tai \n";
       print "  *** DFS epäjatkuvuustekijöille\n";
       print "  *** TOT (kaikki yhteen tiedostoon), MAP kartat\n";
       exit;
       }
    else{
       print "Anna luettava parametri ja tulostusmuoto.\n";
       print "  *** MAC vaikutusaloille  tai \n";
       print "  *** DFS epäjatkuvuustekijöille\n";
       print "  *** TOT (kaikki yhteen tiedostoon), MAP kartat\n";
       exit;
    }
}
elsif($#ARGV<2){
    print "Anna inputfile, luettava parametri ja tulostusmuoto\n";
    print " *** esim. 3DEN, 3TFU tai NTFU\n";
    print " *** TOT (kaikki yhteen tiedostoon), MAP kartat\n";
    exit;
}
my $outflag=$ARGV[2];
my ($Output,$In,$paate);
my $base = $Input ;
$base =~ s/.inp//;
$Input =~  s/inp/out/;

my @rfue;
my ($ri,$rj,$rk);
my $case=$ARGV[1];
my $line;
my $XSname;
my @XSname;
my ($K,$J);
my $NJ=28;
my (@I,$ii,$ix,$ni);
my $kerroin;
my @Items;
my $a=0;
my @XS;
my %XStot;

# ------------------------------------------------------------------------ #
# Luetaan tiedostosta fue.def data, jolla määritellään onko noodissa pa:ta #
open ($In, "<", "fue.def") or die "En löytänyt tiedostoa fue.def\n";
while(my $line =<$In>) {
    @Items= split /\s+/,$line;
    shift @Items;  # poista ylimääräinen tyhjä
#    print "size $#Items, $Items[0],$Items[1],$Items[2],$Items[3]\n";
    $rfue[$Items[0]][$Items[1]][$Items[2]]=[$Items[3]]
}
close $In;
# ------------------------------------------------------------------------ #

print "Luen tiedostoa $Input ja haen sieltä vaikutusalat !\n";
open($In, "<", $Input) or die "Can't open $Input: $!";
print "Avattiin $Input \n";

$ni=0;
while($line=<$In>){
# ------------------------------------------------------------------------ #
# Hae tiedostosta kohta, jossa data alkaa. Lue otsikkorivi ja kerroin, jos #
# se löytyy.  Muuten kerroin on 1.                                         #
    if (($line=~ /PRI.$case - /) && !($line=~ /Note/)) {
       if ($case eq 'MAC'){                        # Vaikutusalojen tapaus #
           $line =~
              /^(.*)(Cross section)(\s+)(\w*.\w)(\s+)(for core plane K
=)(\s+)(\d{1,2})(.*)$/;
           $K=$8;
           $XSname=$4;
           if ($9=~/^(.*)(REAL VALUE )(.*)(\d.\dE\+\d\d)(.*)$/){
              $kerroin=1/$4;
           }
           else{$kerroin=1;}
       }
       elsif($case eq 'DFS'){              # Epäjatkuvuustekijöiden tapaus #
           $line =~
              /^(.*)(Discontinuity Factor)(\s+)(\w*.\w)(\s+)(for Core Plane K
=)(\s+)(\d{1,2})(.*)$/;
           $K=$8;
           $XSname=$4;
           if ($9=~/^(.*)(REAL VALUE )(.*)(\d.\dE\+\d\d)(.*)$/){
              $kerroin=1/$4;
           }
           else{$kerroin=1;}
       }
       else {print "*** Epäkelpo tapaus. Poistun ohjelmasta ***\n";exit;}
# ------------------------------------------------------------------------ #
       $line=<$In>;                   # luetaan rivi jolla I-koordinaatti #
       chomp $line;
       @I=split /\s+/, $line;
       $line=<$In>;                   # luetaan tyhjä rivi
       foreach my $int (1..$NJ){
           $line=<$In>;
           @Items= split /\s+/,$line;
           shift @Items;              # poista ylimääräinen tyhjä rivin alusta
           $J=shift @Items;           # otetaan J-koordinaatti
           # ---------------------------------#
           # reaktorin vasen puoli
           if ($I[1]==1){
              $ii=0;
              foreach my $it (@Items){
                  $ix=$I[$#I]-$#Items+$ii;
                  $XS[$J][$ix]=$kerroin*$it;
                  $ii++;
```

```
               } # end foreach Item
            } # end if I[1]=1
            # ---------------------------------#
            # reaktorin oikea puoli
            else {
               $ii=0;
               foreach my $it (@Items){
                   $ix=$I[1]+$ii;
                   $XS[$J][$ix]=$kerroin*$it;
                   $ii++;
               } # end foreach Item
            } # end else
        } # end foreach 1..28 rivi
# ------------------------------------------------------------------------ #
# Jos myös oikea puoli on luettu, niin sijoitetaan luettu vaikutusala      #
# talteen muuttujaan XStot                                                 #
        if ($I[1]!=1){
            if ($K==1){$XSname[$ni]=$XSname;$ni++}
            foreach $J (1..$NJ){
               foreach my $Ii (1..$I[$#I]){
                   if (defined $XS[$J][$Ii]) {
                      $XStot{$XSname}[$K][$J][$Ii]=$XS[$J][$Ii];
                   }
               }
            }
# ------------------------------------------------------------------------ #
# Tulostetaan parametrikartat 'MAP' optiolla kullekin K:lle omaan fileen   #
            if ($outflag eq 'MAP'){
               if ($case eq 'MAC'){$paate=".xsm";}
               elsif ($case eq 'DFS'){$paate=".dfs";}
               $Output=$base."_".$XSname."_K".$K.$paate;
               $Output =~s /\//-/;
               open(my $Out, ">", $Output) or die "Can't open $Output: $!";
               foreach $J (1..$NJ){
                   foreach my $Ii (1..$I[$#I]){
                      if (defined $XS[$J][$Ii]) {
                          printf $Out "%10.3e",$XStot{$XSname}[$K][$J][$Ii];
                      }
                      else{
                          printf $Out "%10.3s",' ';
                      }
                   } #end foreach I
                   printf $Out "\n";
               } #end foreach J
               close $Out or die "$Out: $!";
            } # end if MAP
# ------------------------------------------------------------------------ #
        } # end if I1!=1
     } # end if PRI.MAC
}
if ($outflag eq 'MAP'){
    print "$case parametrikartat kirjoitettu tiedostoihin *$paate\n";
}
# ------------------------------------------------------------------------ #
close $In or die "$In: $!";

# ------------------------------------------------------------------------ #
# Tulostus optiolla 'TOT' kaikki data yhteen tiedostoon, myös indeksit     #
if ($outflag eq 'TOT'){
    my $Output2=$base."_".$case.".tot";
    open(my $Out2, ">", $Output2) or die "Can't open $Output2: $!";
    print "XSnames: @XSname \n";
    foreach $K (1...25){
       foreach $J (2...27){
           foreach my $I (2..27) {
               printf $Out2 "%4d %4d %4d",$K,$J-1,$I-1;
               foreach my $sig (@XSname) {
                   if ((defined $XStot{$sig}[$K][$J][$I])&&(defined
$rfue[$K][$J][$I])) {
                      printf $Out2 "%10.3e",$XStot{$sig}[$K][$J][$I];
```

```
                    }
                    else {printf $Out2 "%10.3f",0;}
                }
                print $Out2 "\n";
             }
        }
    }
    print "Parametri $case kirjoitettu tiedostoon $Output2.\n";
    close $Out2 or die "$Out2: $!";
}
# ------------------------------------------------------------------------ #
```

A.2: sim_DEN_extract.pl

```
#! /usr/local/bin/perl
#
#---------------------------------------------------------------------#
#                                                                     #
# Tama skripti kasittelee SIMULATE -ohjelman output-filen ja hakee    #
# siita 3D noodikohtaiset tiheydet tai lämpötilat, jotka sen          #
# kirjoittaa kullakin parametrille ja korkeudelle omaan               #
# tiedostoon *.map. Vaihtoehtoisesti se kirjoittaa kaikki datat       #
# yhteen tiedostoo .tot, jossa on myös indeksit rivin alussa.         #
# Kolmantena vaihtoehtona on .trb, joka sisältää datan TRAB-3D:n      #
# muodossa kullekin nipulle, datat ylhäältä alas. Tätä suositellaan   #
# käytettäväksi mm. Xenonille ja Samariumille, jotta saadaan          #
# TRAB-3Dlle tiedostot.                                               #
#                                                                     #
# Input-parametri (h=help):                                           #
#          SIMULATE-input file *.inp                                  #
#          luettava parametri, esim DEN tai TFU                       #
#                                                                     #
# (c) Karin Rantamaki 25.11.2008                                      #
#                                                                     #
# 21.1.2009 Lisätty TRAB-3Dn kaipaama tulostusmuoto                   #
#                                                                     #
#---------------------------------------------------------------------#

use strict;
use warnings;

my $Input = $ARGV[0];
if($#ARGV<0 || $Input =~ 'h'){
    print "Anna SIMULATEn inputfile (.inp), luettava parametri ja tulostusmuoto\n";
    print " *** esim. 3DEN, 3TFU tai NTFU\n";
    print " *** TOT (kaikki yhteen tiedostoon), MAP kartat, TRB Trab-3Dtä varten\n";
    exit;
}
elsif($#ARGV<1){
    if($Input =~ 'h'){
       print "Anna inputfile, luettava parametri ja tulostusmuoto\n";
       print " *** esim. 3DEN, 3TFU tai NTFU\n";
       print " *** TOT (kaikki yhteen tiedostoon), MAP kartat\n";
       exit;
       }
    else{
       print "Anna luettava parametri ja tulostusmuoto.\n";
       print " *** esim. 3DEN, 3TFU tai NTFU\n";
       print " *** TOT (kaikki yhteen tiedostoon), MAP kartat\n";
       exit;
    }
}
elsif($#ARGV<2){
    print "Anna inputfile, luettava parametri ja tulostusmuoto\n";
    print " *** esim. 3DEN, 3TFU tai NTFU\n";
    print " *** TOT (kaikki yhteen tiedostoon), MAP kartat\n";
    exit;
}
my $outflag=$ARGV[2];
my $Output;
my $base = $Input ;
$base =~ s/.inp//;
$Input =~  s/inp/out/;
print "Luen tiedostoa $Input ja haen sieltä vaikutusalat !\n";

open(my $In, "<", $Input) or die "Can't open $Input: $!";
print "Avattiin $Input \n";

my $line;
my $XSname=$ARGV[1];
my ($K,$J,$I);
my ($Imax,$Jmax,$Kmax,$Tmin);
my (@I,$ii,$ix);
my $kerroin;
my @Items;
my $a=0;
my @XS;
my %XStot;
$Imax=0;$Jmax=0;$Kmax=0;

while($line=<$In>){
# ------------------------------------------------------------------------ #
# Hae tiedostosta kohta, jossa data alkaa. Lue otsikkorivi ja kerroin, jos #
# se löytyy.  Muuten kerroin on 1                                          #
    if (($line=~ /PRI.STA $XSname  - /) && !($line=~ /Note/)) {
       if ($line=~/^(.*)(REAL VALUE )(.*)(\d.\dE\+\d\d)(.*)$/){
           $kerroin=1/$4;
       }
       else{$kerroin=1;}
       $line=<$In>;  # luetaan rivi jolla I-koordinaatti
       chomp $line;
# ------------------------------------------------------------------------ #
# Käsitellään I-koordinaatien rivi 3D-muuttujille
       if ($XSname =~ /^3/){
           $line=~/^(\D+)(\d{1,2})(.+)/;
           $J=$2;
           if ($J>$Jmax) {$Jmax=$J;}
           $line=<$In>;
           chomp $line;
           $line =~ s/^\s+//;
           @I=split /\/\d{2}\s*/, $line;
           if ($I[$#I] gt $Imax) {$Imax=$I[$#I];}
           $line=<$In>;  # luetaan tyhjä rivi
       }
# ------------------------------------------------------------------------ #
# Käsitellään I-koordinaatien rivi ND-muuttujille, joilla on heijastinalue
       elsif ($XSname =~ /^N/){
           @I=split /\s+/, $line;
           shift @I;     # poista ylimääräinen tyhjä rivin alusta
           if ($I[$#I] gt $Imax) {$Imax=$I[$#I];}
           $line=<$In>;
           chomp $line;
           $line=~ /^\s\s(\()(\s?\d{1,2})(.*)/;
           $J=$2;
           if ($J>$Jmax) {$Jmax=$J;}
       }
# ------------------------------------------------------------------------ #
# Luetaan data ja käsitellään se
       $line=<$In>;
       while($line=~/^\s+/){
           @Items= split /\s+/,$line;
           shift @Items;    # poista ylimääräinen tyhjä rivin alusta
           $K=shift @Items; # otetaan K-koordinaatti
           if ($K>$Kmax) {$Kmax=$K;}
           if(($XSname eq 'NTFU')&&(!defined $Tmin)){
              $Tmin=$Items[0]; print "Tmin=$Tmin\n";
           }
           # ---------------------------------#
           # reaktorin vasen puoli
           if ($I[0] eq 1){
              $ii=0;
              foreach my $it (@Items){
                  $ix=$I[$#I]-$#Items+$ii;
                  $XS[$K][$J][$ix]=$it;
                  $ii++;
              } # end foreach Item
           } # end if I[0]=1
           # ---------------------------------#
           # reaktorin oikea puoli
           else {
              $ii=0;
              foreach my $it (@Items){
                  $ix=$I[0]+$ii;
                  $XS[$K][$J][$ix]=$it;
                  $ii++;
              } # end foreach Item

           } # end else
           # ---------------------------------#
           $line=<$In>;
       } # end while
    } # end if PRI.MAC
}
# ------------------------------------------------------------------------ #
print "Kmax=$Kmax,Jmax=$Jmax,Imax=$Imax\n";
my $pf;
if    ($XSname =~ 'TFU'){$pf="%10.3g";}
elsif ($XSname =~ 'DEN'){$pf="%10.3f";}
else  {$pf="%10.3e"};
# ------------------------------------------------------------------------ #
# Tulostus optiolla 'TOT' kaikki data yhteen tiedostoon, myös indeksit     #
if ($outflag eq 'TOT'){
    my  $Output2=$base."_".$XSname.".tot";
    open(my $Out2, ">", $Output2) or die "Can't open $Output2: $!";

    foreach $K (1..$Kmax){ #K=0 on heijastin
       foreach $J (1..$Jmax){
           foreach my $Ii (1..$Imax){
              printf $Out2 "%4d %4d %4d",$K,$J,$Ii;
              if (defined $XS[$K][$J][$Ii]) {
                  printf $Out2 $pf,$XS[$K][$J][$Ii];
              }
              else{
                  printf $Out2 $pf,0;
              }
              print $Out2 "\n";
           } #end foreach I
       } #end foreach J
    } # end foreach K
    close $Out2 or die "$Out2: $!";
    print "Vaikutusalat kirjoitettu tiedostoon $Output2.\n";
} #end if TOT
# ------------------------------------------------------------------------ #
# Tulostus karttamuodossa kullekin K:lle omaan tiedostoon                  #
if ($outflag eq 'MAP'){
    foreach $K (1..$Kmax){ #K=0 on heijastin
       $Output=$base."_".$XSname."_K".$K.".map";
       open(my $Out, ">", $Output) or die "Can't open $Output: $!";
       foreach $J (1..$Jmax){
           foreach my $Ii (1..$Imax){
              if (defined $XS[$K][$J][$Ii]) {
                  printf $Out $pf,$XS[$K][$J][$Ii];
              }
              else{
                  printf $Out "%10.3s",' ';
              }
           } #end foreach I
           printf $Out "\n";
       } #end foreach J
       close $Out or die "$Out: $!";
    } # end foreach K
    print "Vaikutusalakartat kirjoitettu tiedostoihin *.map\n";
} # end if MAP
 #  #print "Data tiedostossa $Input alkoi riviltä $a\n";

#---------------------------------------------------------------------#
# TRAB-3Dtä varten halutaan XEN ja SAM-datat kullekin nipulle         #
# ylhäältä alaspäin.  Tulostetaan tässä muodossa tiedostoon .trb      #
# *** KMR 21.1.2009                                                   #
if ($outflag eq 'TRB'){
    my  $Output2=$base."_".$XSname.".trb";
    open(my $Out2, ">", $Output2) or die "Can't open $Output2: $!";
    foreach $J (1..$Jmax){
       foreach my $Ii (1..$Imax){
           if (defined $XS[$Kmax/2][$J][$Ii]) {
              foreach $K (1..$Kmax){ #K=0 on heijastin
                  printf $Out2 $pf."\n",$XS[$K][$J][$Ii];
              }
           } #end foreach I
       } #end foreach J
    } # end foreach K
    close $Out2 or die "$Out2: $!";
    print "Vaikutusalat kirjoitettu tiedostoon $Output2.\n";
} #end if TRB
#---------------------------------------------------------------------#
# Kirjoitetaan fue.def tiedostoon indeksit, joissa on polttoainetta   #
if ($XSname eq 'NTFU'){
    open(my $Out3, ">", "fue.def") or die "Can't open fue.def: $!";

    foreach $K (1..$Kmax-1){
       foreach $J (1..$Jmax){
           foreach $I (1..$Imax){
              if ((defined $XS[$K][$J][$I])&&($XS[$K][$J][$I]!=$Tmin)) {
                  printf $Out3 "%4d %4d %4d %4g\n",$K,$J,$I, 1.0;
              }
           }
       }
    }
    close $Out3 or die "$Out3: $!";
    print "Polttoainealueen indeksit kirjoitettu tiedostoon fue.def\n";
}
close $In or die "$In: $!";
#print "Suljin tiedostot $Input ja $Output \n";
# ------------------------------------------------------------------------- #
```

A.3: sim_KIN_extract.pl

```
#! /usr/local/bin/perl
#
#---------------------------------------------------------------------#
#                                                                     #
# Tama skripti kasittelee SIMULATE -ohjelman k3d output-filen ja      #
# hakee siita betat, lambdat ja 1/v:t, jotka se kirjoittaa kunkin     #
# omaan tiedostoon *.tot.                                             #
#                                                                     #
# Input-parametri (h=help):                                           #
#           SIMULATE-input file *.inp                                 #
#                                                                     #
#                                                                     #
# (c) Karin Rantamaki 9.12.2008                                       #
#                                                                     #
#                                                                     #
#---------------------------------------------------------------------#

use strict;
use warnings;

my $Input = $ARGV[0];
if($#ARGV<0 || $Input =~ 'h'){
    print "Anna SIMULATEn inputfile (.inp)\n";
    exit;
}
elsif($#ARGV<1){
    if($Input =~ 'h'){
       print "Anna SIMULATEn inputfile (.inp)\n";
       exit;
    }
}

my ($Output,$In,$paate);
my $base = $Input ;
$base =~ s/.inp//;
$Input =~  s/inp/k3d/;

my @rfue;
my $line;
my $XSname;
my @XSname=("beta","lambda","vinv") ;
my ($K,$J,$I,$ig);
my (@IGMa,$gi);
my @IGMi=(1,1,3);
my @Items;
my %XStot;

# ------------------------------------------------------------------------ #
# Luetaan tiedostosta fue.def data, jolla määritellään onko noodissa pa:ta #
open ($In, "<", "fue.def") or die "En löytänyt tiedostoa fue.def\n";
while(my $line =<$In>) {
    @Items= split /\s+/,$line;
    shift @Items;  # poista ylimääräinen tyhjä
    $rfue[$Items[0]][$Items[1]][$Items[2]]=[$Items[3]]
}
close $In;
# ------------------------------------------------------------------------ #

print "Luen tiedostoa $Input ja haen sieltä parametrit !\n";
open($In, "<", $Input) or die "Can't open $Input: $!";
print "Avattiin $Input \n";

# ------------------------------------------------------------------------ #
# Luetaan kineettiset parametrit, beta (gi=0), lambda (gi=1), 1/v (gi=2)   #
$gi=0;
while($line=<$In>){
    next if ($line =~ /^TIT/);  # Ohitetaan TIT-kortit
    @Items= split /\s+/,$line;
    shift @Items;               # Poista tyhjä rivin alusta
    $K=shift @Items;
    $J=shift @Items;
    $I= shift @Items;
    $XSname=$XSname[$gi];
# Käsitellään rivi, jos  siinä on polttoainetta
    if (defined $rfue[$K][$J][$I]) {
       $ig=1;                 # viivästyneiden neutronien ryhmä
       foreach my $Items  (@Items){
           $XStot{$XSname}[$K][$J][$I][$ig]=$Items;
           $ig++;
       }
       $IGMa[$gi]=$ig-1;       # parametrin ryhmien määrä
    }
    $gi++;
    if ($gi==3) {$gi=0;}        # resetoi parametrilaskuri
} # end while In
close $In or die "$In: $!";

# ------------------------------------------------------------------------ #
# Kirjoitetaan parametrit omiin tiedostoihinsa                             #
$gi=0;
foreach $XSname (@XSname) {
    $Output=$base."_".$XSname.".tot";
    open(my $Out, ">", $Output) or die "Can't open $Output: $!";
    foreach $K (1...25){
       foreach $J (2...27){
           foreach my $I (2..27) {
              printf $Out "%4d %4d %4d",$K,$J-1,$I-1;
              $ig=1;
              @Items=$XStot{XSname}[$K][$J][$I];
              for $ig ($IGMi[$gi]..$IGMa[$gi]) {
                  if ((defined $XStot{$XSname}[$K][$J][$I][$ig])&&(defined
$rfue[$K][$J][$I])) {
                     printf $Out "%12.5e",$XStot{$XSname}[$K][$J][$I][$ig];
                  }
                  else {printf $Out "%12.5f",0;}
              }
              print $Out "\n";
           }
       }
    }
    print "Parametri $XSname kirjoitettu tiedostoon $Output.\n";
    close $Out or die "$Out: $!";
    $gi++;
}
# ------------------------------------------------------------------------ #
```

A.4: sim_FUEmass.pl

```
#! /usr/local/bin/perl
#
#---------------------------------------------------------------------#
#                                                                     #
# Tämä skripti käsittelee SIMULATE-ohjelman tiedostoja *.out ja       #
# *.sum sekä hyvin alkuvaiheessa linkitettävää tiedostoa fuel.inc     #
# fuel.inc-tiedosto sisältää polttoaineen määrittelyt ja se on saatu #
# TVO:lta (vaihtuu joka syklille, ** HUOM polku! **).                 #
# *.out-tiedostosta luetaan polttoainesegmenttien massat sekä nipun  #
# dimensiot.                                                          #
# fuel.inc-tiedoston avulla lasketaan kullekin käytössä olevalle      #
# nipulle aksiaaliset massat SIMULATEn käyttämälle noodijaolle.       #
# *.sum-tiedostosta luetaan nippujen sijainnit radiaalikartalta.      #
# Lopuksi kirjoitetaan noodikohtaiset massat (g/cm) tiedostoon *.fue #
#                                                                     #
# Input-parametri (h=help):                                           #
#           SIMULATE-input file *.inp                                 #
#                                                                     #
# (c) Karin Rantamaki 11.12.2008                                      #
#                                                                     #
#---------------------------------------------------------------------#


use strict;
use warnings;

my $Input = $ARGV[0];
if($#ARGV<0 || $Input =~ 'h'){
    print "Anna SIMULATEn inputfile (.inp)\n";
    exit;
}
elsif($#ARGV<1){
    if($Input =~ 'h'){
       print "Anna SIMULATEn inputfile (.inp)\n";
       exit;
    }
}
my $home="/nfs/h201/a/users/prokmr";
my $file1="$home/TVO/Simulate/t1/c30/ini/fuel.inc"; # sisältää SEG.LIB ja FUE.ZON
my $file2=$Input; $file2=~ s/.inp/.out/;            # sisältää massat, ja dimensiot
my $file3=$Input; $file3=~ s/.inp/.sum/;        # sisältää polttoainekartan FUE.TYP
my ($In1);
my ($K,$J,$I);
my $line;
my @fuetyp;
my @Items;
my ($seg,$is);
my (@segmass,@segnumber,@segment,@massa);
my $la;
#-------------------------------------------------------------------#
# Luetaan Assembly Pitch, Node Height, nipun korkeus sekä           #
# polttoainesegmenttien  massat tiedostosta *.out                   #
open ($In1, "<", $file2) or die "Can't open $file2: $!\n";
$la=1;
while($line=<$In1>) {
    last if ($line=~ / Assembly Pitch/);      # Hae Assembly Pitch kohta
    $la++;
}
chomp $line;
$line =~ /^(.*)(Assembly Pitch)(\D+)(\d+\.\d+)(.*)$/;
my $asspitch=$4;                              # hae nipun leveys riviltä

while($line=<$In1>) {
    last if ($line=~ / Node Height/);         # Hae Noodikorkeus kohta
    $la++;
}
chomp $line;
$line =~ /^(.*)(Node Height)(\D+)(\d+\.\d+)(.*)$/;
my $dz=$4;                                    # Hae noodikorkeus riviltä

while($line=<$In1>) {
    last if ($line=~ / Core Height/);         # Hae korkeus kohta
    $la++;
}
chomp $line;
$line =~ /^(.*)(Core Height)(\D+)(\d+\.\d+)(.*)$/;
my $Lz=$4;                                    # Hae korkeus riviltä


print "Assembly pitch on $asspitch cm, ";
print "noodikorkeus on $dz cm ja ";
print "sydämen korkeus on $Lz cm\n";

while($line=<$In1>) {
    last if ($line=~ / Segment Name/);         # Hae segmenttien  kohta
    $la++;
}
$line=<$In1>;
$is=0;
while($line=<$In1>) {
    last if !($line =~ / \w+/);               #lue segmentit kunnes tulee tyhjä rivi
    chomp $line;
    @Items= split /\s+/,$line;
    shift @Items;                              # poista tyhjät rivin alusta
    $seg= shift @Items;                        # lue segmentin numero
    $segnumber[$is]=$seg;
    shift @Items;                              # poista sementin nimi
    $segmass[$seg]=(shift @Items);             # lue segmentin massa (g/cc)
    $segmass[$seg]=$segmass[$seg]*$asspitch*$asspitch; # muuta g/cm
    $is++;
    $la++;
#    print "Segmentin $seg massa on $segmass[$seg] g/cm\n";
}

#print "Viimeinen rivi oli: $line";
close $In1 or die "$In1: $!\n";
#-------------------------------------------------------------------#
#-------------------------------------------------------------------#
# Luetaan segmentit ja polttoaineen rakenne SEG.LIB ja FUE.ZON -     #
# korteilta tiedostosta fuel.inc                                    #
# Jaetaan segmentit noodeille                                       #
open ($In1, "<", $file1) or die "Can't open $file1: $!\n";
$la=1;                                        # rivilaskuri
while($line=<$In1>) {
    last if ($line=~ /FUE.ZON/);              # Hae FUE.ZON-kortti
    $la++;
}
print "FUE.ZON löytyi riviltä $la\n";
my ($k1,$k2);
my ($z1,$z2);
#print "Pa\\K";
#for $K (1..25) {printf "%6d", $K;}
#print "\n\n";
while ($line =~ /FUE.ZON/){                  # käy läpi kaikki pa-tyypit
    if($line=~/RAD/){$line=<$In1>;}          # skippaa heijastin
    chomp $line;
    $line =~ s/,//;
    $line =~ s/\///;
#    print $line,"\n";

    @Items = split /\s+/,$line;
    shift @Items;                             # poista 'FUE.ZON'
    my $fuetyp= shift @Items;                 # lue pa-tyyppi
    splice @Items, 0, 3;                      # poista mekaaninen numero,pa:n nimi ja
heijastin
    pop @Items;                               # poista heijastin lopusta

    $z1=shift @Items;                         # Lue eka koordinaatti
    $k1=int($z1/$dz)+1;                       # muuta se noodi-indeksiksi
    $k2=0; $z2=0;                             # Alustetaan ekat z2 ja k2
    $massa[$fuetyp][$k2]=0;                   # sekä noodin k2=0 massa
    my $Ni=$#Items;
    for $is (0.. $Ni/2){                      # käy läpi kaikki segmentit
       $seg=shift @Items;                    # Lue segmentin numero
       $massa[$fuetyp][$k2]+=
           ($k2-$z2/$dz)*$segmass[$seg];     # lisätään jaetun noodin yläosan massa
       $segment[$k2]+=($k2-$z2/$dz)*$seg;
       $z2=shift @Items;                     # ja sen päättymiskoordinaatti
       $k2=int($z2/$dz)+1;                   # muuta se noodi-indeksiksi
       for $K ($k1..$k2-1) {                 # noodien k1...(k2-1) massat
           $segment[$K]=$seg;                # voi sijoittaa suoraan
           $massa[$fuetyp][$K]=$segmass[$seg];
       }                                     # $k2:n massa pitää interpoloida
       $massa[$fuetyp][$k2]=                 # lasketaan noodin alaosan massa
           ($z2/$dz-$k2+1)*$segmass[$seg];
       $segment[$k2]+=($z2/$dz-$k2+1)*$seg;
       $k1=$k2+1;                            # Päivitä seuraavan segmentin alaosan
noodi
    } # end of segmentit
#    print " $fuetyp:";
#    for $K (1..25) {      printf "%6.3g", $segment[$K];$segment[$K]=0;}
#    print "\n     ";
#    for $K (1..25) {      printf "%6.3f", $massa[$fuetyp][$K];}
#    print "\n";
    $line=<$In1>;                             # Lue uusi rivi FUE.ZON kortilta
    $la++;
} # end of FUE.ZON
#------------------------------------------------------------------#
# Nyt on siis kaikki polttoainetyypit käyty läpi ja nipulla         #
# aksiaalinen massa                                                 #
#------------------------------------------------------------------#
close $In1 or die "$In1: $!\n";
#------------------------------------------------------------------#
#------------------------------------------------------------------#
# Luetaan polttoainetyyppi sijaintikartasta FUE.TYP .sum tiedostosta #
open ($In1, "<", $file3) or die "Can't open $file3: $!\n";
#print "Luen tiedostoa $file3\n";
$la=1;                                        # rivilaskuri
while($line=<$In1>) {
    last if ($line=~ /^ FUE.TYP/);            # Hae FUE.TYP-kortti
    $la++;
}
$line=<$In1>;                                 # Lue eka rivi FUE.TYP kortilta
print "FUE.TYP löytyi riviltä $la\n";
```

```
until($line=~ /^ FUE/) {                     # FUE.TYP-kortti, kunnes seuraava FU-
kortti
    chomp $line;                             # remove trailing newline
    @Items= split /\s+/, $line;
    shift @Items;                            # poista tyhjät rivin alusta
    $J= shift @Items;                        # lue rivin numero J-ndeksiksi
    shift @Items;                            # poista ylimääräinen numero
#    my $Ni= $#Items+1;
    for $I (1..$#Items+1){
       $fuetyp[$J][$I]=$Items[$I-1];        # luetaan polttoainetyyppi
    }
    $line=<$In1>;                            # seuraava rivi
}
close $In1 or die "$In1: $!\n";
#-------------------------------------------------------------------#
# Kirjoitetaan tiedostoon *.fue noodikohtaiset massat muodossa       #
# K, J, I, massa                                                    #
my ($Output,$Out);
$Output=$Input;
$Output =~ s/.inp/.fue/;
open ($Out, ">", $Output) or die "Can't open $Output: $!\n";
foreach $K (1...25){
    foreach $J (2...27){
       foreach my $I (2..27) {
           printf $Out "%4d %4d %4d",$K,$J-1,$I-1;
           if ($fuetyp[$J][$I] >1) {
              printf $Out "%8.2f\n", $massa[$fuetyp[$J][$I]][$K];
           }
           else {printf $Out "%8.2f\n",0;}
       }
    }
}
print "Polttoainemassat kirjoitettu tiedostoon $Output\n";
close $Out or die "$Out: $!\n";
```

A.5: sim2trab-kortti.pl

```
#! /usr/local/bin/perl
#
#-------------------------------------------------------------------#
#                                                                   #
# Tama skripti lukee *. tot-tiedostoista tilanmuuttujat ja vaikutus- #
# alat sekä nominaalitapaukselle vielä epäjatkuvuustekijät, 1/v:t ja #
# betat.  Lisäksi luetaan uraanimassat tiedostosta *.fue. *.out-     #
# tiedostosta luetaan lisäksi erilaisia tarvittavia parametreja.     #
# Sitten kootaan kaikki data HEXBU-muotoiseksi korttitiedostoksi.    #
# Enemmän dataa sisältävien korttien kirjoittamiseksi on tehty       #
# aliohjelmat.  Vaikutusalojen sovituskertoimia varten tehtiin Apu-  #
# moduuliin aliohjelma kerroin, joka tekee toisen asteen polynomin   #
# kertoimien sovituksen.                                            #
#                                                                   #
# Input-parametri (h=help):                                         #
#           SIMULATE-input file *.inp                               #
#                                                                   #
#                                                                   #
# (c) Karin Rantamaki 17.12.2008                                    #
#                                                                   #
#                                                                   #
#-------------------------------------------------------------------#

use strict;
use warnings;
use lib "/nfs/h201/a/users/prokmr/bin/";
use Apu;
my $Input = $ARGV[0];
if($#ARGV<0 || $Input =~ 'h'){
    print "Anna perustapauksen SIMULATEn inputfile (.inp)\n";
    exit;
}
elsif($#ARGV<1){
    if($Input =~ 'h'){
       print "Anna SIMULATEn inputfile (.inp)\n";
       exit;
    }
}
my @Pcases=("60","80","100");    # Tehotapaukset
my @Rcases=("aro","ari");        # Säätösauvatapaukset
my @Dcases=("-30n","-60n");      # Tiheystapaukset ("" on peruslasku)
my @Tcases=("-T-50","-T+50");    # Lämpötilatapaukset ("" on peruslasku)
my @params=("MAC","DFS","3TFU","3DEN","beta","vinv");
my @XSname=("D1","D2","SR1","SA1","SA2","NF1","NF2","KN");    # luettavat sigmat
my @DFname=("F1W","F1S","F1E","F1N","F2W","F2S","F2E","F2N"); #luettavat
epäjatkuvuustekijät

my ($Output,$Out);
my $base = $Input ;
$base =~ s/.inp//;
my $line;
my @Items;
my ($K,$J,$I,$ig);
my ($Kmax,$Jmax,$Imax);
my ($Kmin,$Jmin,$Imin);

#$Kmin=3; $Jmin=5; $Imin=11;
$Kmin=1; $Jmin=1; $Imin=1;
$Kmax=25; $Jmax=13; $Imax=26;
#$Kmax=25; $Jmax=$13; $Imax=26;                      # Puolikas taso (yläpuolisko)
#$Kmax=25;$Jmax=$26;$Imax=26;                      # Täysi taso (huom! 2000= (500.
nippu,K=1)

my $bu =0.0;


my (@TFnom,@DEnom,@XSnom,@DFnom,@beta,@vinv);   # nominaalitapauksen parametrit
my (@TFkerr,@XSkerrT,@DEkerr,@XSkerrD);         # kerrointapauksen parametrit


my $title;
my ($maternumb,$nbij,@Umassa);
my ($Pnom,$Tnom,$Dinom);

my $FTEMP=1;
my ($NXE,$NSM);
my (@NCDopp,@NCBor,@NCDT,@NCDe,@NCTe,@NCBu);

#---------------------------------------------------------------------#
# Luetaan Otsikko ja  Uraanimassat                                    #
$Input=$base.".out";
$title=HaeOtsikko($Input);
($Pnom,$Tnom,$Dinom)=FluidParams($Input);

$Input=$base.".fue";
@Umassa=Apu::LueTaulu($Input);
#---------------------------------------------------------------------#
# Luetaan perustapauksen parametrit TFU,DEN,MAC,DFS, beta ja 1/v

foreach my $params (@params) {
    $Input=$base."_$params.tot";
    @Items=Apu::LueTaulu($Input);
    if ($params eq '3TFU')   {@TFnom=@Items;}# print "TFU=$TFnom[1][1][12]\n";}
    elsif($params eq '3DEN') {@DEnom=@Items;}# print "DEN=$DEnom[1][1][12]\n";}
    elsif($params eq 'MAC')  {@XSnom=@Items;}
    elsif($params eq 'DFS')  {@DFnom=@Items;}# print "DF=$DFnom[1][1][12][0]\n";}
    elsif($params eq 'beta') {@beta =@Items;}# print "B=$beta[1][1][12][0]\n";}
    elsif($params eq 'vinv') {@vinv =@Items;}# print "1/v=$vinv[1][1][12][0]\n";}
    else {print "Epäkelpo parametri\n";}
}

#---------------------------------------------------------------------#
# Luetaan kerrointapauksen parametrit ensin TF ja sitten jäähdytteen #
# tiheysmuutoksen parametrit                                         #
$ig=0;
foreach my $Tcase (@Tcases) {
    $Input=$base.$Tcase."_3TFU.tot";    # tiedosto jossa on 3D TF
    @Items=Apu::LueTaulu($Input);
    $TFkerr[$ig]= [@Items];
    $Input=$base.$Tcase."_MAC.tot";     # ja vastaavat vaikutusalat
    @Items=Apu::LueTaulu($Input);
    $XSkerrT[$ig]=[@Items];
    $ig++;
}
$ig=0;
foreach my $Dcase (@Dcases) {
    $Input=$base.$Dcase."_3DEN.tot";    # tiedosto jossa on 3D tiheydet
    @Items=Apu::LueTaulu($Input);
    $DEkerr[$ig]=[@Items];
    $Input=$base.$Dcase."_MAC.tot";     # ja vastaavat vaikutusalat
    @Items=Apu::LueTaulu($Input);
    $XSkerrD[$ig]=[@Items];
    $ig++;
}

#--------------------------------------------------------------------#
# Määritellään FITCO- kerroinmatriisin sekä NOMCOM-kortin parametrit #
$NXE=0;$NSM=0;
foreach $ig (0..6){
    $NCDopp[$ig]=2;
    $NCBor[$ig]=0;
    $NCDT[$ig]=0;
    $NCDe[$ig]=2;
    $NCTe[$ig]=0;
    $NCBu[$ig]=0;
}
    my $KSM=0; my $KXE=0;
    my $CB=0;
    my $alfa=0;
    my $modc=0;
    my $nhet=8;

#--------------------------------------------------------------------#
# Kirjoitetaan korttitiedosto                                        #

$Output= $base.".ntr";
open ($Out, ">", $Output) or die "Can't open $Output: $!\n";
my ($case,$m0);
if ($base =~ /aro/){
    $case='unrodded';
    $m0=-1;
}
elsif ($base =~ /ari/){
    $case='rodded';
    $m0=0;
}

foreach $K ($Kmin..$Kmax) {
    $nbij=1;
    $maternumb=$m0+$K*1000;
    foreach $J ($Jmin..$Jmax) {
       foreach $I ($Imin..$Imax) {
           if ($TFnom[$K][$J][$I]>0){
              $maternumb+=2;
              printf $Out "\/CASE\/   '$title, (I,J,K)=($I,$J,$K), $case'\n";
              printf $Out "\/MATER\/ %7g %8.3e %3d\n",
                           $maternumb,$Umassa[$K][$J][$I],$FTEMP;
              printNOMCOM($Out,$K,$J,$I);
              printFITCO($Out);
              printDIFCON($Out,$K,$J,$I);
              printDOPLER($Out,$K,$J,$I);
              printMODER($Out,$K,$J,$I);
           }
```

```
         }
      }
}
close $Out or die "$Out: $!\n";
print "HEXBU-muotoiset vaikutusalat kirjoitettu tiedostoon $Output!\n";
#----------------------------------------------------------------------#
# End of program sim2trab-kortti.pl                                    #
#----------------------------------------------------------------------#
#----------------------------------------------------------------------#
# Aliohjelmamäärittelyt                                                #
#----------------------------------------------------------------------#
sub printFITCO
{
#----------------------------------------------------------------------#
# Kirjoittaa FITCO-kortin. Parametrit määritetty pääskriptissä         #
    my $Out=shift;
    print $Out "/FITCO/   $NXE $NSM   @NCDopp    DOPPLER\n";
    print $Out "                $NCBor    BORON\n";
    print $Out "                @NCDT     MOD DENS AND TEMP\n";
    print $Out "                @NCDe     MOD DENS \n";
    print $Out "                @NCTe     MOD TEMP \n";
    print $Out "                @NCBu     MOD BUCKLING \n";

}

sub printNOMCOM
{
#----------------------------------------------------------------------#
# Kirjoittaa NOMCOM-kortin. Hakee nominaalitapauksen parametrit        #
# Pnom,Tnom,Dinom FluidParams-aliohjelman avulla                       #
    my $Out=shift;
    my $K=shift;
    my $J=shift;
    my $I=shift;
#     my $Input=$base.".out";
#     my ($Pnom,$Tnom,$Dinom);
#     ($Pnom,$Tnom,$Dinom)=FluidParams($Input,$K,$J,$I);
    printf $Out "/NOMCON/  PNOM=%7.3f  KXE=%2d KSM=%2d ", $Pnom,$KXE,$KSM;
    printf $Out " TFNOM=%6.1f TMNOM=%6.1f TINOM=%6.1f",
$TFnom[$K][$J][$I],$Tnom,$Tnom;
    print $Out "\n          ";
    printf $Out "CBNOM=%6.1f  DMNOM=%6.3f   DINOM=%6.3f",
$CB,$DEnom[$K][$J][$I],$Dinom;
    print $Out "\n          ";
    printf $Out "ALFA=%6.3f   MODC=%2d      NHET=%2d\n",$alfa,$modc,$nhet;
}

sub printDIFCON
{
#----------------------------------------------------------------------#
# Kirjoitaa DIFCON-kortin. Pitää huolta myös vaikutusalojen tulostus-#
# järjestyksestä, joka poikkeaa HEXBUn ja SIMULATEn välillä            #
# Kortille kirjoitetaan ensin vaikutusalat, sitten epäjatkuvuus-       #
# tekijät, 1/v:t ja lopuksi betat                                      #
# SF:n tilalle kirjoitetaan nuSF ja epsilonin tilalle Kappa/nu.        #
# TRAB3D:hen tehtävä vastaavat korjaukset (MA 22.1.2009)               #
    my $Out=shift;
    my $K=shift;
    my $J=shift;
    my $I=shift;
    my (@XSout1,@XSout2);
    @XSout1=(0,3,5,5);   #("D1","SA1","NF1");
    @XSout2=(1,4,6,6);   #("D2","SA2","NF2");
#     my $Kappa = 1.0; # CMS-nimi, HEXBUnimi=epsilon Pitäisi selvittää arvo
    my $Kappa = $XSnom[$K][$J][$I][7];
    printf $Out "/DIFCON/%12.1f", $bu;
    foreach my $ig (@XSout1){
       printf $Out "%12.4e",$XSnom[$K][$J][$I][$ig] ;
    }
#     printf $Out "%12.4e",  $XSnom[$K][$J][$I][5]*$XSnom[$K][$J][$I][7]; #SF1
```

```
    printf $Out "%12.4e\n        ",$XSnom[$K][$J][$I][2]; #SR12
    foreach my $ig (@XSout2){
       printf $Out "%12.4e",$XSnom[$K][$J][$I][$ig] ;
    }
#    printf $Out "%12.4e",$XSnom[$K][$J][$I][6]*$XSnom[$K][$J][$I][7];  # SF2
    printf $Out "%12.4e\n        ", $Kappa;


    @XSout1= (0,1,2,3); # nopeat epäjatkuvuustekijät F1W,F1S,F1E,F1N
    @XSout2= (4,5,6,7); # hitaat epäjatkuvuustekijät F2W,F2S,F2E,F2N
    foreach my $ig (@XSout1) {
       printf $Out "%12.4e",$DFnom[$K][$J][$I][$ig] ;
    }
    print $Out "\n        ";
    foreach my $ig (@XSout2) {
       printf $Out "%12.4e",$DFnom[$K][$J][$I][$ig] ;
    }
    print $Out "\n        ";

    printf $Out "%12.4e%12.4e",$vinv[$K][$J][$I][0],$vinv[$K][$J][$I][1];
    print $Out "\n        ";
    foreach my $ig (0..5){
       printf $Out "%12.4e",$beta[$K][$J][$I][$ig];
    }
    print $Out "\n";

}

sub printDOPLER
{
#-------------------------------------------------------------------#
# Kirjoittaa DOPLER-kortin polttoaineen lämpötilan kertoimille.      #
# Sovituskertoimet saadaan Apu::kerroin -moduulin avulla.            #
# Polttoaineen lämpötilakertoimet lasketaan neliöjuuri Tf:stä.       #
    my $Out=shift;
    my $K=shift;
    my $J=shift;
    my $I=shift;
    my (@c,@dy,@dx);
    my @XSout1=(0,3,5,2,1,4,6);   #("D1","SA1","NF1");

    printf $Out "/DOPLER/%12.1f", $bu;

    for my $ig (0..1) {
       $dx[$ig]=sqrt($TFkerr[$ig][$K][$J][$I])-sqrt($TFnom[$K][$J][$I]);
       }
    my $pi=1;

    foreach my $ig (@XSout1){
       for my $gi (0..1) {
           $dy[$gi]=$XSkerrT[$gi][$K][$J][$I][$ig]-$XSnom[$K][$J][$I][$ig];
       }
       @c=Apu::kerroin(@dy,@dx);

       printf $Out "%12.4e%12.4e", @c;
       $pi++;
       if ($pi==3){print $Out "\n        ";$pi=0;} # Rivin vaihto joka 6. kertoimen
jälkeen

    }
    print $Out "\n";
}

sub printMODER
{
#-------------------------------------------------------------------#
# Kirjoittaa MODER-kortin polttoaineen lämpötilan kertoimille.       #
# Sovituskertoimet saadaan Apu::kerroin -moduulin avulla.            #
# Moderaattorille diffuusiokertoimet lasketaan käyttäen 1/D-arvoja.  #
    my $Out=shift;
    my $K=shift;
    my $J=shift;
    my $I=shift;

    my (@c,@dy,@dx);
    my @XSout1=(0,3,5,2,1,4,6);   #("D1","SA1","NF1");

    printf $Out "/MODER/2%12.1f", $bu;

    for my $ig (0..1) {
       $dx[$ig]=$DEkerr[$ig][$K][$J][$I]-$DEnom[$K][$J][$I];
       }
    my $pi=1;
    foreach my $ig (@XSout1){
       if ($XSname[$ig]=~ "D"){
#           print "$XSname[$ig]: Käytetään 1/D:tä\n";
           for my $gi (0..1) {
              $dy[$gi]=1/$XSkerrD[$gi][$K][$J][$I][$ig]-1/$XSnom[$K][$J][$I][$ig];
           }
       }
       else {
           for my $gi (0..1) {
              $dy[$gi]=$XSkerrD[$gi][$K][$J][$I][$ig]-$XSnom[$K][$J][$I][$ig];
           }
       }
       @c=Apu::kerroin(@dy,@dx);
       printf $Out "%12.4e%12.4e", @c;
       $pi++;
       if ($pi==3){print $Out "\n        ";$pi=0;} # Rivin vaihto joka 6. kertoimen
jälkeen
    }
    print $Out "\n";

}

sub HaeOtsikko
{
#----------------------------------------------------------------------#
# Haetaan otsikko *.out-tiedostosta                                    #
    my $Inp=shift;
    open (my $In, "<", $Inp) or die "En löytänyt tiedostoa $Out: $!\n";
    my $line;
    my $otsikko;
    while ($line=<$In>) {
       last if($line =~ /TIT.PRO/);
    }
#    $line=~ /(\s+)('TIT.PRO')(\s+)([\w+\s*]+)(.*)/;
    $line=~ /(.*)(\')([\w+\s*]+)(\'.*)/;
#    print "rivi: $line\n";
#    print "eka=$1,toka=$2,kolmas=$3\n";
    $otsikko=$3;
    print "Otsikko on \'$otsikko\'\n";
    close $In or die "$In: $!\n";
    return($otsikko);
}

sub FluidParams
{
#----------------------------------------------------------------------#
# Haetaan Pnom=Pthermal/Core mass, Tnom, Dinom *.out-tiedostosta       #
    my $Inp=shift;
#    my $K=shift;
#    my $J=shift;
#    my $I=shift;
    open (my $In, "<", $Inp) or die "En löytänyt tiedostoa $Out: $!\n";
    my $line;
    my ($pi,$ti,$ci);
    my ($Pth,$MC,$TMI,$DMI);
    $pi=0;$ti=0;$ci=0;
    while ($line=<$In>) {
        if(($pi==0) && ($line =~ /Thermal Power /)){
            $pi++;
            $line =~/(.*)(\s{2,})(\d+\.\d+)(\D+)(\s+)(\d+\.\d+)(.*)/;
            $Pth=$3;
        }
        elsif(($ci==0) && ($line =~ /Core Loading/)){
            $ci++;
            $line =~/(.*)(\s{2,})(\d+\.\d+)(\D+)(\s+)(\d+\.\d+)(.*)/;
            $MC=$6;
        }
        elsif(($ti==0) && ($line =~ /Saturated Fluid/)){
            $ti++;
            $line=~

        /(\d+\.\d+)(\s{3,})(\d+\.\d+)(\s{3,})(\d+\.\d+)(\s{3,})(\d+\.\d+)(\s{3,})(.*)
/;
            $TMI=$1;
            $DMI=$7;
        }
    } # end while
#     if($K==1 && $J==int($Jmax/2) && $I==int($Jmax/2)){
    print "Terminen teho on $Pth MW\n";
    print "Sydämen massa on $MC tonnia\n";
    print "Jäähdytteen lämpötila on $TMI ja tiheys $DMI \n";
#           }

    return ($Pth/$MC,$TMI,$DMI);
}
```

A.6: Trab3DNeutr.pl

```
#! /usr/local/bin/perl
use strict;
use warnings;
use lib "/nfs/h201/a/users/prokmr/bin/";
use Apu;

my $Input = $ARGV[0];
if($#ARGV<0 || $Input =~ 'h'){
    print "Anna SIMULATEn inputfile (.inp)\n";
    exit;
}
elsif($#ARGV<1){
    if($Input =~ 'h'){
       print "Anna SIMULATEn inputfile (.inp)\n";
       exit;
    }
}

my ($base,$Pcase);

($base,$Pcase) = ($Input=~/(\w+)-(\w+)(.)/);
#print "$base, $Pcase\n";

my $path;
my ($Output,$Out,$In);

my $Npin=500;    # Nippujen määrä koko reaktorissa
my $sym= 2;      # Symmetria 1= koko reaktori,2=puolikas
my $maxK=25;     # noodien lukumäärä z-suunnassa
my $maxJ=13;     # puolikas taso
my $maxI=26;
my $blnk="";
my (@bu,@tfu);
my $Ni;
my ($Ii,$Ji,$Ki);
my $pi;
my @nshift=(0,-1,-3,-1,-1,-1,-1,0,0,-1,-1,0,0);  # nippujen aloituspaikat
my @nass=(6,8,14,16,18,20,22,22,22,24,26,26,26); # nippujen määrä rivillä
my $CRDmap="CRD-groups-OL1.txt";
my @CRDpos;
my $CRDinv;
my ($line,$Lz);

$Output=$base."-$Pcase.ntr";
open ($Out, ">", $Output) or die "Can't open $Output: $!\n";
$path="P".$Pcase;
$Input=$path."/".$base."-$Pcase-aro_3TFU.tot";
@tfu=Apu::LueTaulu($Input); # TFU > 0 osoittaa nipun paikan
#-------------------------------------------------------------------#
# Säätösauvojen sisääntyöntö SIMULATEsta, 100=täysin ulkona         #
# Syvyys tarkastettava joka tehtävälle erikseen                     #
$CRDpos[0]=0;
print " *** Tarkistithan säätösauvojen syvyyden! *** \n";
$CRDpos[1]=100;          # pikasulkusauvat
foreach $Ii (2..5){
    $CRDpos[$Ii]=100;}   # ulkokehän sauvat
$CRDpos[6]=100;
$CRDpos[7]=100;
$CRDpos[8]=100;
$CRDpos[9]=100;
if ($Pcase==60) {
    $CRDpos[10]=92;
    $CRDpos[11]=62;
    $CRDpos[12]=24;      # M2--ryhmä
    $CRDpos[13]=16;      # M1--ryhmä
    $CRDpos[14]=4;       # puoliryhmä
    $CRDpos[15]=4;       # ulompi ryhmä
}
elsif($Pcase==80)   {
    $CRDpos[10]=100;
    $CRDpos[11]=78;
    $CRDpos[12]=40;      # M2--ryhmä
    $CRDpos[13]=32;      # M1--ryhmä
    $CRDpos[14]=4;       # puoliryhmä
    $CRDpos[15]=4;       # ulompi ryhmä
}
elsif($Pcase==100)   {
    $CRDpos[10]=100;
    $CRDpos[11]=100;
    $CRDpos[12]=100;     # M2--ryhmä
    $CRDpos[13]=100;     # M1--ryhmä
    $CRDpos[14]=44;      # puoliryhmä
    $CRDpos[15]=4;       # ulompi ryhmä
}
$CRDpos[16]=4;           # sisempi ryhmä
$CRDpos[17]=4;           # keskisauva
$CRDinv=56*$CRDpos[1];
foreach $Ii (@CRDpos){$CRDinv+= 4*$Ii};
$CRDinv-=3*$CRDpos[17];
print " *** CRD-inventaari on $CRDinv *** \n";

#-------------------------------------------------------------------#
# Luetaan reaktorin korkeus out-tiedostosta                         #
$Input=~ s/_3TFU.tot/.out/;
open ($In, "<", $Input) or die "Can't open $Input: $!\n";
while($line=<$In>) {
    last if ($line=~ / Core Height/);         # Hae korkeus kohta
}

chomp $line;
$line =~ /^(.*)(Core Height)(\D+)(\d+\.\d+)(.*)$/;
$Lz=$4;                                       # Hae korkeus riviltä

#-------------------------------------------------------------------#
# Kirjoitetaan OPERAT-kortti
print $Out "/OPERAT/  ";
shift @CRDpos;  # poistetaan 0-elementti
$pi=10;
```

```
foreach $Ii (@CRDpos){
    printf  $Out "%6.2f ", (1-$Ii/100)*$Lz/100; # syvyys metreissä
    $pi+=7;
    if ($pi>=73) {
       print $Out "\n          ";
       $pi=10;
    }
}
print $Out "\n";

#-------------------------------------------------------------------#
# Kirjoitetaan DELAYE korttiin lambdat, jotka luetaan *-aro.out:ista #
#
while($line=<$In>) {
    last if ($line=~ /KIN.0-D/);          # Hae kohta
}
$line=<$In>;                              # Betat
$line=<$In>;                              # lambdad

if($line=~ / Lambda/) {
    @Items= split /\s+/,$line;
    pop @Items;    pop @Items;   pop @Items; # Poistetaan lopusta tekstit
    shift @Items;  shift @Items;             # ja alusta lisäksi tyhjät
    print $Out "/DELAYE/  \n";
    print $Out "@Items ";
    print $Out "\n";
}

close $In  or die "$In: $!\n";
#-------------------------------------------------------------------#
# Kirjoitetaan LAYOUT-kortti
print $Out "/LAYOUT/ ";
print $Out " NSYM=1 ";   # Symmetria 0=koko, 1=yläpuolisko,, 2=1/4
print $Out " NLAY=0 ";   # Symmetria-akselin paikka 0=nippujen välissä
print $Out " NBCON=0\n"; # Symmetriareuna 0=periodic, SELVITÄ
# rivin suhteellinen aloituspaikka ja nippujen määrä rivillä
print $Out "         NPOS=";
$pi=15;
foreach $Ji (1..$maxJ) {
    printf $Out "%3d %3d ", $nshift[$Ji-1], $nass[$Ji-1];
    $pi+=9;
    if ($pi>=70) {printf $Out "\n%15s"," "; $pi=15;}
}
print $Out "\n";

#-------------------------------------------------------------------#
# Kirjoitetaan POSITION-kortti
print $Out "/POSITN/ \n";
$blnk="    ";
$Ni=1;
foreach $Ji (1..$maxJ) {
    $pi=1;
    foreach $Ii (1..$maxI) {
       if ($tfu[1][$Ji][$Ii]>0){
           printf $Out "%4d", $Ni;
           $Ni++
           }
       else{print $Out "$blnk";}
       $pi+=4;
       if ($pi>=75) {print $Out "\n"; $pi=1;}
    }
    print $Out "\n";
}

#-------------------------------------------------------------------#
# Kirjoitetaan säätösauvadata RODPOS-korttiin käytetään säätösauva   #
# karttaa
print $Out "/RODPOS/  0\n";
open ($In, "<", $CRDmap) or die "Can't open $CRDmap: $!\n";
$pi=1;
```

```
while (my $line = <$In>) {
    print $Out "$line";
    $pi++;
    last if ($pi>$maxJ);
}
print  $Out "/RODCON/  ";
print  $Out "NCRA=0 ";           # crd-tyyppi,0=BWR,-1=sormi
printf $Out "HLROD=%4.1f ",$Lz;  # säätösauvan pituus (tässä reaktorin korkeus)
print  $Out "MRROD=1";           # materiaalinron lisäys unrodded->rodded
print  $Out "\n";

#------------------------------------------------------------------#
# Kirjoitetaan BOUNDR-kortti heijastinmateriaaleille               #

print $Out "/BOUNDR/  1 2 \n";                # ylä- ja alareunan heijastin

$pi=3;
$Ni=12;
my $Nbl;
$blnk="";                                     # eka rivi on käsiteltävä eri
tavalla
foreach $Ii (1..$Ni-1) {$blnk=$blnk."  ";}    # luodaan rivin alun blankko
print $Out "$blnk";
foreach $Ii (1..$nass[0]) {
    printf $Out "%2d", $pi;
}
print $Out "\n";                              # eka rivi käsitelty

foreach $Ji (1..$maxJ-1){
    $blnk="";
    $Nbl=$Ni-2;
    if ($nshift[$Ji]<-1) {
       $Nbl+=(1+$nshift[$Ji]);
    }

    foreach $Ii (1..$Nbl) {$blnk=$blnk."  ";}    # luodaan rivin alun blankko
    # käsitellään muut rivit
    print $Out "$blnk";                          # printtaa alun blankot
    if ($nshift[$Ji]==-1 || $nshift[$Ji-1]==1 ){$pi=4;} # Kulmapiste
    else {$pi=3;}                                # ei-kulmapiste
    print $Out " $pi";                           # printtaa arvo
    if (($nshift[$Ji]<-1) || ($nshift[$Ji-1]>1)) { # jos useampi arvo rivillä
       my $Mi;
       if (abs($nshift[$Ji]) > $nshift[$Ji-1]) {
           $Mi=abs($nshift[$Ji]);}
       else {$Mi=$nshift[$Ji-1];}
       foreach $Ii (2..$Mi-1) {
           print $Out " $pi";
       }
       $pi=4;
       print $Out " $pi";
    }
    foreach $Ii (1..$nass[$Ji-1]) {              # nippujen blankot
       print $Out "  ";
    }
    foreach $Ii (1..abs($nshift[$Ji])) {
       if ($Ii==1) {$pi=4;}
       else {$pi=3;}
       printf $Out " $pi";
    }
    if ($nshift[$Ji]==0) {
       $pi=3;
       printf $Out " $pi";
    }
    print $Out "\n";
    if ($Ni>0){$Ni+=$nshift[$Ji];}
}      # end for Ji=0->maxJ-1
# Viimeisen rivin käsittely
$Nbl=$Ni-2;
if ($nshift[$maxJ-1]<-1) {
    $Nbl+=(1+$nshift[$maxJ-1]);
}
$blnk="";
foreach $Ii (1..$Nbl) {$blnk=$blnk."  ";} # luodaan rivin alun blankko
print $Out "$blnk $pi";
foreach $Ii (1..$nass[$maxJ-1]) {  # nippujen blankot
    print $Out "  ";
}
print $Out " $pi\n";

# printtaa viimeinen rivi, jos koko reaktori
if($sym==1) {
    print $Out "$blnk";
    foreach $Ii (1..$nass[0]) {
       printf $Out "%2d ", $pi;
    }
    print $Out "\n";
}

#---------------------------------------------------------------------#
# Tähän väliin pitäisis tulla XSDATA-kortti
print $Out "/XSDATA/  ";
print $Out "NXSDAT=0 ";
print $Out "\n";

#---------------------------------------------------------------------#
# Kirjoitetaan nippumäärittelyyn tarvittava ASSTYP-kortti            #
#    print $Out "/ASSTYP/";
foreach  $Ni (1..$Npin/$sym) {
    printf $Out "/ASSTYP/ %3d", $Ni;  # number of fuel assembly type
    $pi=0;
    $blnk="";
    for ($Ki= $maxK; $Ki>=1;$Ki--){   # material number from top to bottom
       printf $Out "$blnk %6d",$Ki*1000+(2*$Ni-1);
       $pi++;
       if ($pi==7) {
           $blnk="            ";
           print $Out "\n$blnk";
           $pi=0;
           $blnk="";
       } # end if
    } # end for Ki
    print $Out "\n";
} # end for Ni
print $Out "/END4/\n";
#---------------------------------------------------------------------#
#---------------------------------------------------------------------#
# Kirjoitetaan nippujen lataukseen tarvittava LOAD-kortti            #
#---------------------------------------------------------------------#
# Palamaa ei laitetakaan LOAD-kortille tässä tapauksessa             #
#    $Input=$path."/".$base."-$Pcase-aro_3EXP.tot";
#    @bu=Apu::LueTaulu($Input);
$Ni=1;
$blnk="";
foreach $Ji (1..$maxJ) {
    foreach $Ii (1..$maxI) {
       if ($tfu[1][$Ji][$Ii]>0){
           printf $Out "/LOAD/ %3d %3d %3d", $Ni,$Ni,$Ni ; # assembly number,
location number
           $blnk="";                                     # and assembly type
           $pi=0;
           $Ni++;

#---------------------------------------------------------------------#
# Palamaa ei laitetakaan LOAD-kortille tässä tapauksessa             #
#             for ($Ki= $maxK; $Ki>=1;$Ki--){
#                 printf $Out "$blnk %6g", 1000*$bu[$Ki][$Ji][$Ii];
#                 $pi++;
#                 if ($pi==7) {
#                    $blnk="                  ";
#                   print $Out "\n$blnk";
#                   $pi=0;
#                   $blnk="";
#                 } # end if
#             } # end for Ki
#--------------------------------------------------------------------#
          print $Out "\n";
      } # end if fuel
   } # end foreach Ii
} # end foreach Ji
print $Out "/END5/\n";
#--------------------------------------------------------------------#
#    $Input=$path."/".$base."-$Pcase-aro_3SAM.tot";
#    @sam=Apu::LueTaulu($Input);
```

A.7: Apu.pm

```
#! /usr/local/bin/perl

use strict;
use warnings;
#use lib "/nfs/h201/a/users/prokmr/bin";

package Apu;
```

Part of file removed

```
sub kerroin
{
#--------------------------------------------------------------#
# Tämä aliohjelma tekee laskee toisen asteen polynomin         #
# dy = c1(dx) + c2 (dx)^2 kertoimet c1 ja c2, kun annetaan     #
# dy:t ja dx:t (2kpl kumpaakin).
    my ($dy1,$dy2,$dx1,$dx2)=@_;

    my @kerroin;
#    print "dy1=$dy1,dy2=$dy2,dx1=$dx1,dx2=$dx2\n";

    $kerroin[1]=($dy2/$dx2-$dy1/$dx1)/($dx2-$dx1);
    $kerroin[0]=$dy1/$dx1-$kerroin[1]*$dx1;
#    print "C1=$kerroin[0],C1=$kerroin[1]\n";
    return @kerroin;
}
#----------------------------------------------------------------------------#

sub LueTaulu
{
#--------------------------------------------------------------------#
# LueTaulu(filename) lukee tiedostosta "filename" taulukon muotoa    #
#                     K,J,I,Taulukko[K][J[I][G] G=1..                #
    my $Input1=shift;
    my (@Items,@Taulukko);
    my ($I,$J,$K,$ig);
    print "Luen tiedostoa $Input1 ja haen sieltä parametrit !\n";
    open(my $In1, "<", $Input1) or die "Can't open $Input1: $!";

  while(my $line=<$In1>){
      @Items= split /\s+/,$line;
      shift @Items;
      $K=shift @Items;
      $J=shift @Items;
      $I= shift @Items;
      if ($#Items>0) {  # jos taulukossa on useampi esim. vaikutusala
         $ig=0;
         foreach my $Items (@Items){
             $Taulukko[$K][$J][$I][$ig]=$Items;
             $ig++;
         }
      }
```

```
         else {
            foreach my $Items (@Items){$Taulukko[$K][$J][$I]=$Items;}
         }
     } # end while In

       close $In1 or die "$In1: $!";
       return (@Taulukko);
   }

   1;
```

Attachment B: Input files

B.1: sim-base-KMR2.inp

```
'COM' Tiedosto on saatu 21.11.2008 Simo Verralta.
'COM' Tehonnoston suunnittelulasku
'COM' KMR muokannut tiedostoa

'RES' '/cms/t1/c30/res/dist-boc.res' 0.0/

'ITE.ADP' 'OFF'/
'PRI.STA' 50*' '/
'TLM.EDT' 'ON' 50*' '/
'CMS.EDT' 'OFF'/

'TIT.CAS' ' ** T1 c28 26v-01 NORMAALI             PATTERN: 8212'/
'TIT.REF'     0    -1.0000/
'COR.MWT'  375.00,3700.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE', 0 -2/
'DEP.FPD' 8/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100  20 100  60 100 100             3*0/
'CRD.POS', 3 2*0,         100 100   4 100   4 100   4 100 100         2*0/
'CRD.POS', 4 1*0,     100 100   0 100   4 100   4 100   0 100 100     1*0/
'CRD.POS', 5 1*0,     100   4 100   4 100   4 100   4 100   4 100     1*0/
'CRD.POS', 6      100  60 100   4 100   4 100   4 100   4 100  20 100    /
'CRD.POS', 7      100 100   4 100   4 100   4 100   4 100   4 100 100    /
'CRD.POS', 8      100  20 100   4 100   4 100   4 100   4 100  60 100    /
'CRD.POS', 9 1*0,     100   4 100   4 100   4 100   4 100   4 100     1*0/
'CRD.POS',10 1*0,     100 100   0 100   4 100   4 100   0 100 100     1*0/
'CRD.POS',11 2*0,         100 100   4 100   4 100   4 100 100         2*0/
'CRD.POS',12 3*0,             100 100  60 100  20 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN: 8212'/
'TIT.REF'     0    -1.0000/
'COR.MWT'  325.00,3650.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE', -10/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100  20 100  60 100 100             3*0/
'CRD.POS', 3 2*0,         100 100   4 100   4 100   4 100 100         2*0/
'CRD.POS', 4 1*0,     100 100   0 100   4 100   4 100   0 100 100     1*0/
'CRD.POS', 5 1*0,     100   4 100   4 100   4 100   4 100   4 100     1*0/
'CRD.POS', 6      100  60 100   4 100   4 100   4 100   4 100  20 100    /
'CRD.POS', 7      100 100   4 100   4 100   4 100   4 100   4 100 100    /
'CRD.POS', 8      100  20 100   4 100   4 100   4 100   4 100  60 100    /
'CRD.POS', 9 1*0,     100   4 100   4 100   4 100   4 100   4 100     1*0/
'CRD.POS',10 1*0,     100 100   0 100   4 100   4 100   0 100 100     1*0/
'CRD.POS',11 2*0,         100 100   4 100   4 100   4 100 100         2*0/
'CRD.POS',12 3*0,             100 100  60 100  20 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN: 8772'/
'TIT.REF'     0    -1.0000/
'COR.MWT'  675.00,3850.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE', -1/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100  80 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100   4 100   4 100   4 100 100         2*0/
'CRD.POS', 4 1*0,     100 100  60 100   4 100  24 100  60 100 100     1*0/
'CRD.POS', 5 1*0,     100   4 100   4 100   4 100   4 100   4 100     1*0/
'CRD.POS', 6      100 100 100  24 100   4 100   4 100   4 100  80 100    /
'CRD.POS', 7      100 100   4 100   4 100   4 100   4 100   4 100 100    /
```

```
'CRD.POS', 8      100  80 100   4 100   4 100   4 100  24 100 100 100    /
'CRD.POS', 9 1*0,     100   4 100   4 100   4 100   4 100   4 100     1*0/
'CRD.POS',10 1*0,     100 100  60 100  24 100   4 100  60 100 100     1*0/
'CRD.POS',11 2*0,         100 100   4 100   4 100   4 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100  80 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:9660'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 1250.00,4000.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -1/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100   4 100   4 100   4 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100  66 100  92 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100   4 100   4 100   4 100   4 100   4 100     1*0/
'CRD.POS', 6      100 100 100  92 100  36 100  36 100  66 100 100 100    /
'CRD.POS', 7      100 100   4 100   4 100   4 100   4 100   4 100 100    /
'CRD.POS', 8      100 100 100  66 100  36 100  36 100  92 100 100 100    /
'CRD.POS', 9 1*0,     100   4 100   4 100   4 100   4 100   4 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100  92 100  66 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100   4 100   4 100   4 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:9996'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 1500.00,4100.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'COM' 'COR.TIN'  271.20/ CORE INLET TEMP. CELSIUS -**- DISABLED - USING INTERNAL
HEAT BALANCE
'DEP.HRS' 'AVE',  -1.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  14 100   4 100  22 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100  90 100 100 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100  22 100   4 100   4 100   4 100  14 100     1*0/
'CRD.POS', 6      100 100 100 100 100  60 100  60 100  90 100 100 100    /
'CRD.POS', 7      100 100   4 100   4 100   4 100   4 100   4 100 100    /
'CRD.POS', 8      100 100 100  90 100  60 100  60 100 100 100 100 100    /
'CRD.POS', 9 1*0,     100  14 100   4 100   4 100   4 100  22 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100 100 100  90 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100  22 100   4 100  14 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10028'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 1500.00,4000.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -1.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  16 100   4 100  24 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100  92 100 100 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100  24 100   4 100   4 100   4 100  16 100     1*0/
'CRD.POS', 6      100 100 100 100 100  62 100  62 100  92 100 100 100    /
'CRD.POS', 7      100 100   4 100   4 100   4 100   4 100   4 100 100    /
'CRD.POS', 8      100 100 100  92 100  62 100  62 100 100 100 100 100    /
'CRD.POS', 9 1*0,     100  16 100   4 100   4 100   4 100  24 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100 100 100  92 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100  24 100   4 100  16 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'WRE' 'Tehonnosto.res'/
'STA'/
```

```
'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10092'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 1625.0,4550.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -2.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  20 100   4 100  28 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100  96 100 100 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100  28 100   4 100   4 100   4 100  20 100     1*0/
'CRD.POS', 6      100 100 100 100 100  66 100  66 100  96 100 100 100    /
'CRD.POS', 7      100 100   4 100   4 100   4 100   4 100   4 100 100    /
'CRD.POS', 8      100 100 100  96 100  66 100  66 100 100 100 100 100    /
'CRD.POS', 9 1*0,     100  20 100   4 100   4 100   4 100  28 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100 100 100  96 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100  28 100   4 100  20 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10156'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 1750.00,5150.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -2.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  24 100   4 100  32 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100  32 100   4 100   4 100   4 100  24 100     1*0/
'CRD.POS', 6      100 100 100 100 100  70 100  70 100 100 100 100 100    /
'CRD.POS', 7      100 100   4 100   4 100   4 100   4 100   4 100 100    /
'CRD.POS', 8      100 100 100 100 100  70 100  70 100 100 100 100 100    /
'CRD.POS', 9 1*0,     100  24 100   4 100   4 100   4 100  32 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100  32 100   4 100  24 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10204'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 1875.00,5800.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -2.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  28 100   4 100  36 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100  36 100   4 100   4 100   4 100  28 100     1*0/
'CRD.POS', 6      100 100 100 100 100  74 100  74 100 100 100 100 100    /
'CRD.POS', 7      100 100   4 100   4 100   4 100   4 100   4 100 100    /
'CRD.POS', 8      100 100 100 100 100  74 100  74 100 100 100 100 100    /
'CRD.POS', 9 1*0,     100  28 100   4 100   4 100   4 100  36 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100  36 100   4 100  28 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10252'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2000.00,6450.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -2.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  32 100   4 100  40 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
```

```
'CRD.POS', 5 1*0,      100  40 100   4 100   4 100   4 100  32 100      1*0/
'CRD.POS', 6       100 100 100 100 100  78 100  78 100 100 100 100 100     /
'CRD.POS', 7       100 100   4 100   4 100   4 100   4 100   4 100 100     /
'CRD.POS', 8       100 100 100 100 100  78 100  78 100 100 100 100 100     /
'CRD.POS', 9 1*0,      100  32 100   4 100   4 100   4 100  40 100      1*0/
'CRD.POS',10 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS',11 2*0,          100 100  40 100   4 100  32 100 100          2*0/
'CRD.POS',12 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS',13 5*0,                      100 100 100                      5*0/
'WRE'/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10316'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2125.00,7000.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -2.0/
'CRD.POS', 1 5*0,                      100 100 100                      5*0/
'CRD.POS', 2 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS', 3 2*0,          100 100  36 100   4 100  44 100 100          2*0/
'CRD.POS', 4 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS', 5 1*0,      100  44 100   4 100   8 100   4 100  36 100      1*0/
'CRD.POS', 6       100 100 100 100 100  82 100  82 100 100 100 100 100     /
'CRD.POS', 7       100 100   4 100   8 100   4 100   8 100   4 100 100     /
'CRD.POS', 8       100 100 100 100 100  82 100  82 100 100 100 100 100     /
'CRD.POS', 9 1*0,      100  36 100   4 100   8 100   4 100  44 100      1*0/
'CRD.POS',10 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS',11 2*0,          100 100  44 100   4 100  36 100 100          2*0/
'CRD.POS',12 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS',13 5*0,                      100 100 100                      5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10348'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2187.50,7200.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -1.0/
'CRD.POS', 1 5*0,                      100 100 100                      5*0/
'CRD.POS', 2 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS', 3 2*0,          100 100  38 100   4 100  46 100 100          2*0/
'CRD.POS', 4 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS', 5 1*0,      100  46 100   4 100  10 100   4 100  38 100      1*0/
'CRD.POS', 6       100 100 100 100 100  84 100  84 100 100 100 100 100     /
'CRD.POS', 7       100 100   4 100  10 100   4 100  10 100   4 100 100     /
'CRD.POS', 8       100 100 100 100 100  84 100  84 100 100 100 100 100     /
'CRD.POS', 9 1*0,      100  38 100   4 100  10 100   4 100  46 100      1*0/
'CRD.POS',10 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS',11 2*0,          100 100  46 100   4 100  38 100 100          2*0/
'CRD.POS',12 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS',13 5*0,                      100 100 100                      5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10372'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2250.00,7500.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -1.0/
'CRD.POS', 1 5*0,                      100 100 100                      5*0/
'CRD.POS', 2 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS', 3 2*0,          100 100  40 100   4 100  46 100 100          2*0/
'CRD.POS', 4 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS', 5 1*0,      100  46 100   4 100  12 100   4 100  40 100      1*0/
'CRD.POS', 6       100 100 100 100 100  86 100  86 100 100 100 100 100     /
'CRD.POS', 7       100 100   4 100  12 100   4 100  12 100   4 100 100     /
'CRD.POS', 8       100 100 100 100 100  86 100  86 100 100 100 100 100     /
'CRD.POS', 9 1*0,      100  40 100   4 100  12 100   4 100  46 100      1*0/
'CRD.POS',10 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS',11 2*0,          100 100  46 100   4 100  40 100 100          2*0/
'CRD.POS',12 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS',13 5*0,                      100 100 100                      5*0/
```

```
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10420'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2250.00,7450.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -2.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  44 100   4 100  46 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100  46 100   4 100  16 100   4 100  44 100     1*0/
'CRD.POS', 6      100 100 100 100 100  90 100  90 100 100 100 100 100    /
'CRD.POS', 7      100 100   4 100  16 100   4 100  16 100   4 100 100    /
'CRD.POS', 8      100 100 100 100 100  90 100  90 100 100 100 100 100    /
'CRD.POS', 9 1*0,     100  44 100   4 100  16 100   4 100  46 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100  46 100   4 100  44 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10468'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2250.00,7350.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -2.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  46 100   4 100  48 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100  48 100   4 100  20 100   4 100  46 100     1*0/
'CRD.POS', 6      100 100 100 100 100  94 100  94 100 100 100 100 100    /
'CRD.POS', 7      100 100   4 100  20 100   4 100  20 100   4 100 100    /
'CRD.POS', 8      100 100 100 100 100  94 100  94 100 100 100 100 100    /
'CRD.POS', 9 1*0,     100  46 100   4 100  20 100   4 100  48 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100  48 100   4 100  46 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10516'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2250.00,7200.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -2.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  46 100   4 100  52 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100  52 100   4 100  24 100   4 100  46 100     1*0/
'CRD.POS', 6      100 100 100 100 100  98 100  98 100 100 100 100 100    /
'CRD.POS', 7      100 100   4 100  24 100   4 100  24 100   4 100 100    /
'CRD.POS', 8      100 100 100 100 100  98 100  98 100 100 100 100 100    /
'CRD.POS', 9 1*0,     100  46 100   4 100  24 100   4 100  52 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100  52 100   4 100  46 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10556'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2250.00,6950.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -2.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  48 100   4 100  54 100 100         2*0/
```

```
'CRD.POS', 4 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS', 5 1*0,      100  54 100   4 100  28 100   4 100  48 100      1*0/
'CRD.POS', 6       100 100 100 100 100 100 100 100 100 100 100 100 100    /
'CRD.POS', 7       100 100   4 100  28 100   4 100  28 100   4 100 100    /
'CRD.POS', 8       100 100 100 100 100 100 100 100 100 100 100 100 100    /
'CRD.POS', 9 1*0,      100  48 100   4 100  28 100   4 100  54 100      1*0/
'CRD.POS',10 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS',11 2*0,          100 100  54 100   4 100  48 100 100          2*0/
'CRD.POS',12 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS',13 5*0,                      100 100 100                      5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10604'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2250.00,6650.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -2.0/
'CRD.POS', 1 5*0,                      100 100 100                      5*0/
'CRD.POS', 2 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS', 3 2*0,          100 100  52 100   8 100  54 100 100          2*0/
'CRD.POS', 4 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS', 5 1*0,      100  54 100   4 100  32 100   4 100  52 100      1*0/
'CRD.POS', 6       100 100 100 100 100 100 100 100 100 100 100 100 100    /
'CRD.POS', 7       100 100   8 100  32 100   4 100  32 100   8 100 100    /
'CRD.POS', 8       100 100 100 100 100 100 100 100 100 100 100 100 100    /
'CRD.POS', 9 1*0,      100  52 100   4 100  32 100   4 100  54 100      1*0/
'CRD.POS',10 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS',11 2*0,          100 100  54 100   8 100  52 100 100          2*0/
'CRD.POS',12 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS',13 5*0,                      100 100 100                      5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10604'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 1725.00,4000.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -0.75/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10604'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2250.00,7000.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -0.75/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10636'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2250.00,6650.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -1.5/
'CRD.POS', 1 5*0,                      100 100 100                      5*0/
'CRD.POS', 2 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS', 3 2*0,          100 100  52 100  12 100  56 100 100          2*0/
'CRD.POS', 4 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS', 5 1*0,      100  56 100   4 100  34 100   4 100  52 100      1*0/
'CRD.POS', 6       100 100 100 100 100 100 100 100 100 100 100 100 100    /
'CRD.POS', 7       100 100  12 100  34 100   4 100  34 100  12 100 100    /
'CRD.POS', 8       100 100 100 100 100 100 100 100 100 100 100 100 100    /
'CRD.POS', 9 1*0,      100  52 100   4 100  34 100   4 100  56 100      1*0/
'CRD.POS',10 1*0,      100 100 100 100 100 100 100 100 100 100 100      1*0/
'CRD.POS',11 2*0,          100 100  56 100  12 100  52 100 100          2*0/
'CRD.POS',12 3*0,              100 100 100 100 100 100 100              3*0/
'CRD.POS',13 5*0,                      100 100 100                      5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10644'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2250.00,6500.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
```

```
'DEP.HRS' 'AVE',  -1.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  52 100  14 100  58 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100  58 100   4 100  34 100   4 100  52 100     1*0/
'CRD.POS', 6      100 100 100 100 100 100 100 100 100 100 100 100 100    /
'CRD.POS', 7      100 100  14 100  34 100   4 100  34 100  14 100 100    /
'CRD.POS', 8      100 100 100 100 100 100 100 100 100 100 100 100 100    /
'CRD.POS', 9 1*0,     100  52 100   4 100  34 100   4 100  58 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100  58 100  14 100  52 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' ' ** T1 c30 26v-01 NORMAALI             PATTERN:10664'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2250.00,6300.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -1.0/
'CRD.POS', 1 5*0,                     100 100 100                     5*0/
'CRD.POS', 2 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS', 3 2*0,         100 100  52 100  15 100  59 100 100         2*0/
'CRD.POS', 4 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS', 5 1*0,     100  59 100   4 100  35 100   4 100  52 100     1*0/
'CRD.POS', 6      100 100 100 100 100 100 100 100 100 100 100 100 100    /
'CRD.POS', 7      100 100  15 100  35 100   4 100  35 100  15 100 100    /
'CRD.POS', 8      100 100 100 100 100 100 100 100 100 100 100 100 100    /
'CRD.POS', 9 1*0,     100  52 100   4 100  35 100   4 100  59 100     1*0/
'CRD.POS',10 1*0,     100 100 100 100 100 100 100 100 100 100 100     1*0/
'CRD.POS',11 2*0,         100 100  59 100  15 100  52 100 100         2*0/
'CRD.POS',12 3*0,             100 100 100 100 100 100 100             3*0/
'CRD.POS',13 5*0,                     100 100 100                     5*0/
'STA'/

'TIT.CAS' '** t1 c30 26v-01 Normaali             PATTERN:10996'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2300,6700.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]  DOME
PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -1.0/
'STA'/

'TIT.CAS' '** t1 c30 26v-01 Normaali             PATTERN:10996'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2350.00,7000.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -4.0/
'STA'/

'TIT.CAS' '** t1 c30 26v-01 Normaali             PATTERN:10996'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2400.00,7250.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -4.0/
'STA'/

'TIT.CAS' '** t1 c30 26v-01 Normaali             PATTERN:10996'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2450.00,7500.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -4.0/
'STA'/

'TIT.CAS' '** t1 c30 26v-01 Normaali             PATTERN:10996'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2500.00,7750.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -4.0/
'STA'/
```

```
'TIT.CAS' '** t1 c30 26v-01 Normaali            PATTERN:10996'/
'TIT.REF'     0    -1.0000/
'COR.MWT' 2500.00,7900.00, 0.70000E+07/  POWER [MwTH], FLOW [kg/s], Pressure [Pa]
DOME PRESSURE=   70.00 BAR
'DEP.HRS' 'AVE',  -8.0/
'STA'/

'COM' Lisätty säätösauvakuvio 100%, otettu sim-dep.inp-tiedostosta
'COM' M.Paajasen ohje 01/2009 private communication
'TIT.CAS' '*** OL1 C30 ***' /
'TIT.REF'       1        0.000 /
'COR.MWT' 2500.00 , 7800.00 7.00000e+06 /
'CRD.SYM'  4 1/
'COM' Steps Withdrawn:   100 is fully withdrawn
'COM'      Row
'CRD.POS'  1  5*100                     100 100 100                        5*100/
'CRD.POS'  2  3*100             100 100 100 100 100 100 100                3*100/
'CRD.POS'  3  2*100         100 100 100 100   4 100 100 100 100            2*100/
'CRD.POS'  4  1*100     100 100 100 100 100 100 100 100 100 100 100        1*100/
'CRD.POS'  5  1*100     100 100 100   4 100  44 100   4 100 100 100        1*100/
'CRD.POS'  6        100 100 100 100 100 100 100 100 100 100 100 100 100        /
'CRD.POS'  7        100 100   4 100  44 100   4 100  44 100   4 100 100        /
'CRD.POS'  8        100 100 100 100 100 100 100 100 100 100 100 100 100        /
'CRD.POS'  9  1*100     100 100 100   4 100  44 100   4 100 100 100        1*100/
'CRD.POS' 10  1*100     100 100 100 100 100 100 100 100 100 100 100        1*100/
'CRD.POS' 11  2*100         100 100 100 100   4 100 100 100 100            2*100/
'CRD.POS' 12  3*100             100 100 100 100 100 100 100                3*100/
'CRD.POS' 13  5*100                     100 100 100                        5*100/
'DEP.HRS'  'AVE'        -1.0000/
'WRE'/
'STA'/

'COM' Ajetaan täyden tehon säätösauvakuviolla  50 tuntia
'TIT.CAS' '*** OL1 C30 *** Xe-poltto'/
'TIT.REF' 1 0.0/
'COR.MWT' 2500.00 , 7800.00, 7.00000e+06 /
'DEP.HRS' 'AVE',  69.0, -10.0, 119.0/
'STA'/

'STOP'/
'END'/
```

B.2: trans-60-aro.inp

```
'COM' **** OL1 C30 pumpputransientti ***
'COM' ---------------------------------------------------------|
'COM' Tiedosto, jolla lasketaan vaikutusaloja OL1 C30 aikana |
'COM' tapahtuneelle pumpputransienttilaskulle.              |
'COM' Tämä on perustapaus 60% teholla,                      |
'COM' kaikki säätösauvat ulkona                             |
'COM' ---------------------------------------------------------|

'RES' '/nfs/h201/a/users/prokmr/Toimeksiannot/STUK/OL1-pumpputransientti/base/Tehonnosto.res' 16.0000/

'PRI.LIS' 'ON'/                      * Tämä pitää olla, jotta PRIMAC kirjoittaa
'PRI.STA' 50*' '/  '3RPF'

'TIT.PRO' 'OL1 C30 pumpputransientti'/
'TIT.CAS' 'Perus ARO' /

'STA'/

'SAV.BAS','DEN','TFU','RPF'/         * Talleta tila
'SAV.LOK','HYD','FPD','TFU','XEN'/

'STA'/
```

```
'DEP.FPD', 0/
'USE.BAS' 'DEN', 1.0, 0.0,
          'TFU', 1.0, 0.0
          'RPF', 1.0, 0.0 /              * pidä TFU ja DEN  vakiona
'COM'

'CRD.ARO'/

'PRI.STA' '3DEN','3TFU','3XEN','3SAM','3EXP'/
'PRI.MAC' 'FULL' /                       * Vaikutusalat
'PRI.DFS' 'FULL'/                        * Epäjatkuvuustekijät
'KIN.EDT' 'ON' '1-D'/                    * Kirjoittaa kineetisen datan .kin-fileen

'STA'/

'END'/
```

B.3: trans-60-ari.inp

```
'COM' **** OL1 C30 pumpputransientti ***
'COM' ------------------------------------------------------|
'COM' Tiedosto, jolla lasketaan vaikutusaloja OL1 C30 aikana |
'COM' tapahtuneelle pumpputransienttilaskulle.              |
'COM' Tämä on perustapaus 60% teholla,                      |
'COM' kaikki säätösauvat sisään                             |
'COM' ------------------------------------------------------|

'RES' '/nfs/h201/a/users/prokmr/Toimeksiannot/STUK/OL1-pumpputransientti/base/Tehonnosto.res' 16.0000/

'PRI.LIS' 'ON'/                          * Tämä pitää olla, jotta PRIMAC kirjoittaa
'PRI.STA' 50*' '/  '3RPF'

'TIT.PRO' 'OL1 C30 pumpputransientti'/
'TIT.CAS' 'Perus ARI' /

'STA'/

'SAV.BAS','DEN','TFU','RPF'/             * Talleta tila
'SAV.LOK','HYD','FPD','TFU','XEN'/

'STA'/

'DEP.FPD', 0/                 * käytä edellisiä Xe,I,PM ja Sm-pitoisuuksia
'USE.BAS' 'DEN', 1.0, 0.0,
          'TFU', 1.0, 0.0
          'RPF', 1.0, 0.0 /              * pidä TFU, DEN ja RPF  vakiona
'COM'

'CRD.ARI'/

'PRI.STA' '3DEN','3TFU','3XEN','3SAM','3EXP'/
'PRI.MAC' 'FULL' /                       * Vaikutusalat
'PRI.DFS' 'FULL'/                        * Epäjatkuvuustekijät
'KIN.EDT' 'ON' '1-D'/                    * Kirjoittaa kineettisen datan .kin-fileen

'STA'/

'END'/
```

B.4: trans-80-aro-T-50.inp

```
'COM' **** OL1 C30 pumpputransientti ***
'COM' ------------------------------------------------------|
'COM' Tiedosto, jolla lasketaan vaikutusaloja OL1 C30 aikana |
'COM' tapahtuneelle pumpputransienttilaskulle.              |
'COM' Tämä on 80% teholla, -50K lämpötila                   |
'COM' kaikki säätösauvat ulkona                             |
'COM' ------------------------------------------------------|

'RES' '/nfs/h201/a/users/prokmr/Toimeksiannot/STUK/OL1-pumpputransientti/base/Tehonnosto.res' 24.0000/

'PRI.LIS' 'ON'/                        * Tämä pitää olla, jotta PRIMAC kirjoittaa
'PRI.STA' 50*' '/  '3RPF'

'TIT.PRO' 'OL1 C30 pumpputransientti'/
'TIT.CAS' 'ARO, TFU-50K' /

'STA'/

'SAV.BAS','DEN','TFU','RPF'/           * Talleta tila
'SAV.LOK','HYD','FPD','TFU','XEN'/

'STA'/

'DEP.FPD', 0/               * käytä edellisiä Xe,I,PM ja Sm-pitoisuuksia
'USE.BAS' 'DEN', 1.0, 0.0,
          'TFU', 1.0, -50.0
          'RPF', 1.0, 0.0 /            * pidä TFU, DEN ja RPF  vakiona
'COM'

'CRD.ARO'/

'PRI.STA' '3DEN','3TFU','3XEN','3SAM','3EXP'/
'PRI.MAC' 'FULL' /                     * Vaikutusalat
'PRI.DFS' 'FULL'/                      * Epäjatkuvuustekijät
'KIN.EDT' 'ON' '1-D'/                  * Kirjoittaa kineettisen datan .kin-fileen

'STA'/

'END'/
```

B.5: trans-80-aro-T+50.inp

```
'COM' **** OL1 C30 pumpputransientti ***
'COM' ------------------------------------------------------|
'COM' Tiedosto, jolla lasketaan vaikutusaloja OL1 C30 aikana |
'COM' tapahtuneelle pumpputransienttilaskulle.              |
'COM' Tämä on 80% teholla, +50K lämpötila                   |
'COM' kaikki säätösauvat ulkona                             |
'COM' ------------------------------------------------------|

'RES' '/nfs/h201/a/users/prokmr/Toimeksiannot/STUK/OL1-pumpputransientti/base/Tehonnosto.res' 24.0000/

'PRI.LIS' 'ON'/                        * Tämä pitää olla, jotta PRIMAC kirjoittaa
'PRI.STA' 50*' '/  '3RPF'

'TIT.PRO' 'OL1 C30 pumpputransientti'/
'TIT.CAS' 'ARO, TFU+50' /

'STA'/

'SAV.BAS','DEN','TFU','RPF'/           * Talleta tila
```

```
'SAV.LOK','HYD','FPD','TFU','XEN'/

'STA'/

'DEP.FPD', 0/                  * käytä edellisiä Xe,I,PM ja Sm-pitoisuuksia
'USE.BAS' 'DEN', 1.0, 0.0,
          'TFU', 1.0, 50.0
          'RPF', 1.0, 0.0 /               * pidä TFU, DEN ja RPF  vakiona
'COM'

'CRD.ARO'/

'PRI.STA' '3DEN','3TFU','3XEN','3SAM','3EXP'/
'PRI.MAC' 'FULL' /                        * Vaikutusalat
'PRI.DFS' 'FULL'/                         * Epäjatkuvuustekijät
'KIN.EDT' 'ON' '1-D'/                     * Kirjoittaa kineettisen datan .kin-fileen

'STA'/

'END'/
```

B.6: trans-100-ari-30n.inp

```
'COM' **** OL1 C30 pumpputransientti ***
'COM' --------------------------------------------------------|
'COM' Tiedosto, jolla lasketaan vaikutusaloja OL1 C30 aikana |
'COM' tapahtuneelle pumpputransienttilaskulle.               |
'COM' Tämä on 100% teholla, 30% tiheys                       |
'COM' kaikki säätösauvat sisään                              |
'COM' --------------------------------------------------------|

 'RES' '/nfs/h201/a/users/prokmr/Toimeksiannot/STUK/OL1-pumpputransientti/base/Tehonnosto.res' 69.0000/

'PRI.LIS' 'ON'/                           * Tämä pitää olla, jotta PRIMAC kirjoittaa
'PRI.STA' 50*' '/  '3RPF'

'TIT.PRO' 'OL1 C30 pumpputransientti'/
'TIT.CAS' 'ARI, 30% DEN' /

'STA'/

'SAV.BAS','DEN','TFU','RPF'/              * Talleta tila
'SAV.LOK','HYD','FPD','TFU','XEN'/

'STA'/

'DEP.FPD', 0/                  * käytä edellisiä Xe,I,PM ja Sm-pitoisuuksia
'USE.BAS' 'DEN', 0.30, 0.0,
          'TFU', 1.0, 0.0
          'RPF', 1.0, 0.0 /               * pidä TFU, DEN ja RPF  vakiona
'COM'

'CRD.ARI'/

'PRI.STA' '3DEN','3TFU','3XEN','3SAM','3EXP'/
'PRI.MAC' 'FULL' /                        * Vaikutusalat
'PRI.DFS' 'FULL'/                         * Epäjatkuvuustekijät
'KIN.EDT' 'ON' '1-D'/                     * Kirjoittaa kineettisen datan .kin-fileen

'STA'/

'END'/
```

B.7: trans-100-ari-60n.inp

```
'COM' **** OL1 C30 pumpputransientti ***
'COM' ------------------------------------------------------|
'COM' Tiedosto, jolla lasketaan vaikutusaloja OL1 C30 aikana |
'COM' tapahtuneelle pumpputransienttilaskulle.              |
'COM' Tämä on 100% teholla, 60% tiheys                      |
'COM' kaikki säätösauvat sisään                             |
'COM' ------------------------------------------------------|

'RES' '/nfs/h201/a/users/prokmr/Toimeksiannot/STUK/OL1-pumpputransientti/base/Tehonnosto.res' 69.0000/

'PRI.LIS' 'ON'/                           * Tämä pitää olla, jotta PRIMAC kirjoittaa
'PRI.STA' 50*' '/  '3RPF'

'TIT.PRO' 'OL1 C30 pumpputransientti'/
'TIT.CAS' 'ARI, 60% DEN' /

'STA'/

'SAV.BAS','DEN','TFU','RPF'/              * Talleta tila
'SAV.LOK','HYD','FPD','TFU','XEN'/

'STA'/

'DEP.FPD', 0/                 * käytä edellisiä Xe,I,PM ja Sm-pitoisuuksia
'USE.BAS' 'DEN', 0.60, 0.0,
          'TFU', 1.0, 0.0
          'RPF', 1.0, 0.0 /               * pidä TFU, DEN ja RPF  vakiona
'COM'

'CRD.ARI'/

'PRI.STA' '3DEN','3TFU','3XEN','3SAM','3EXP'/
'PRI.MAC' 'FULL' /                        * Vaikutusalat
'PRI.DFS' 'FULL'/                         * Epäjatkuvuustekijät
'KIN.EDT' 'ON' '1-D'/                     * Kirjoittaa kineettisen datan .kin-fileen

'STA'/

'END'/
```