Jül-4294

Institute for Energy Research (IEF)
Safety Research and Reactor Technology (IEF-6)

TINTE - A two-dimensional code for reactor dynamics

H. Gerwin, W. Scherer, A. Lauer, G. Strydom

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
FORSCHUNGSZENTRUM

TINTE - A two-dimensional code for reactor dynamics

*H. Gerwin, W. Scherer, A. Lauer, G. Strydom**

* *Pebble Bed Modular Reactor (Pty) Ltd, South Africa*

Berichte des Forschungszentrums Jülich; 4294
ISSN 0944-2952
Institute for Energy Research (IEF)
Safety Research and Reactor Technology (IEF-6) Jül-4294

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek, Verlag
D-52425 Jülich · Bundesrepublik Deutschland
☎ 02461 61-5220 · Telefax: 02461 61-6103 · e-mail: zb-publikation@fz-juelich.de

Contents

FIGURES

TABLES

ABREVIATIONS

This list contains the abbreviations used in this document.

Abbreviation or Acronym	Definition
AVR	Arbeitsgemeinschaft Versuchsreaktor
CPU	Central Processing Unit
FZJ	Forschungszentrum Juelich
HTGR	High Temperature Gas Cooled Reactor
ISR	Institut fuer Sicherheitsforschung und Reaktortechnik
PBMR	Pebble Bed Modular Reactor
SANA	Selbsttaetige Abfuhr der Nachwaerme
TINTE	**TI**me Dependent **N**eutronics and **TE**mperatures
VELUNA	Versuchsanlage fuer Lufteinbruch mit Naturkonvektion
VSOP	Very Superior Old Programs

1. INTRODUCTION

The TINTE main documentation consists of three parts, the first two of which/1/,/2/ have been published (in German) in the late eighties. In the first part the problems of modelling the nuclear and thermo-gas-dynamic behaviour of the primary circuit of a high temperature gas cooled reactor (HTGR) have been discussed in detail. It has been explained how the multi-connected system can be decomposed into single tasks to be solved separately. The solution of the total system is thus found by iteration of the partial results.

In the second part of the documentation some major applications of TINTE are demonstrated. Among them the analyses of dynamic reactor experiments performed at the AVR reactor /4/ are of special interest. These results play a major role in the TINTE validation process, and the very good conformance obtained with the experimental data validate the TINTE calculations to a considerable extent.

Earlier post-calculations of the AVR experiments with a previous version of the TINTE code have been described in /5/. Moreover, the basic algorithms as used in TINTE together with some applications have been shown in /4/.

Since not all of the capabilities of TINTE could be addressed in these analysis, the validation process was continued, e.g. with the evaluation of the VELUNA corrosion experiments /6/.

An addendum to the principal considerations of /1/ has been added as a supplement to /2/. Here the gas flow in an optional 1-D component and flow network is described, which may be used to enhance the 2-D reactor model for special situations. This flow network was necessary to model non-central pipes and other three-dimensional gas flow paths. It allows the description of co-axial pipes and a lumped parameter simulation of the primary side of heat exchangers or steam generators. One example in /4/ shows that under certain limitations even a simulation of a gas-gas heat exchanger and the incorporation of the secondary loop is possible with the aid of that flow-network.

In this addendum the possibility is also introduced to calculate the pressure inside the reactor from a given (fixed or variable in time) gas inventory. This is of relevance for accident analyses, where a failure of the pressure enclosure is assumed. If the pressure increases significantly gas may be removed from the system by burst discs or safety valves.

This document starts with a description of the TINTE code structure (Section 4), while Section 5 is dedicated to the description and interpretation of the main input data. Section 6 deals with the preparation of the nuclear data base, the generation of the cross-section polynomial expansions and the necessary interface codes. Aspects included here are the evaluation of nuclide vectors (prepared by burn-up codes) and the preparation of spectrum calculations with variation of temperatures, buckling and concentrations for spectrum relevant nuclides. User notes on the code installation and calculational procedures are presented in Section 7, while Sections 8 and 9 discuss the TINTE control options and output data options, respectively. Section 10 lists the changes made in the TINTE source code over the years.

The report also includes in the Appendices some newer algorithms for the treatment of special situations, while a description of the correlations used for the heat capacities and thermal conductivities are also given. Of special note here is Appendix E, which lists the detail of the ROMO model newly implemented in TINTE in 2004.

2. TINTE PROGRAM STRUCTURE

The partial problems of a full time-dependent, coupled neutronic and thermo-gas reactor calculation are solved separately in TINTE. The disadvantage of this procedure is that the overall solution of the multi-connected system can only be found by iteration; while on the other hand a modular code structure can now be achieved. The TINTE calculational flow is indicated in Figure 1, where Δt_{Ni} represents the nuclear time step width (see Section 4.3), and Δt_T the temperature calculation time step width (see Section 4.4).

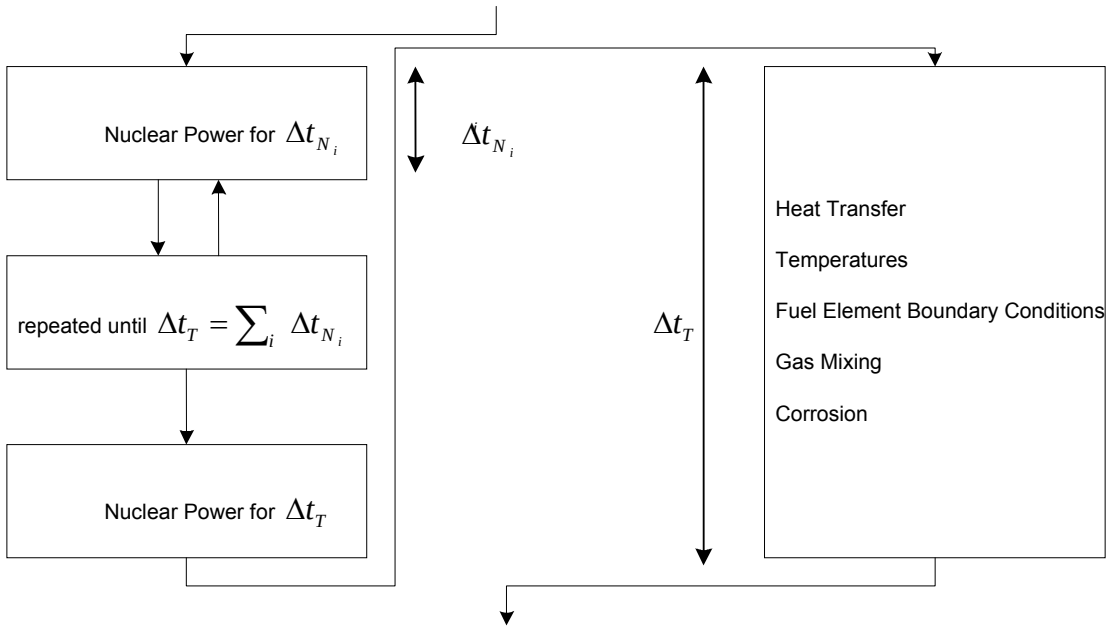

Figure 1: **Modular structure of TINTE - The diagram shows the principal time discretization.**

Most of the TINTE source is coded in FORTRAN-77, but some of the later subroutines and extensions have been added in FORTRAN-90 format. The calculations are generally performed using REAL*8 floating point variables, which was experienced to be mandatory to achieve a reasonable accuracy. Only the long input arrays for the nuclear cross-sections are stored in REAL*4 variables for memory saving aspects, and because the data accuracy for this data is not better than the REAL*4 level in any case.

In FORTRAN-77 all dimensions have to be fixed at compile time, which does not support a problem dependent variable dimensioning. The only way to overcome this deficiency at least partially is to store all multi-dimensional arrays successively on a large 1-D container array, and to pass the starting addresses of the individual arrays to the subroutines where the data are needed. TINTE uses this technique for all problem dependent arrays. The address calculation is performed in a way that easily allows for later modifications.

Except locally used variables like scalars or fixed-length arrays, all data are stored in this 1-D container-array. This is performed by the EQUIVALENCE statement for scalars, and by address passing as described above for all arrays. Data are divided into three logical groups corresponding to their location on the container vector, namely:

1. Data that describe the system (titles, control flags, accuracy limits, array lengths, geometry and material distribution data, material properties and nuclear cross-sections), including all necessary information for the pseudo-variable dimensioning.
2. Data that describe the current system status (temperatures, neutron fluxes, heat

source distributions, concentrations for delayed neutron production, for iodine, xenon and other delayed absorbers with their precursors, for decay heat production, gas temperatures and gas concentrations).

3. Working data arrays, which are used in specific programme areas only, and which may overlap for storage savings reasons.

Two different forms of restart options are provided in TINTE. In both cases the data of group 1 are stored as a first restart record. The second record contains the data of group 2 to characterise the current status of the system. The **documenting restart** describes the reactor transient over time. At the beginning of the transient the first restart record is written and at certain points in time (as defined by the user), the second restart record is repeatedly written, i.e. the existing restart file is appended by that record. Thus, a restart of a TINTE calculation is possible at each of these time points. This restart option needs a larger amount of storage for the restart file.

NOTE: The documenting restart should only be used for transient calculations; if the calculation contains an equilibrium case, this option does not always work properly.

The **normal** or **'overwriting' restart** overwrites the existing restart file completely at the user specified time points. Following an equilibrium calculation, it is performed by default if IERST ≠ 0 (see 'Initialisation Records', Section 6). During a transient calculation it is performed only if no documenting restart has been asked for. A restart is possible only from the last restart time point, and not from any of the previous specified time points.

Additionally to the restart output at the user defined time points, restart data is written at the end of a TINTE calculation (depending on the output options). This holds true for a normal end of a transient, except in the case where explicitly no output at all is asked for (i.e. *ZEITS* = -4 (see Section 8). The normal restart file is also written if a TINTE job ends because of lacking CPU- or REAL-time, or if it is forced to end by the "HE" command from the terminal during an interactive session.

TINTE contains all necessary coding in FORTRAN-77 format. The only additional operating-system dependent code is related to access the CPU- and real-time.

2.1 FLOW SCHEME OF A TINTE CALCULATION

Due to historical reasons the term 'card' is often used in this document. This is a synonym for "input record", which is not necessary limited to 80 columns.

2.1.1 Initializing Steps

TINTE starts by calling the subroutine TIRCPU, which manages the time control and time limitations that have been read from the first control card (see Section 8 for a description o the control options). Together with the actual CPU read-out, the final CPU time will be defined. The same procedure is done with the (optional) REAL-time limitations.

In the following calls to the subroutine the actual CPU- and REAL-time are compared to the final values, and both the expired CPU-time and the minimum of remaining CPU- and REAL-time (CPRE) are calculated.

After this initialisation step control is passed to the input control subroutine. Upon return from this subroutine the group 1 data (system defining data) is known, and in the case of a restart the group 2 data (current status of system) is also known. Only then a transient may be initiated immediately. If new input has been read for the group 2 data, only estimated values are available either from the input or from the additionally read restart file. It is however

possible to modify data in the .tn3 or .tn4 files and then start a transient from a restart file using the modified .tn3 and/or .tn4 data. The restriction is on the modification of the dimensioning arrays.

Upon return from the input control subroutine, the subroutine TIRSTI is called. Here the information is read to control the time-dependent development of some system parameters (also known in TINTE as a parameter “ramp”). These ramps always start from their current values; the input routine asks for the final value and the final time. TIRSTI also interprets general control statements given by the user, as indicated in Section 8. According to the ramps defined, and using information from the previous time intervals, the time step lengths for the nuclear and the temperature calculations are determined here. Also new (or extrapolated) values for anticipated nuclear items at the end of the time interval are prepared at this point, and stored in an auxiliary holder array. After the control items have been read in TIRSTI three options are possible:

- A restart output (i.e. a 2-D array output for the current situation including a heat balance table), followed by
    - a return to the TIRSTI call, or
    - the end of the calculation (optionally after output of scalar transient data)
- An equilibrium calculation.
- The start of a transient.

2.1.2 Output of the Restart Data and the 2-D Arrays to a File

For a documenting restart the second restart record containing the data for the current status is added. Otherwise the total restart output (after a REWIND command for the restart data file) is newly written. The output of the 2-D arrays is performed in subroutine TIW2DF. This subroutine is called for each of the 2-D arrays. TIW2DF writes the arrays to the standard output device (using a format that is by default adapted to a line length of 256 columns) and (if specified by output trigger IPP2D > 0) in binary format on the file IPP2D for later graphics processing.

Finally a detailed heat balance table is written to the standard output file. This table contains information on heat production in the different material regions, and on the heat transfer between them. The regions are identified by their material numbers. After the heat balance table has been written, TIRSTI is recalled or the calculation ends, depending on the value of the control parameter ZEITS (= -2 or -3, see Section 8). At the end of a transient some of the scalar variables calculated during the transient is written to the output file as well.

2.1.3 The Equilibrium Calculation

For an equilibrium case (ZEITS = -1) an iterative procedure between the nuclear and temperature calculations is performed. At the beginning of the iteration the accuracy limits are strongly relaxed and later on successively aggravated, depending on the level of convergence already achieved. After each iteration the remaining CPU-time CPRE is tested. A monitor output is performed for both the nuclear and the temperature calculations. The iteration procedure is finished if:

- The required convergence is reached.
- CPRE is less than 1.5 times the CPU-time consumed by the last iteration step.
- The user aborts the run with control command “HE” (see section 8.3 Table 26).

2.1.4 The Transient Calculation

For transient calculations (ZEITS > 0), TIRSTI prescribes the time steps, where the temperature time step is greater or equal to the nuclear time step. In the nuclear module it is first tested whether the previous time step has been aborted because of a too large fission power change during this time interval, in which case the working arrays are reset to the values of the beginning of the time interval. Upon the next TIRSTI call, a shorter nuclear time step is then based on the extrapolated power change, and the nuclear calculation is repeated with the shorter time step.

After a successful calculation of the nuclear time step, it is tested whether the end of the temperature interval has already been reached or whether a non-allowable power increase is expected for the next nuclear time interval. If both these criteria are negative, the nuclear time step is newly adjusted and the nuclear calculation proceeds. The length of the temperature time step is finally set to the sum of the previously calculated nuclear time steps, and the temperature module is called. After the end of a temperature time step a monitor output is written, and the next time step starts (see Figure 1 for a graphic representation of this process).

At arbitrary time points the user can request restart outputs, optionally combined with the output of the 2-D field arrays and the heat balance tables (ZEITS = -2, see Section 8 for more detail).

2.2 THE INPUT SUBROUTINES

The input of the data describing the system is controlled by the subroutine TIANFA. The following actions are performed:

- Initialisation of the predefined variables.
- Reading of the initialisation cards.
- Optional reading of a restart file and/or
- Reading of card input (see Section 6). This is mandatory for a new case (i.e. without a restart file input) or may serve as corrective action for some input data.

After reading of the initialisation cards the further input procession is structured into several steps:

1. Reading of variables to define the necessary dimension lengths with simultaneous temporary storage of the input blocks 1 to 5; the input stream of the (large) block 6 (NQ) is rewound after reading of its first part and read again later.
2. Address calculation for the positioning of the variable length field arrays in the container vector in COMMON-block TINCOM.
3. Final storage of the temporarily stored and the remaining input data, together with a first analysis and initialisation of items to describe the current system status.

If the initialisation cards ask for reading of an existing restart file, a different procedure takes place:

1. Start reading the restart file. Normally all information to start the calculation is available there. However it is possible to take some changes in the system into account by reading additional card input. Care has to be taken in such a case, not to change any dimension length (recalculation and readjusting of the addresses is not implemented in TINTE yet). There is one exception to this rule: For the nuclear cross section data the number of polynomial expansion coefficients may be reduced.
2. The new dimension lengths (if applicable) are compared to those from the restart file and if different special comments are given.
3. There is no new address calculation and no initialisation is performed because normally the information from the restart file is more accurate.
4. If any discrepancy has been detected with respect to the dimension boundaries, the calculation is terminated.

Before analysing the input, TIANFA calls the subroutine TIVERS. This subroutine identifies the TINTE version being used for the structure of the restart file, and is stored in a new restart file. On each reading of a restart file the version data are compared.

A predefinition of some variables follows (control flags, accuracy limits, etc.); those data may be overwritten by the user input. If the restart reading option IIRST has a valid (positive) value, the first record of the restart is read, the length of the second record is calculated and then this record is read as well. In case of a documenting restart, the appropriate value for the requested time point is selected. After this the reading and temporary storage of the input blocks 1 to 5 starts (if the file numbers ISTEU, IMESH, ITMAP, ITMEI and INMAP on the second initialisation card have been explicitly set to positive numbers, or for IIRST = 0 have been left to the predefined values).

The input data are tested on the necessary dimension boundaries. If IIRST > 0, the calculated dimensions are compared to those from the restart file, and any difference will result in an error stop of the code, with the exception of the reduction of dimensioning space in the nuclear cross section block, as described earlier.

Before the appropriate input routine for each particular block is activated, subroutine TIRTTL is called. This routine searches and analyses the corresponding title card on whether formatted or format-free input reading shall be performed. If title cards are found in the input stream that have not been requested the associated blocks are neglected. Those title cards are written to the standard output device to facilitate the input control for the user.

2.3 THE NUCLEAR CALCULATION

The nuclear calculations are controlled by the subroutine TINUKL. Figure 2 indicates the components of the nuclear calculation. The nuclear (fission) power calculation starts by calculating the temperature distribution inside the fuel elements by using extrapolated boundary conditions for the heat transfer to the gas. From this fuel element temperature data the moderator and fuel temperatures are derived, which are used to calculate the nuclear cross-sections. During the neutron flux iteration the Xe-135 concentrations are adjusted, and in transient cases an iteration between the power distribution and the fuel element temperatures takes place.

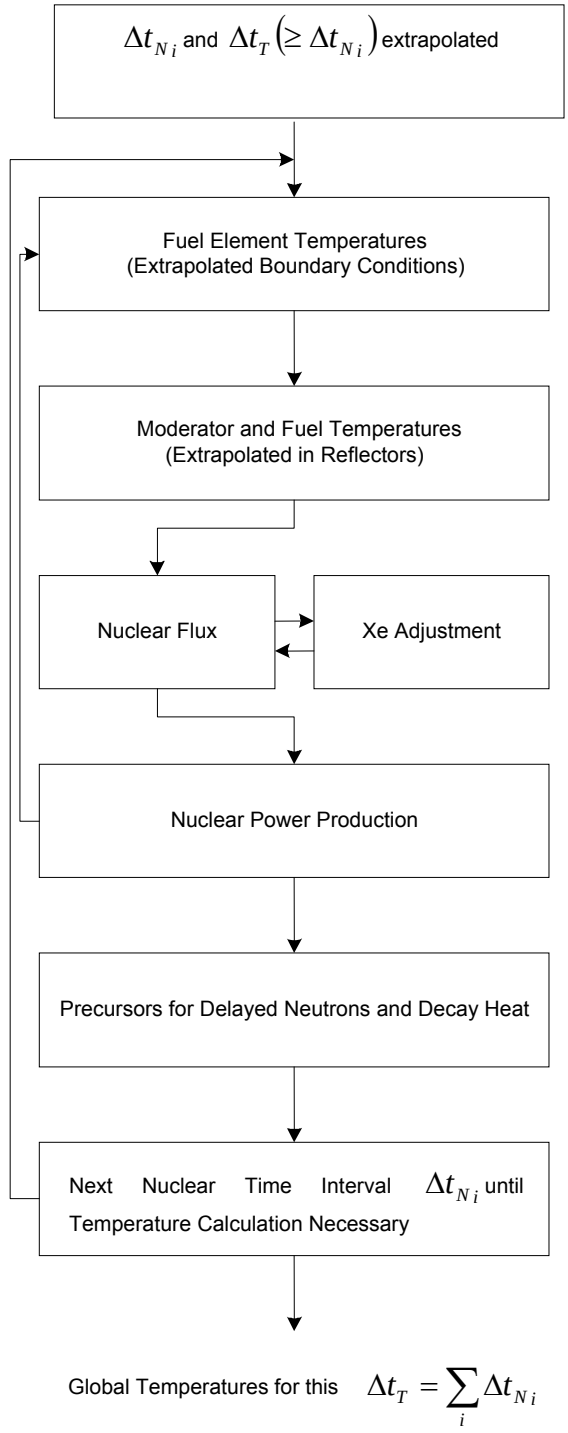

Figure 2: **Calculating the nuclear power distribution.**

At the beginning of each nuclear calculation, the subroutine TITKBE calculates the detailed heterogeneous temperature distribution in the fuel element zones, and these results are used to pre-calculate the fuel and moderator temperatures as averages over the fuel element zones. The equilibrium and transient calculations are controlled by the subroutines TINKEF and TINDYN, respectively. An iterative process between the fission power and fuel temperature is used during the transient, since the newly calculated power distribution (using extrapolated heat transfer boundary conditions at the fuel element surface) may significantly influence the temperature distribution inside the fuel elements.

After convergence of this process, the nuclear state at the end of the nuclear time interval will be fixed, and the nuclear heat production is added to the current temperature interval. As

described in /1/ (Section 4.5), a subdivision of the heat source into a local (inside the fuel elements) and a non-local part is made, and for both parts the spatial distributions are calculated.

It may happen during the transient calculation that the nuclear time interval provided in TIRSTI produces a change in the fission power that is too large. If this occurs, the nuclear calculation is repeated after a correction of the time step size.

Although the nuclear calculations for the equilibrium and transient cases are controlled by different subroutines, they are performed using the same algorithms and the same subroutines. This was done to minimise the numerical uncertainties that could occur when a converged equilibrium case is used as the starting point for a subsequent transient calculation.

In the calculation of the nuclear heat sources, delayed processes are of importance. These are mainly:

- Delayed neutrons.
- Decay heat, especially early after a fission event (including heat sources from breeding events).
- Production and loss of Xe-135 and of other strong neutron absorbers.

The production terms are related to the decay of the short-lived precursor nuclides, with a probability distribution strongly dependent on the parent heavy metal isotope.

Due to the lack of more detailed information, and with the large calculational effort required to track all possible delayed neutron precursors and decay heat producers, these precursors are collected into groups with similar decay constants, with measured or calculated yields.

For a decay precursor group structure of 24 groups (see /1/, section 4.1), a set of precursor data is predefined in TINTE. The data for the reactions I-135 $\rightarrow$ Xe-135, Pm-149 $\rightarrow$ Sm-149, Pm-151 $\rightarrow$ Sm-151 and Eu-151 $\rightarrow$ Gd-151 have also been predefined (total produced heat per fission, subdivisions into prompt, prompt-local and delayed-local parts, and the decay data for Th-232, Pa-233, U-239 and Np-239).

The data of the decay heat production are taken from /7/, subdivision data is extracted from /8/, and other information came from ENDF/B-IV /9/,/10/, (Tab. III, page 61), /11/,/12/, and the nuclide chart /13/.

Using these predefined data, effective yields and heat production terms are calculated for each nuclear material in the system. The dependency of these data on temperature and buckling is neglected, or may be partly considered by an iterative improvement of the reference parameters in the construction of the nuclear data base.

2.3.1 The Equilibrium Solution

The objective of the nuclear equilibrium calculation is the determination of the eigenvalue and eigenflux of the diffusion equation for the system under consideration. According to the solution algorithm, the so-called leakage-iteration process (see /14/), axial and radial local neutron leakage are determined as coupling terms for the 1-D flux iterations to finally get the 2-D r-z flux array.

After the determination of moderator and fuel temperatures from the temperature pre-calculation, the equilibrium calculation starts in the subroutine TINKEF with the evaluation of the nuclear cross-sections for these temperatures, and other parameters.

At the first call to TINKEF (except when reading a restart file) the fluxes and leakages have to be initialised. This is done in several steps:

- A radial 1-D calculation is performed in mid-height of the core using a small, fixed axial leakage.
- The resulting radial leakage is used in the axial calculation at mid-radius.
- The newly resulting axial leakage is used in the next round of calculations for all radial positions.
- The initialisation flux is obtained by the product of the radial and axial transversal terms.

Before the flux iteration procedure is started, feedback terms are evaluated, either with the initialisation flux or the last calculated flux. Although this is not necessary for the equilibrium cases, the procedure is included here to stay as close to the transient calculation methodology as possible. The feedback terms consist of a source term made up by the 1/v cross-section for each energy group, and for each spatial mesh. It is based on an artificial time interval, which is chosen to be very large at the beginning to accelerate convergence (during the iteration process this time interval is reduced down to 1 minute). This method does not influence the final solution but improves the convergence of the problem.

The flux iteration according to the leakage iteration method is performed in the subroutine TINITR. Within TINITR, the subroutine TINF1D is called for alternating radial and axial 1-D calculations at all iteration meshes. After each iteration step the acceleration parameter Υ (see /1/ Section 3.6.) is re-evaluated and the flux height is re-normalised. After a sufficient convergence has been reached, the subroutine TINJXE is called to readjust the concentrations of Xe-135 and other (delayed) strong absorbers. The iteration of the nuclear equilibrium calculation is terminated after reaching either the user defined accuracy level, or after 10 iterations.

TINKEF provides a monitor output on the flux convergence, and controls an additional repetition of the iteration with the newly determined feedback terms. More than one feedback repetition has been found to be impractical, because the convergence of the temperatures is much slower and produces larger flux changes.

After the flux calculation has finished, the fluxes and leakages are transferred from the working storage to their final places and the heat production is calculated. Additionally all data used in the transient calculation are determined, among them the equilibrium concentrations of the delayed neutron precursors and the decay heat producers.

2.3.2 The Transient Calculation

The time-dependent calculation is based on the conditions at the beginning of a time interval, and the goal is to determine the conditions at the end of this time interval. After the pre-calculation of fuel and moderator temperatures the process is controlled by TINDYN. The calculations are performed using the same subroutines as in the equilibrium case.

The evaluation of the parameters that are constant during the time step (performed in subroutine TINKNS) is followed by the calculation of the coupling terms in subroutine TINRKT. These terms consist of the l/v cross-section related source terms, and the source terms from the delayed neutrons.

The source terms are reduced with increasing time step length and therefore the maximum allowable nuclear time step is about 60 seconds, except in special cases (when the nuclear power is prescribed, or if an external source determines the flux in zero power analyses).

The flux iteration is once again performed in the subroutine TINITR. In contrast to the equilibrium case, the number of iterations is not limited, and the re-normalisation of the flux is

omitted (except in prescribed power calculations). The subroutine TINJXE again evaluates the time-dependent concentrations of Xe-135 and the other strong delayed absorbers. In TINJXE a special method to minimise the numerical errors is used (see /1/ Section 3.7, and A. Lauer /15/)). This method has been adapted to work with small time intervals by appropriate expansions of the exponential functions.

After convergence of the flux iteration a preliminary heat production shape is calculated. Together with the new boundary conditions at the fuel element surface, new fuel and moderator temperatures are determined, where after the flux calculation is repeated. After this process has converged the nuclear calculation for this nuclear time step is finally finished. Corresponding to the methodology followed in the equilibrium case, fluxes and leakages are moved to their storage arrays, and the data for the feedback terms and heat production distributions are prepared.

After possibly several nuclear time intervals, the cumulative nuclear heat production is finally determined. If applicable, the nuclear calculation continues with another nuclear time step until the end of a temperature time step is reached or until the change in fission power surpasses the prescribed limit.

2.4 THE TEMPERATURE CALCULATION

The temperature calculations due to the nuclear and additional heat sources are controlled by the subroutine TISPKR. The temperature calculations require many nested iteration processes between calculations of the gas flow and the gas temperatures, as well as between the heterogeneous fuel element temperature and the homogeneous solid material temperature calculations. In the latter case a heat transfer rule at the fuel element surface is determined during the nuclear calculations and is used in the form of an extrapolation. If more than one gas type is present, gas mixing calculations are also included, and the sources and sinks due to graphite corrosion are determined as well. Figure 3 indicates the temperature calculation process followed in TINTE.

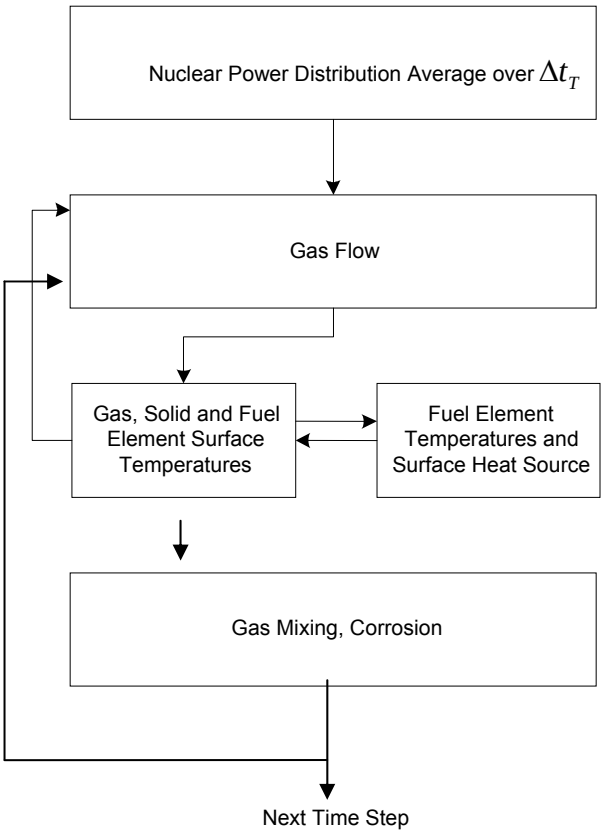

Figure 3: The temperature determination process in TINTE

The temperature determination is subdivided into four to five steps:

1. Determination of the coolant flow paths and rate under given boundary conditions, fixed temperatures and gravity.
2. Determination of the heat flow from the fuel element surface to the gas, and to the surrounding (fuel-) elements.
3. Determination of the heat conduction (including radiation) for the solid materials and inside the pebble bed.
4. Determination of the convective heat transport, i.e. the heat transport to the coolant and the heat transport within the coolant according to the flow distribution.
5. Determination of gas mixing and corrosive gas sources and sinks in case of multiple gas components.

All these aspects are strongly coupled, and because different solution algorithms are needed for the various partial problems, a simultaneous direct solution method is not currently possible. Interdependency arises as follows:

- The gravity influence on the flow depends on the gas density, and that in turn depends on the gas temperature, pressure and composition.
- The flow velocity influences both the heat transfer rates and the heat transport.
- The heat flow at the fuel element surface influences the fuel element surface temperature.
- The corrosion interactions influence the flow and the temperatures.

In the two-phase model (solid-coolant) the temperature calculations for both phases are performed in parallel (after the gas flow and the heat transfer from the fuel element surface have been determined). Unfortunately the gas flow is strongly influenced by temperature changes, which makes an iterative process necessary at this point. The consideration of the heterogeneous temperature distribution inside the fuel elements makes the problem even more complex. The biggest convergence problems arise for example if corrosive effects take place due to a massive air ingress. The convergence problems are caused by strong changes in the corrosive heat sources, for example when oxygen is transported to areas where hydrogen or carbon monoxide is still present (explosive mixture).

After the calculation of the gas and solid temperatures, an estimation is performed on the change of the boundary conditions at the fuel element surface. This result is used for the pre-calculation of the fuel and moderator temperatures for the next nuclear time step.

Note that the output data consists of the average temperatures, overall heat flows and heat balances for all materials present in the system.

2.4.1 The Coolant Flow

The dependency of the flow resistance on the flow velocity makes the flow calculation a non-linear problem, and despite the usage of a direct solution algorithm an iteration process is necessary. Due to the special solution method (see /1/ Section 7.3) used in TINTE, this iteration is needed only at the beginning of a calculation, since later in a transient the changes become sufficiently small.

The flow calculation is also iterated with the gas temperature determination, since changes in the gas temperature affects the gas flow directly. This is especially important if the flow is mainly governed by natural convection phenomena.

It has been found that for an accurate description of a system like a nuclear reactor, two-dimensional modelling is not always sufficient. It is impossible for example to describe in 2-D space the flows that cross in 3-D space without contact. Non-central and radial pipes may also be modelled in a crude way only. Therefore TINTE has been augmented by an additional 1-D flow network (see /2/ Appendix A2). This allows the modelling of components like burst-discs and safety valves as well. Restrictions for the proper operation of such components are described in Section 6.

In general, helium is used as a coolant in the pebble bed high temperature reactors which are the main focus of the TINTE code. As the inertia forces are small in this light gas, TINTE assumes a quasi-static treatment for the gas flow. Inside the reactor the pressure drop takes mainly place in the pebble bed. Since a detailed description of the flow between the pebbles is not possible, the pressure drop is taken from experimental results. In all other regions the pressure drop is sufficiently small so that it can be described by the rules of pipe flow. In large cavities this is a very crude approximation, and only gives reasonable results if the cavity is located between regions with a larger pressure drop (e.g. the void on top of a pebble bed core).

These assumptions finally lead to a diffusion-type equation for the calculation of the pressure drop, from which the gas mass flow is derived by a gradient procedure. Since the accuracy requirement for the pressure field is extremely high, an iterative solution algorithm has been abandoned and a direct solution method was chosen. The temperatures at the mesh boundaries, which are responsible for the calculation of the buoyancy forces, are influenced by flow changes. Therefore an extrapolation algorithm has also been implemented which stabilises the flow, especially in regions where the coolant is almost stagnant.

After the constants for the pressure field differential equation have been determined, the linear equation system for the flow meshes in the 2-D grid and the 1-D flow network are

established. Meshes coupled to only two neighbours are treated separately (the elimination processes needed here are trivial). This results in a reduced range of the system to be solved when using the Gaussian algorithm. The axial flow component is finally determined from the pressure drop and buoyancy data, and the radial flow component is derived by using the flow continuity equation.

2.4.2 Temperatures

2.4.2.1 The Gas Temperature

The problems in discretising the convection equation have been discussed in detail elsewhere /1/. For the leakage-iteration process a system of equations is established in rows and columns where there is gas flow for the boundary and central temperatures for each mesh (see /1/ Section 8.2 and the Appendix). As the convection heat transfer is strongly coupled to the solid temperatures via the fuel element surface temperatures, the equation system for the gas temperatures and the heat conduction in the solids can be simultaneously solved.

2.4.2.2 The Solid Temperature

The solution of the difference equations for the solid temperature (see /1/ equations 6.27 and 6.28) includes coupling terms to the time interval beginning. These coupling terms are set to zero for equilibrium calculations, but during the transient phase the material properties are evaluated at the time interval beginning and end. The subroutines XLAMT (where thermal conductivity values are calculated - see Appendix D) and WKPT (where specific heat capacity values are calculated - see Appendix C) include the information from /16/ and /17/ (see Section 4.4.3). Heat transport by radiation is accounted for in cavities and gaps, but contrary to the normal TINTE conventions this procedure needs the temperatures at the mesh boundaries (and not in the mesh centres).

A special method was introduced in TINTE to achieve this goal without too much effort, as described in Appendix A.

In core regions the temperatures inside the fuel elements may be calculated in detail (i.e. the heterogeneous temperature option). TINTE previously identified the 'global' (average) solid temperature with the fuel element surface temperature, and a time delay caused by heat capacity storage effects were taken into account only in the heterogeneous temperature calculation. In the global solid temperature calculation no heat capacity was used, but this approach caused numerical problems in certain situations (e.g. simulation of a loss of coolant accident), because the convergence stabilising effect of the heat capacity terms were missing in these examples. This was overcome by defining the global solid temperature as the temperature of the second-last shell (in the spatial discretisation of a fuel element), instead of the exact surface shell temperature. In doing so, the heat capacity of the last shell is available in the global solid temperature calculation, which leads to a vastly improved stabilisation of the global solid temperature calculation.

The determination of the delayed heat sources is performed in the subroutine TITKBE, which has already been used for the calculation of the fuel- and moderator temperatures in the nuclear module. For each material mesh the heterogeneous temperature distribution inside

the fuel elements is derived from the internal heat sources, and the boundary conditions (as calculated by the subroutine TITHET) from the effective heat transfer coefficient between the gas and its solid neighbours, and the effective average temperature in the neighbouring meshes. In the first call to TITHET this calculation is done with fixed (or extrapolated) heat transfer coefficients, while the effective neighbouring temperatures are iteratively determined from the surface temperatures of the surrounding fuel elements.

Using the delayed heat sources, the constants in the equation system for the global solid temperatures are corrected in regions with a heterogeneous temperature calculation. The equation system is solved simultaneously with the gas temperature system, as mentioned above. After reaching a certain convergence of the radial and axial solutions, the delayed heat source is incorporated into the iterative process, while repetitive calls to TITHET update the heat transfer using the data from the global temperature field.

In equilibrium cases all heat produced inside a fuel element has to leave that element, so that the iteration of the heat sources for the global temperature field are not necessary, and all TITHET calls are skipped. Only one call is made at the end of TITKBE to generate the boundary conditions and the temperature distribution inside the fuel elements as a base for the next fuel and moderator temperature calculation.

2.4.2.3 The temperature distribution inside fuel elements

The global temperatures in a pebble bed should theoretically be identified with the surface temperatures of the fuel elements, because interactions with the coolant, neighbouring fuel elements and reflectors take place there. However, due to the reasons discussed above, the temperatures of the second-last shell in the pebble model are used instead. From the coupling between the effective neighbouring pebble temperature (as calculated in the global temperature solving process) and the inner heat sources, the spatial temperature distribution in the pebbles is calculated (see /1/ Chapter 5). For transient cases the time dependency of the heat source to the last pebble shell is determined as well.

The heat conductivity equation for a sphere may easily be discretised using a standard finite difference scheme. However, at high fuel element power (e.g. during fast reactivity transients), a substantial number of meshes is necessary to obtain the exact temperature distribution inside a fuel element. As this algorithm is very often used in TINTE, an additionally solution method has been implemented, based on the analytic solution for the equilibrium case, where it is exact. It has been found that for most transients the temperature shape is modelled with sufficient accuracy. The advantage of this additional model is the possibility of a very rough subdivision of the sphere, which may be based on material properties only. The size of the last (i.e. outer) shell should be taken to be around 1 mm for a 6-cm sphere (i.e. 10% of the heat capacity are used for the global calculation). The geometry dependent data matrices can be calculated in advance during the input processing, so that only the inversion of a low-order matrix has to be performed.

Both these methods are programmed in TITKBE, with the course-mesh method (based on the analytic solution method) being the default method used. Note that a switch to the finite difference method may not be done by an input option in the present TINTE version; it can only be performed by setting the internal variable DIFF = .TRUE., and recompiling the code).

The present version of TINTE also allows the user to define several pebble types that differ in heat production and heat conductivity. By this method a multi-pass fuelling scheme with pebbles of different burn-up occurring at the same place, and even mixtures of fuel elements and pure graphite spheres, may be treated. This allows for better results of maximum fuel temperatures, especially in fast power excursions. By calling an internal loop within TITKBE sequentially for each pebble type, and by taking into account the heat transport by radiation between the different pebble types, an iterative process is started until convergence is

reached for the radiation transfer (see Appendix B for more detail). Note that a reduction in the number of fuel “batches” (or types) will usually result in a reduction in calculation time (especially during DLOFC scenarios), without paying a penalty in the accuracy of the maximum fuel temperatures.

2.4.2.4 Gas mixing and Corrosion

Gas mixing is a diffusion process for which a partial differential equation has to be solved. This type of equation (which is similar to other partial differential equations often used in TINTE) is valid for each component of the gas, assuming that the gas mixture is stagnant. If the gas mixture is in motion, an additional convection-type equation must also be solved. Spatial differences in the gas type constituents are due to different sources, and a variation of these gas sources with time leads to a frontal movement of the different concentrations, super-imposed by an uncertainty caused by diffusion effects. To describe these effects the mixing process has to be treated in a time-dependent manner.

Since the sum of all the gas components must be unity for NGAS components, the mixing equation has to be solved (NGAS – 1) times. For a single-component gas, this part of the algorithm may be skipped. The gas sources may be defined by the user, or they may be caused by corrosion effects. TINTE has been extended /18/ to deal with chemical processes that occur in large air and water ingress scenarios in HTRs, and the latest versions of TINTE (from 2007 onwards) can handle air and water ingress into a full reactor model successfully. The equations to describe the appropriate chemical reactions as functions of temperature, pressure and gas-component mixture have been implemented.

2.4.3 Material Properties

2.4.3.1 Specific Heat Capacity

The specific heat capacity of materials is dependent on temperature. The necessary laws (or correlations/rules) have been implemented from work performed at FZJ /16/, in the form of a data library in the subroutine WKPT. The specific heat capacity data “rules” are summarised in Appendix C.

2.4.3.2 Thermal Conductivity

The thermal (or heat) conductivity of materials is a function of temperature, while in nuclear reactors the conductivity is also dependent on the neutron dose rate and on the irradiation temperature. Heat conduction is therefore a non-linear, time-dependent problem. The complex heat conductivity functions were implemented into a data library from previously compiled information /17/, and to facilitate comparison with the THERMIX code, the library subroutine XLAMT has been transferred to TINTE, and amended by additional conductivity functions and materials.

The most important additional data are the tables based on measurements by Binkele /19/ for the heat conductivity of pebble graphite, depending on temperature and fast neutron dose. This data may also be replaced by user-defined input data, e.g. for the analysis of experiments with different graphite types (see ILAM = 52 in Appendix D), or to deal with pebbles from other materials or sizes.

A special model has been developed for the thermal conductivity of reflector graphite (described in Appendix E). This model is used by specifying material conductivity type 3, 83, 84, 201 or 210. In this case the irradiation temperature is relevant (i.e. the temperature that the solid had been irradiated at), as well as the current solid temperature and the irradiation

dose. The initial solid temperature guess (variable TVOR) is used for this purpose in the materials under consideration. In practice one estimate the irradiation temperature at the beginning, and after a converged equilibrium run have been achieved, replace these “guess” values by the average material temperatures as indicated in the TINTE equilibrium output. It is therefore assumed that the irradiation temperature is equal to the equilibrium operating temperature, which is a normally a good approximation.

2.4.3.3 Material Properties for Gases and Gas Mixtures

These material properties are used in flow and gas temperature calculations. The flow calculation needs the viscosity, and for natural convection cases the gas densities are of importance. In temperature and convection calculations, the heat conductivity and heat capacity are needed, while the velocity of gas mixing is dominated by gas-in-gas diffusion constants. These data have been collected in a computer code /20/ that has been directly implemented into TINTE.

2.4.3.4 Flow-Resistance and Heat-Transfer Rules

The calculation of the flow resistance for the pebble bed is performed according to the safety-related rules (KTA) for high-temperature reactors /21/. For regions with gas flow the rules as given in /1/ (Section 7.1) and /2/ (Appendix A1.3) are used (these rules are mainly taken from /23/).

3. TINTE INPUT DESCRIPTION

3.1 INTRODUCTION

The TINTE input consists of 7 blocks which collect the input parameters for several fields of interest. These are:

Block 1 : General Control Parameters.

Block 2 : Geometry and Spatial Mesh Definitions.

Block 3 : Material Assignment to the Mesh grid for Thermal-fluid Calculation.

Block 4 : Material Description for Thermal-fluid Materials.

Block 5 : Material Assignment to the Mesh Grid for Nuclear Calculations.

Block 6 : Nuclear Cross Section Data Base.

Block 7 : Control Commands for the Programme Operation.

The input data of blocks 1 to 5 are usually stored in a file with the extension **'.tn3'**. (However, block 1 for instance may also be submitted as console input on unit 5 if ISTEU – see Table 2 – specifies that. In this case any block 1 present in the .tn3 file is ignored.). They must follow the same sequence as described in sections 5.1 to 5.6. Blocks not used may be omitted, and if they are present out of sequence in the input stream, they are neglected by the code. The cross section data in block 6 have to be stored in a file with the extension **'.tn4'**. Block 7 data may either be fed in from the console in an interactive way by the user, or it may be stored in a file with the extension **'.tn1'**.

Due to the historical development of the code, in the following description the words 'card' and 'card image' are often used instead of 'record'. The input files must consist of fixed length records with at least 80 characters (i.e. in most of the records the data have to be placed on the first 80 positions). This is mandatory also in case of the modified format-free reading as described below. Only in the few cases of 'unrestricted format-free' reading is the record length unlimited.

Each of the data blocks begins with a title record. These title records serve as reading controls, and must start with a 3-character identification sequence. The rest of the title record may be used for comments by the user to describe the problem under consideration. The first two characters of the title record characterise the block, while the third defines the format of the input data. The **%** character is used to define formatted reading (where a blank field is interpreted as zero), while the ***** character defines a “modified format-free” reading procedure. In this latter case the data are not fixed to certain field lengths, and the parameters have to be separated by a colon or one or several blanks. Repeated data may be written in the form n*d (e.g. 5*0.0 instead of 0.0 0.0 0.0 0.0 0.0). Due to the internal programme interpretation, input data are read card by card- except those that are especially labelled as 'unrestricted format-free' data blocks. A restriction compared to the normal format-free reading was however necessary: the number of input items per card is fixed. If there is a deviation from the number of required items per card, the additional data are discarded, and missing data is added as zeros. On the other hand, this restriction allows the user to omit trailing zeros, similar than the rules for formatted input.

Each block is terminated with at least one blank card. These card(s) are necessary to define the block end, the readability of the input is enhanced, and later programme developments may use the blank cards as wildcard options.

Blocks 1 to 6 describe the reactor system under consideration. The contents will be saved in the restart file so that they need not to be repeated if a restart file is read. However, they may be read-in again in a restart case to replace certain input data. To accomplish this task, the

re-reading of the blocks is triggered by corresponding initialisation data in the additional block “Initialisation Records”. Note that non-used blocks may remain in the input stream, as they will automatically be ignored, according to their identification. The search and identification of the title cards is performed in subroutine TIRTTL, where the title cards are output to the display and output devices after being read. The title cards are also saved in the restart file (with default extension “**.rtn**”), and output after every reading of this file.

The reading of the first 6 blocks is performed in two steps: First the data are tested for necessary and/or consistent dimension boundaries, and then the first 5 blocks are stored in an intermediate storage array. The address calculation for the problem dependent variable dimensioning is then performed, before the data are written to their final destination storage arrays. Block 6 (normally very large) is read twice before the data are stored finally.

Due to the checks on consistent array boundaries, corrective updating of the data in a restart case must not lead to any changes in the array boundaries. The input is always tested on this requirement, and if discrepancies arise between the original and restart array boundaries, the calculation will be terminated. Block 7 controls the programme execution, and is mandatory for every case. The data in this block may be given by the user from the terminal in case of an interactive operation, or it may alternatively be given as a data file.

3.2 INITIALISATION RECORDS

The 7 input blocks are preceded by initialisation records that are read format-free from the standard input (FORTRAN file 5) in a dialog menu. This procedure has been found to be useful, since special calculation options may be controlled in this way without modification of the underlying input data. The **first initialisation record** provides information on the data files for input and output purposes (e.g. the user is asked to define the names of the .tn3, .tn4 and .tn1/.rtn input files, if provided). Note that when writing this file-name record, the missing file items have to be marked by a * character (see Section 6 for more detail).

The **second initialisation record** controls the input and output (I/O) requests. The control flags are integer numbers and their interpretation is given in Table 1.

Table 1: Control Flag Values for the Second Initialization Record

Flag Value	Interpretation
Integer number < 0	The corresponding input/output request is suppressed.
Integer number = 0	The flag default value for input or output control is used.
Integer number > 0	The entered number indicates a FORTRAN unit number that will be used for input or output.

Table 2 gives the flag default values and their interpretation by TINTE.

As an example, assume that the files labelled “PBMR400.tn3” and “PBMR400.tn4” contain the TINTE model input data.

Table 2: Initialisation Control Flag Interpretation

No.	Flag Name	Value	Meaning
1	IIRST	0	TINTE run is NOT started from restart file.
2	IORST	0	“Documenting” restart file will NOT be created.
3	IERST	10	“Overwriting restart file” WILL be created. Since the user did not specify the restart file name, TINTE will create it as PBMR400.RTN.
4	ISTEU	3 or 0 if IIRST≠0	Read Block 1 (starting with string ST*) from FORTRAN device 3, i.e. file PBMR400.tn3. If 0, reads Block 1 from restart file in IIRST. Note: If a calculation is done not using one of the blocks, a -1 has to be given at the corresponding place. For example, if temperature calculations are performed without any nuclear data, there is no .tn4 file available nor is an INMAP block present. Therefore, INMAP=-1, ISIGM=-1 must be given.
5	IMESH	3 or 0 if IIRST≠0	Read Block 2 (starting with string GM%) from FORTRAN device 3, i.e. file PBMR400.tn3. If 0, reads Block 2 from restart file in IIRST.
6	ITMAP	3 or 0 if IIRST≠0	Read Block 3 (starting with string TZ*) from FORTRAN device 3, i.e file PBMR400.tn3. If 0, reads Block 3 from restart file in IIRST.
7	ITMEI	3 or 0 if IIRST≠0	Read Block 4 (starting with string TM*) from FORTRAN device 3, i.e file PBMR400.tn3. If 0, reads Block 4 from restart file in IIRST.
8	INMAP	3 or 0 if IIRST≠0	Read Block 5 (starting with string NZ*) from FORTRAN device 3, i.e file PBMR400.tn3. If 0, reads Block 5 from restart file in IIRST.
9	ISIGM	4 or 0 if IIRST≠0	Read Block 6 (starting with string NQ% from FORTRAN device 4, i.e file PBMR400.tn4. If 0, reads Block 6 from restart file in IIRST.
10	IZEIT	5 (or 1, if .tn1 file name is given at first card)	Read Block 7 (starting with string TT*) from FORTRAN device 5, i.e. console. If 1, reads Block 7 from file with extension ” .tn1”.
11	IPRINT	15	Produces general print format output file. The user can select the file name when starting TINTE from the DOS command prompt, e.g. PBMR400.out.
12	IPMAP	15	User entered 0, but since an output file has been created (PBMR400.O), TINTE prints in it a picture of reactor nodalization meshes.
13	IPCOMP	8	Produces a file with extension “.ttb” that is used in a procedure to optimize the nuclear cross-sections. User entered default value, i.e. 0, so TINTE creates file PBMR400.TTB on FORTRAN device #8.
14	IPZEIT	9	Produces a file with extension “.ptr” that contains transient scalar output for post-processing as graphics. User entered default value, i.e. 0, so TINTE will create file DUMMY.PTR on FORTRAN device 9, but will NOT write in it.

15	IPMON	0	User entered default value, i.e. 0, so TINTE does NOT create a file with detailed iteration monitoring. Note that this option produces a huge output file with detailed iteration monitoring. To avoid memory storage problems, this must be used carefully!
16	IPMOFU	15	User entered 15, so TINTE writes in output file PBMR400.O (FORTRAN device#15) moderator and fuel temperatures.
17	IPGASD	15	User entered 15, so TINTE writes in output file PBMR400.O (FORTRAN device#15) gas pressure, mass flow rate, and flow paths.
18	IPGAST	15	User did not enter device number, so TINTE entered 0 which signals the use of the flag default value 15. As a result, TINTE writes in the output file PBMR400.O (FORTRAN device #15) gas temperatures.
19	IPFEST	15	User did not enter device number, so TINTE entered 0 which signals the use of the flag default value 15. As a result, TINTE writes in the output file PBMR400.O (FORTRAN device #15) the temperatures of solids, i.e. pebbles, reactor component, etc.
20	IPPOW	15	User did not enter device number, so TINTE entered 0 which signals the use of the flag default value 15. As a result, TINTE writes in the output file PBMR400.O (FORTRAN device #15) data describing the convective heat transfer, heat sources (local, non-local, and external), and a heat balance table for all materials.
21	IPP2D	12	User did not enter device number, so TINTE entered 0 which signals the use of the flag default value 12. As a result, TINTE creates a file with extension “P2D” (FORTRAN device #12) that contains two-dimensional array with data for post-processing as graphics.

Examples:

1. An input line for a normal run, using all the default options in TINTE, could be written as

0 0 10 12*0 3*15

2. An input line for a run without any neutronics input (e.g. for pure TH benchmarks):

2*0 10 5*3 2*-1 2*0 15 3*0 2*15

3. An input line for a run without any temperature input (e.g. for pure neutronics benchmarks):

2*0 10 2*3 2*-1 8*0 2*15

The “documenting” (IORST) restart file describes the reactor transient over time. A restart record is written at the beginning of the transient. Other restart records are written at times indicated by the user. Every new record is appended to the previous record– this allows a TINTE restart at any of these times, but can result in producing a very big file that may exceed the available memory. The “overwriting” (IERST) restart file deletes the existing restart file at times defined by the user. During a transient calculation, the “overwriting” restart file can be used only if no “documenting” restart file has been requested. Since the “overwriting” keeps only the last restart record, a TINTE restart is possible only from the last restart time point. Simultaneous use of both restart options, i.e. IIRST and IERST, makes

sense only if both equilibrium and transient calculations have to be performed simultaneously in a single TINTE run. The equilibrium results will then be saved on IERST, i.e. in an “overwriting” restart file. After an output is sent to IORST – this is done only during a transient calculation – IERST $\neq$0 will be discarded.

If TINTE is run from a restart file, i.e. IIRST > 0, a check is made to establish if it is a “documenting” restart file, (i.e. has been generated with flag IORST >0) or an “overwriting” restart file, (i.e. it has been generated with IERST>0).

If the restart file is a “documenting” one, a list is given with the times when restart records have been made. The user has to choose the time point when the next restart record has to be made. If the restart file is recognized as an “overwriting” one, it is immediately read as input by TINTE.

An additional output file exists that does not have any I/O request flag; it is the console output file with the standard FORTRAN unit number 6. Note that output control flag values (FORTRAN unit numbers) of items no. 12, 16 and 17 can be chosen equal to that of no. 11, in order to collect all print output into one single file.

These data items have to be placed on a single record. If less than 21 items are present, the missing will be set to zero. If all predefined data should hold, a blank card will be sufficient. Following these initialisation records the corresponding restart file or/and the 6 input blocks are read.

The **third and last** initialisation record asks for limitations in the computer time and for the print output line width for the run under consideration. CPU time and/or real time can be given (in minutes). A zero or blank (RETURN) indicates no limitation. If the remaining time is less than 1.5 times the time span used for the last time-step the code finalises the task with normal output and writes a final restart file, if asked for. The third item on this card defines the line width for 2D-array print formatting in output file IPRINT. A zero or blank (RETURN) indicates the default value of 256.

3.3 BLOCK 1: GENERAL CONTROL DATA

Note: In Sections 5.3 to 5.6, examples of the input blocks are given at the end of each Section.

The title card of this block starts with 'ST%' or with 'ST*'. The remainder of the card may be filled with arbitrary text. For formatted input (title card starts with “ST%”) the first card of the block is read with format (8A1,2X,6E10.0). The eight CHARACTER-Variables are used to initialise the following LOGICAL control variables, which are predefined with. TRUE..

They will be changed to .FALSE. only if a “F” (capital!) is read in; for any other input they remain .TRUE.. The meaning of the control variables for .TRUE. is listed in Table 3.

Table 3: Control Flag Interpretations for Input Block 1

No.	Control Variable	Default Value	Description
1	NUKL	.T.	The nuclear programme module is called.
2	MOFU	.T.	The calculations for the moderator and fuel temperatures are performed.
3	TXTR	.T.	The boundary conditions in calculating the moderator and fuel temperatures are linearly extrapolated in time. Choosing .FALSE. here results in constant boundary conditions.
4	TEMP	.T.	The temperature programme module is called.

No.	Control Variable	Default Value	Description
5	SGAS	.T.	The gas flow is calculated. The .FALSE. option may only be used in a restart case or together with TGAS =.FALSE, but convergence cannot be guaranteed.
6	TGAS	.T.	The gas temperature is calculated.
7	TBEK	.T.	The calculation of the fuel element temperature is repeated during the global solid temperature calculation. The .FALSE option means that the heterogeneous temperature calculation is not repeated; instead the heat source is taken from the calculation of the fuel and moderator temperatures. (A .FALSE. value for this option makes sense only in equilibrium calculations: during transients it produces the heat source at the end of the time interval instead of the average).
8	TFST	.T.	The solid temperature is calculated. The option of .FALSE. here means that the initial temperature values (TVOR) for the solids remains unchanged.

Even if the title card starts with “ST*”, these 8 CHARACTER-variables are read in formatted format.

Examples:

1. The input flags for the pure neutronics Dodd’s benchmark are: TTTTTTFF.

2. The input flags for a pure TH conduction problem are: FFFTTTTT

3. The same pure TH problem can also be run with any convective heat calculations: FFFTTFFT

The format-free reading in this case starts from column 11 of the first card and is performed separately for each of the other cards; omitted items will not start reading of the next card but the corresponding parameters will be interpreted as zeros (i.e. the default values will be used). The following six variables (with the default values in brackets) are indicated in Table 4.

Table 4: First Card Variables of Input Block 1

No.	Variable	Default Value and Unit	Description
1	STATPW	1.0	The total nuclear power of the reactor [MW] (without any 'imposed' power assigned to certain materials like the (negative) power of a steam generator region, or the decay heat power in the discharge tube).
2	FXSKF	1.0E-10 [fissions/s]	See next parameter.
3	FXSOR	0.0 [neutrons/cm^{3}s]	The fixed external neutron source, which stabilises the neutron flux at a certain (low) level in case of reactor shut-down. This is necessary to avoid numerical problems and to allow a reactor restart at any time. The source is assumed to be homogeneously distributed all over the core. The total strength is either FXSKF times the total nuclear fission rate or (if FXSOR $\neq$ 0) equal to FXSOR.
4	DLTFT	2.0 K	See next parameter.
5	DLTGT	5.0 K	DLTFT and DLTGT are the limits for the maximum expected solid and gas temperature changes during a single time step. They are used to define the time step length.

No.	Variable	Default Value and Unit	Description
6	DLTON	0.05	This factor is the limit for the maximum relative power change per time step. It is used to define the nuclear time step length, to test the actual power change in the nuclear time step (if necessary with recalculation of the neutronics using a smaller time step), and to control the endpoint of the temperature interval if a desired power change is reached.

Note that omitted input values or zeros results in the old (default or restart) values. This holds true for the second card as well (Table 5), where accuracy limits are read in format (7E10.0).

Table 5: Second Card Variables of Input Block 1

No.	Variable	Default Value and Unit	Description
1	EPSNI	2E-5	The relative convergence of the nuclear calculation with given temperature boundary conditions (transient calculations). User Hints: 1. If in DLOFC calculations re-criticality is expected a value of about 1E-9 should be given here. This will lead to a much more stable transient with smooth re-criticality oscillations. However, this tight criterion might lead to instabilities in other transients. 2. Tighter values for this parameter (and the next one EPSNO) can be used to improve stability and convergence for large input models, e.g. changing from 2E-5 to 1E-6..
2	EPSNO	lE-4	The relative (outer) convergence of the equilibrium nuclear calculation (steady state calculations).
3	EPSFT	0.2 K	The solid temperature relative convergence.
4	EPSBT	0.2 K	The fuel element temperature relative convergence (for the convergence test of the solid/fuel element iteration the value EPSBT /16 is used).
5	EPSGT	0.2 K	The gas temperature relative convergence.

The third card may be blank, if the cooling gas is pure helium and the standard control for the gas properties shall be used. This card is structured the same way as card 1 (8Al,2X,I10.0) and starts again with eight CHARACTER-variables, where an input “F” changes the .TRUE. predefined data to .FALSE. The items are indicated in Table 6.

Table 6: Third Card Variables of Input Block 1

No.	Variable	Default Value	Description
1	RGAS	.T.	The real-gas correction factor according to Redlich and Kwong is applied to the general gas equation.
2	STKO	.T.	The gas viscosity is calculated by applying the Stockmeier correction to the Lenard-Jones potentials.
3	WILK	.T.	The viscosity and heat conductivity of gas mixtures interaction parameters are calculated according to Wilke (.F. calculates according to Henning-Zipperer).

No.	Variable	Default Value	Description
4	WEMA	.T.	Gas diffusion coefficients are calculated according to Weissman and Mason (.F. calculates according to Chapman and Enskog) .
5	DIFU	.T.	Diffusion is accounted for in the calculation of the intermixture of different gases.
6	REAG	.T.	Chemical reactions in the gas phase are calculated.
7	REAC	.T.	Chemical reactions in the solid phase are calculated.
8	DFFE	.T.	Variable presently not used.

After these control parameters the item MGAS (default value = 1) is read. This integer defines the dimension of fields related to the gas concentrations. Each gas type under consideration has a fixed place in these fields, as follows:

1. Helium
2. Nitrogen
3. Oxygen
4. Carbon-Monoxide
5. Carbon-Dioxide
6. Steam (H_2O)
7. Hydrogen
8. Fluid water (droplets or condensed; option not yet working properly!).

MGAS may have values from 1 to 8. For MGAS = n the gases from 1 to n are considered in the calculation. The gases from (n+1) to n = 8 must not be given as sources, nor be the results of chemical reactions or condensation. On the other hand, the gases from 1 to n may exist individually or as mixture; they may be mixed and may react chemically. MGAS should be kept as small as possible to avoid large memory demands.

EXAMPLE

An example of the ST* input block is indicated below.

```
ST* Project A; Dose=18FPY, Excore power=TINTE, CRheight from VSOP-data
TTTTTTTT   400.0       0          0        2.       5.         .5
.00002      .0001     .1         .1       .1
TTTTTFFT   7
```

3.4 BLOCK 2: GEOMETRY AND SPATIAL MESH DEFINITIONS

The title card of this block starts with GM% or with GM*. The calculations for the solid temperature are performed in the total defined mesh grid. The gas temperature and the fluid flow are calculated only in those meshes that are declared as being flow meshes. For memory saving reasons this may usually be done in a smaller mesh grid that has to be a subset of the total mesh grid. This holds true for the nuclear calculations and the heterogeneous temperature calculations as well.

The definition of the mesh grid includes the mesh boundaries in the axial and radial direction, an optional division in finer meshes and information of the type of calculations to be performed in the mesh. The defined mesh grid is therefore a “material” mesh grid and should

be constructed so that for both the thermal-fluid, and the nuclear calculations, a well-defined material assignment is possible, and that temperatures, flows and power densities need not be unduly homogenised. In defining the boundaries of this mesh grid it is not necessary to account for a sufficient small discretising with respect to the finite difference solution of the differential equations. A smaller mesh subdivision can be introduced by the user, in which the basic leakage iteration process is used only for the 1- D calculations.

For formatted input ('GM%') the reading is done per card for I = 1 to 6, with format (6(E9.0,I2,I2)):

- **XGM (I): the start co-ordinate of the mesh in cm.**
- **NDX (1,I): a flag to define the type of calculation to be performed in this mesh**. The flag is constructed as sum of the possible sub-flags, defined as follows:

  0: The solid temperature is calculated or prescribed.

  1: The mesh may be a gas flow mesh.

  2: Nuclear calculations may be done in the mesh.

  4: A heterogeneous temperature calculation may be done in the mesh.
- **NDX (2,I)**: the coarse mesh will be subdivided into NDX (2,I) fine-mesh intervals, in which the 1-D leakage-iteration calculations will be performed. Studies performed with TINTE indicate that the number of fine meshes needed depends on the mesh location: meshes at the grid border, or meshes next to areas where large material changes take place, need a higher number of fine meshes for good convergence. As a rough guide, a coarse mesh subdivision into 3 fine meshes is usually adequate, but the number of fine meshes is only limited to 99 when using the GM% format. For very small coarse meshes (< 10 mm) only one fine mesh should be used.

NDX (1,I) and NDX (2,I) are used to define the dimensioning boundaries for the space dependent arrays and their positioning vs. the temperature fields. For a later interpretation XGM (I) and NDX (2, I) are intermediately saved. The free-format read-in (GM*) is performed card-by card, with 6 meshes per card. On the last card for the r and z co-ordinates trailing zeros need not be given; they are amended by the code. The limitation of 6 groups per card is necessary, because at the reading time the total number of items to be read is not yet known. The input is given first for the radial dimension until a NDX (2, I) is found to be zero. The input for the z-dimension is given in the same way on a new card. Note that the positive direction of the z-axis is assumed to show downward (i.e. in the direction of gravity). The input co-ordinates must be in monotonically increasing order (e.g. from r = 0 cm to r = 300 cm). They are tested on this requirement, and where applicable an error code is set and the programme is terminated.

The dimensioning boundaries are used for the array address calculations. IF IIRST > O, they will be compared with the data on the restart file, and any inconsistency will lead to a commented error stop. Note that the number of meshes defined as “nuclear” meshes (i.e. with NDX (1,I) $\geq$ 2) must match with the number of rows and columns in the NZ* data block (see Section 5.7).

EXAMPLE

An example of the GM% input block is indicated below. Note the required six-card format, and the subdivisions defined per coarse mesh.

```
GM%
     0.0 3 3    10.0 3 3    41.0 3 3   73.6  3 3   80.55 3 3   92.05 3 3
    99.0 3 1   100.0 7 3   109.0 7 3  121.7  7 3  134.4  7 3  150.6  7 3
   163.3 7 3   176.0 7 3   185.0 3 1  186.0  3 3  192.95 3 3  204.45 3 3
   211.4 3 3   225.0 3 3   243.6 3 3  260.6  3 5  275.0  1 5  287.5  1 3
   292.5 1 5   310.0 0 3   328.0 0 3  462.3  0 3  463.3
-951.8  0 3 -950.8  0 3 -620.8  0 3 -536.8  1 5 -278.8  1 3 -243.8  3 3
-228.8  3 3 -203.8  3 3 -193.8  3 3 -153.0  3 3 -113.0  3 3  -78.5  3 3
   0.0  7 3   17.0  7 3   72.0  7 3  124.06 7 3  166.7  7 3  211.7  7 3
 261.7  7 3  311.7  7 3  361.7  7 3  411.7  7 3  461.7  7 3  511.7  7 3
 561.7  7 3  611.7  7 3  661.7  7 3  711.7  7 3  761.7  7 3  811.7  7 3
 861.7  7 3  911.7  7 3  961.7  7 3 1011.7  7 3 1050.5  7 3 1105.25 7 3
1160.0  3 5 1220.0  3 3 1242.0  3 3 1257.0  3 3 1302.0  3 3 1362.0  3 3
1377.0  3 3 1392.0  1 3 1472.0  1 3 1592.0  1 3 1664.0  1 3 1687.0  1 3
1712.0  1 3 1724.5  1 3 1775.5  1 5 1975.5  0 3 2044.5  0 3 2154.5  0 3
2155.5
```

3.5 BLOCK 3: MATERIAL ASSIGNMENT TO THE MESH GRID FOR THERMAL-FLUID CALCULATION

The title card of this block starts with TZ% (fixed format input) or with TZ* (unlimited format-free input). After the input of the previous GM data block the total number of meshes is known (MRM.MZM, with MRM = number of radial meshes, and MZM = number of axial meshes). Now materials have to be assigned to these (coarse) meshes. For the thermal fluid-dynamics calculations this is done by reading MZM times a record with MRM numbers, with each number in either fixed format (18I4), or unlimited format-free. A data array, (LTRZ (J,I), with J = 1,MRM) is then formed, using the identification numbers of materials for the thermal- fluid calculations along row I (I = 1, MZM), each I starting on a new record. For so-called node-materials (see further below), or for meshes with predefined gas parameters, the material identification number must be negative. The negative sign identifies the mesh as node-mesh, while the absolute value identifies the material number. Note that a node-material may occupy only one mesh in the mesh grid.

The data for the LTRZ array are temporarily saved initially, and is only assigned to the final destination after the address calculation has been completed. For the gas-fluid sub-mesh grid (i.e. the grid formed by materials defined as materials with gas flow), the material vector LGRZ is extracted using the information from the GM data block.

For the modelling of additional gas flow paths, an “external” 1-D flow network may be defined. This network consists of “components” and “nodes”. The coupling of this network to the 2-D coarse mesh grid is obtained by defining “coupling-nodes”, i.e. special materials which may be assigned to only one mesh in the grid, and whose identification number in the LTRZ array must be negative. For each of the components in the 1-D flow network a single card is required, which is read either with format (4I5,5E10.5) or format-free. The 1-D flow network card variables are listed in Table 7 below.

Table 7: 1-D Flow Network Card Variables

No.	Variable	Description
1	K1	The unique material number of the component. This must be different from all numbers in the mesh grid and different from other component and node numbers.
2	K2	First material number connected by the 1-D component.

No.	Variable	Description
3	K3	Second material number connected by the 1-D component. These numbers may be the negative numbers of the coupling nodes, or additional unique positive numbers of further network nodes. If K1 is simulating a burst disk or a safety valve, K2 may not be used to define the reference pressure. The node K3 will not in this case be part of the calculations; it is merely used to define the (time-dependent) outer pressure.
4	K4	(Optional). The number of another material (component or arbitrary material in the mesh grid) being coupled thermally to the component. If no such coupling shall be considered, set K4 either 0 or equal to K1. If a coupling is defined, K4 has to be identified in the following block (TM) as having time-dependent solid properties. This may be obtained by defining there a (marginal) heat source for K4. It is not allowed to use K4 as material with predefined temperatures.
5	A1	The gas flow cross-section area of the 1-D component (for cross-flow tubes incl. the tube bundles). Making a component acting as a valve is controlled by time-variable ζ-values. Give either here the actual cross-section and in the TM block the actual ζ value, or give 1 here and ζ /(cross-section)^2 in the TM block. For a component without flow (e.g. a closed valve, but not for burst-discs or over-pressure valves), a value of zero may be given here. It is converted internally to ζ = 1 and ζ /(cross-section)^2 = 10^{20} cm^{-2}.
6	A2	The (gas) flow length of the component in cm.
7	A3	The difference in height between the nodes K3 and K2. If K2 is located higher than K3, A3 is positive. Note: In calculations with natural convection: equalizing the pressure is only correct between nodes of equal height.
8	A4	The solid surface (cm^2) in contact with the gas for the calculation of the heat transfer to the components solid material. For cross-flow tubes this figure is ignored and will be determined internally.
9	A5	The thermal coupling of the component to material K4 (≠ 0 and ≠ K1). It is the ratio of the effective component surface to the effective heat transport length (component wall thickness).
10	A6	The volume of the component. If set to zero, this figure is calculated as (A1.A2).

The input for the network components ends with a blank card. There is no strong ordering for the component data necessary, but some information data in the iteration monitor are dependent on the ordering shown here. For the first component defined, the pressure drop is given (DP21 in the iteration monitor output), for the first node (K2) the mass flow source (MP1 in the iteration monitor output), for the nodes K2 and K3 and for the node K2 of the second component the gas temperatures (TK1, TK2 and TK3 in the iteration monitor output). Note that it is mandatory to define at least one 1-D component at the end of the TZ block (probably without gas-flow), which implies the definition of at least two coupled nodes!

Lastly, the largest absolute value among the input figures of LTRZ, Kl, K2 and K3 is determined, and designated as the variable MLT. This number is used in dimensioning activities, but it is not necessary that all material numbers smaller than MLT are present in the problem under consideration. Note that 1-D nodes and components not defined in the TZ block are not allowed to be present in the TM block.

EXAMPLE

An example of a section of the TZ* input block is indicated below. An 1-D node (coupling materials -8 and -9) is shown in the last input card at the bottom of the block.

```
TZ*

 52  48  48  48  58  48  62 48 48 48 48 48 48 48  48  48  64  48  48  48
 52  48  48  48  58  48  62 48 48 48 48 48 48 48  48  48  64  48  48  48
 10  51  51  51  58  54  61 54 54 54 54 54 54 54  54  54  64  66  66  67
 10  51  51  51  58  53  60 55 55 55 55 55 55 55  63  63  64  66  66  67
 10  51  51  51  58  53  60 56 56 56 56 56 56 56  63  63  64  66  66  67
 10 130 157 184 211 238 265 57 57 57 57 57 57 57 292 319 346 373 400 427
 10 131 158 185 212 239 266  1  1  1  1  1  1  1 293 320 347 374 401 428
 10 132 159 186 213 240 267  1  1  1  1  1  1  1 294 321 348 375 402 429
 10 133 160 187 214 241 268  1  1  1  1  1  1  1 295 322 349 376 403 430
 10 134 161 188 215 242 269  1  1  1  1  1  1  1 296 323 350 377 404 431
 10 135 162 189 216 243 270  1  1  1  1  1  1  1 297 324 351 378 405 432
 10 136 163 190 217 244 271  1  1  1  1  1  1  1 298 325 352 379 406 433
 10 137 164 191 218 245 272  1  1  1  1  1  1  1 299 326 353 380 407 434
 10 138 165 192 219 246 273  1  1  1  1  1  1  1 300 327 354 381 408 435
 10 139 166 193 220 247 274  1  1  1  1  1  1  1 301 328 355 382 409 436
 10 140 167 194 221 248 275  1  1  1  1  1  1  1 302 329 356 383 410 437
 10 141 168 195 222 249 276  1  1  1  1  1  1  1 303 330 357 384 411 438
 10 142 169 196 223 250 277  1  1  1  1  1  1  1 304 331 358 385 412 439
 10 143 170 197 224 251 278  1  1  1  1  1  1  1 305 332 359 386 413 440
 10 144 171 198 225 252 279  1  1  1  1  1  1  1 306 333 360 387 414 441
 10 145 172 199 226 253 280  1  1  1  1  1  1  1 307 334 361 388 415 442
 10 146 173 200 227 254 281  1  1  1  1  1  1  1 308 335 362 389 416 443
 10 147 174 201 228 255 282  1  1  1  1  1  1  1 309 336 363 390 417 444
 10 148 175 202 229 256 283  1  1  1  1  1  1  1 310 337 364 391 418 445
 10 149 176 203 230 257 284  1  1  1  1  1  1  1 311 338 365 392 419 446
 10 150 177 204 231 258 285  1  1  1  1  1  1  1 312 339 366 393 420 447
 10 151 178 205 232 259 286  1  1  1  1  1  1  1 313 340 367 394 421 448
 10 152 179 206 233 260 287  1  1  1  1  1  1  1 314 341 368 395 422 449
 10 153 180 207 234 261 288 25  1  1  1  1  1 22 315 342 369 396 423 450
 10 154 181 208 235 262 289 24 23  1  1  1 20 21 316 343 370 397 424 451
 10 155 182 209 236 263 290 26 26 26 26 26 26 26 317 344 371 398 425 452
 10 156 183 210 237 264 291 27 27 27 27 27 27 27 318 345 372 399 426 453
 10  31  32  33  34  35  36 28 28 28 28 28 28 28  37  38  39  40  41  42
 10  31  32  33  34  35  36 28 28 28 28 28 28 28  37  38  39  40  41  42

 92 -9  -8  0  100.  200.  200.  0.  0.  0.
```

3.6 BLOCK 4: MATERIAL DESCRIPTION FOR THERMAL-FLUID MATERIALS

3.6.1 General Input Data

The title card of this block starts with TM% or TM*. For each of the material identification numbers given in the previous TZ block, the material properties are defined in the TM block. For each of these materials one or more cards are read, depending on whether the material is defined in the GM block as a material with gas flow (NDX $\geq$ 1), or whether a heterogeneous temperature calculation must be performed (NDX $\geq$ 4). The order of these groups of cards is arbitrary. On the first card of each set the information with respect to the solid temperature calculation is given (this card is mandatory for all material definitions). The following data are read (for formatted input (TM%)) in the format (A8,I4,4(I3,E8.5),3E8.5). The first card variables are listed in Table 8. .

Table 8: First Card Variables for the TM Block

No.	Variable	Description
1	TXT	An arbitrary text to identify the material. This is used only for information purposes. Note that the TXT field can only be 8 characters long.
2	NR	The (positive) material identification number. It may not be greater than the absolute value of the largest MLT in the previous block.
3	MAT	A number that identifies the type of material. The options are: 0: a (solid) material without gas flow 1: a pebble bed with gas flow -1: a pebble bed like 1, but additionally a heterogeneous temperature calculation for the temperature distribution inside the pebbles is performed. (Regions with MAT = -1 should be given small material identification numbers NR to reduce the size of the restart file and to save working memory). 2: a vertical boundary layer between the pebble bed and radial reflector. This may be used only between these two materials, otherwise problems might occur! It should be used as a single layer without any subdivision for a fine mesh, ands the mesh size must be less than 5 mm in width if possible. 3: vertically oriented flow tubes, e.g. the riser tubes in the side reflector. 4: a cavity with gas flow in all directions. In TINTE the number of meshes inside the cavity is not limited. The pressure drop and heat transfer are calculated according to tubular flow. The main difference of 4 to 3 is the possibility of horizontal flow. The flow calculation here is performed using a very crude approximation to the Navier-Stokes equation. The results are valid only if a large flow resistance exists in the cavity, or if the cavity is located between regions with large flow resistance. 5: a simulation model for azimuthal non-flow elements that have to be simulated by cylindrical regions. The azimuthal arrangement is taken into account by introducing a radial motility without a pressure drop and without thermal interaction (e.g. the AVR nose model). 7: aligned cross-flow tube bundles, e.g. in steam generators or heat exchangers. 8: same as 7, with bundles offset. 10: a network node. The data for the solid properties are ignored. For the gas properties, the initial temperature and information on pressure or mass flow sources may be given. 11: a burst disc or an over-pressure valve. In the current version only one such a component is allowed in the system! 12: the external node of a burst disc or an over-pressure valve. 13 : a tubular component. 17: similar to 7, but in the I-D flow network. 18: similar to 8, but in the I-D flow network
4	TVOR	The initial solid temperature guess in °C. TVOR must be given for all materials (except MAT=10) as an initial guess. For materials with the thermal conductivity treated according to rules #3, 83, 84, 210 and 210 (see Appendices D and E), TVOR is used as the irradiation temperature. This makes an iterative process necessary. After the first equilibrium calculation TINTE gives the region averaged temperatures in the data output. Inserting these solid temperature results here will lead to a rapidly converging iteration. In regions without calculation of the solid temperature, i.e. with time-independent input temperatures, the value -(1000. +T) is given here, with T being the temperature for equilibrium conditions (e.g. a constant boundary condition of 20 ^{o}C will be given as -1020).

No.	Variable	Description
5	IRHOC	The identification number of a rule to calculate the specific heat capacity (see Appendix C). This parameter, as well as the next parameter RHOC, are ignored if MAT = -1. In this case the heat capacity is accounted for during the heterogeneous temperature calculation. Nevertheless, a value has to be given here for cases that involve graphite corrosion, since it is used to identify the type of graphite in contact with the gas. (The corrosion cases will only work for materials # 7,8 201 and 210).
6	RHOC	A parameter related to the specific heat capacity. If IRHOC = 0, this is the specific heat times density times solid material fraction (unit $Ws/cm^3/K$). If IRHOC ≠ 0, RHOC is a corrective multiplication factor to the calculated heat capacity. Note: Despite the fact that for MAT=-1 IRHOC and RHOC are ignored for the calculation of the heat capacity, it plays an important role for corrosion calculations. In this case, RHOC has to be set equal to the bed filling factor to calculate the amount of graphite in contact with the gas correctly.
7	ILAM	The identification number of a rule to calculate the heat conductivity (see Appendix D). If ILAM < 0, the absolute value is interpreted as a rule number, and simultaneously a flag is set to read an additional input block KG* after the TM block. The KG block is used to input a temperature and dose-dependent set of conductivity data (see the discussion later in this Section w.r.t. the KG block).
8	ALAM	A parameter related to the heat conductivity. If ILAM = 0, this is the heat conductivity times solid material fraction (unit W/cm/ K) . If ILAM ≠ 0, then in some cases ALAM is interpreted as a dose-dependent conductivity (W/cm/K), or it is simply a corrective multiplication factor to account for dilution effects etc. For rules that describe the conductivity of a mixture, e.g. a pebble bed, the material fraction has to be set to 1. In all cases an input of 0 is internally modified so that the default values are used. In cavities with gas flow and without internal structures, it is recommended to homogenise small layers from adjacent regions using their conductivity rules, and applying a large dilution factor via ALAM. (In cavities radiative heat transfer is important, and a very small heat conductivity is avoided by using this “dilution” method). Note: Errors may occur in special situations in tubes and gaps, caused by the volume-oriented heat transport in TINTE. This may be (partly) overcome by using special settings of ALAM and ALPHA (shown on the next cards).
9	IDIR	The direction and type of radiative heat transfer. The options are: 0: no radiation. 1: mainly axial radiation. 2: mainly radial radiation (In both cases half of the calculated effective conductivity is assumed for the other direction (see /1/ Chapter 6.1)). 3: radiation exclusively in the axial direction. 4: radiation exclusively in the radial direction.
10	REPS	The effective radiation emission constant. For parallel slabs with physical emissivities ϵ_1 and ϵ_2 separated by a small distance, the effective radiation emission constant is $REPS = 1/((1/\epsilon_1+1/\epsilon_2) -1)$ (1) For a cylindrical annular gap with surfaces F1 and F2, it is $REPS = 1/(1/\epsilon_1+(F_1/F_2) . (1/\epsilon_2-1))$ (2) It is not possible to use REPS as a radiative boundary condition. For regions without solid temperature calculation it is ignored.

No.	Variable	Description
11	DOS	The fast neutron dose (unit 1E21 EDN) of the material. This figure is used to calculate the heat conductivity in graphite materials. For MAT = -1 it is used only for the effective pebble bed conductivity. For the conductivity *inside* the pebbles, this value is only used if no input is found concerning the mixture of pebble types (see "multi-fuel element option" later in this Section); otherwise the pebble conductivity is calculated from data and dose values provided there.
12	WPR	An external heat source (unit MW). If the material is identified as having variable solid parameters, WPR is not a guess for the nuclear heat production, but an additional source beside the nuclear heat sources, used to describe the decay heat production in a fuel discharge tube, or to simulate the (negative) heat source in the steam generator. Note that WPR is ignored if TVOR < 0.
13	ALPFL	The heat transfer coefficient (unit W/cm^2/K) in regions with TVOR < 0 (see /1/ eq. 6.29). A very small ALPFL results in an *adiabatic* boundary condition with respect to the solid heat conduction, while a large ALPFL make the surface temperature identical to the average temperature in that region. ALPFL is needed only in regions with time-dependent temperature specifications; however it is also necessary for regions with WPR ≠ 0, in case of a later re-definition. An input ALPFL = 0 is internally redefined to ALPF L = 1, and for components of the 1-D flow network ALPFL is ignored.

If MAT ≠ 0, a second card follows that describes the gas flow and heat transfer parameters. The data are read in fixed format (for TM%) in the format (16X,8E8.5)). The second card variables are listed in Table 9.

Table 9: Second Card Variables for the TM Block

No.	Variable	Description
1	GAST	The gas temperature in °C. For source regions this is the gas inlet temperature, otherwise it is an initial guess.
2	FFK	The solid filling factor in the region, i.e. the fraction of solid material. The porosity is (1- FFK). For tube bundles (MAT = 7, 8, 17, 18), FFK is ignored; in these cases the porosity is calculated internally.
3	DHYD	The hydraulic diameter in cm. (see definition in Chapter 7.1). The options are: For MAT = 1, -1, 2: DHYD is the pebble diameter For M AT = 3, 4, 13: DHYD is 4(volume/surface ratio) of the flow paths For M AT = 7, 8, 17, 18 DHYD is the outer diameter of the tubes.
4	ZETA	An additive term to the flow resistance parameter caused by curvatures of the flow path. The options are: For tube bundles (MAT = 7,8,17,18) ZETA must be the tube distance (centre to centre in each layer) divided by DHYD. In tube components (MAT = 13) ZETA is used to control valve functions time variable. For a component used as burst disc or safety valve (M AT = 11): ZETA is the over pressure for opening [bar]. If during a transient the pressure difference of the neighbouring nodes surpasses this value (this works only if the pressure calculation is performed via the coolant inventory!), the burst disc will completely open immediately. For a safety valve the opening starts and the loss of inventory is simulated.

No.	Variable	Description
5	PHI	A factor to the resistance parameter to account for different shapes of the tube for laminar flows. The options are: 1.0 for circular tubes 0.89 for square tubes 1.5 for an infinite slab For rectangular cross-sections it varies between these values. Input PHI = 0 is internally converted to PHI = 1. For tube bundles (MAT = 7, 8, 17, 18) PHI is interpreted as the distance of the tube layers (centre to centre) divided by DHYD. For a component with MAT = 11 PHI is a pressure difference, and is used to identify that component as either burst disc or safety valve. A value smaller than ZETA (attention 0→ 1!) identifies the component as burst disc, while a value larger than ZETA identifies the component as safety valve. Although this is not realistic, a burst disc will close again, if after opening the pressure difference falls below this value. To simulate a realistic burst disc set ZETA to the pressure of the outer node (MAT=12), i.e. normally 1 bar. In this case PHI is the overpressure for full opening. Safety valves with a hysteresis (closing pressure smaller than opening pressure) have shown to inhibit convergence, and cannot be used in TINTE.
6	ALPHA	A factor to adjust the calculated heat transfer coefficient. ALPHA = 0 is internally converted to ALPHA = 1. So, if necessary use a small positive value for ALPHA. For MAT = 2, ALPHA is the roughness factor as defined in Chapter 6.5. This ALPHA value is also used to correct the calculated mass transfer coefficient in corrosion calculations.
7	PGAS	If the material contains a gas source or a gas sink, one of the variables PGAS or QGAS has to be given. It is assumed by the solution algorithm that this material is present in one single mesh or one component of the 1-D flow network only. For forced convection at least two of those regions have to be present in the system; for natural convection one of them will suffice. In at least one of the source/sink regions the reference pressure has to be defined by PGAS, a pressure specification [bar]. If PGAS is given for more than one material, the reference pressure has to be indicated by a negative value. For a node of the 1-D flow network with MAT = 12, which is defined as K3 for a component with MAT = 11 (burst disc or safety valve), PGAS is the external pressure. The node K2 before that component may not be used to specify the reference pressure during a transient with coolant inventory specification. Another possibility to identify a material as source/sink, is to define a mass flow for it.
8	QGAS	A source of gas mass-flow [kg/s]. In TINTE, more than one gas source is possible, and more than one gas sink is possible as well. This option however has to be used with care!

Use of one of the parameters PGAS or QGAS characterises a material as being controllable in transients with respect to pressure, mass-flow and gas-inlet temperature. For nodes in the 1-D flow network this is the case by default.

EXAMPLE

An example showing 6 materials of a TM* input block is indicated below.

```
TM*
COMP2     2    4    581    201   0.9206   201   1.0     0      0.0          0.0
                    522          0.9           10.0
COMP3     3    3    493    201   1.0189   201   1.0     0      0.0          0.0
                    488          0.99           1.0          0.16
COMP4     4    4    317    0     38.8E-03 4     1.0     1      0.7841
                    300          1.0E-05        164.5        0.0     0    0    0.0
COMP5     5    4    351    0     38.1E-03 4     1.0     1      0.7841
                    311          1.0E-05        272.4        0.0     0    0   66.75
COMP6     6    3    537    0     28.8E-03 4     1.0     0      0.0          0.0
                    505          1.0E-05        20.0         0.0
COMP7     7    4    751    201   0.9835   201   1.0     0      0.0          0.0
                    696          0.9615         1.4          0.0
```

3.6.2 Input Data for Material -1 (Heterogeneous Temperature Calculation)

For MAT = -1 additional input is read for the heterogeneous temperature calculation. The detailed geometry of the pebble is read in the same way as in input block GM, either in free format (TM*) or with format (6(E9.0,I1,I2)) (TM%). The input is read in the following sequence:

- **XGM(I):** the starting co-ordinate of the mesh [cm], beginning in the pebble centre with radius 0.
- **NDX(1,I):** currently not used. For free-format input an arbitrary dummy character has to be placed, e.g. ‘0’ or ‘, , ‘ or ‘, ,’.
- **NDX(2,I**): the number of fine meshes in this mesh. Temperature gradients and the dependency of the heat conductivity on temperature will be improved. Similar to the GM block input data is given for up to six meshes on one card. Reading of this card stops if NDX(2,I) = 0.

The fuel mesh structure may be as coarse as the physical material structure of the fuel element allows. Mostly a subdivision into three coarse meshes will be sufficient: the fuel pellet, the fuel free shell (without a small (1 mm) outer shell), and the 1 mm outer shell (implemented to improve the convergence of the overall 2-D r-z conductivity calculation). In the present version of TINTE all fuel elements in the system must have the same geometry. This implies in case of more than one material with MAT = -1 that the above card must be identical for each of them. The composition for the different shells however may be different. If regions with MAT = -1 differ in the composition of the elements but not in the pebble radius, this situation may be solved by introducing a common fine heterogeneous mesh structure for all of them.

The current version of TINTE also allows the description of different types of fuel elements that are present in a pebble-bed reactor with multi-pass fuel management. These elements differ mainly in the burn-up and the accumulated fast neutron dose. The heat conductivity degradation due to fast neutron irradiation and the reduction of the power production due to increasing burn-up thus form counterbalancing effects on the fuel temperature development. As these different types of fuel elements have all the same geometry, and are assumed to have the same heat capacity, and follow the same conductivity rules (but with different doses), no additional input is necessary.

The additional fuel element type information is given in the NQ data input block, and is automatically recognized by the code. This treatment also works for “non-fuel” mixtures such as pure graphite spheres, as proposed for some HTR designs.

Also, the different power productions from the different fuel elements results in different surface temperatures, and radiative heat transport will occur between them. This effect is also accounted for in TINTE (see Appendix B).

For each of the above defined coarse meshes inside the fuel elements the material properties are read from consecutive cards. Starting with the innermost mesh, the card variables (indicated in Table 10) are read either format-free (TM*) or with the fixed format (10X,2(I2,E8.1),E10.3) (TM%) .

Table 10: Heterogeneous Temperature Card Variables for Fuel Pebble Layers

No.	Variable	Description
1	IRHOC	Identical to the description in Table 8.
2	RHOC	Identical to the description in Table 8.
3	ILAM	Identical to the description in Table 8.
4	ALAM	Identical to the description in Table 8.
5	QDEN	The distribution of local nuclear heat production (see /1/ Chapter 4.5) within the fuel element meshes. A zero value indicates a mesh without fuel. Otherwise QDEN must be given relative to the fission densities. Normalisation is automatically performed by the code.

Note that the dose information of these cards is also used for the fuel elements if no information concerning a mixture of fuel-elements and/ or graphite elements is found at the end of the .tn4 file (in the NQ data block). Also note the difference between the heat conductivity inside the fuel element, and the effective heat conductivity of the pebble bed, which is defined on the first card of the group.

The specifications for the heat capacity (IRHOC and RHOC) given on the first card of the group will be discarded and be replaced by the data read here (in Table 10), as the heat capacity is accounted for in the heterogeneous calculation. The homogenized heat capacity of the outer shell is used for the homogeneous 2- D calculation.

The input is read until NR = 0 (i.e. a blank card) is found.

EXAMPLE

An example of the heterogeneous input cards in the TZ* input block is indicated below. The first 2 input lines are the global pebble bed parameters (see the paragraphs after the example), while the last 4 input lines indicate the data for the various shells within a single fuel pebble (discussed in Table 10 and in the paragraphs before the example).

```
CORE1     1  -1   828.0 201   0.9943  55  1       0    0.      0.83204    0.
                 806.8 0.61     6.
                 0.   0 5   2.5   0 3   2.9   0 2   3.0
                       201   0.9943 -52  1.        1.
                       201   0.9943 -52  1.        0.
                       201   0.9943 -52  1.        0.
```

3.6.3 Additional Input Data

- **Corrosion Data:** Although an equilibrium case does not exist if graphite corrosion is possible, TINTE allows calculation of an equilibrium situation even with oxygen present in the system and activated corrosion options. During the equilibrium calculation the corrosion option is automatically deactivated. Starting a transient from such a situation will not be possible in general, because the extremely fast chemical reactions will inhibit the convergence of the code.
- **Detail Temperature Field on a Separate Grid:** If necessary, it is possible to define a refined temperature field for special purposes. If any TINTE activity starts with the calculation of a self-consistent equilibrium situation, the temperature guess given above for each material is adequate. The detailed temperature field will be calculated

iteratively by the code. There are however situations, where a transient does not start from an equilibrium status, e.g. in evaluating certain experimental data. In these cases it may be necessary to have more detailed temperature information at the beginning of a transient. For these purposes an independent temperature mesh grid can be defined. In contrast to the standard TINTE algorithms, this grid is edge-centred, i.e. the temperatures are given at the edges of the grid. All input in this case is format-free. On the first card the independent temperature mesh radii are read. Note that zeros have to be replaced by 10^{-30}, because the number of non-zero items is counted. On the next card the axial data are given. For each axial position a card is then read, giving the temperatures at each radial position. A blank card ends this additional input. Later on, from this newly-defined temperature field the temperatures for the standard TINTE mesh grid are interpolated. Note however that this option is not sufficiently tested at this stage!

- **Block K7*:** If a conductivity rule from 70 to 76 is found in the TZ block, the heat conductivity for a bed with arbitrary pebble filling is calculated. The data necessary are read in an additional block with title card K7*. Data are read format-free but with card structure. The first card gives information on the conductivity of the pebble material. Either a negative number -ILAM (where ILAM is the identification number of an already existing conductivity rule for the calculation of the pebble conductivity), followed by ALAM (with the same interpretation as before), or a set of up to 10 items is read. The order of the input is: (TLEIT(I), I=1,IL(Ie10)), a series monotonic increasing temperatures in °C, for which on the next card (ALEIT(I), I=1I,IL), the same number of corresponding conductivities [W/cm/K] follow. A dependency on fast neutron dose is not implemented as this option is mostly used in connection with non-irradiated experimental materials. In both cases a conductivity table is established. These data are used in calculating the effective pebble-bed conductivity according to rules 71 to 76. Additionally they are directly available under rule 70.
- **KG* Block:** The KG* block is used to contain additional or modified parameters for pebble bed and fuel pebble data. The title card of this “modified” block is KG*, and it does not form part of the TM block (i.e. it follows the TM block, and is separated by a blank line).

  First Part: If at least one of the ILAM values figures above were negative (i.e. a negative thermal conductivity rule), modified pebble bed parameters and/or heat conductivities for graphite are read in unlimited format-free. (Note that this is not applicable for the ILAM value sin the shells of MAT = -1 materials). For the fixed temperatures 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 and 2500 (°C), and the fixed fast neutron doses 0.0, 0.259, 0.982, 1.753, 3.582 (10^{21} EDN), the new heat conductivity values is read in the following format: **(TDGST(J,I), J=1,11), I=1,5).** This heat conductivity values then replace the data of conductivity rule #52.

  (This modified conductivity rule is currently available under rule #52, and the new data can also be used as the graphite conductivity when calculating the effective pebble-bed conductivity in rules #53-#55, and #63-#65).

  Second Part: On this optional card modifications to the pebble bed parameters might be made. If the card is missing, the default values are used. The card variables (indicated in Table 11) are read either format-free (TM*) or with the fixed format (10X,2(I2,E8.1),E10.3) (TM%) .

Table 11: KG* Block Input Card Variables for General Pebble Bed Parameters

No.	Variable	Description
1	DK	The fuel sphere diameter [Default: 6.0 cm].
2	FF	The filling factor of the pebble bed [Default: 0.61].
3	PP	The gas pressure in the pebble bed [Default: 1.0 bar]. This pressure can be obtained from the output of a converged equilibrium calculation.
4	PHI	The (material dependent) flattening parameter for the pebbles. Known values exist for steel (0.0013), ceramics (0.0077) and copper (0.0253). For electro-graphite the value of 0.0165 may be used. The default value used here is 0.0165.
5	GFAKO	The gas fraction in the calculation of the effective conductivity. This is used for test purposes to see the influence of the gas conductivity on the effective conductivity. The default value used here is 1.0, but a value of -1.0 can be used to flag a read-in of pebble emissivities (see Third Part below).

Third Part: If GFAKO <0, the final two cards of the KG* block contain information on the emissivity of the fuel pebble material. n the first of these cards a series, **(TEPS(I), I=1,IE**(≤10)), consisting of up to 10 positive, monotonic increasing temperatures in ^{o}C, is read, while the last card gives the corresponding emissivities for the material under consideration **(EPS(I), I=1,IE)**.

EXAMPLE

An example of the KG* block is indicated below. The structure of the block shows 11 columns for the 11 fixed temperatures, and 5 rows for the 5 fixed doses in lines 2-6. Line 7 shows the modified pebble bed parameters (discussed in Table 11), where the -1.0 value of GFAKO results in the last two lines being read in for the emissivities of the fuel pebbles.

```
KG* Lambda for 11 Temp and 5 fixed doses, Rule No.52 negative!
.4940,.4460,.4115,.3790,.3555,.3300,.3100,.2875,.2740,.2625,.1486,
.3100,.3060,.2900,.2740,.2635,.2525,.2445,.2335,.2260,.2190,.1486,
.2025,.2160,.2170,.2135,.2090,.2015,.1945,.1895,.1845,.1815,.1414,
.1510,.1660,.1710,.1740,.1735,.1725,.1720,.1715,.1710,.1705,.1364,
.1211,.1359,.1465,.1500,.1506,.1486,.1472,.1423,.1415,.1408,.1296
6   0.61  87.0  0.0165   -1.0
100,  200, 300, 400, 500, 600, 700, 800, 900, 2500
0.85, .85, .85, .85, .85, .85, .85, .85, .85, .85
```

If MGAS = 1 has been given on the third card of the ST Block (see below Table 6) a blank card follows to indicate the end of the TM block. Otherwise the equilibrium case gas-mixture is read in here. Either format-free (TM*) or with format (8E10.0) (TM%), the relative molar fractions for the constituents are given. The normalisation is performed internally.

EXAMPLE

An example of the last line of the TM block is indicated below.

1. 0. 0. 0. 0. 0.

3.7 BLOCK 5: MATERIAL ASSIGNMENT TO THE MESH GRID FOR NUCLEAR CALCULATIONS

The title card of this block starts with NZ% or NZ*. This block defines the nuclear materials for that part of the grid identified in the GM* block where nuclear calculations have to be

done. The dimensions of the rows (MNR) and the columns (MNZ) of the NZ block is calculated from the data in the GM* input block, where the meshes are defined as nuclear meshes (see Section 5.4). If a mismatch inn dimensioning occurs between the GM* and the NZ* input sets, the TINTE calculation will stop. Also note that the “nuclear material” numbers used in the NZ* block are independent from those defined for thermal calculations (i.e. in the TZ* block). The definition of layers is done similar to that in section 5.4 by MNZ times reading of MNR positive or negative numbers, either (NZ%) in format (18I4), or unlimited format-free (NZ*) as follows: (LNRZ(J ,1), J=1,MNR), where identification numbers of materials for the nuclear calculations are read in along a row I (I=1, MNZ), each I starting on a new record. The positive numbers in LNRZ refer to sets of nuclear cross-sections, defined in the next (NQ%) block, while the negative material numbers are reserved for nuclear materials that consist of a mixture of sets of cross sections.

There is a correspondence between the cross-section set numbers here and in the .tn4 file. Each set number given here has to be found in the .tn4 file, but there may be set numbers in the .tn4 file not used here. However, the highest number found in the .tn4 file has to be part of the NZ block.

EXAMPLE

An example showing a section of a NZ* input block is indicated below.

```
NZ*
  240  214  214  214  214  214  214  214  214  214  214  214  -1   214
  240  241  241  241  241  241  241  241  241  241  241  241  -2   241
  242  243  243  243  243  243  243  243  243  243  243  243  -2   243
  171  274  282   15   39   63   87  111  135  159  296  304  -18  -28
  171  274  282   16   40   64   88  112  136  160  296  304  -19  313
  171  260  278  290  290  290  290  290  290  290  300  300  212  317
  171  260  278  290  290  290  290  290  290  290  300  300  212  317
  171  221  223  224  224  224  224  224  224  224  225  225  211  226
  -1   209  1.    183
  -2   209  1.    184
  -3   209  1.    185
  -4   209  1.    186
  -5   209  1.    187
  -6   209  1.    188
```

Among the materials with negative numbers the one with the largest absolute value (e.g. -28 in the block above) has a special meaning: when evaluating the transient neutron flux it is assumed that at this location a flux detector is located. This is the reason why at least one negative number has to be present in the NZ* block.

For defining the dimensions of the NZ* block the largest (MMM) and the absolute value of the smallest (MVH) material number are extracted first. These two numbers define the total number of materials with fixed cross-sections, and the number of materials in which cross sections may be mixed and/or changed.

For materials with varying cross section compositions (negative material numbers), the basic cross section sets and the sets for superposing are defined with the following series of numbers:

(-I, LCNC(I,0), (CONC(I,J,1), LCNC(I,J), J = 1, jmax)),

where

I = the absolute value of a negative material number from LNRZ

LCNC(I,J) = one of the “overlay cross section sets”

CONC(I,J,1) = the fractional part of concentration of the corresponding cross section set.

EXAMPLE

```
-1   209  1.    183
-2   209  0.6   184   0.2   185
```

This example shows that basic cross section set for nuclear material -1 is nuclear material #209 (as defined in the next NQ% data block). This basic cross section set is overlaid (or superimposed) with the cross section set from nuclear material #183. The fraction of overlay is 1.0, i.e. the cross section data for nuclear material -1 is completely obtained from nuclear material #183. This overlay fraction can be changed during transient events (see Section 8). The second line shows that the basic cross section set for nuclear material -2 is also nuclear material #209, but here the basic cross section set is overlaid with 60% of the cross section set from nuclear material #184, and 20% of the cross section set from nuclear material #185. The cross section data for material -2 therefore consists of 20% #209 data, 20% #185 data and 60% of #184 data. (Also note that there is no open line/card between this section of the input, and the “map” of the NQ* block- see the previous example).

For each negative material number appearing in the block LNRZ, up to 6 sets of data per card can be read in an arbitrary sequence, either in format (8F7.0,I2) (for NZ%) or format-free (for NZ*). Zeros at the end of the cards are added, if necessary. Cross sections are determined from the adjusted fractions (as indicated above), but this is not possible for the average burn-up. This factor is determined with the basic cross section composition, which fixes the precursor-concentrations after the equilibrium calculation. The sum of the overlay parts has to be identical to 1, therefore for J = *jmax* down to 1, the *CON(I, J*, 1) are corrected by

$$CON(I,J,1) = \max(0, \min(CON(I,J,1 - \sum_{k>0} CON(I,K,1)))$$

and additionally *CON(I, 0*,1) is adjusted according to

$$CON(I,0,1) = \max\left(0,1 - \sum_{k>0} CON(I,K,1)\right).$$

jmax is defined by reading *LCNC(I,jmax)* > 0 and *LCNC(I,jmax* + 1) = 0. Therefore, when *jmax* = 7 then one card is needed, when jmax = 15 a second card becomes necessary, and so on. For defining the dimension limits, the largest J has to be found, for which LCNC(I, J) is different from zero.

This overlay input is finished, when LCNC(..., 0) = 0 (blank card). In regions with variable compositions, additional previously unused cross section sets might appear. Now the maximum MLL of cross section sets to be read is determined as the maximum of MMM and the largest number of the LCNC.

3.7.1 Final Input Data

The final NZ* input (and the final input for the .tn3 file) is used to group material meshes into leakage iteration meshes (see /1/ Section 3.4.) for memory saving reasons. The data is read in two cards, consisting of numbers whose sum is MNR for the columns and MNZ for the rows. Each set is read from a new card, either in format (14I5) (for NZ%) or in free format (for NZ*), with an arbitrary number of items per card. Both series of numbers are terminated by a

zero. The length of the series determines the number of iteration columns MRI and rows MZI. The items of these series are (ND(J), J=1, MRI resp. MZI), meaning ND(J) material meshes will be grouped into leakage iteration mesh J.

The leakage iteration calculation is then performed using the leakage rates from the newly defined “coarse mesh” rows and columns, instead of those from the original “finer” material mesh. This course mesh procedure has been proven to work well for nuclear calculations, as opposed to the temperature calculations (where this approach is not followed). Care has to be taken in applying this method in the core regions, as the accuracy of the heat production (needed in material meshes) is degraded.

EXAMPLE

The example below indicates the grouping of the 1-D iteration meshes. Note that the core region is not divided into coarser meshes (i.e. the 14 radial and 25 axial core regions are kept as single 1-D iteration meshes), and the 0 to end the input cards. This block of input is separated by a single open (empty) line from the NZ* block, but technically still forms part of the NZ* block.

```
14*1  2  1  2  3  0
3  3  25*1  3  2  2  0
```

USER NOTE:

*As indicated above, the two lines shown in the example are also the last input lines in the .tn3 file. The .tn3 file should be closed off with **two** open (empty) lines after the final input data, or run-time problems will occur.*

3.8 BLOCK 6: NUCLEAR CROSS SECTION DATA

The title card of this block starts with NQ%, i.e. only fixed format input is possible. The data in this block is read in by the subroutine TIRTTL, when called from the routine TIANFA. Prior to the detailed reading of this block the information on the highest order of the polynomial expansion of the cross sections has to be extracted for dimensioning purposes. For simplicity and because of the large size of this block the data are not stored intermediately during the first reading. Instead the data set is read twice.

The block consists of three parts: In the first part the nuclear cross-sections and their polynomial expansions are given. In the second part information is read for the treatment of large cavities or holes (non-isotropic diffusion regions) (see /24/). The third part contains information on the treatment of the decay heat, and optionally on multi-fuel element properties.

Since it is assumed that the first part of the block (nuclear cross-sections and their polynomial expansions) will be generated mostly automatically by some pre-processor code, a format-free input option is not implemented. The description of the structure of this first part is intended to enable the user to write a customised interface programme or to modify the polynomial expansion. For non-isotropic diffusion regions like large voids and cavities the user must still provide the necessary information by hand, as described below.

The input for the first part includes the data for all nuclear materials as described in Section 5.7. The cross-section sets are made up of 3 “groups” of cards: a group of cards for the cross-section sets, a second group of cards detailing information related to the reference conditions, and a final group of cards for each expansion coefficient, for parameters deviating

from the reference values. The parameters included are the fuel temperature, the moderator temperature, the xenon concentration and the buckling. The feedback of operational state parameters on the total cross-section at a specific operational reactor state is then made up as follows:

$$XS = A_0 + \{(A_1 \Delta T_{\mathrm{mod}} + (A_1 \Delta T_{\mathrm{mod}})^2 + ...) + (B_1 \Delta T_{fuel} + (B_1 \Delta T_{fuel})^2 + ...) + ...\}$$

Additionally (when chemical and corrosion studies are performed), polynomial expansion coefficients with respect to the concentrations of hydrogen and nitrogen in the cooling gas, and with respect to the reduction of graphite mass due to corrosion, may also be provided.

The first three cards of each group are similar structured. On the first two cards data for the fast and thermal energy groups (respectively) are read with format (I3,1X,7I1,2X,I1,6E11.0), shown here in Table 12.

Table 12: Cross- Section and Expansion Coefficients Data for the Fast and Thermal Groups: First 2 Cards

Variable	Description
ISIG	The material identification number of the cross-section set
J(I), I=1,7	The identification numbers for the type of the group. If all J(I) ≡ 0 then this is the reference group, otherwise J(I) defines the type of the expansion coefficient, i.e.: J(1): expansion coefficient of order J(1) for the deviation in moderator temperature J(2): expansion coefficient of order J(2) for the deviation in fuel temperature J(3): expansion coefficient of order J(3) for the deviation in relative xenon concentration J(4): expansion coefficient of order J(4) for the deviation in buckling J(5): expansion coefficient of order J(5) for the deviation in N_2 concentration J(6): expansion coefficient of order J(6) for the deviation in H_2 concentration J(7): expansion coefficient of order J(7) for the deviation in graphite burn-up, which has to be reduced by the concentration increase of CO and CO_2. (Note that no serial expansions with mixed parameters are implemented; i.e. from all J(I) only one value may differ from zero).
IG	The energy group to which the following macroscopic cross sections expansion coefficients of material ISIG apply: SIGTR, the transport cross-section SIGMA, the absorption cross-section NUSIGF the neutron production cross-section RVAU, the 1/v cross-section SIGF the fission cross-section, from which the heat production is calculated SIGR, the removal (scattering) cross-section to the other energy group.

EXAMPLE

An example of the first 6 lines from a NQ% input block is indicated below. Note that all 4 cards that can occur are shown here, and not only the first 2 cards as described in Table 12 above. The first input line starts with the reference group data (since all J(I)=0)) for group 1 (i.e. the fast group), for cross-section set #1, followed by the 6 expansion coefficients described in Table 12. The second input line gives the data for the second group (i.e. the

thermal group). The variables of the third input line are described in Table 13 below the example, and the variables of the fourth input line in Table 14. The fifth input line in this example then starts with the data for the expansion coefficients for the deviation in moderator temperature (for the first energy group), since J(I)=1000000.

```
NQ%: PBMR400-SAR, 6 batches, VSOP data of 6 September 2004, iter 4.
  1          1 1.4545E-01 3.3116E-04 1.2311E-04 1.6951E-08 4.8342E-05 5.6538E-04
  1          2 2.1647E-01 3.3803E-03 5.7111E-03 2.2602E-06 2.2483E-03 0.0000E+00
    1.8400E+02 1.5412E+06 4.2665E+04 2.8685E+03 6.2274E+04 0.0000E+00 0.0000E+00
   0.000000.000000.000030.000090.004430.168760.000070.051580.000020.000000.19852
  1 1000000  1 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
```

From the third card the following items are read in format (3X,7E11.0) (see Table 13):

Table 13: Cross- Section and Expansion Coefficients Data for the Fast and Thermal Groups: Card 3

Variable	Description
ABBR	The average burn-up in material zone ISIG (in [MWs/cm3]). This item is relevant only in the reference card group (i.e. all J(I) = 0).
SIGMX(I), I=1,4	The thermal microscopic cross- sections or expansion coefficients of ^{135}Xe and the other strong absorbing fission products ^{149}Sm, ^{151}Sm and ^{157}Gd, with similar delayed build-up and similar (but smaller) reactivity impact as ^{135}Xe.
QREF(I), I=1,2	QREF(1) is used only for J(4) > 0. In this case it is the reference buckling of the fast group and QREF(2) is the reference buckling of the thermal group. In all other cases QREF(2) is the reference value of the corresponding variable under consideration.

If the card group belongs to the reference group (i.e. all J(I)=0), an additional fourth card follows (independent of whether fissions take place in the material ISIG or not). On this card information is given about the relative distribution of the fissions between different fissionable nuclei, and on the ratio of breeding captures to fissions. From these data the effective heat production per fission is derived. This figure is assumed to be independent of the expansion parameters, therefore this card is read only for the reference situation. In format (11F7.3) the items read are indicated in Table 14.

Table 14: Cross- Section and Expansion Coefficients Data for the Fast and Thermal Groups: Card 4

Variable	Description
XX(I), I=2,12	XX(I), I=2,10 are the relative fissions of ^{232}Th, ^{233}U, ^{234}U, ^{236}U, ^{238}U, ^{239}Pu, ^{240}Pu, ^{241}Pu and ^{242}Pu. The portion of the most significant nuclide ^{235}U is calculated from $XX1 = 1 - \sum_{I=2}^{10} X(I)$ XX(11) and XX(12) are the ratio of neutron captures in the breeding nuclides ^{232}Th and ^{238}U to the total number of fissions.

A special treatment should be applied for cavity regions that are to be treated with diffusion theory. If in those regions the concentrations of N_2 and H_2 are sufficiently small, no spectrum calculations should be performed. Instead the method as described in /24/ should be preferred.

For cylindrical medium height cavities all reaction cross sections should be either set to zero (or in case of inside structures to some homogenised value of these); the diffusion constant

in the radial direction should be set to 1/10 of the cavity radius, and in axial direction the diffusion constant should be 1/2 of the cavity radius.

It is good practise to set the SIGTR in both energy groups to SIGTR=1/(3 *D), with the diffusion constant set to the cavity radius (in cm). All other input fields in this 4-card group (except ISIG and IG) should be set to zero. The first part of this block must be closed by a blank (ISIG = 0) card.

EXAMPLE

In this example, the first part of the modified cavity correction factors is shown. Material numbers 169 to 171 represents the cavity areas. Since the input is read in fixed format, the positioning of the open spaces and lines are important.

```
169 0000000  1 1.8018E-03
169 0000000  2 1.8018E-03


170 0000000  1 3.3333E-02
170 0000000  2 3.3333E-02


171 0000000  1 3.3333E-02
171 0000000  2 3.3333E-02
```

The second part of this block is used for information on neutron streaming effects in holes and cavities. This normally asks for a non-isotropic description of the diffusion constant, like in cavities treated according to /24/, or in regions with pronounced holes like the radial reflectors containing the raiser channels. Of course, isotropic gaps like in the pebble bed may be treated here as well, if not already done in the spectrum calculations.

Each corrective input consists of a group of cards. The data of the first card are in format (4I3). For the (consecutive) materials starting with identification number **IS1** and ending with identification number **IS2,** the following corrections to the diffusion constant (given as transport cross sections in the first part above) are to be made starting with energy group **IG1** and ending with energy group **IG2**.

The correction factors are given on the next two cards (card no. 2 and card no. 3 of the group) for the radial and then the axial direction in format (2E10.0):

> **XFAKT(I), I=IG1,IG2** correction factors for the diffusion constant in radial direction and in axial direction.

For a (standard) cavity the second card of the group contains the input “0.1000” twice and the third card contains the input “0.5000” twice (see Example below), since both the fast and thermal SIGTR the value of 1/(3*cavity-radius) has been set. These numbers are valid for mid height central cavities. For extremely flat or extremely high cavities slightly modified values may be derived from /24/. For non-central holes more cavities correct figures have to be obtained by special theories. This second part of the block is closed by a blank (IS1=0) card.

EXAMPLE

This example shows the second part of the modified cavity correction factors input.

```
0
170171  1  2
 .0400      .0400
 .6000      .6000
```

```
169169  1  2
 .1000     .1000
 .5000     .5000
```

The third part of the block contains information on a detailed description of the decay heat production, and optionally on the details of a mixture of pebbles with different burn up. The first card of this part contains data (format 4E10.0) which are necessary for the application of DIN 25485 /7/. The description and default values of the variables contained in this section are shown in Table 15.

Table 15: Decay Heat Production Parameters

Variable	Default Value	Description
APD	3.0	Average power density in the core [MW/m^3]
SM	7.0	Heavy metal content in fuel pebbles with radius 3 cm [g/pebble]
BU	80.0	Average final burn-up [GWd/tU]
PUEBH	0.5	Difference between average temperature in the coated particle kernel and the adjacent matrix graphite, divided by the local power density at equilibrium conditions. The dimension of PUEBH is: [°C/(MW/m^3]. PUEBH is used to influence the dynamics of the coated particles in fast reactivity transients.

The second card allows for an arbitrary correction of the calculated TINTE decay heat and to describe the fuel element irradiation history. The data are in format (3E10.0,I10,2E10.0) and the parameters are indicated in Table 16.

Table 16: Decay Heat Modification Parameters

Variable	Default Value	Description
FNACHW	1.0	Modification factor to the decay heat, calculated by TINTE
PNACHW	-1.	Average power level in previous fuel element history. Possible options are: PNACHW = -1 – power is calculated for an equilibrium condition. PNACHW = Q – where Q > 0 is the power level [MW/m^3] specified by the user and used by TINTE to calculate the decay heat.
TNACHW	-1.0	TNACHW has a dimension of time in [sec]. This is the time during which all materials have been burnt in the equilibrium flux. The LIFE-history from VSOP99 is extended by this time. All following data on this card are ignored. Give a zero here to take over the VSOP99 data without any change. Entering TNACHW >0 is an obsolete option and should not be used. (It was used to construct an approximate life history for the fuel elements, before the option was available that allows TINTE to take over the detailed life history information from VSOP99).

EXAMPLE

This example includes input for the variables listed in Table 15 and Table 16.

```
4.5          9.          92.477       1.
1.0         -1.       -0.00000E0
```

Note that these 3 input blocks (as shown here for the previous 3 example input blocks) must be inserted directly after the last input line of the “punch.pol” (see section 6.6) file information in the .tn4 file, without an open line. Part 3 of the block above is also directly followed by the first line of the “life” history input (see the discussion below). The open lines between the blocks are also important (see Section 6.6 for a discussion on the creation of a full .tn4 file).

If the TNACHW entry on the second card is a negative number, TINTE reads on the following cards the VSOP99 fuel irradiation history. One card is read for each time interval in the fuel irradiation history, as it has been calculated by the VSOP99 code. The group of cards for each material must be entered in chronological order, starting with the most recent time interval, and the cards for one material must not be mixed with cards of another material. An example is given in Table 17.

EXAMPLE

This input line is the first line of the “life.tn4” file, created by the interface process (see Section 6.6). Note that no spaces occur between the entries.

```
1  1 4.80000E+00 3.31179E+020.000000.751100.190250.055820.000000.283910.99717
```

Table 17: Example of Fuel Irradiation History Input

Variable	Value	Description
IR	1	Material identification number .
M	1	Sequential number of the time interval. (The most recent interval is listed first).
DELTZ	4.80000E+00	Length of the time interval [days].
PZ	3.31179E+02	Average number of fissions/ sec*cm^3 in this interval, divided by rated power [W] .
X3	0.00000	Fraction of fissions in U^{233} in this time interval and material.
X5	0.75110	Fraction of fissions in U^{235} in this time interval and material.
X9	0.19025	Fraction of fissions in Pu^{239} in this time interval and material.
X1	0.05582	Fraction of fissions in Pu^{241} in this time interval and material.
B2	0.00000	Number of breeding captures in Th^{232} per fission in this time interval and material.
B8	0.28391	Number of breeding captures in U^{238} per fission in this time interval and material.
X8	0.99717	This is the remaining fraction of fissions in this time interval for this material. This figure is interpreted as fast fissions in U^{238}. This input is currently ignored, and calculated directly according to the formula X8 = 1- (X1+X3+X5+X9).

This input segment is closed with a blank card (IR=0).

Note: TINTE requires at least one card as input, so, in case of non-irradiated core, at least one card with “0” on it should be entered.

Multi-Fuel Elements Options.

Should the input contain non-blank cards after the description of the fuel irradiation history, they will be interpreted by TINTE as data for a mixture of fuel pebbles. This information is used only for materials from TINTE input block TM with material description MAT=-1. As for the fuel irradiation history, this information must currently be supplied to TINTE by the VSOP99 code (see section 6.6), or given as input manually. For each of the cards the variables is read, as indicated in the example given in Table 18.

EXAMPLE

The following line is an input line describing a mixture of fuel pebbles.

```
1  20.166666670.22114619 0.0111889
```

Table 18: Input for a Mixture of Pebble Types

Variable	Value	Description
IMAT	1	Identification number of pebble bed material region, IMAT=1.
ITYP	2	Identification number of pebble type (i.e. ITYP=2) in material region 1.
BANT	0.16666667	Fraction of pebbles of type ITYP=2 in material region 1.
FANT	0.22114619	Fraction of local fission power in pebbles of type ITYP=2
DOS	0.0111889	Fast neutron dose [10^{21}*EDN] of pebbles of type ITYP=2. Negative values indicate absence of fuel in the pebbles, i.e. they are graphite dummy pebbles.

This information is used to calculate the actual heat conductivity for the different types (e.g. in multi-pass cycles). Because of the different fission power for the different fuel types this leads to different central temperatures in the pebbles.

For each material IMAT the sum of all fractions BANT and FANT must add up to unity. The input to the block is closed by at least one blank card.

4. GENERATION OF NUCLEAR CROSS SECTIONS FOR USE IN TINTE, AND THE COUPLING WITH VSOP99

As already described in section 5.8, the nuclear cross sections are given in TINTE as polynomial expansions. The determination of the coefficients of this expansion with respect to fuel- and moderator- temperature, to xenon-concentration, to buckling, to the concentrations of nitrogen and hydrogen and with respect to the graphite corrosion asks for significant effort and may thus be performed only for a few cases without the assistance of additional computer codes.

The spectrum calculation to generate the 2-energy-group-cross- sections is the central part of these activities. A computer code is appropriate only if it can perform a single spectrum calculation with sufficient high speed, because the parameter variation asks for a great number of runs for each material zone. It would be useful to have an auxiliary code which reads the reference input for the spectrum code and generates the input stream for that code for all the varied parameters. A post-processing code should read the cross-sections that were produced by the spectrum code over the parameter variation field and the provide the expansion coefficients directly in the form to be used in TINTE by applying some least square fit algorithm.

The input for the spectrum code is composed of the nuclide vectors in each material zone and by the corresponding reference data for the parameters mentioned above. These reference data may be estimated in the first run. A later TINTE calculation writes the new reference data to the output file IPCOMP. To improve the polynomial expansion these new data should iteratively be inserted into the spectrum input with the aid of another auxiliary programme.

The nuclide vectors cannot be calculated by TINTE, but have to be taken from a burn-up code. Because of the general complexity of transferring the information from that burn-up code to the spectrum code under consideration, this task should again be accomplished by a special service code. If, as usual, the pebble flow pattern of a pebble-bed reactor is treated in detail in a burn-up code, a very fine spatial mesh is required. Such a fine mesh is not favourable for TINTE especially considering the gas- flow calculations. Inside the many core meshes the pressure is coupled to each of the four neighbour meshes, which blows up the equation system drastically and leads to both a high storage requirement and long computing times in TINTE. Therefore the definition of a special TINTE adapted mesh would be preferred. A transformation code should be available to move the nuclide vectors from the burn-up code to the spectrum code by appropriate averaging procedures.

4.1 THE SPECTRUM CODE

Any spectrum programme with high operating performance may be used. The possibility to separately calculate the thermal absorption cross sections for Xe-135, Sm-149, Sm-151 and Gd-157, as well as the fission portions for the heavy metals including a proper output format has probably to be added to the code.

To perform the calculations reported in /4/, the spectrum code TISPEC has been used, which is based on the MUPO /25/ program. For this spectrum code a set of auxiliary codes as described above has been developed. A detailed description may be found in /26/. TISPEC reaches a high speed performance by using an algorithm that solves directly the 0-dimensional diffusion equation with arbitrary leakages for 43-energy groups and replaces an explicit resonance calculation by a look-up table. Self-shielding and other heterogeneity effects may be accounted for by input of energy dependent disadvantage factors.

4.2 THE AUXILIARY CODE TO VARY THE EXPANSION PARAMETERS

This pre-processing code deals with the spectrum code reference input. It defines the multi-dimensional mesh and the sampling points in the space of the expansion parameters by estimating the boundaries and by preferring the range close to the reference values. After the output of a copy of the reference input the reference values are replaced by each reasonable combination of parameter data and the so varied input block is written to the final input stream of the spectrum code. Usually the thus generated input consists of some 10000 spectrum calculations for the whole system under consideration.

4.3 THE AUXILIARY CODE TO PERFORM THE POLYNOMIAL EXPANSION

This code takes the output of the spectrum programme and reads the cross-sections generated for all combinations of parameters. The polynomial expansion may be performed using a least square-fit procedure. Therefore the solution of a linear equation system is necessary. To establish the equation matrix, the values of the concurring functions for the parameter combinations have to be calculated.

4.4 THE ITERATION OF THE REFERENCE DATA

In defining the sampling points for the expansion mesh the whole region for a parameter has to be covered. Thereby it is possible to guess the reference values for a system for the first TINTE run. The accumulation of the sample points close to the reference data implies a better accuracy of the polynomial expansion in the vicinity of that reference point. After the first TINTE run calculated reference points are available and it is suggested to repeat the process of generating the cross-section data base using these improved figures. TINTE supports this method by writing the material averaged parameter data for the reference points to output file IPCOMP. These data may be imported to the spectral code input stream by some auxiliary programme.

4.5 COUPLING TO BURN-UP CODE SYSTEM

The coupling of the TINTE input to the results of other code systems, e.g. burn-up codes, may be supported to some extent by additional auxiliary procedures. Especially the transfer of the nuclide vector information to the above discussed support codes is identified to be a promising candidate. Other parts of the TINTE input seem to be difficult to automates. From the results of the burn-up code, e.g. available in a restart file, an auxiliary file may be extracted containing the information for the spectrum calculations. The evaluation of that file requires a varying effort mainly depending on whether the spatial mesh of the burn-up code can directly be used in TINTE or has to be replaced by another (less detailed) description. The following two examples explain what is meant here.

4.5.1 The Coupling to VSOP99

VSOP99 /27/ is a programme system for core design and fuel cycle analysis. TINTE can use part of the results of VSOP, mainly the nuclide vectors for a certain burn-up situation as input to the spectrum calculations, and the power history of the fuel elements for the determination of the proper decay heat production. The transfer of this data from VSOP99 to TINTE is almost completely automatic.

4.5.2 Identical spatial mesh

In many cases the detailed pebble flow close to the bottom reflector is of minor importance for the burn-up, so that a relative coarse mesh is sufficient. In these cases the burn-up mesh may directly transferred to TINTE and the material distribution may remain.

A special subroutine in VSOP99 in this case generates an output file to be used by the appropriate pre-processing code for TINTE. Here the nuclide identification numbers used in VSOP are replaced by those of TISPEC. Other information necessary to the TISPEC is added as well.

4.5.3 Changing the spatial mesh

The detailed treatment of the fuel flow-lines is one of the options in VSOP99. Burn- up and nuclide vectors are then calculated in a curved geometry mesh. In these cases a direct transfer of them to TINTE is not possible. However, in VSOP99 the diffusion calculation is performed using a rectangular mesh grid. This is managed within VSOP99 by special procedures involving transfer matrices. These procedures may be used to transfer the VSOP data, relevant to TINTE, to the same rectangular mesh. This includes nuclide vectors, the life history data for generating the decay heat precursors, and additionally all information on multiple-type fuel elements. The auxiliary file generated by VSOP99 in this case is amended by appropriate information.

4.6 PROCEDURE TO PRODUCE THE .TN4 FILE FROM THE VSOP99 OUTPUT FILES "NAKURE" AND "TINTE"

The .tn4 file is created by the user by performing a set of operations on two output files that are generated by the VSOP99 code: "TINTE" and "NAKURE". The following procedure can be used to create a .tn4 file:

1. Create a working subdirectory c:\PRODUCE-TN4.

2. Create in c:\PRODUCE-TN4 the subdirectories:

   c:\PRODUCE-TN4\VSOP-DATA

   c:\PRODUCE-TN4\SPECT

   c:\PRODUCE-TN4\TISPEC

   c:\PRODUCE-TN4\STEADY-STATE

3. Create in directory c:\PRODUCE-TN4\STEADY-STATE the subdirectories:

   c:\PRODUCE-TN4\STEADY-STATE\ITERATION-1

   c:\PRODUCE-TN4\STEADY-STATE\ITERATION-2

   c:\PRODUCE-TN4\STEADY-STATE\ITERATION-3

   c:\PRODUCE-TN4\STEADY-STATE\ITERATION-4

   c:\PRODUCE-TN4\STEADY-STATE\ITERATION-5

   Note: The production of a .tn4 file is an iterative process and may require as many as 5 iteration steps. The data for each iteration step will be stored in separate directories.

4. Copy the VSOP99 files “TINTE” and “NAKURE” into the directory

c:\PRODUCE-TN4\VSOP-DATA

5. Copy the .tn3 file (e.g. PBMR400.tn3) that corresponds to the VSOP input data into the subdirectory

c:\PRODUCE-TN4\STEADY-STATE

Note: The user could create the .tn3 file by using the VSOP defined model as a starting point.

6. Run the procedure “INTERF.EXE”:

Go to subdirectory c:\PRODUCE-TN4\VSOP-DATA

6.1. Copy the VSOP99 files “TINTE” and “NAKURE”, as well as the file “INTERF.EXE” in the directory c:\PRODUCE-TN4\VSOP-DATA.

6.2.From the DOS prompt, enter:

c:….\ interf

6.3. TINTE displays the following messages:

Beginn INTERF-Rechnung

Titel des von VSOP erstellten TINTE - Files:

TINTE-LIB., CASE 2006, 400 MW, 6X

Beginn der Umrechnung von VSOP - auf TINTE - Maschen

Daten des TINTE - Files

Daten des NAKURE - Files

Daten zur Mischung von BE/Typen

Die VSOP-Rechnung beruht auf maximal 6 Durchlaeufen

(Batches of fuel pebbles of up to 6 passes through the core have been identified).

Wieviele BE-Typen sollen beruecksichtigt werden?

(How many Fuel elements types should be considered?).

Note: The maximum number the user can enter is the number of fuel passes through the core, in this case: 6.

Ende der Umrechnung von VSOP - auf TINTE - Maschen

(End of conversion of VSOP meshes to TINTE meshes).

Daten fuer Output-Files erstellen

Welche KAPPA -Werte sollen benutzt werden?

(0.0000000E+00 fuer Th232 und 0.4000000 fuer U238)

The user enters: 0.0 and 0.4, with a space in-between the two numbers.

Beginn TISPEC - Input erstellen

Material 169 scheint ein Hohlraum zu sein (conc(c)= 5.01419E-09),

aus dem Input-Stream entfernen? 0<=>ja, sonst nein

(Material 169 seems to be a cavity. Must it be deleted from the input stream?)

The user enters: 0 , since material 169 represents indeed a cavity (as an example- cavities can be represented by any number!).

If the material identified is not a cavity, press enter without any number input.

TISPEC - Input erstellt.

Beginn LIFE.tn4 - File erstellen

LIFE.tn4 - File erstellt

**** ENDE DER RECHNUNG ****

Two new files have now been produced:

LIFE.tn4 - the second section of the new .tn4 file.

TISPEC.INP - input file for the procedure TISPEC.

Note: The file LIFE.tn4 contains data needed to calculate the residual decay power levels from the fuel: the irradiation history of the fuel, fraction of fissions in various nuclides, etc.

6.4 Copy file LIFE.tn4 into subdirectory c:\PRODUCE-TN4\STEADY-STATE

6.5 Copy file TISPEC.INP from directory c:…\PRODUCE-TN4\VSOP-DATA

to directory c:…\PRODUCE-TN4\SPECT.

Note that due to the VSOP input specifications, interf.exe might not pickup some of the void regions of the VSOP input file. The void regions that remain in the TISPEC.INP might be identified by the very low Carbon concentration. The following is an example of an ordinary reflector region copied from TISPEC.INP

```
****  MODUL  **REFL444***              0.  444                8
0
2 25 43
10.         0.77300E+03  0.0        1.0            1.
8.40000E+01 7.62460E-08
2.00000E+01 7.28030E-02 7.73000E+02 3.88000E+00
1.93000E+02 0.00000E+00 7.73000E+02 6.60000E+02
8.00000E+01 0.00000E+00
```

If this was a void region the highlighted Carbon concentration would change as follows:

```
****  MODUL  **REFL444***              0.  444                8
0
2 25 43
10.         0.77300E+03  0.0        1.0            1.
8.40000E+01 7.62460E-12
2.00000E+01 7.28030E-02 7.73000E+02 3.88000E+00
1.93000E+02 0.00000E+00 7.73000E+02 6.60000E+02
8.00000E+01 0.00000E+00
```

6.6 Go to subdirectory c:…\PRODUCE-TN4\SPECT

7. Run the procedure “SPECT.EXE”:

7.1. From the DOS prompt:

c:..\PRODUCE-TN4\SPECT\SPECT0 TISPEC.INP

<enter>

c:..\PRODUCE-TN4\SPECT\tispec.exe

<enter>

Note: The files “spect0.bat” and “tispec.exe” should be located in this folder.

7.2. A new directory, TISPEC.TMP is produced which contains the following files:

OUTPUT.TSP - output from SPECT

PUNCH.TSP

TISPEC.HSP

TISPEC.INI

If a message "Normal End" is written by TINTE, then copy the file TISPEC.INP to subdirectory c:\PRODUCE-TN4\TISPEC and go to that directory.

7.3. Should the procedure end with message "Abnormal End", open OUTPUT.TSP.

7.4. Go to file end of file OUTPUT.TSP and back page to see the component which caused the error message. Most often, the component that has caused an “Abnormal end “ message is a mixture of too much reflector graphite and too little fuel material. Assume it is component "23", and assume component “24” is pure reflector material. Since component “23” contains very little fuel and is mostly reflector graphite, it can be replaced entirely by another component that contains only reflector graphite, i.e. by component “24”.

7.5. Edit file TISPEC.INP to remove the cause for the error.:

Open file TISPEC.INP and find the component ("23") that caused the error. Select the block of lines with data for component 23 and then delete them. Select the block of lines with data for component 24 and then copy them below existing component “24”. Change in the copied lines both strings "24" to "23".

7.6. Save the edited file TISPEC.INP as TISPEC-EDIT1.INP in subdirectory c:\PRODUCE-TN4\SPECT.

7.7 Repeat procedure SPECT0 (point 7.1 above) with the corrected TISPEC-EDIT1.INP file. If a message "Normal End" is produced", then copy file TISPEC-EDIT1.INP to subdirectory c:\PRODUCE-TN4\TISPEC and go to that directory.

8. Run the Procedure “TISPEC.exe”

8.1. From the DOS prompt:

c:\PRODUCE-TN4\TISPEC\TISPEC0-am.bat TISPEC.INP SPECLIBN-r04.lib

Note: The files “SPECLIBN-r04.lib” and “tispec0-am.bat” and “tispec.exe” should be located in this folder. SPECLIBN-r04.lib is the libraries name that is used for the specific version. Different libraries can be used with the different versions and there is no restriction on the library name.

A new directory, “TISPEC.TMP”, is produced which contains the following files:

FT07.TMV

FT19.TMV

OUTPUT.TMV

OUTPUT.TSP

PUNCH.POL

PUNCH.TSP

ISPEC.HSP

TISPOL.OUT

TISVOR.HCP

8.2. Copy the file PUNCH.POL into the directory c:\PRODUCE-TN4\STEADY-STATE.

8.3. Go to the directory c:\PRODUCE-TN4\STEADY-STATE.

9. Produce the first iteration for the .tn4 file

9.1. The following files now must be in c:\PRODUCE-TN4\STEADY-STATE subdirectory:

c:\PRODUCE-TN4\STEADY-STATE\PBMR400.tn3

c:\PRODUCE-TN4\STEADY-STATE\LIFE.tn4

c:\PRODUCE-TN4\STEADY-STATE\PUNCH.POL

c:\PRODUCE-TN4\STEADY-STATE\TDB2.EXE

c:\PRODUCE-TN4\STEADY-STATE\TISPEC.INP

c:\PRODUCE-TN4\STEADY-STATE\TINTE.exe

where the .tn3 file must be the correct file for the problem.

9.2. Create a new file c:\PRODUCE-TN4\STEADY-STATE\PBMR400-ITER_0.tn4

9.3. Enter in it, on the first line in the first column:

NQ%

followed by an arbitrary string of up to 80 characters on the first line, used primarily as a comment statement by the user.

9.4. Copy into the file PBMR400-ITER_0.tn4, from the second line, in the first column, the contents of the file PUNCH.POL.

9.5. At the end of the file PBMR400-ITER_0.tn4 input lines describing the void regions in the .tn3 file should be added, since TINTE uses a special method to calculate neutron streaming effects in void areas. The user should identify all void areas in the NZ block of the .tn3 file- these material numbers must correspond to the numbers that were identified by TINTE in point 6.3 above. As an example, assume that nuclear material numbers 169, 170 and 171 were identified as void regions. The input by the user will then look as follows:

*WARNING: The .tn4 file is read in fixed FORTRAN format, which means that open spaces or lines in the wrong place could produce erroneous results. (Remember that TINTE by default interprets all empty lines and spaces as 0 values). In this example, the character * will be used to indicate a space or an open line. The sample input (from the PBMR 400.tn4 file) starts with the last two lines of the "punch.pol" file, and ends with the first two lines of the "life.tn4" file (see 9.6 below).*

```
322 5000000  2-1.7053E-17-3.7631E-20 0.0000E+00-5.7484E-22 0.0000E+00
               0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
169*0000000**1*1.8018E-03
169*0000000**2*1.8018E-03
**********************************************************************
**********************************************************************
170*0000000**1 3.3333E-02
170*0000000**2 3.3333E-02
**********************************************************************
**********************************************************************
171*0000000**1 3.3333E-02
171*0000000**2 3.3333E-02
**********************************************************************
**********************************************************************
0
170171**1**2
 .0371******.0371
 .6431******.6431
169169**1**2
 .0955******.0955
 .5053******.5053
**********************************************************************
4.8********9.*********92.******1.
 1.0*******-1.******-1.18000E6********10*******3E5*****3.5E6
   1  1 4.80000E+00 3.31179E+020 .000000 .751100 .190250 .055820 .000000
   1  2 5.12431E+00 5.55419E+010 .000000 .723020 .214540 .059430 .000000
```

Note that the user has to create all the normal (i.e. not italic) typeface entries (see Section 5.8). This central data block is the only place in the .tn4 file (apart from the title line) where the user has to write input. The rest of the file consists only of the "punch.pol" data in the first data block, the "life.tn4" data in the last data block, and some very important spaces in-between.

9.6 Copy the contents of the file "LIFE.tn4" after the last line with data for the void components. Note that an empty line does <u>not</u> occur between the two data blocks!

9.7 Save the file as c:\PRODUCE-TN4\STEADY-STATE\PBMR400-ITER_0.tn4.

9.8 Run TINTE, using the .tn3 and the zero-th iteration .tn4 file as input files (see Section 5.2).

9.9 If the run was successful, TINTE has now produced a file PBMR400-ITER_0.TTB. The idea now is to create better converged .tn4 files by creating new “punch.pol” files after each TINTE iteration run.

9.10 Run Procedure TDB2: From the DOS prompt:

C:\….\TDB2

Once inside the TDB2 dialog, enter

PBMR400-Iter_0 TISPEC.INP TISPEC-Iter_1.INP

The TDB2 code produces a file TISPEC-Iter_1.INP.

9.11 Copy this file in directory : c:\…\PRODUCE-TN4\STEADY-STATE\TISPEC\. Copy into the same directory the files

TISPEC0-am.bat

SPECLIBN-r04.lib

TISPEC.EXE

TISPOL.EXE

TISVOR.EXE

9.12 Run the TISPEC procedure:

c:...\TISPEC0-am.bat TISPEC-Iter_1 SPECLIBN-r04.lib

9.11 It will produce a directory TISPEC.TMP and a file PUNCH.POL contained in it. Copy the file PUNCH.POL into the subdirectory

c:\PRODUCE-TN4\STEADY-STATE\ITERATION-1

9.12 Go to subdirectory c:PRODUCE-TN4\STEADY-STATE\ITERATION_1

10. Produce second iteration for file PBMR400.tn4

10.1. Open the file PBMR400-ITER_0.tn4 and delete all lines from the second one down to line above the one starting with the void regions data (i.e. delete the first iteration PUNCH.POL data block).

10.2. Copy the contents of file PUNCH.POL into file PBMR400-ITER_0.tn4 starting after first line.

10.3. Save the file as PBMR400-ITER_1.tn4.

10.4. Run TINTE, using the .tn3 and the first iteration .tn4 file as input files.

11. Iteration Success Criterion

The above iteration procedure should be repeated until a PBMR400_ITER.tn4 file is produced that results in an equilibrium TINTE run that converges after a few time steps. The .tn4 file can now be renamed to something more suitable, and used for further calculations.

5. TINTE CALCULATIONAL PROCEDURE

5.1 PROCEDURE TO PRODUCE A TINTE STEADY-STATE (OR EQUILIBRIUM) CALCULATION

General Notes

1. Prior to calling any of the TINTE programs, issue the command TiPS once in the WINDOWS session. This must be done from a DOS window or WINDOWS shortcut, but not from a secondary command processor like Norton Commander.
2. To terminate a TINTE calculation (while still busy) in a controlled way, the "HE" command can be used by typing "HE" from the same directory where the calculation was started from. TINTE will then end the calculation prematurely and generate a final output dataset, including an updated restart file.
3. To produce various ASCII plot data outputs from the binary .ptr or .p2d data files, the command: "TPInt <.tn3 file-name> <.ptr/.p2d file-name>" can be used. (For help with these -post-processing functions, type: "TPInt /?"). The commands "ptrb2a <.ptr file-name> <new_output file-name>" and "p2db2a <.p2d file-name> <new file-name>" convert the binary .ptr and p2d files directly to a readable ASCII format, if needed for data manipulation.

A TINTE equilibrium calculation is required to define the initial conditions from which any transient calculation will commence. Therefore, it is often advisable to execute first a separate TINTE run with only a steady-state calculation and thereby to produce a restart file that will then be used to start the transient computations in subsequent TINTE calculations.

Step 1: Input Data Preparation

The input files with extensions .tn3 and .tn4 are needed for every nuclear steady-state calculation (except if it is a restart of a n earlier case). The input files must all be placed in the same directory, and the output file generated will also be stored in this directory. (For illustration purposes it is assumed that TINTE is located in the directory "c:\TINTE"). Note that for a steady-state calculation of a problem that contains no nuclear fuel, the .tn4 file is not needed.

Step 2: Start TINTE

TINTE is initialised from the DOS command prompt by typing:

c:\TINTE> tinte test.out

where the user specifies the TINTE output file name (here called "test.out"). The output file can have any suffix that the user specifies.

Step 3. Enter the input file specifications

TINTE displays the following message after step 2 is completed:

Eingabe der Namen:

→ *der .tn3 - Datei*

→ *der .tn1 - Datei*

→ *einer .rtn - Datei*

der .tn4 - Datei

der .p2d - Datei

(mindestens einen mit der → markierten angeben!)

The user is asked to enter the names of the .tn3 (geometry and materials input), .tn1 (time and code option control input), .rtn (restart data output for transient phase), .tn4 (nuclear data input) and .p2d (plotting 2-D data output) files. Note that the sequence of filenames is fixed, and if any filename is not specified, an asterisk * character must be typed in the input line. Spaces must also be inserted between every filename. The .rtn and .p2d filename specifications are optional when performing an equilibrium run, since TINTE will use the .tn3 filename to create default filenames for these two files. Note that the .tn1 and .tn4 files are also optional: the .tn1 data can be entered interactively from the console, while the .tn4 data is only needed if a nuclear cross-section dataset exists (i.e. when nuclear fuel is present in the model). For example, the input line

*PBMR400 * * PBMR400 **

results in the following TINTE response:

Verknuepfung von Fortrannummern und Dateien

3 PBMR400.tn3

4 PBMR400.tn4

7 fort.7

8 PBMR400.ttb

9 dummy.ptr

10 PBMR400.rtn

11 dum11.rtn

12 dummy.p2d

15 test.out

This step is used by TINTE to couple Fortran device numbers to the filenames specified. For example, the file pbmr400.tn3 is now defined in the TINTE code as Fortran device #3, while the data output file ("test.out") is defined as device #15.

Warning: *TINTE will exit at this point with the following error message:*

```
KEINE GUELTIGE TITELKARTE GEFUNDEN
  IFILE,KARTE           4
 NZ*
```

if the .tn3 or .tn4 files specified can not be found in the directory where the runs are performed. The number following the KARTE text indicates the FOTRAN device number that is missing (see step 4 below). For example, 1,3,4 or 10 implies a missing tn1, tn3, tn4 or rtn file. It will also then create ***empty*** *files with the names that the user specified, which might complicate further calculations if these empty files are not deleted before the next calculation is attempted. Also note that TINTE could exit if the user specifies an output file name that already exists, i.e. overwriting of existing output files is prohibited by TINTE, for data security reasons, under certain operating systems.*

Step 4. Initialization of input and output controls

The next TINTE message shows which file is assigned to a particular FORTRAN device:

Verknuepfung von Fortrannummern und Dateien

3 PBMR400.tn3

4 PBMR400.tn4

7 fort.7

8 PBMR400.ttb

9 dummy.ptr

10 PBMR400.rtn

11 dum11.rtn

12 dummy.p2d

15 PBMR400.o

The above TINTE message means:

- File PBMR400.tn3 is assigned to device # 3
- File PBMR400.tn4 is assigned to device # 4
- File FORT.7 is assigned to device # 7
- File PBMR400.TTB is assigned to device # 8
- File DUMMY.PTR is assigned to device # 9
- File PBMR400.RTN is assigned to device #10
- File DUM11.RTN is assigned to device #11
- File DUMMY.P2D is assigned to device #12
- File PBMR400.O is assigned to device #15

By default, FORTRAN device #5 is the console.

It should be noted that TINTE uses also two additional FORTRAN devices, units 5 and 6, the usual console input and output streams. Unit 5 is used for interactive input of the control commands (see section 8) if a .tn1 file is not used. (This input can also be redirected from a file through a piping clause at the commandline). Unit 6 contains the console output of TINTE which essentially is a compacted version of the print file output (on unit 15, by default); it does not comprise the larger data arrays like the 2D-fields, the heat balance tables and the final transient history tables.

TINTE input prompt:

BITTE EIN- /AUSGABE-FILES ZUORDNEN: (0 ==> Vordef., >0 : File, <0 : unbenutzt)

1. IIRST (0) RESTART-INPUT

2. IORST (0) RESTART-OUTPUT (DOK.)

3. IERST (0) RESTART-OUTPUT (NORMAL)

4. ISTEU (3 BZW. 0 (1.>0)) "ST" INP.

5. IMESH (3 BZW. 0 (1.>0)) "GM" INP.

6. ITMAP (3 BZW. 0 (1.>0)) "TZ" INP.

7. ITMEI (3 BZW. 0 (1.>0)) "TM" INP.

8. INMAP (3 BZW. 0 (1.>0)) "NZ" INP.

9. ISIGM (4 BZW. 0 (1.>0)) "NQ" INP.

10. IZEIT (5 BZW. 1 (tn1 vorh.) "TT" INP

11. IPRINT (15) PRINTER-OUTPUT

12. IPMAP (0) REAKTOR-BILD (p)

13. IPCOMP (8) X-ITERATIONSDATEN

14. IPZEIT (9) TRANS.DATA (binary)

15. IPMON (0) VIEL OUTPUT (p)

16. IPMOFU (0) FUEL/MOD.-TEMP. (p)

17.IPGASD (0) STROEMUNGS-OUTP.(p) ggf. + "KG" INP.

18. IPGAST (15) GAS-TEMPERATUR

19. IPFEST (15) FESTSTOFF-TEMP.

20. IPPOW (15) WAERMEQUELLEN

21. IPP2D (12) 2-D OUTPUT (binary)

N.b.: (p) ist optionaler Printer-Output und kann z.B. 15 zugeordnet werden.

Example User input:

*2*0 10 5*3 4 2*0 15 3*0 2*15*

NOTE: All 21 flags have to be entered as a single record. If less than 21 items are present, the missing will be entered by the code as zeros. If the user wants to use only the flags' default values, a blank card should be entered as an initialization record. The following input line means the same as the input given above, except for the reactor map (point 12 in the list above) that will not be shown:

*2*0 10 12*0 15*

TINTE prompt (on the input line *2*0 10 5*3 4 2*0 15 3*0 2*15*):

RESTART EINGABE	*IIRST = 0*
RESTART AUSGABE DOKUMENTIEREND	*IORST = 0*
RESTART AUSGABE UEBERSCHREIBEND	*IERST = 10*
EINGABE DER STEUERGROESSEN	*ISTEU = 3*
EINGABE DER GEOMETRIE	*IMESH = 3*
EINGABE DER TEMPERATURMATERIALIEN	*ITMAP = 3*
EINGABE DER T.MAT.-EIGENSCHAFTEN	*ITMEI = 3*
EINGABE DER NUKLEARMATERIALIEN	*INMAP = 3*
EINGABE DER NUKL. QUERSCHNITTE	*ISIGM = 4*
EINGABE ZEITABHAENGIGER GROESSEN	*IZEIT = 5*
ALLGEMEINE AUSGABEEINHEIT	*IPRINT = 15*
AUSGABE EINES REAKTORBILDES	*IPMAP = 15*
AUSGABE FUER JEDE COMPOSITION	*IPCOMP = 8*
AUSGABE ZEITABHAENGIGER GROESSEN	*IPZEIT = 9*
AUSGABE DES ITERATIONSMONITORS	*IPMON = 0*
AUSGABE DER MOD.U.FU.TEMPERATUREN	*IPMOFU = 15*

AUSGABE VON GASDRUCK UND -STROEM. *IPGASD = 15*

AUSGABE DER GASTEMPERATUR *IPGAST = 15*

AUSGABE DER FESTSTOFFTEMPERATUR *IPFEST = 15*

AUSGABE DER LEISTUNGSVERTEILUNG *IPPOW = 15*

PLOTAUSGABE DER 2D-FELDER *IPP2D = 12*

The TINTE FORTRAN numbers are expalined in Table 19.

Table 19: TINTE Flag Detail

No.	Flag Name	Value	Deflt. Value	Meaning
1	IIRST	0	0	TINTE run is NOT started from restart file.
2	IORST	0	0	"Documenting" restart file will NOT be created.
3	IERST	10	0	"Overwriting restart file" WILL be created. Since the user did not specify the restart file name, TINTE will create it as PBMR400.RTN.
4	ISTEU	3	3 (0)[1]	Read Block 1 (starting with string ST*) from FORTRAN device 3, i.e file PBMR400.tn3.
5	IMESH	3	3 (0)[3]	Read Block 2 (starting with string GM%) from FORTRAN device 3, i.e file PBMR400.tn3.
6	ITMAP	3	3 (0)[3]	Read Block 3 (starting with string TZ*) from FORTRAN device 3, i.e file PBMR400.tn3.
7	ITMEI	3	3 (0)[3]	Read Block 4 (starting with string TM*) from FORTRAN device 3, i.e file PBMR400.tn3.
8	INMAP	3	3 (0)[3]	Read Block 5 (starting with string NZ*) from FORTRAN device 3, i.e file PBMR400.tn3.
9	ISIGM	4	4 (0)[3]	Read Block 6 (starting with string NQ% from FORTRAN device 4, i.e file PBMR400.tn4.
10	IZEIT	0	5 (1)[2]	Read Block 7 (starting with string TT*) from FORTRAN device 5, i.e. console (since user did not specify a .tn1 file at step 3).
11	IPRINT	0	15	Print output in FORTRAN device 15, i.e. file PBMR400.O.
12	IPMAP	15	0	User entered 15, thus TINTE prints a picture of reactor nodalization into output file PBMR400.O.
13	IPCOMP	0	8	User entered default value, i.e. 0, so TINTE creates file PBMR400.TTB on FORTRAN device #8.
14	IPZEIT	0	9	User entered default value, i.e. 0, so TINTE creates a file with extension "PTR" on FORTRAN device 9 for the binary time-dependent scalar variables output for post-processing as graphics. Since no .tn1 file name was specified in step 3, this file is named DUMMY.PTR. For a steady-state calculation, TINTE will NOT write anything in it.
15	IPMON	0	0	User entered default value, i.e. 0, so TINTE does NOT create a file with detailed iteration monitoring.
16	IPMOFU	15	0	User entered 15, so TINTE writes in output file PBMR400.O (FORTRAN device#15) moderator and fuel temperatures.

[1] 0 is the default value if IIRST > 0 (see item no. 1)

[2] 1 is the default value if a .tn1 file is specified

No.	Flag Name	Value	Deflt. Value	Meaning
17	IPGASD	15	0	User entered 15, so TINTE writes in output file PBMR400.O (FORTRAN device#15) gas pressure, mass flow rate, and flow paths.
18	IPGAST	0	15	User did not enter device number, which signals the use of the flag default value 15. As a result, TINTE writes in the output file PBMR400.O (FORTRAN device #15) gas temperatures.
19	IPFEST	0	15	User did not enter device number, which signals the use of the flag default value 15. As a result, TINTE writes in the output file PBMR400.O (FORTRAN device #15) the temperatures of solids, i.e. pebbles, reactor component, etc.
20	IPPOW	0	15	User did not enter device number, which signals the use of the flag default value 15. As a result, TINTE writes in the output file PBMR400.O (FORTRAN device #15) data describing the convective heat transfer, heat sources (local, non-local, and external), and a heat balance table for all materials.
21	IPP2D	0	12	User did not enter device number, which signals the use of the flag default value 12. As a result, TINTE creates a file with extension “P2D” (FORTRAN device #12) that contains two-dimensional array with data for post-processing as graphics. For the transient part of TINTE computations, this output can be directed by the user through the TINTE control commands (see section 8.3).

No.	Flag Name	Value	Deflt. Value	Meaning
1	IIRST	0	0	TINTE run is NOT started from restart file.
2	IORST	0	0	“Documenting” restart file will NOT be created.
3	IERST	10	0	“Overwriting restart file” WILL be created. Since the user did not specify the restart file name, TINTE will create it as PBMR400.RTN.
4	ISTEU	3	3 (0)[3]	Read Block 1 (starting with string ST*) from FORTRAN device 3, i.e file PBMR400.tn3.
5	IMESH	3	3 (0)[3]	Read Block 2 (starting with string GM%) from FORTRAN device 3, i.e file PBMR400.tn3.
6	ITMAP	3	3 (0)[3]	Read Block 3 (starting with string TZ*) from FORTRAN device 3, i.e file PBMR400.tn3.
7	ITMEI	3	3 (0)[3]	Read Block 4 (starting with string TM*) from FORTRAN device 3, i.e file PBMR400.tn3.
8	INMAP	3	3 (0)[3]	Read Block 5 (starting with string NZ*) from FORTRAN device 3, i.e file PBMR400.tn3.
9	ISIGM	4	4 (0)[3]	Read Block 6 (starting with string NQ% from FORTRAN device 4, i.e file PBMR400.tn4.
10	IZEIT	0	5 (1)[4]	Read Block 7 (starting with string TT*) from FORTRAN device 5, i.e. console (since user did not specify a .tn1 file at step 3).
11	IPRINT	0	15	Print output in FORTRAN device 15, i.e. file PBMR400.O.

[3] 0 is the default value if IIRST > 0 (see item no. 1)

[4] 1 is the default value if a .tn1 file is specified

No.	Flag Name	Value	Deflt. Value	Meaning
12	IPMAP	15	0	User entered 15, thus TINTE prints a picture of reactor nodalization into output file PBMR400.O.
13	IPCOMP	0	8	User entered default value, i.e. 0, so TINTE creates file PBMR400.TTB on FORTRAN device #8.
14	IPZEIT	0	9	User entered default value, i.e. 0, so TINTE creates a file with extension “PTR” on FORTRAN device 9 for the binary time-dependent scalar variables output for post-processing as graphics. Since no .tn1 file name was specified in step 3, this file is named DUMMY.PTR. For a steady-state calculation, TINTE will NOT write anything in it.
15	IPMON	0	0	User entered default value, i.e. 0, so TINTE does NOT create a file with detailed iteration monitoring.
16	IPMOFU	15	0	User entered 15, so TINTE writes in output file PBMR400.O (FORTRAN device#15) moderator and fuel temperatures.
17	IPGASD	15	0	User entered 15, so TINTE writes in output file PBMR400.O (FORTRAN device#15) gas pressure, mass flow rate, and flow paths.
18	IPGAST	0	15	User did not enter device number, which signals the use of the flag default value 15. As a result, TINTE writes in the output file PBMR400.O (FORTRAN device #15) gas temperatures.
19	IPFEST	0	15	User did not enter device number, which signals the use of the flag default value 15. As a result, TINTE writes in the output file PBMR400.O (FORTRAN device #15) the temperatures of solids, i.e. pebbles, reactor component, etc.
20	IPPOW	0	15	User did not enter device number, which signals the use of the flag default value 15. As a result, TINTE writes in the output file PBMR400.O (FORTRAN device #15) data describing the convective heat transfer, heat sources (local, non-local, and external), and a heat balance table for all materials.
21	IPP2D	0	12	User did not enter device number, which signals the use of the flag default value 12. As a result, TINTE creates a file with extension “P2D” (FORTRAN device #12) that contains two-dimensional array with data for post-processing as graphics. For the transient part of TINTE computations, this output can be directed by the user through the TINTE control commands (see section 8.3).

Step 5.

TINTE prompt:

Bitte CPU- und Real-Rechnen-Zeit-Schranken (min), sowie ggf. Zeilenlaenge eingeben (0 ⇔ keine Beschrankung (bzw. 256 C/Z)) .

This record establishes the limits for maximum allowed CPU time or real time (both measured in minutes) to be used for the TINTE run. The third item is the line width of the print output, for formatting the 2D-arrays (default = 256). Should no such limits or width change be desired, the user must enter 0 or a blank line here.

User Input:

Press “Enter key” to input an blank line

Step 6.

TINTE prompt:

Zeitsteuerungsinput von TT oder TT% !!!*

Enter bei Leerzeile bricht Rechnung ab!

User Input:

*TT**<User may enter maximum 80-characters>

Step 7.

TINTE prompt:

Bitte Zeitsteuerinput:

User Input (See Section 8.3):

–1

Step 8.

TINTE prompt:

The code will commence a calculation to achieve a steady-steate condition.

Step 9

When the calculation ends, TINTE will terminate the console output by a message:

Waerme-Bilanz …. (S.Printer-Output!)

Bitte Zeitsteuer-Input:

User Input (See Section 8.3):

–4

TINTE prompt:

TINTE: STOP in Subroutine TIRSTI:

Steuerparameter –4 gelesen

Ausfuehrlichere Meldung siehe eventuell im Printer-Output!

The code will write a restart file (e.g. PBMR400.RTN). The user should then press <enter> to exit the DOS screen.

Note: If a continous steady-state and transient calculation is required, the user could continue the run by using further control commands (see section 8).

5.2 PROCEDURE TO PRODUCE A TINTE TRANSIENT CALCULATION

All TINTE transient calculations (i.e. calculations where a variable, or more than one variable, changes in value over time) must be preceded by a complete steady-state calculation, or at least use the steady-state restart file as a starting point for the transient. The easiest way of

controlling the transient calculation is by specifying the transient control parameters in an input file with the mandatory extention of .tn1.

The input can of course also be given by the user in real time during the transient, but since typical transients takes 2-4 hours to complete, this is not always a practical solution. The syntax for the transient calculation is identical to the steady-state part, but a wide variety of transient control options are availble to the user (see Section 8).

Step 1: File specifications

The input files with extensions .tn3 and .tn4 are again needed for the transient calculation, but since the contents of these files are contained in the steady-state restart file, there are 2 options available to the user. Recalling that the input file sequence must be

Eingabe der Namen:

→ *der .tn3 - Datei*

→ *der .tn1 - Datei*

→ *einer .rtn - Datei*

der .tn4 - Datei

der .p2d - Datei

(mindestens einen mit der → markierten angeben!)

the first option input line would look as follows:

*PBMR400 scram * PBMR400 **

resulting in the following TINTE response:

Verknuepfung von Fortrannummern und Dateien
1 scram.tn1
3 pbmr400.tn3
4 pbmr400.tn4
7 fort.7
8 pbmr400.ttb
9 scram.ptr
10 pbmr400.rtn
11 scram.rtn
12 scram.p2d
15 scram.out

The user will then have to commence with a steady-state calculation (using control command "-1"), and directly thereafter start with the transient run, if the control options in the .tn1 file are correctly specified (see section 8).

The second option is to start from the restart file that was created at the end of the steady-state calculation. In this case the input line would look as follows:

** scram pbmr400 * **

and it would result in the following TINTE response:

Verknuepfung von Fortrannummern und Dateien
1 scram.tn1
7 fort.7
9 scram.ptr
10 pbmr400.rtn
11 scram.rtn
12 scram.p2d
15 restart.out

The next input line form the user in this case should indicate that a restart file is being used:

*10 0 11 12*0 15*

TINTE will now use the file on device 10 (i.e. pbmr400.rtn) as the input restart file for the transient calculation, and will use the file on device 11 (scram.rtn) as the overwriting restart file during the transient run. In this second case, the .tn1 file would not contain the "-1" control command for the steady-state calculation (otherwise it would be repeated unnecessarily in the restart case).

Both of these options would result in the same output, except that time is saved by not re-doing the steady-state calculation again every time that the user wants to start a transient calculation.

5.3 SAMPLE ERROR AND WARNING MESSAGES

This section includes some examples of the TINTE error messages that could be experienced during the execution of the code. It is by no means intended as an exhaustive list of error messages, and the remedies suggested are also not the only possible solutions to the errors. Very few error messages in the TINTE code are in English, so a first step towards successful resolution of the errors is creative use of a German (technical) dictionary. Note that not all error messages appear in the screen dump of the active TINTE window; most of then will be in the main output file. It is also instructive at times to trace the error message back to the source files, and try to figure out if it is a format or missing data problem.

- *TINTE Message:*

  ```
  Die gelesenen angaben zu den iterationsmaschen sind fehlerhaft,
  die summe ergibt x, statt y. (dies zieht wahrscheinlich
  folgefehler nach sich)

  (The iteration mesh inputs are incorrect: the sum equals x, in
  stead of y. This is probably a source for follow-up errors from
  an earlier error).
  ```

  There is a mismatch between the GM block input (or the NZ block) and the final 2 cards in the tn3 file, where a coarser nuclear mesh can be used. Check that the total number of *r* and *z* meshes match between the two blocks. A related (usually follow-up) message to this problem is

  ```
  Eingabefehler bei den nuklearen iterationsmaschen: x (dies ist
  wahrscheinlich ein folgefehler)

  (Input error for the nuclear iteration mesh x. This is probably
  a follow-up error from an earlier error).
  ```

- *TINTE Message:*

  ```
  Die querschnittskarte x fuehrt zu dimensionsueberschreitungen.

  (The cross-section card x resulted in dimensional overwriting).
  ```

  This message points to a NQ (tn4 file) block input card for a nuclear material in the NZ mesh. Check that the number of NQ materials is not larger than the number of NZ materials, or if the format of the identified nuclear material card is specified correct. The number of NQ materials needs not to be equal to the number of NZ materials.

However each NZ material must be found in NQ and the highest NQ material number must be found in the NZ block.

An open line, or the lack of one at specific positions, can also lead to this message.

- *TINTE Message:*

```
Die life-history fehlt.
```

The decay heat data from the VSOP interface „Life“ file is missing. An open line might cause this error as well, in stead of a continuous block of data where the code expects it.

- *TINTE Message:*

```
Es wurde kein gasdruck vorgegeben, keine gasstroemung und
konvektion moeglich.
```

A reference system pressure was not specified, so no gas flow or convective calculations are possible. Related pressure or flow rate error messages are

```
1. Stroemungsberechnung fuehrt zu negativem druck in masche
j,i=x,y. Konvergenz im gewaehlten maschennetz nur schwer zu
erreichen.

(Gas flow calculations resulted in a negative pressure in mesh
j,i. Convergence in the given mesh structure can only be
obtained with difficulty, or not at all).

2. Gasdruck, -Temperatur oder -Zusammensetzung sind fehlerhaft

(gas pressure, temperature or combination is wrong)
```

(here usually a subroutine name and a position number is printed, which can be traced in the titini.f source file).

and

```
3. Unsinniger druck berechnet

(Invalid/non-physical pressure calculated)
```

The last three examples can occur when an isolated flow mesh is specified with all-solid neighbors (i.e. a material type 3 or 4 surrounded by material types 0), or where very large or very small pressure gradients build-up in the flow mesh. The mesh index j,i are a possible clue to where the problem occurs, but be aware that the specified mesh might not be the true culprit, but only the final victim of bad input data. These errors (especially the second one) may be very time-consuming to solve!

- *TINTE Message:*

```
0benoetigter speicherplatz x, vorhandener speicherplatz y.
```

This error occurs when the total memory array requirement x, is larger than the current TINTE array size y. It can be fixed by decreasing the number of fine and/or coarse meshes in the model. (There is still a bug in the calculation of memory requirement. Trouble may occur even if the indicated x is less than y. A bugfix is in progress).

- The following TINTE error messages are all related to input for time or variable parameter control ramp errors(see Section 8):

```
1. Eingabefehler, fuer das material x sind keine zeitabhaengigen
parameter vorgesehen
```

(Input error: no time-dependent parameters can be specified for material x).

2. Zeits,zeitz,wertz,iort,iart nicht identifizierbar auf karte

(Variables Zeits,zeitz,wertz,iort,iart (see Table 20) is not identifiable on the input card).

3. Es wurde ein ungueltiger befehl eingegeben.

(A invalid command is given).

All these errors can be solved by reviewing the input, as discussed in Section 8.

- The following warning occurs when a prompt jump is made in any time-dependent variable (i.e. ZEITZ < ZEITS in Table 20):

  Sprunghafte veraenderung auf karte x kann konvergenzschwierigkeiten verursachen

  It will allow continuation of the TINTE run, but convergence problems may be expected.

6. TINTE CONTROL COMMANDS

After the six input blocks described in Section 5 have been given, the TINTE calculations may start. With this seventh time control block the calculation is controlled. This is done by means of specifying linear time dependencies (ramps) for the variable, externally defined parameters or by means of direct control commands defining the type of the calculation. The title card of this block starts with either TT% (indicating formatted input lines are following with format: E10.0, 2E8.0, 2I2) or TT* (free- format input lines with 5 items on each). It is read in subroutine TIRTTL, called from the routine TIRSTI. The TINTE transient and code control commands can be stored in an input file with the mandatory extension .tn1, or the user can enter the control commands interactively from the console. The user can also optionally specify any text input of up to 80 characters after the TT* or TT% string, usually to describe the transient case. General comments can be entered by the user from the 41 to 132 columns of each input line. Note that any text or number in the first 40 columns will be interpreted by TINTE as code input, which might result in erroneous calculations. Input lines starting with one of the 5 characters #, *, !, c, or C in the first column and any text on the remainder of the record are considered as **comment lines** and can be interspersed with the control command records.

A description of all possible control commands used in TINTE is given below.

6.1 COMMAND FORMAT

TINTE uses a code command format consisting of 5 variables, separated by spaces between the numbers. The variables are (in sequence):

ZEITS ZEITZ WERTZ IORT IART

and are explained in Table 20 below.

Table 20: Transient Control Variables

No.	Variable	Description
1	ZEITS	ZEITS is a control parameter with various meanings, depending on its relation to the parameter TIMET (the time since the start of transient (seconds)). The options are: ZEITS > TIMET: The transient is calculated until TIMET equals ZEITS, at which time the given command will be executed. ZEITS = TIMET: The command is executed. ZEITS = 0: The command is executed immediately after being read. TIMET > ZEITS > 0: The command is ignored. ZEITS < 0: The command is executed immediately.
2	ZEITZ	Defines the end-time of the command, measured in seconds from the beginning of the transient. ZEITZ may also be a control flag in some cases. The options are: ZEITZ < ZEITS: Causes a prompt jump in value of ramp variable. It should be used only for equilibrium calculations. ZEITZ = -1: This special command will keep the value of a variable from this point forward constant.
3	WERTZ	Defines the final value of the ramp variable which increases linearly from the initial value over the time interval (ZEITZ - ZEITS).

No.	Variable	Description
4	IORT	IORT <> 0: Defines the material identification number (as identified in the TZ* block of the .tn3 file) where the ramp variable is being changed. IORT = 0: Used for global ramps only, e.g. changes in K_{EFF} , reactor power or control rod position.
5	IART	Defines the type of ramp or control command and is explained in the section below.

6.2 NUCLEAR AND THERMAL RAMPS

Ramps in the values of nuclear or thermal parameters can be applied in TINTE. The ramp control commands specify **linear** changes in a chosen variable over the time period given by (ZEITZ – ZEITS). The ramp is completed when the variable reaches the value specified by the parameter WERTZ. However, a new ramp on the same variable is allowed to start, even before the first ramp has finished. As a result, a **polygonal** rather than linear time dependence for the variable will be established. Setting ZEITZ = -1 keeps the parameter constant.

6.2.1 Nuclear Ramps

6.2.1.1 Nuclear Ramps on Fractions of Materials with Overlaying Cross-Section Sets

The control commands used in nuclear ramps describe “movement” of materials, e.g. control rods, by superposition of sets of nuclear cross-section data in the respective nuclear meshes. In this case, the five items of the control commands are described in Table 21.

Note: With the implementation of the Linear Rod Motion model ROMO (see section 8.2.1.2 and Appendix E), these individual material ramps need no longer be used for control rod movements. Rather, the newer nuclear ROMO ramp 4 (see below) can be used to model such movements much simpler as well as physically more precise, since the approximate rod insertion position, and not the material concentration, is varied directly.

Table 21: Nuclear Ramp Control Variables

No.	Variable	Description
1	ZEITS	Time at which the nuclear ramp starts (seconds).
2	ZEITZ	Defines the end time of the ramp.
3	WERTZ	Final value of the overlaying cross-section concentration at the ramp’s end.
4	IORT	A negative number that is the identification number of the material with the overlaying cross-sections sets.
5	IART	A positive number that is the overlay position number in the .tn3 file of material which cross-section set will be mixed with the overlay material.

EXAMPLE

A possible TINTE definition of an aggregate control rod region in the .tn3 file is the following: Assume a definition of material – 4 as a material with basic cross-section set defined by nuclear material 190, and three overlaying cross-sections, defined by materials 187, 188, and 189. The input line in the NZ* block of the .tn3 file will look as follows:

-4 190 0.2 187 0.3 188 0.5 189

These individual entries are explained in Table 22.

Table 22: Nuclear Ramp Control Example

Entry	Description
- 4	Material identification number – should be included in TINTE input block NZ in the .tn3 file.
190	Identification number of the material with the basic cross-sections set. Assume this material to be graphite with certain density. Material 190 cross-section set is described in TINTE input block NQ in the .tn4 file.
0.2	Fraction of cross-section set of material 187 equal to 20%.
187	Identification number of the first material (i.e. overlay position number 1) with set of overlaying cross-sections. Assume this to be graphite with the same density as in material 190 mixed with absorber with nuclei density N_1.
0.3	Fraction of cross-section set of material 188.
188	Identification number of the second material (i.e. overlay position number 2) with set of overlaying cross-sections. Assume this to be graphite with the same density as in material 190 mixed with absorber with nuclei density N_2.
0.5	Fraction of cross-section set of material 189
189	Identification number of the third material (i.e. overlay position number 3) with set of overlaying cross-sections. Assume this to be graphite with the same density as in material 190 mixed with absorber with nuclei density N_3.

A possible TINTE nuclear ramp could then be:

0. 100. 0 –4 1

which will result in reducing the concentration fraction of material 187 (overlay position number 1) cross-sections from 0.2 to 0.0 (i.e. from 20% to 0%), and increasing the fraction of material 190 cross-sections from 0.0 to 0.2, over a period of 100 seconds. The fractions of the cross-sections of materials 188 and 189 remain unchanged. (Note that the total of all the defined fractions must always add up to 1.0. Since the sum of the fractions of the overlaying materials 187, 188, and 189 add up to 1, this means that prior to the ramp the fraction of the cross-sections set of the basic material 190 was zero).

6.2.1.2 Global Nuclear Ramps and Control Rod Movements

In addition to simulation of Control Rods movements by individual material overlay ramps, TINTE provides a way to change the reactor power by executing ramps on parameters like the effective multiplication factor, K_{EFF}, the equilibrium fission power, or the desired fission power. The ROMO feature provides two further “global” ramps: one to simulate an uniform rod motion as a whole and one to conduct a completely automated permanent reactivity control by continuous rod movements in either direction. All these global ramps have a common structure, as indicated in Table 23.

Table 23: Global Nuclear Ramp Control Variables

No.	Variable	Description
1	ZEITS	Time at which the nuclear ramp starts (seconds).
2	ZEITZ	Defines the end time of the ramp.
3	WERTZ	Final value of a factor or a parameter.
4	IORT	IORT = 0 indicates a global ramp.
5	IART	The options are: IART = 1: Indicates a multiplicative K_{EFF} ramp. IART = 2: Indicates a multiplicative ramp on the value of the equilibrium reactor fission power. IART = 3: Indicates a ramp on the final end value of the equilibrium reactor fission power (in MW). IART = 4: indicates a rod bank movement ramp (with linear motion in time) on the value of the rod insertion position (in cm), comprising any range of rod regions which have been pre-defined by an option "-11" input set (see section 8.3: command ROMO) IART = 5: indicates a reactivity control ramp or permanent criticality control on the value of the external reactivity demand (in %) needed to execute a given power ramp or keep constant a fixed power level (see previous ramp types: IART = 2 or 3). The reactivity demand ramp ends at time ZEITZ, but the rod control still continues then until either the equilibrium power option is switched off, or a rod motion ramp with IART = 4 is started.

Note:

1. The options IART = 2 and IART = 3 should not be used simultaneously with fluid ramps (i.e. when IART = 6 or IART = 9), since the feedback on the fission power from changes of the coolant flow or temperature will be in conflict with direct control over the fission power by a different ramp.

2. If a global parameter is ramped to new value, this value is kept for the duration of the transient, until it is changed again by a new ramp.

3. The ramps for K_{EFF} and reactor power cannot be used simultaneously. Starting one ramp type will automatically terminate the other one if present.

4. The same holds for the two rod ramp types 4 and 5.

5. Before a ramp type 5 can be issued, a ramp type 2 or 3 must first have been started.

6. Detailed information and examples how to apply the ROMO model and use the rod ramps of type 4 and 5 is given in Appendix E.

EXAMPLES

```
0.  100.  1.006  0  1
```

For the time period from T_1 = 0.0 sec until T_2 = 100 sec, the equilibrium neutron production term is multiplied by a factor of 1.006, i.e. the equilibrium k-eff value is increased by a factor 1.006 (or 0.6%).

0. 100. 1.10 0 2

For the time period from time T_1 = 0.0 sec until time T_2 = 100 sec, the equilibrium reactor fission power will be increased by a factor 1.10, i.e. from its steady-state value (100%) to 110%.

0. 100. 320. 0 3

For the time period from time T_1 = 0.0 sec until time T_2 = 100 sec, the reactor fission power will be decreased from its steady-state value (e.g. 400 [MW]) to 320. [MW].

Note: For examples of the control rod ramps 4 and 5, refer to Appendix E.

6.2.2 Ramps for Solid Materials Parameters

In some TINTE materials descriptions, the temperature may be given as an input parameter. In this case, the temperature can be changed by using a ramp to simulate a change in the heat transfer boundary condition. Some TINTE materials can also be defined to have an external heat source. The value of the heat source can also be modified by using a ramp to simulate a change in the heat production rate in the component. In tube components (i.e. with MAT = 13), the ZETA variable can also be changed by a ramp to allow a control over the fluid flow rate inside such component.

The opening pressure for a burst disc or a safety valve (MAT = 11) can also be modified by using a ramp. The TINTE parameters involved in the ramps on solid materials are indicated in Table 24.

Table 24: Solid Material Ramp Control Variables

No.	Variable	Description
1	ZEITS	Time at which the ramp starts (seconds).
2	ZEITZ	Defines the end time of the ramp, except for the case when IART = -4, when ZEITZ is not the time, but a parameter.
3	WERTZ	The value reached at the end of the ramp (temperature, external power (MW), final ZETA or final overpressure).
4	IORT	Material identification number of the component that is undergoing the ramp.
5	IART	The options are: IART = -1: Indicates a temperature ramp. IART = -2: Indicates a heat source ramp. IART = -3: Indicates a ZETA ramp in a tubular component (MAT=13). IART = -4: Indicates a Pressure jump for burst disc or safety valve (MAT=11). The options here are: If ZEITZ= -1: Then WERTZ is the pressure difference between the nodes of the component at which opening starts. If ZEITZ= -2: Then WERTZ is the pressure difference between the nodes of the component at which full opening is achieved.

EXAMPLE

Temperature Ramp

Assume a material definition with temperature given as an external parameter:

BORDER 99 0 -1050. 0 0 0 0. 0 0 0 0 1.e8

This TINTE material definition represents a boundary condition with constant temperature 50°C. The TINTE ramp:

0. 100. 85. 99 -1

will result in increasing the temperature in component 99 from its current value (50°C) to 85°C.

EXAMPLE

Heat Source Ramp

Assume a definition of a material with an external heat source:

PDCH.LU 50 1 645. 7 0.61 55 1.0 0 0 0.7 1.8e-2

640. 0.61 6.0

Material 50 has an external heat source of 1.8E-02 MW. The TINTE ramp:

0. 100. 1.0e-2 50 -2

will result in changing the heat source in material 50 from its current value (1.8e-2 MW) to 1.0E-02 MW over the time period from the current transient time (indicated by ZEITS = 0.0 seconds) until the transient time of 100 seconds.

EXAMPLE

ZETA Ramp

Assume a definition of a tube:

49 -57 -42 36 80. 1800. -1800. 1.13E5 5.65E5

TUBE.UP 49 13 230. 11 .2 11 0.2 0 0.

180. 0.2 5. 0.0001

Component 49 is defined as a tube (MAT = 13). It is connected to a node from the 2-D grid (-42) and to a node (-57) from an external 1-D flow network. The ZETA parameter of component 49 is ZETA = 0.0001. The TINTE ramp:

0. 100. 0.001 49 -3

will result in increasing the ZETA parameter of material 49 from its current value (i.e. 0.0001) to 0.001 over the time period from 0.0 seconds until 100 seconds.

EXAMPLE

Opening a safety valve by using a ramp

Assume that component 71 is defined as a safety valve (MAT = 11) and is connected to a material –42 from the 2-D grid and to a node 99 from an external network:

71 -42 99 0 10. 100. –100. 0. 0.

The TINTE ramp:

10. –1 80. 71 -4

will result in safety valve 71 beginning to open when the pressure difference between the 2-D component (-42) and the 1-D network (99) exceeds 80 bar.

The TINTE ramp:

10. –2 90. 71 -4

will result in safety valve 71 being fully open when the pressure difference between the 2-D component (-42) and the 1-D network (99) reaches 90 bar.

6.2.3 Ramps for Gas Flow Parameters

There are many ways in which the user can influence the TINTE gas flow calculation with respect to the gas pressure, gas inlet temperature in source regions, gas mass flow rate, and pressure drop across components. The TINTE parameters involved in the ramps on gas materials are indicated in Table 25.

Table 25: Gas-Flow Ramp Control Variables

No.	Variable	Description
1	ZEITS	Time at which the ramp starts (seconds).
2	ZEITZ	Defines the end time of the ramp, except for the case when IART = 5,8,9,12 when ZEITZ is not the time, but a parameter.
3	WERTZ	The final value reached at the end of the ramp, or a control parameter.
4	IORT	Material identification number that has a gas flowing through it. Note that for materials with ID numbers < 0, the absolute value should be given.
5	IART	This control item characterizes the ramp type. See below the table for the possible options.

For IART, the input options are:

- IART = 1: Specification of the gas inlet temperature [deg C] in gas source regions. In gas sinks these ramps are ignored. Nevertheless, it makes sense to specify the temperature in sink regions, especially if flow reversal cannot be ruled out.
- IART = 2: Specification of the pressure difference against the reference pressure [mbar]. The reference pressure, however, must have been defined in another material region.
- IART = 3: Specification of the mass flow source [kg/sec].
- IART = 4: Specification of the volumetric flow source relative to the steady-state value [fraction relative to the equilibrium source flow rate].
- IART = 5: Specification of the pressure [bar]. The reference pressure can be redefined here and moved to another material region. In such a case, previous specifications of pressure differences will have different meanings. They should have been switched by ramps on the total pressure. For a ramp defined with ZEITZ = -1, WERTZ = -1 and IART = 5 for an arbitrary node, its pressure will be defined as the new reference pressure. As for all pressure ramps, the reference pressure can be changed later by using a ramp in which WERTZ is given as the new value of the reference pressure. Instead of specifying the reference pressure, it can be calculated by TINTE from the coolant inventory and the coolant temperature. This TINTE feature is activated by using ZEITZ = -2; WERTZ = 0; and IART = 5. In this ramp an arbitrary node can be specified in IORT in which the pressure will be treated as the reference one.

  Note: Only this type of pressure control should be used to allow modelling of burst discs and opening and closing of safety valves in a transient.

Prior to using the following options (IART > 5) for IORT, a partner node should be defined using IART = 12. A partner node is a node for which differences of certain variables are

defined. If no partner node is defined, then TINTE assumes as a partner node the one in which the reference pressure is defined (which usually is the main reactor inlet or outlet node).

- IART = 6: Specification of the relative power removed by convection. Used to adjust the mass flow between the node indicated in IORT and the partner node. The temperature “gas-outlet – gas-inlet” difference is determined by TINTE from the reactor power and is kept constant.
- IART = 7: Specification of the relative power removed by convection. Used to adjust the gas inlet temperature (Tin), while keeping the mass flow between the node indicated in IORT and the partner node constant to achieve constant power.
- IART = 8: Specification of the relative power removed by convection. Used to adjust the gas inlet temperature, while keeping the volumetric flow between the node indicated in IORT and the partner node constant. The inlet gas temperature in this case can only be adjusted within the limits [250 °C to 330 °C]. Should the inlet gas temperature reduce below the lower limit, its value is made equal to 250 °C, and TINTE then adjust the gas mass flow for constant power. Should the inlet gas temperature exceed the upper limit, its value is made equal to +330 °C, and TINTE then adjusts the gas mass flow for constant power. The upper and lower limits can also be adjusted: The new desired value can be set on WERTZ , using ZEITZ = -1 (minimum gas-inlet temperature), -2 (the maximum gas-inlet temperature) and -3 (the minimum gas-exit temperature).
- IART = 9: Specification of the gas inlet temperature with respect to the current gas exit temperature in the partner node in according with the following relation:

  $T(IORT) = \alpha * T(MPART) + \delta T$

  where

  T(IORT): Gas temperature in the material node specified in IORT.

  T(MPART): Gas temperature in the partner node α (it has a predefined value of 1).

  If ZEITZ = -2: α is equal to the value given by WERTZ.

  If ZEITZ = -1: δT is adjusted to the current status without changing the value of α.

  If ZEITZ > 0: δT is specified in a ramp.
- IART = 10: Specification of the blower power, relative to its equilibrium value. It is assumed that the blower is located between the component specified by IORT and the partner node. Note that IORT may not be the reference pressure node.
- IART = 11: Specification of a limitation in the blower power, relative to its equilibrium value. It is assumed that the blower is located between the component specified by IORT and the partner node. Note that IORT may not be the reference pressure node.
- IART = 12: Definition of a partner node to be used in ramp options IART = 6 to IART = 11. The idea is, if IORT is specified a source node, then the coolant is taken out of the partner node after its temperature has been changed. This allows simulation of gas heat-up inside a gas blower (see above option IART = 9), expansion in a gas turbine or compression in a compressor.

  The definition of a partner node is reflexive – if the gas flow direction is reversed, the relation between the partner node and the one specified in IORT is preserved. The partner node relation plays a special role in the pressure calculation from the gas

inventory balance. The way a partner node relation works is that a loss of gas in one of the partnered nodes results in a gas gain in the other. Therefore, to calculate the gas inventory correctly, the user has to remember that gas mass can be gained, or lost, only in nodes that have *not* been partnered. The identification number of the partner is specified in the ramp definition by WERTZ.

Cancelling an existing partnership is done by a ramp with WERTZ=0. If MGAS > 1 and WERTZ = 0 (i.e. cancelling a partnership or defining a new gas composition), it is necessary to read an additional card. From this card the MGAS numbers are read (format-free) and interpreted as mole fractions for the corresponding gas types. Normalisation is performed internally.

Establishing a partnership between nodes with materials defined with a mixture of gases (i.e. MGAS>1), will result in transferring between them not only the gas mass, but also the exact *gas mixture* composition. Should the partnered nodes be a gas sink and a gas source, it may result in some cases in the transfer of large masses of gas instantaneously. For a more realistic transfer time, use ZEITZ to specify the time in which the transfer takes place.

(Comment: the delay time option for gas mixtures is not yet tested completely. Thus it is better to simulate the external volume by additional meshes to account for the temperature changes as well).

Options 6 to 11 have been introduced to simulate some special reactor operating situations. Option 9 can be used with α = 1 to simulate a heat exchanger, while options 6 to 8 allow for models without an external heat sink, where the heat produced is transported by convection to the gas sink region.

User Notes:

1. Modelling Experience for loops with steam generator models:

If the mass flow must be stopped and the pressure simultaneously be reduced significantly, start the mass flow ramp first and then (1s later) the pressure ramp. The following more complex techniques can also be used:

a) Switch to a relative volume flow and then reduce volume flow and pressure or

b) Define a partner node and reduce the reference pressure and pressure difference to the partner. This is the best option for modelling natural convection effects.

2. The partnership and the coupling of the partners (tested up to now is only IART=9) is written to the restart file. So the information must not be repeated in the .tn1 file in case of a later restart. However if in a restart case the .tn3 file is read again additionally to the restart file, the partnership and the IART=9 coupling is lost. The gas temperature in the source region (assumed to be one of the partners) is then set to the figure given in the .tn3 file.

3. In cases with more than one gas loop (e.g. PBMR with core barrel cooling) and a partner definition for the main loop for calculations involving inventory control, a partnership has to be established in the second loop as well. Only then the mass flow in the second loop is preserved if gas mixing changes occur in the main loop. This principle must be applied for example where water ingress in the main loop occurs, to preserve the mass flow in the core barrel cooling loop.

EXAMPLE

Assume materials –56 and –57 have been defined as a gas source and a gas sink in a gas circulation system for which no heat is transferred out and a steady-state gas temperature is established. In this case, the gas temperature in the sink (-57) will be equal to gas temperature in the source (-56). The TINTE ramp

0 -1 56 57 12

will establish a partnership between both nodes. The TINTE ramp

0 -1 0 56 9

will specify the gas inlet temperature in node 56 with respect to the current gas exit temperature in the partner node 57 according to the relation:

$$T(IORT) = \alpha * T(MPART) + \delta T$$

where

T(IORT): Gas temperature in material 56,

T(MPART): Gas temperature in the partner node 57, and

α : Has a value of 1

In this case, since ZEITZ= -1, then the parameter δT =1, since α =1. As a result the gas inlet temperature in node 56 is equal to the gas outlet temperature in node 57.

6.3 THE EXECUTION OF TINTE CONTROL COMMANDS

These commands are given by specifying negative numbers. If ZEITS <0, then ZEITS is the control command and it is executed each time it is read by TINTE. If ZEITZ <0, then ZEITS specifies the time when the command, specified by ZEITZ, is executed. WERTZ is used only if ZEITZ has a value in the range from -5 to -12, as well as IORT = 0 and IART = 0. The first group of available Control Commands Options are indicated in Table 26.

Table 26: TINTE General Control Commands: Group 1

Command	Description
-1	Start an equilibrium calculation. After that is finished, the TINTE output is governed by command -2. This option cannot be used within a transient calculation (otherwise, the reactor time will be reset to zero and a new transient calculation will be initiated after this equilibrium calculation).
-2	TINTE writes an overwriting or documenting restart file (depending on what choice the user selected), and the 2-D arrays in the the . .p2d and in the printer output files.
-2.1	TINTE writes only a restart file.
-2.2	TINTE writes only the 2-d arrays in the .p2d and in the print output files, but does not write a restart file.
-2.3	TINTE writes a single 2-d array (the solids' temperatures) in the .p2d and in the output files. It does not write a restart file.
-2.4	TINTE writes a restart file and the 2-d arrays in the .p2d file. It does not write the 2-D arrays in the output file.
-3	TINTE writes a restart file and the 2-d arrays in the .p2d and in the print output files, prints transient history tables (if any transient part exists) and exits, i.e. ends the calculation.

Command	Description
-3.1	TINTE writes only a restart file and the history tables and exits, i.e. ends the calculation.
-3.2	TINTE writes only the 2-d arrays in the .p2d and in the print output files and prints the history tables; it does not write a restart file; and it exits, i.e. ends the calculation.
-3.3	TINTE writes a single 2-d array (the solids' temperatures) in the .p2d and in the print output files and prints the history tables. It does not write a restart file; and it exits, i.e. ends the calculation.
-4	TINTE exits, i.e. ends the calculation and does not write any output.
HE	If this command entered from the console in the directory, where TINTE was started, TINTE immediately prepares a restart file for the last time step that is calculated for the transient and stops the run. It is possible to restart the run by using this file from the time step that is calculated last.

EXAMPLE

The following text can occur in the .tn1 file or may be entered from the console:

```
600.    -2          Time 600 sec
3600.   -2           time 1 h
10800. -2            time 3 h
14400. -2            time 4 h
18000. -2            time 5 h
```

The option selected (-2) results in the output data being written in a restart file, as well as the 2-D arrays in the .p2d and in the output files, at the indicated times (600 seconds, 1 hour, 3 hours, etc.). Note that the comments only start after column 40 in the .tn1 file.

The second group of available Control Commands Options are indicated in Table 27 below. They are mostly used to speed up the calculation time during a transient, or to limit the time step widths during especially fast transients (e.g. rod ejection).

Table 27: TINTE General Control Commands: Group 2

Command	Description of the meaning of item WERTZ for that command
-5	New upper limit for the temperature calculation time step. Default value is 60 seconds. Should the fission power be influenced by the temperature, this parameter should not exceed 300 seconds. In cases when the fission power is no longer dependent on the temperature (i.e. shutdown reactor), this parameter can be as large as 20 minutes.
-6	New lower limit for the temperature calculation time step. Default value 1.0 sec.
-7	New upper limit for the nuclear calculation time step. Default value is 60 sec.
-8	New lower limit for the nuclear calculation time step. Default value is 0.1 sec.
-9	New lower limit for the Xenon-135 re-calculation time span. Default value is 0.0 sec. (This command should be used only for testing purposes).
-11	This command is used to enter special control options.

EXAMPLE

The following text can occur in the .tn1 file:

```
0.
0  -7    5                              max. Schrittweite, Nuclear
0  -5   10                              max. Schrittweite, Temperatur
600.    -2                              Time 600 sec
0  -7   60                              max. Schrittweite, Nuclear
0  -5   60                              max. Schrittweite, Tempera
3600.   -2                              time 1 h
```

The example above results in a maximum nuclear time step width of 5 seconds, and a maximum temperature time step width of 10 seconds, at current transient time (t = “0” seconds). This time step specifications continue until t = 600 seconds, when output is written (see example above), and the time step widths are changed to 60 seconds each. This value will then be applied for the duration of the transient. Note that since this is only a maximum time step width indication, TINTE will automatically optimise the real time step width at every iteration point to suit the convergence criteria imposed on the numerical routines.

Time-step control as recommended by FZJ:

As TINTE calculates average values over the time step and not point values at the end of the step, it is not necessary to choose extremely small time steps for slow transients - and it is not recommended. With each time step calculation a certain small error results from numeric uncertainty and other reasons. This error depends slightly on the time step length, of course, but it does not go to zero by decreasing the step length. Therefore, as an example, if the user go down from the recommended value of 1 minute to 0.01 seconds, an additional 6 000 solutions are calculated with each one containing a small error and which may sum up to distort the transient. In FZJ’s experience there are only a few cases where time steps below one second are good.

There are situations where time steps have to be reduced, but the automatic step length adjustment of TINTE does this. Of course this is not always done in the best way. If the step length is oscillating between large and very small steps, it makes sense to go down with the step length to an average value. But in the vast majority of cases the very specific time step control of the code outperforms any user attempts to direct it.

There is one situation, however, where small time steps must be specified by the user: at the very beginning of a fast change, especially in nuclear quantities, which is started by a ramp input during the transient. In this case it is sufficient to force the first time step of the change to be adequate for its time scale (e.g.: by an input line containing only a ZEITS value). That may be for instance a 1 second step at a control rod ramp or even a 0.01 second step in the case of a very fast rod ejection from full power. TINTE will therefore produce the best results when the time step limitation is used only if necessary to describe fast changes, or when the step length is oscillating, otherwise (due to delayed neutrons) 60 seconds represents an optimal step size.

An example of this method is shown below, where the 10.1 second time specification is used to “warn” Tinte of the large change that is coming with the -11 NKNV switch. The time step after the -11 NKNV switch is also given at 10.2 seconds, to “force” a time step of 0.1 seconds before and after the large change.

EXAMPLE

The following text can occur in the .tn1 file:

```
-1


10.0
10.1


-11
NKNV


10.2
15.0
```

The control command -11 is used for the input of special control options in TINTE. The special controls may be divided into two groups: the first group commands change parameters for the duration of the transient unless these parameters' changes are reset again, while the commands of the second group modify some special coefficients. These commands are read from a separate new card in FORTRAN format A4.

The next field is read in format A1 and its value indicates whether the corresponding parameters are to be set (A1='blank') or are to be reset (A1<>'blank') (see the Example below Table 28). Should additional parameters are to be used, these are read from the same card starting from column No.6. Several special options can be specified one after the other, and a blank line must always be used to end the action(s) of command –11.

The Group #1 commands, given after entering control line –11, are described in Table 28 below.

Table 28: TINTE Control Commands for the -11 Option: Group 1

Command	Description
GAVO	This option may be used to change the gas coolant mixture, if MGAS>1. From the next card, non-negative numbers are read and are interpreted as new gas molar fractions. From the following cards material numbers (one per card) are read. All meshes of these materials will have the new gas mixture. A blank line terminates the reading. This is a "at a time" action, i.e. it occurs immediately after the reading of the input. The meshes are filled with the new gas mixture, and in a following transient this gas is as usual transported through the system. Even if the new gas mixture was assigned to the source, the source gas mixture will return to the original during subsequent time steps (i.e. the gas is introduced, and transported out of the system). In the 2-D output fields the gas molar fraction changes will only be seen if the gas was already present prior to the GAVO command.

Command	Description
LPOW	The nuclear heat source is assumed to be a local heat source in the pebbles. The non-local part is set to zero. This command may have up to four options: 1) No parameters entered: All non-local heat sources are assumed to be local heat sources. 2) Parameter No.1 <>0 – neutron moderation heat sources remain as non-local heat sources. 3) Parameter No.2 <> 0 – heat sources from (n,γ) and (n, α) reactions remain as non-local heat sources. 4) Parameter No.3 <>0 - γ transport stays non-local.
NBUK	In the process of evaluating the nuclear cross-sections, their dependence on the buckling is not completely taken into account and the calculated buckling is equal to a reference value with uncertainty DBUK. The quantity DBUK is read as a parameter that can be used to resolve convergence problems. In many cases, setting DBUK = 1.0E-4 can help achieving convergence.
NKNV	When using this option, all calculations involving gas, i.e. gas flow, heat transfer by convection, etc. are omitted. Therefore, any interaction between the gas and the solids is ignored. This option is particularly useful when simulating in a conservative way an accident as “Depressurized Loss Of Forced Cooling.”
NRFT	Temperature calculation is performed only in the core. In all other materials, e.g. reflector, reactor internals, etc. the temperatures are kept equal to the values assigned to them as initial values.
NXBE	In an equilibrium nuclear calculation, the adjustment of the concentrations of Iodine-135, Xenon-135 is skipped. Should transients be initiated, however, the calculation of their concentrations is resumed immediately or after a certain time interval (see above ZEITZ = -9).
ROMO	This option is used to define or re-define the parameters of the “linear rod motion model” the user wants to apply in executing the control rod ramps (global nuclear ramps of type 4 and 5; see section 8.2.1.2). These data comprise mainly the enumeration and grouping of the pertaining overlay regions, their rod length values and their (exponential) interpolation parameters for the absorber concentrations. The details of this input which extends over three additional records are explicitly described in Appendix E.
STXT	After a specified change of reactor parameters, it normally lasts for about 2 days or more until a new equilibrium status has been reached, even if the change itself has been finished in minutes. The reasons are the xenon dynamics caused by power level changes and the delayed temperature adjustment due to the large heat capacity in the pebble bed core. This command allows the calculation of the new equilibrium in a short time. The transient xenon dynamics and the transient solid temperature calculations are skipped. Instead the new equilibrium Xenon-concentration is calculated and the heat capacity is set to zero in the temperature calculations, as well as the delayed heat source calculation at the fuel element surface. The interim values in this transient of course do not have any physical meaning. During this procedure the multiplication factor of the nuclear calculation remains fixed, while the other figures, e.g. the reactor power are adjusted to the new situation. The difference against an equilibrium calculation for the new situation is, that therefore a power must be given and the multiplication factor is calculated as opposed to the situation here. Fast changes in reactor parameters sometimes lead to strong changes in Xenon- concentrations and /or strong changes in the reactor temperatures, which may inhibit the convergence of the code. Therefore a retardation constant (l/s) may be used as given in XTZK, which is read as a parameter to this option. Comment: This option should be used with care, unless the differences to a normal transient have been understood completely!

EXAMPLE

The following text can occur in the .tn1 file:

```
-1
0 -2

-11
NKNV

-11
LPOW  0  0  1

1.                                  Scram by CRs insertion over 1 sec
1.  2  1.0  -6  1                   Mat#6  First overlay = 100%
1.  2  1.0  -7  1                   Mat#7  First overlay = 100%
1.  2  1.0  -8  1                   Mat#8  First overlay = 100%
1.  2  1.0  -9  1                   Mat#9  First overlay = 100%
```

The example above results in the following actions:

- An equilibrium calculation is performed.
- Output from the equilibrium calculation is written as specified at t = 0 seconds.
- The special command NKNV is used to stop all gas interactions.
- The special command LPOW is used to define that only γ transport stays non-local.

The above commands belong, as it has been mentioned already, to Group#1 of special commands and present program control options.

There exists a second group of special commands which serve to execute special program actions. These Group #2 commands, given after entering control word –11, are described in Table 29. This set of commands is ended by entering a blank card, too.

Table 29: TINTE Control Commands for the -11 Option: Group 2

Command	Description
STAB	After a change in a material mixture has been defined, e.g. simulation of a control rod movement, the effective multiplication factor is recalculated without recalculating the new xenon-135 concentration and temperatures. By ignoring those feedbacks, the reactivity effect from moving control rods is calculated in a more conservative way. To avoid convergence problems, the control rods movement should be better split into several steps and the calculated new fluxes are maintained. A reset to the old situation (e.g. by STAB0) is not useful: the old flux distribution has to be recalculated completely or read again from a restart file. This option may also be used to adjust a concentration of a neutron poison to a given control rod reactivity. In this case it must be assumed, that the ratio of the fast and thermal absorption remains constant and that there is no temperature dependence of this cross-section.

Command	Description
TEKO	This option may be used to determine local temperatures and concentration coefficients to use in point kinetics computer programs. Under command TEKO all temperature and absorber calculations are skipped. Instead of them, arbitrary changes in temperatures and/or absorber concentrations are read from cards that follow command TEKO. The entries on these cards are: J1 – radial mesh of the nuclear mesh grid where the changed data input starts J2 – the radial mesh of the nuclear mesh grid where the changed data input ends. I1 – the axial mesh of the nuclear mesh grid where the changed data input starts I1 – the axial mesh of the nuclear mesh grid where the changed data input ends. DTF – the change in the fuel temperature [deg C] DTM – the change in the moderator temperature [deg C] Should hydrogen and nitrogen be specified (i.e. MGAS > 1) and appropriate nuclear cross-sections expansion coefficients are available, then the following quantities can be calculated: DH2 – the change in partial pressure [bar] of hydrogen gas at the current gas temperature DN2 – the change in partial pressure [bar] of nitrogen gas at the current gas temperature DGA – the change in graphite burn-up [%]. More than one of the above cards may be entered. If commands refer to the same meshes, the effects will be compounded. The input is ended by entering a blank card. Using the above modified data the effective multiplication factor is recalculated while keeping the Xenon-135 concentration constant. Therefore, the change in the multiplication factor is due to change in the local temperature or concentration only. After each coefficient calculation the fluxes are reset to the old values. However, the temperatures and the concentrations are not reset and must be manually reset before new coefficients are calculated. If a subsequent equilibrium or transient calculation is performed, this is not necessary, because these events start with a re-calculation of these items.
XEEF	This option is used to calculate the reactivity effect of Xenon. The multiplication factor is re-calculated for a changed xenon absorption cross-section. All temperatures remain at their current values. The factor XEEF is multiplied with the xenon absorption cross-section. It has been experienced that great changes could result in convergence problems. A reduction to e.g. vanishing xenon should therefore be performed only in several steps. The newly calculated fluxes are preserved; thus the old situation has to re-calculated or re-read from a restart file.

Each of these commands from the second group initializes a direct action as described above, or is used to reset the changed parameters. In this case TINTE either stops, or, in the case of command TEKO, other commands may be entered and TINTE can continue the calculation. If other commands than these here described (for option -11) are given, they will be ignored.

7. TINTE OUTPUT DESCRIPTION

The TINTE print output file generally contains the following main data blocks, depending on the amount of output data requested by the user (see Section 6.3).

The additional TINTE console output is very similar, but does not contain the larger fields like the 2-d arrays, region heat balance and final transient history tables.

Output Block #1: Print of TINTE-User dialog**.**

Output Block #2: Print of the Input Blocks ST, GM, TZ, TM, and NZ, as contained in the .tn3 file.

Output Block #3:

This block starts with the following TINTE message:

> *NUKLEARE QUERSCHNITTE (Nuclear Cross-sections)*
>
> *Es wurden Angaben zu einer Mischung aus 2 BE Typen gefunden.*
>
> *(Input for 2 fuel element types were found)*
>
> *Temperatur-Rechnung. (Temperature calculation)*
>
> *Das Volumen Der Gebiete Mit Heterogener Rechnung Betraegt 8.52720e+07 Cm**3, Das Der Einzelnen Regionen. (The volume of the model area that requires heterogeneous temperature calculations is:)*

This means that TINTE has determined that the total volume of materials with heterogeneous temperature calculation is:

$$\mathbf{V_{TOTAL\text{-}TINTE}} = 8.52720\text{E+}07\ \text{cm}^3.$$

Since the heterogeneous temperature calculation has been specified only for materials in the core fuel region, the above figure is the total core volume. One should note here that this figure is usually slightly greater than the actual core volume, since TINTE ignores the graphite fraction in the conus region in the bottom of the core. The error as a result of this apprimation is acceptably small. The above TINTE messages are followed by a listing of the volumes of all the materials declared in input block TZ. Output Block #3 ends with TINTE message:

> *Die Nachwaermeberechnung erfolgt mit LIFE-Daten aus VSOP (the decay heat calcualtions using LIFE data from VSOP follows)*
>
> *Daten fuer BE-Mischung eingelesen (Data for fuel elements read)*
>
> *BENOETIGTER SPEICHERPLATZ (*8) 1703982 (required memory size)*
>
> *VORHANDENER SPEICHERPLATZ (*8) 2000000 (available memory size)*

Output block #4

If a .tn3 file was read, this block starts with the TINTE message:

1 MASCHENBILD DES REAKTORS FUER T I N T E - RECHNUNG

0 TEMPERATUR - MATERIALBELEGUNG

The message is followed by a printout of the coarse and fine mesh grids, both axial and radial, that are used by TINTE for the temperature calculations.

Output Block #5

If the NZ block of the .tn3 file was read (nuclear regions definition), the next Output Block #5 is the coarse and fine mesh grids, both axial and radial, that are used by TINTE for the nuclear calculations.

It starts with the message:

1 MASCHENBILD DES REAKTORS FUER T I N T E - RECHNUNG

0 NUKLEARE MATERIALBELEGUNG

Output Block #6

Output block #6 starts with the TINTE message:

BITTE CPU- UND REAL-RECHENZEIT-SCHRANKEN (MIN), sowie ggf. Zeilenlaenge

EINGEBEN (0 <=> KEINE BESCHRAENKUNG (bzw. 132 C/Z))

Output block #6 contains the output from TINTE equilibrium calculation if it is executed. The output parameters of each line of the iteration monitor, sent both to the output file and to the console, are described in Table 30.

Table 30: TINTE Equilibrium Iteration Monitor Output Description

Parameter	Description
CPUS	CPU time [sec]
IFG	Number of iterations done by TINTE for solid-to-gas heat tansfer calculation.
ITF	Number of iterations done by TINTE for the solids temperature calculation.
VFT	Maximum change [deg C] of solids' temperature between two iteration steps.
IGS	Number of iterations done by TINTE for gas flow calculation.
VGT	Maximum change [deg C] of gas temperature between two iteration steps.
DP21	Pressure drop between second and first components of external flow network In this case: Pressure drop [mbar] between reactor outlet and reactor inlet.
P	Pressure [bar] in second component of external flow network. In this case: pressure at reactor outlet.
TK1	Gas temperature [deg C] in the first node of the 1-D flow network(in most cases the reactor outlet- see Table 7).
TK2	Gas temperature [deg C] in the second node of the 1-D flow network(in most cases the reactor inlet).

Parameter	Description
TK3	Gas temperature [deg C] in a material selected in the .tn3 file, e.g.: the CBCS inlet.
MP1	Gas mass flow rate [kg/sec] in material reactor inlet.
WQK	Power [%] transferred by convection from fuel to gas. Note that this figure is never 100%, since radiation and conduction heat transfer also plays a small role. A typical steady-state value is WQK = 99.55%, which means that convective heat transfer is responsible for transporting 398 MW of the 400 MW thermal power generated by the reactor.
BZTM	Maximum fuel sphere centre temperature [deg C].
ITN	Number of iterations done by TINTE for the nuclear calculation within one temperature iteration step.
VNL	Maximum change [%] of reactor fission power between two iteration steps.
WQNF	Reactor fission power [%]. The global fission power is calculated as the *equilibrium power which would be attained* for the current neutron flux. (During a transient, this field is a measure of the current fission rate in relation to the equilibrium fission rate – see Table 31). The difference between WQNF and WQNT (which occurs only during transients) should be noted by the user (see Table 31).
KEFF	Effective multiplication coefficient.

There are nine more columns to the right from the KEFF one, but only in the printer output file. The output information in these coluimns is used to indicate for which material number, and in what coarse and fine mesh number, the maximum changes occur in the solid temperatures, the gas temperature, and the nuclear fission power. TINTE Output Block #6 ends with the messages:

0STATIONAERE RECHNUNG KONVERGIERT

(The steady-state calculation converged).

0GENAUIGKEITSSCHRANKEN: EPANI EPSNO EPAFT EPABT EPAGT

2.E-05 1.E-04 2.E-01 2.E-01 2.E-01

(Various convergence criteria used are listed; cf. input block ST).

0T I N T E - RESTARTAUSGABE NACH STATIONAERER RECHNUNG

0RESTART-FILE 10 GESCHRIEBEN UND ABGESCHLOSSEN

(Restart file after equilibrium calculation written and closed).

TINTE Output Block #7

It contains the two-dimensional data arrays, that could occur in both the output and the .p2d binary files, if the user specified these options. The data arrays are as follows (in sequence as listed by TINTE):

- ***MOD. TEMP [K]*** Radial and axial mesh grid values of the average moderator temperature, used for nuclear feedback calculations [K]. This data is given in the nuclear material mesh grid, i.e. the grid defined in the NZ block of the .tn3 file. inside the core area, it is calculated as the average temperature of the fuel and fuel-free zones in a sphere (i.e. the average of all fine shells inside a sphere), over all fuel batches in a spatial mesh. Outside the core area, this field is equal to the “Feststoff” or “solid” temperature.
- ***FUEL TEMP [K]*** Radial and axial mesh grid values of the average fuel temperature, used for Doppler feedback calculations [K]. This data is given only in the core area,

i.e. areas that contain fuel. It is calcualted as the average temperature of the fuel zones in a sphere (i.e. the average of all fuel-containing fine shells inside a sphere), over all fuel batches in a spatial mesh. (The difference with respect to the Moderator temperature field above should be noted).

- ***RELATIVER GAS – DRUCK [mbar]*** Radial and axial mesh grid values of the gas pressure [mbar], relative to the reference pressure (which usually is defined at the reactor outlet).
- ***GAS – STROMDICHTE [KG/M**2*SEC]*** Radial and axial mesh grid values of the gas mass flowrate per unit area [kg/(m^2*s)].
- Picture of the radial and axial gas flowrate vectors (if selected as an option).
- ***RADIALER-GASSTROM [KG/SEC/MASCHE]*** Radial mesh grid values of the gas mass flowrate [kg/s] (if selected as an option).
- ***AXIALER-GASSTROM [KG/SEC/MASCHE]*** Axial mesh grid values of the gas mass flowrate [kg/s].
- ***RAD.-KONVEKTION [MW/MASCHE]*** Radial and axial mesh grid values of the convective heat transferred in the radial direction [MW/mesh].
- ***AX.-KONVEKTION [MW/MASCHE]*** Axial mesh grid values of the convective heat transferred in the axial direction [MW/mesh].
- ***RAD. GAS-RANDTEM. /C*** Radial mesh grid values of the gas temperature for gas flowing in the radial direction [deg C]. Note that these temperatures are indicated on the mesh boundaries as axially averaged temperatures.
- ***AXI. GAS-RANDTEM. /C*** Axial mesh grid values of the gas temperature for gas flowing in the axial direction [deg C]. Note that these tempeeratures are indicated on the mesh boundaries as radially averaged temperatures.
- ***GAS – TEMPERATUREN /C*** Radial and axial mesh averaged values of the gas temperature [deg C].
- ***WAERMEUEBERGANGSZAHL ALPHA*F/V*** Radial and axial mesh grid values of the convective heat transfer coefficient [W/cm^3*K].
- ***HE (N2, O2, CO, CO2, H2O, H2) - MOLENBRUCH IM GAS*** Mole fraction of the specific gas in the gas mixture.
- ***GAS-KONZENTRAZION*** Radial and axial mesh grid values of the gas concentration [mol/gas-m^3].
- ***GRAPHIT - KORROSION / (MOL/M**3)*** Amount of corroded graphite.
- ***FESTSTOFF-TEMPERATUREN*** Radial and axial mesh grid values of the solid materials temperatures [deg C]. These temperatures are mesh-averaged data, and is used to define the interface temperature with the solid-to-gas heat transfer routines. The field can be divided in 3 sections. In the case of the fuel in the core region, this field is euqal to the average fuel sphere surface temperature field for all fuel batches. Outside the core region, the Feststoff temperaure field is equal to the Moderator field, up to the boundary of the neutronic mesh. Outside the neutronic mesh, it is the only field that indicates solid temperatures for metal and other ex-core structures.

- ***Temperaturen am BE-Rand /C "1"*** Radial and axial mesh grid temperature values [deg C] of type 1 pebbles at the surface of the fuel bearing part of the fuel sphere (R = 2.5 cm)

- ***Temperaturen am BE-Rand /C "2"*** Radial and axial mesh grid temperature values [deg C] of type 2 pebbles at the surface of the fuel bearing part of the fuel sphere (R = 2.5 cm)

  Note: The number of output fields here depends on the number of fuel types indicated by the .tn4 file.

- ***Temperaturen am BE-Zentr. /C "1"*** Radial and axial mesh grid values of type 1 pebbles central temperatures [deg C]

- ***Temperaturen am BE- Zentr. /C "2"*** Radial and axial mesh grid values of type 2 pebbles central temperatures [deg C]

(Note that the average values of the fuel surface and center temperatures for all batches are *not* equal to the "Moderator" or "Fuel" fields for all batches, since the averages for the "Moderator" and "Fuel" fields are calculated from the fine shell temperatures inside the fuel spheres, which is not included in the current TINTE output).

- ***LEITFAEHIGKEIT RADIAL*** Radial mesh grid values of thermal conductivity for heat transfer in radial direction [W/cm*K]. Various correlations are used to calcuate these values, e.g. the Zehner-Schluender correlation for conductivity inside the pebble bed.

- ***LEITFAEHIGKEIT AXIAL*** Axial mesh grid values of thermal conductivity for heat transfer in axial direction [W/cm*K].

- ***KONVEKTIV ABGEFUEHRTE WAERME*** Radial and axial mesh grid values of the heat density transferred by convection [W/cm^3]. Note that this is a heat density value, and not a heat per mesh value. Data will only be given in the meshes that were defined as gas flow meshes.

- ***LOKALE NUKLEARE WAERMEQUELLE*** Radial and axial mesh grid values of the heat (or power) density that was produced in the fuel by nuclear fission [W/cm^3].

- ***alle nicht-lok. Waermequellen*** Radial and axial mesh grid values of all non-local heat density sources in [W/cm^3], produced from all processes other than fission nuclear reactions, e.g. neutron absorption, fast neutrons thermalization, gamma-quanta absorption. This data are given in the core region, as well as in the reflector regions.

- ***davon Gamma-Transport*** Radial and axial mesh grid values of non-local heat density sources [W/cm^3], produced from gamma-quanta absorption only.

- ***schneller Fluss*** Radial and axial mesh grid values of the fast neutron flux [$1/cm^2$].

  Note: The fast neutron flux in TINTE is neutrons with energies in the range 3.06 eV – 10 MeV.

- ***thermischer Fluss*** Radial and axial mesh grid values of thermal neutrons flux [$1/cm^2$].

  Note: The thermal neutron flux in TINTE is neutrons with energies less than 3.06 eV.

- ***Xe-Konzentration*** Radial and axial mesh grid values of Xe^{135} concentration [1/barn*cm].

- ***Jod-Konzentration*** Radial and axial mesh grid values of I^{135} concentration [1/barn*cm]

- ***VORGEGEBENE WAERMEQUELLE*** Radial and axial mesh grid values of additional (pre-defined) heat source densities [MW/cm^3]. Note that this is per definition non-nuclear sources, and it can be pre-defined for any solid material in the .tn3 file.

- ***KOMPONENTEN – NETZWERK*** External one-dimensional network description.

Output Block #8

TINTE Output Block #8 starts with the message :

MITTLERE TEMPERATUREN, WAERME-QUELLEN, -STROEME UND -BILANZEN

and contains detail information for each material defined in TINTE Input Block TM, such as:

- Solid material temperature [deg C]: Average value for all nodes with the given material number.
- Gas temperatures [deg C]: Average values for all nodes with the given material number.
- Local nuclear heat sources in the material [MW].
- Non-local nuclear heat sources in the material [MW].
- Heat transfer by convection [MW].
- Heat transfer by conduction and radiation [MW].
- Heat transfer between the material and other materials, adjacent to it [MW].

TINTE Output Block #9

It starts wth message :

INSTATIONAERE RECHNUNG

This output block contains the TINTE output for the non-steady state (or transient) calculation (i.e. all data generated after t = 0 seconds). The time step monitor lines output parameters description is almost identical to that of the steady-state iteration monitor lines output (see Table 30), except that some output fields are unique to the transient calculation. The output fields that occur only during the transient calculation are summarised in Table 31.

Table 31: TINTE Transient Time Step Monitor Lines Output Description

Parameter	Description
REAKT	Reactivity [mNile] needed to keep the reactor at the k-eff value as calculated during the steady-state run. This output field only occurs when a global reactor parameter ramp is performed.
OMNL	Inverse reactor period [1/s]. This field replaces the KEFF field during transients, and is an indication of the varaitions that occurs in the system's global power density rate. The global inverse reactor period ω at any location in the reactor is defined /51/ such that the power density within the reactor changes according to the function $\dot{Q}''' = \dot{Q}'''_0 e^{\omega t}$ $\frac{\partial \dot{Q}'''}{\partial t} = \omega \dot{Q}'''_0 e^{\omega t} = \omega \dot{Q}'''$ $\therefore \omega = \frac{1}{\dot{Q}'''} \frac{\partial \dot{Q}'''}{\partial t}$

Parameter	Description
WQNT	Total reactor thermal power [%]. The total power is calculated by summing the prompt and decay power to obtain the *actual* power produced in the reactor at any given time. This is in contrast to the fission power WQNF (Table 30), which is calculated as the *equilibrium power which would be attained* for the current neutron flux. This field is a measure of the current fission rate in relation to the equilibrium fission rate. For time-dependent cases where power is increasing or has increased, the fission power will generally be greater than total power because the decay heat will be over-predicted (it is assumed to react promptly whereas in the real case decay heat will react very slowly to changes in power level). For decreasing power calculations, the fission power will be less than total power for the same reason. It is suggested that *power* related plots be taken from the total and decay heat outputs, and that fission power (WQNF) be treated as an approximate measure of the current *flux* level.
DETFL	Detected fission power [%]. This field represents the flux detector location as specified by the user in the NZ datablock of the .tn3 file.
RODP	Actual control rod bank insertion position [cm]. (Output only during control rod motion ramps and during automated rod control “ramps”).

During the transient calculation, TINTE writes into the output file the same output blocks #7 and #8 as for the equilibrium run, but with the results from transient calculation at that specific timepoint. The user specifies these output timepoints by selecting the correct option in the .tn1 file (see control commands, section 8.3)..

Output Block #10

If TINTE has successfully completed a run and has reached the terminating control command “-3” (or –3.x, see section 8.3), it produces the final restart and/or p2d output fields and then prints an overview of any transient parts computed by issuing a series of transient history tables of all major scalar reactor variables for all temperature time steps.

After that, the run is ended with the final message:

TINTE – Programmende

If the user has chosen to end the .tn1 (or interactive console) input by the control command “-4” (see section 8.3), no transient history or further output is produced and a successful TINTE run is then simply terminated by the final message:

Tinte: STOP in Subroutine TIRSTI:

Steuerparameter -4 gelesen.

8. TINTE VERSION HISTORY

8.1 CODE VERSIONS RELEASED BY FZJ

This section indicates the history of the TINTE code's development by the ISR group at FZJ. Only the major releases are shown here.

8.1.1 Versions from 1985 until March, 2001

- TINTE was used only internally at FZJ / ISR, on various computer systems.
- The last internal WINDOWS version was tinte.exe ver. 301, a FORTRAN console application.

8.1.2 First External Release Version, Rel. 1.1, Vs. 301c, April 2001

- The first TINTE release version for outside of FZJ, under bilateral agreement (executable code version).
- Tinte.exe output extended to a new compacted and 132 column wide console output, independent from the default printer file output.
- Printer file name could be specified via a call argument.
- Optionally the usage of the default printer-file-only output listing was also possible.
- Various small corrections and cosmetic changes in the source code was made.
- Version was backwards compatible with previous input, output, and restart files.
- New / updated batch files "Tinte.bat" and "TiPS.bat" implemented as user front ends under WINDOWS.
- Auxiliary data reporting and plot interface routines ptrplot.exe, p2dplot.exe, and dreidimplot.exe included, with updated user front end batch file TPint.bat.
- A new TINTE installation directory structure was created.

8.1.3 Rel. 1.2, Vs. 501x, June 2001

- Second TINTE release version under bilateral agreement.
- Some changes in the correlations for calculating heat capacities and heat conductivities of graphite and concrete.
- Additional optional output of the 2d-fields of xenon and iodine densities on the printer output file and on the 2d plot data file.
- Additional output of the grid mesh elements with the largest convergence errors at the time step iteration reports on the printer output file.
- Lengthy heat balance reports removed from the console outputs.
- Backwards compatibility of input, output and restart files.
- Minor corrections and enhancements in the associated batch files Tinte.bat, TPInt.bat, and TiPS.bat.

8.1.4 Rel. 1.3, Vs. 801a, September 2001

- Third TINTE release version under bilateral agreement (tinte.exe dated on 05/09/01, secondary codes dated 28/09/01).
- Some corrections in modelling the heat transfer in cavities
- Additional optional output of the 2d-field of heat transition numbers (alpha* values) on the printer output file and on the 2d plot data file.
- Minor corrections (for option IPP2D=0) and enhancements at the user interface parts of the code (prompting for and documenting of the time control inputs).
- Backwards compatibility of input, output and restart files.
- Enhancement of the batch file Tinte.bat to support multiple TINTE versions.
- Adaptation of the plot interface routines to add the new 2d-array of alpha* values at the 2d plot data output file.
- A new utility P2dB2A.exe to convert the binary 2d plot data files to ASCII text files, if desired (unsupported program).
- Additional release of interface routines VSOP.(99) – TINTE and of TINTE nuclear data base preparation programs, consisting of five executables with seven associated batch script files and two nuclear data libraries (routines and data files unsupported).

8.1.5 Rel. 1.4, Vs. 1002xp, October 2002

- Fourth release version and first source code release.
- Another small correction at modelling the heat transfer in cavities.
- Enhancements in the chemical part of TINTE in parallel to calculations done for corrosion scenarios.
- Inclusion of gas mixing effects in TINTE flow network components.
- Obsolete pressure output figure in the time step monitor deleted and clearer naming of a 2D heat source output field.
- Adding of a new graphite type with material number 84 to the materials table.
- Slightly changed functions of the TINTE options STAB, TEKO and XEEF for deterring special k-effective effects by now including buckling changes.
- Enhancement of the TEKO option for calculating temperature feedback coefficients by now accepting general limiting temperatures for the reactor materials while uniformly incrementing all the region temperatures.
- Adaptations of the TINTE console output file to advanced features of the newer operating systems WINDOWS 2000 and WINDOWS XP.
- Auxiliary data reporting and plot interface routine ptrplot.exe extended for a larger time step number and partly recoded for more flexibility.
- Minor corrections in the batch file TiPS.bat.
- Extra version of the batch file Tinte.bat for systems WINDOWS 2000 and XP.

8.1.6 Rel. 1.41, Vs. 1202xp, January 2003

- Included a source code release update.
- Correction of a bug occurred at the heat transfer calculation for some special non-nuclear application (e.g. NACOK analysis).

8.1.7 Rel. 1.42, February 2003

- Update of only one binary file.
- Replacement of the obsolete CD version of interf.exe (unsupported program) by a newer version adapted to a change in the VSOP interface files structure effected in October 2001 (MIX vector).

8.1.8 Rel. 1.5, Vs. 403xp, April 2003

- Complete release, including TINTE source code.
- Corrections and extensions in the heat transfer parts of TINTE for some "exotic" application cases (i.e. without any pebble fuel elements, mainly).
- Corrections for computing the gas pressure in the case of a given Helium inventory and at the operation of safety valves.
- Enhancements in checking user inputs and computations for physically reasonable pressure and temperature data, with additional program warnings.
- A correction for the option to consider only a part of the non-local heat production by nuclear fissions.
- An extension of the VSOP-TINTE interface routine interf.exe (unsupported program) to allow the usage of disadvantage factors for cross-sections, also in regions outside the core (e.g. with Boron sticks in the reflectors).

8.1.9 Rel. 2.02, Vs. 204d, 27 February 2004

This is the TINTE release used for most 2006 and 2007 SAR calculations, and was a full source code and .exe release. Changes include:

- Extension of the TINTE code: implementation of a virtual linear rod motion model (called ROMO), which permits a simulation of uniform rod motion and can provide continuous reactivity control by automatic rod bank movements.
- Enhancement of the convergence of the corrosion chemistry by several code improvements
- Minor code changes: added option of comment lines in time control input, adding some more error messages
- Corrections in some of the batch script files to permit blanks in the name of the TINTE installation path
- Enhanced comment text in batch script TiPS.BAT concerning the possible alternatives to using TiPS, for the WIN 9x and WIN XP/2k operating systems
- Correction in (unsupported) auxiliary routine interf.exe to adjust the number of pebble types better to the "batches" of the VSOP code.
- A change in (unsupported) auxiliary routine tdb2.exe to increase the number of allowable VSOP material regions

- Addition of a set of sample inputs for using TINTE with the new ROMO feature, in the CD directory ROMO-samples
- All previous (external) TINTE release versions have now been collected in the sub-directory bin\versions as a convenience for the user interested in comparing results of different editions; they can be used at any time in concurrence with the latest release via an optional parameter of the Tinte batch script
- Some new correlations for the calculation of specific heat capacities and thermal conductivities for steel and graphite materials were added, according to the new data developed by M Mitchell and G Haag /46/. The following new correlations rules were implemented into the TINTE source code:

  1. Rule #201: The “Mitchell-Haag” correlations for the thermal conductivity and heat capacity of SGL grade A reactor graphite.

  2. Rule #202: Two new fits for the core barrel stainless steel thermal conductivity and heat capacity calculations.

  3. Rule #203: Two new fits for the reactor pressure vessel (RPV) stainless steel thermal conductivity and heat capacity calculations.

8.1.10 Rel. 2.03, 8.03.04

Replacement of the auxiliary routine interf.exe (unsupported program) by the currently valid version, which has two changes: a correction for disadvantage factors in the reflector, and enhancements of the treatment of V.S.O.P. "batches" (batch volumes).

8.1.11 Rel. 2.1, 11.05.04, TINTE vs. 504

- Update of Tinini source file & Tinte.exe.
- Corrections for special option STXT, convergence criterion in transients, and nuclear effect of corrosion.

8.1.12 Rel. 2.11, 20.09.04

- Update of three auxiliary routines.
- Correction of the TINTE nuclear data base preparation program tispec.exe (unsupported program) for spectrum regions with negative buckling values enhancement of the V.S.O.P.-TINTE interface routine interf.exe unsupported program) for reflector region spectrum calculations by applying internal mode option 10 rather than 9.
- Auxiliary data reporting and plot interface routine p2dplot.exe extended to 50 functions per diagram and smaller adjustments (WIN-XP etc.) done

8.1.13 Rel. 2.2, 12.11.04, TINTE vs. 504a

Update 6 TINTE source files & Tinte.exe.

- A correction for the burst disks behaviour after they have opened.
- A small enhancement at the calculation of the coefficients of the chemical computations (i.e.: the corrosion calculation)
- Adding of an input data check for the temperatures of coupled regions

- Reduction (using a logarithmic scheme) of the number of potential warnings about "temperature out of range" at the optional calculation of temperature dependent heat conductivities in the correlations submitted by PBMR Ltd. in 11/2003.

8.1.14 Rel. 2.21, 3.12.04, TINTE vs. 504b

- A minor update of 4 TINTE files & Tinte.exe
- Minor correction of the dose interpolation algorithm for the temperature dependent graphite conductivity
- Addition of a few console output messages for the nuclear data inputs read
- Suppression of some duplicate console output messages while reading the input data sections

8.1.15 Rel. 2.3, 11.03.05

A complete CD rel., updt. 4 aux. codes & nuclear lib

- Extension of the existing nuclear cross-section libraries by addition of a series of scattering tables for H2O and for D2O. The new library is named SPECLIBN-r04.lib (see in subdirectory tinte\lib).
- Replacement of 4 of the V.S.O.P.-TINTE interface and TINTE nuclear data base preparation routines (unsupported programs), i.e.: interf.exe, tisvor.exe, tispec.exe and tdb2.exe, by new versions which support (only) the new type of nuclear data library with water scattering tables included, as mentioned before.

8.1.16 Rel. 2.31, 17.05.05

Replacement of 4 of the V.S.O.P.-TINTE interface and TINTE nuclear data base preparation routines (unsupported). The WINDOWS XP tool setx.exe from Microsoft has been added to directory etc\WIN_XP. It should be copied to the TINTE installation directory tinte, on computers running under the systems WIN 2K or WIN XP. Setx.exe provides a similar function for WIN XP as WINSET.EXE for WIN 9x. WINSET.EXE is already used with TINTE under the operating systems WIN 9x. The batch script file TiPS.BAT has been updated accordingly. When using TiPS.bat, no settings in autoexec.bat or in the WINDOWS registry is now required under WIN 2K/XP for TINTE any more.

8.1.17 Rel. 2.4, 10. 1.06, TINTE vs. 504e

Release 2.4 of tinte.exe brings no other code changes except for providing a TINTE version with the total problem data storage increased from 2000000 to 5000000 8-byte-words, to allow for computing of very large reactor models with more spatial meshes, nuclear and temperature regions and/or material compositions than possible until now. (Version created on behalf of a request from PBMR Ltd).

- Auxiliary data reporting and plot interface routines (unsupported "TPI" package) ptrplot, p2dplot and dreidimplot (resp. t3dplot) reworked (v61):
- increased array dimensions now supporting "big" TINTE cases,

  current limits at the three codes are:

  - max. number of possible time steps on ptr file: 100000
  - max. number of possible time points on p2d file: 500

- max. number of 1d function tables that can be produced per run: 100
- max. number of TINTE mesh r/z-coordinates: 200 (each)
- max. number of equidistant r/z-coordinates for 3d plots: 500 (each)
- modifications to now permit both the interactive usage (as until today) and batch mode application (to speed up mass production of plot data)
- enhanced file handling, better error reporting and added information.
- Sample inputs for the batch mode use of the TPI routines newly included.

8.1.18 Rel. 2.5, 8.04.06, TINTE vs. 504f

- New auxiliary TINTE Plot Interface (TPI) manager program DiaManT released ("Diagram Manager for Tinte", vs. 1.2, a program in REXX language), as an add-on of not supported software beyond the TINTE license.
- It provides an alternative method to the interactive usage mode of the TPI routines via the old TPInt.bat script, permitting now the easy creation of the various TPI routines plot data sets in a batch mode and with only a single input file for any number of diagrams, thus much speeding up mass production of TINTE diagrams
- The free REXX interpreter Regina from http://regina-rexx.sf.net is needed to execute the Diamant program.
- Optimisation of the three TPI routines ptrplot, p2dplot and dreidimplot (t3dplot) for better usage in batch mode: added messages and omitting of error promting, a new option /B to enforce batch mode, specification of function types and transient time points also possible via their true names and values rather than only through current list numbers (TPI v63)
- Correction of the max. fuel temperature output in TINTE for cases with more than one fuel element type present, when small time steps are used.
- Enhancing transient computation performance by limiting the extrapolation of heat sources at the internal temperature pre-calculations allowing now also jumps with the rod movement ramps available (ROMO).
- Allowing prompt jumps with the rod movement ramps (ROMO).

8.1.19 Rel. 3.0, 22.06.06, TINTE vs. 306

- Renewed corrosion chemistry of TINTE, with improved algorithms and numerical procedures for solving the pertinent non-linear chemical equations of graphite corrosion by air and water ingress at high temperatures, which were new developed continuously over the past two years. The convergence of many transient computations has been increased and the computing times could been reduced considerably.
- In addition to the successful recalculation of the VELUNA and NACOK experiments, now gas ingress incidents at the PBMR can also be tracked numerically.
- Access to SGL grade A graphite via material type 201 has been added to the corrosion part
- In connection with the chemistry, the calculation of the gas flow processes and of the gas mixing by diffusion has also been enhanced.
- The convection incorporates now a space-dependent variable Cp as may be necessary for the gases other than Helium; the stability and accuracy of the mass flow is increased through a refined calculation of the pressure at mesh boundaries.

- The convergence of the neutron flux has been improved by more frequent recalculations of the mesh leakages.
- This release represents the second major code change since the first external release of TINTE in 2001, after the addition of the virtual rod movement and control feature, as the first major extension at release version 2.

8.1.20 Rel. 3.01, 2.07.06, TINTE vs. 306a

- Reduced number of warning messages in a corrosion chemistry routine
- Modified handling of a critical relation at the water shift reaction

9. APPENDIX A: IMPROVEMENT OF RADIATION TRANSFER CALCULATIONS IN TINTE

The treatment of heat radiation in a gap needs the temperature at the boundaries of the gap. Since TINTE temperatures are calculated in the mesh centres, this information is not available directly. In earlier TINTE versions the temperature of the adjacent meshes were used to calculate an effective conductivity for the gap. It was supposed, that the error would remain small because usually the mesh centred temperatures at one side are too low and at the other are too high and only the sum of this temperatures and the sum of their squares is necessary in this process. Meanwhile it has been proven that in certain cases the effective conductivity becomes too large in this way.

The error may be minimized, if the adjacent meshes at both sides of the gap are chosen as small as possible. A way to avoid this restriction may be found by the following ideas:

1. The boundary temperatures might be calculated in addition to the centre temperatures as it is done for the fluid. This is time-consuming and the boundary temperatures have to be stored on the restart file.
2. Instead of using the temperature in the material mesh, the temperature in the adjacent fine mesh could be used. In this way the error can only be reduced to some extent.
3. Finally there is a possibility to extract the boundary temperatures from the existing mesh temperatures when generating the constants for the equations to be solved. This avoids storing them in the restart file.

The third idea has been chosen for improvement in TINTE. Let the gap be surrounded by materials with a heat conductivity λ and not by a region with a given temperature. The heat flow at both sides of the gap can be taken to be the same as the radiative heat flow. Let T_1 and T_2 be the next mesh temperatures at both sides of the gap, calculated at the distances d_1 and d_2 from the gap with the conductive λ_1 and λ_2 in their meshes. For T_L and T_R, the boundary temperatures, the following equations can be written with respect to heat flow:

$$\left(T_1 - T_L\right)\lambda_1 / d_1 = \in C_S \left(T_L^4 - T_R^4\right)$$

and

$$(T_R - T_2)\lambda_2 / d_2 = \in C_S (T_L^4 - T_R^4)$$

Replacing the unknown T_L and T_R by

$$T_S = \left(T_L + T_R\right)/2$$

and

$$T_D = \left(T_L - T_R\right)/2$$

and using the relation

$$a^4 - b^4 = (a-b)(a+b)(a^2+b^2) = 2(a-b)(a+b)\left(\left(\frac{a+b}{2}\right)^2 + \left(\frac{a-b}{2}\right)^2\right),$$

one gets the relations

$$T_1\lambda_1 / d_1 + T_2\lambda_2 / d_2 - T_D\left(\lambda_1 / d_1 - \lambda_2 / d_2\right) - T_S\left(\lambda_1 / d_1 + \lambda_2 / d_2\right) = 0$$

and

$$T_1 - T_2 - 2T_D = 8 \in C_s\left(\lambda_1 / d_1 + \lambda_2 / d_2\right) T_D T_S \left(T_S^2 + T_D^2\right)$$

The first equation is linear in the new variable T_S and T_D. However, it is not suitable to be solved with respect to T_D, as it would have to be divided by a difference which might become zero. But it is well suited to be solved with respect to T_S:

$$T_S = \frac{T_1\lambda_1 / d_1 + T_2\lambda_2 / d_2}{\lambda_1 / d_1 + \lambda_2 / d_2} - T_D \frac{\lambda_1 / d_1 - \lambda_2 / d_2}{\lambda_1 / d_1 + \lambda_2 / d_2}$$

The second equation is of the third degree in both variables. Gathering the constants by:

$$A = 1/\left(4 \in C_S\left(\lambda_1 / d_1 + \lambda_2 / d_2\right)\ \right)$$

gives

$$\left(T_1 - T_1\right) A/2 = T_D\left(A + \left(T_S^2 + T_D^2\right)\right),$$

or

$$T_D = \frac{\left(T_1 - T_2\right) A/2}{A + \left(T_S^2 + T_D^2\right)},$$

a relation very well suited for a recursive determination of T_D. Together with the linear relation for T_S a solution can be found in less then 5 iteration steps, within a range of less than 1°C. If *s* is the width of the gap, the effective conductivity is then found as

$$\lambda_s = 4 \in C_{S.} s T_S \left(T_S^2 + T_D^2\right)$$

If a region adjacent to the gap has a predefined temperature and a heat transfer law j = -α(T - T_{rand}), then the corresponding λ/d is to be replaced by that α in the above definition of A.

These modifications have been added to TINTE. The tests have shown that the new method satisfies even strong accuracy requirements.

10. APPENDIX B: RADIATION HEAT TRANSFER BETWEEN DIFFERENT TYPES OF FUEL ELEMENTS

Radiation between arbitrary ordered surfaces, each of them at a constant temperature, may be investigated by using configuration factors. The configuration factor φ_{jk} from the surface unit O_j to the surface O_k gives the fraction of the radiation, emitted per surface unit from O_j, which arrives at O_k (see /23/ chapter Ka(9)). In a closed space, all radiation emitted from O_j has to reach any surface, the sum of φ_{jk} therefore has to be equal to 1. Furthermore the configuration factors fulfil a reciprocity condition $O_j . \varphi_{jk} = O_k . \varphi_{jk}$.

In pebble bed of different spheres all with the same diameter, in which the probability of finding the type of fuel element *i* is p_i, φ_{jk} will be proportional to the total surface of the target type k, which can be seen from *i,* i.e with the factor α to be adjusted, $\varphi_{jk} = \alpha . p_k$. If the probability of finding pebble types is normalized to 1, α has to be 1, as well, and

$$\varphi_{jk} = p_k .$$

The radiation emitted from a surface j with temperature T_j is $E_j = \sigma . \epsilon_j . T_j^4$ per surface unit. Here $\sigma = 5.67.10^{-8}$ / $Wm^{-2}K^{-4}$ is the Stefan-Boltzmann radiation constant and ϵ_j the emissivity of this surface. When this radiation reaches an other surface i, it is either absorbed or reflected with the probability ratio ϵi $(1 - \epsilon_i)$. To find the contribution of this reflection from other areas (according to /23/, chapter Ka(16)-(18)), the following consideration is done:

Let the incoming radiation at surface O_j be H_j, B_j the emitted, inclusively the reflected radiation, all per surface unit. Then

$$H_j = \frac{1}{O_j} \sum_l \varphi lj . B_l . O_l = \sum_l \varphi jl . B_l$$

$$B_j = E_j + (1 - \epsilon_j) H_l = E_j + (1 - \epsilon_j) \sum_l \varphi jl . B_l$$

holds true and

$$Q_j = B_j - H_j$$

becomes the net radiation emitted per surface unit. From the relation $\varphi_{jk} = p_k$ found above, it follows

$$H_j = \sum_j p_j . B_j ,$$

i.e. H_j takes the same value for all types of fuel spheres, and the form

$$B_j = E_j + (1 - \epsilon_j) \sum_l p_l B_l$$

results. The last relation is a system of linear equations according to the B_j. A solution can be found by multiplying with p_j and summing over all j:

$$\sum_j p_j B_j = \sum_j p_j E_j + \left(\sum_j p_j E_j + (1 - \in_j) \right) \sum_l p_l B_l ,$$

or introducing

$$\bar{\in} = \sum_j p_j \in_j ,$$

$$\sum_j p_j B_j = \sum_j p_j E_j + \left(1 - \bar{\in} \right) \sum_l p_l . B_l ,$$

and further

$$H_j = \sum_l p_l B_l = \frac{1}{\bar{\in}} \sum_n p_n E_n$$

and finally

$$B_j = E_j + \frac{1 - \in_j}{\bar{\in}} \sum_n p_n E_n$$

From this net radiation per surface unit of the fuel sphere type is found as

$$Q_j = E_j - \frac{\in_j}{\bar{\in}} \sum_n p_n E_n ,$$

a value which may become positive or negative, depending on whether the direct emission or the absorption dominates. Its average vanishes. In TINTE instead of the heat transfer per surface unit the heat flow per volume unit is necessary. Assuming a filling factor f, the volume occupied by one fuel sphere is

$$V = \frac{4\pi R^3}{3f}$$

and its surface $O = 4\pi R^2$.

Statistically the type j surface fraction in this volume is therefore $O_j = 4\pi R^2 p_j$, resulting in

$$q_j = \frac{O_j}{V} Q_j = \frac{3f}{R} p_j \sigma \in_j \left(T_j^4 - \frac{1}{\bar{\in}} \sum_n p_n \in_n T_n^4 \right)$$

This heat flow density has to be taken into account when calculating the temperature for every fuel sphere type. It has no influence on the average temperature of the region investigated, but it has influence on the temperature of the different fuel element types, especially with respect to the maximum temperature in the centre. The following transformation may give better convergence:

$$q_j = \frac{3f}{R} p_j \sigma \frac{\in_j}{\bar{\in}} \sum_n p_n \in_n \left(T_j^2 + T_n^2\right)\left(T_j + T_n\right)\left(T_j - T_n\right),$$

as the formulation may be interpreted as a boundary condition of the third kind.

11. APPENDIX C: HEAT CAPACITY RULES IN WKPT

Heat capacity is used per volume unit in TINTE. The unit for this quantity in TINTE is:

$$[p.c] = \left[\frac{W\,\mathrm{sec}}{cm^3 K}\right],$$

The temperature is supplied in °C.

In the subroutine WKPT some of these values were compiled by /16/ and /17/. For use with TINTE further rules have been added.

The rules are called by an identification number IRHOC. Its meaning is:

6 Steatit (Magnesuim-Silikat) with the temperature dependency

$$pc(T) = 2.7\left(0.7641 + 0.00157T\right)$$

7 Reactor graphite (information from HRB) with density 1.75 g/ cm^3:

$$pc(T) = 1.75\left(0.645 + 3.14\left(\frac{T}{1000}\right) - 2.809\left(\frac{T}{1000}\right)^2 + 0.959\left(\frac{T}{1000}\right)^3\right)$$

for temperatures below 1200 °C.

8 Carbon brick with a temperature dependency like reactor graphite, but with density 1.55 g/ cm^3:

$$pc(T) = 1.55\left(0.645 + 3.14\left(\frac{T}{1000}\right) - 2.809\left(\frac{T}{1000}\right)^2 + 0.959\left(\frac{T}{1000}\right)^3\right)$$

for temperatures below 1200 °C.

11 Stainless steel (V2A) (information from HOESCH):

$$pc(T) = 7.8\left(0.4533 + 0.382\left(\frac{T}{1000}\right)\right)$$

for temperatures up to 1200 °C.

12 Thermal shield (information from HRB):

$$pc(T) = 3.5 + 4.1\left(\frac{T}{1000}\right) - 1.1\left(\frac{T}{1000}\right)^2$$

38 CERAFELT, density 192 kg/ m^3, (information from producer FA Johnsmanville):

$$pc(T) = 0.192\left(0.84 + 0.287\left(\frac{T}{1000}\right)\right)$$

for temperatures up to 1260 °C.

39 Feuerleichtstein SILCAREF 140-08 (resp. 140-87B) with density 860 kg/ m^3
(information from producer SILCA mbH):

$$pc(T) = 0.860\left(0.7246 + 0.9749\left(\frac{T}{1000}\right) - 1.02\left(\frac{T}{1000}\right)^2 + 0.39\left(\frac{T}{1000}\right)^3\right)$$

for temperatures up to 1400 °C.

57 Concrete according to /30/, p.12, Tab. 3.1 and p. 13, Tab. 3.2:

$$p(T) = \begin{cases} 2.40 - 0.0002T & : \ 0^oC \le T \le 500^oC \\ 2.315 - 1.7711E10^{-4} e^{8.8781E10^{-3}\cdot T} & : \ 500^oC \le T \le 950^oC \\ 1.50 & : \ 950^oC \le T \le 1100^oC \\ 0.62 + 8.0E10^{-4}T & : \ 1100^oC \le T \le 1350^oC \\ 1.7 & : \ 1350^oC \le T \end{cases}$$

$$c(T)=\begin{cases}0.8237+1.3346E10^{-3}T-9.8457E10^{-7}T^2 & : \ 0^oC\le T\le 600^oC\\ 1.09+3.0E10^{-4}T & : \ 600^oC\le T\le 900^oC\\ -1.34+3.0E10^{-3}T & : \ 900^oC\le T\le 950^oC\\ 2.8163-1.375E10^{-3}T & : \ 950^oC\le T\le 1350^oC\\ 0.96 & : \ 1350^oC\le T\end{cases}$$

58 The average heat capacity of pre-stressed concrete. According to /30/, p.24, it is an average from that of concrete and that of iron. 10% iron are supposed. For concrete, density and capacity are taken from the item above, for iron the rules from /30/, p.16, Tab. 3.5 and 3.6 are used:

$$pFe(T)=\begin{cases}7.84-3.2432E10^{-4}T & : \ 0^oC\le T\le 715^oC\\ 8.00-5.0E10^{-4}T & : \ 715^oC\le T\le 1600^oC\end{cases}$$

$$cFe(T)=\begin{cases}0.47+4.35E10^{-5}T+2.75E10^{-7}T^2 & : \ O^0C\le T\le 800^oC\\ 0.68 & : \ 800^oC\le T\le 1510^oC\\ 0.85 & : \ 1510^oC\le T\le 1600^oC\end{cases}$$

resulting in: $pc(T)=0.9\,p_{concrete}(T)c_{concrete}(T)+0.1pFe(T)cFe(T)$

12. APPENDIX D: HEAT CONDUCTIVITY IN XLAMT

The heat conductivity is calculated as

$$[\lambda] = \left[\frac{W}{cmK} \right]$$

in XLAMT as it is used in TINTE. The temperature is given in °C.

Part of the rules used has been compiled by using /16/ and /17/. Some of them have been applied to specific investigations, but they might be of general interest as well. For investigations with TINTE, some other rules have been added.

The rules are called by an identification number ILAM.

The conductivity rules available are:

1 Molten sodium with the temperature dependency:

$$\lambda(T) = 0.0173\left(54.306 - 18.78\left(\frac{1.8T + 32}{1000} \right) + 2.0914\left(\frac{1.8T + 32}{1000} \right)^2 \right)$$

for temperatures 90 °C ≤ T ≤ 1300 °C.

2 Dose dependent matrix graphite (The irradiation temperature is assumed to be equal to the

actual temperature T):

$$\lambda(T, DOS) = 1.2768\left(\frac{-0.03906\left(\frac{T}{1000}\right) + 0.06829}{DOS + 0.1931\left(\frac{T}{1000}\right) + 0.105} + \left(\frac{T}{1000} \right) + 0.042 \right)$$

for 450 °C ≤ T ≤ 1300 °C, and [DOS]=[10^{21} EDN].

3 Reflector graphite with a complex approximation regarding temperature and dose. The conductivity of the non-irradiated graphite a 40 °C may be given as a factor to characterize the material. The item TVOR, which normally is used as an initial guess for the temperature iteration, is interpreted in this case as the irradiation temperature.

If a zero is given, 1.2 W /(cm K) is assumed.

4 Stagnant helium at 1 bar

$$\lambda(T) = 2.97E10^{-5}(T + 273)^{0.69} + \frac{9.32E10^{7}T}{T^{5} + 4.29E10^{14}}$$

5 Global conductivity for a bed of 6 cm graphite spheres, as used by Petersen under certain as

assumptions /16/ concerning the graphite spheres conductivity. It is derived from the experiments of Robold /31/:

$$\lambda(T) = 1.9E10^{-5}(T - 150)^{1.29}$$

for T ≥ 250 °C.

6 Global conductivity for investigations of a special experiment using a bed of steatite spheres.

7 Reactor graphite with

$$\lambda(T, \lambda_0) = \lambda_0\left(1 - 1.084\left(\frac{T}{1000}\right) + 0.743\left(\frac{T}{1000}\right)^2 - 0.213\left(\frac{T}{1000}\right)^3\right)$$

for T ≤1700 °C. In contrast to rule **3** the dose dependency has to be considered in the factor λ_0

8 Carbon brick with

$$\lambda(T) = 0.05 + 0.03\left(\frac{T}{1000}\right)$$

for T ≤1000 °C, as proposed in the thesis of Lukascewicz /32/ (see also rule **42**).

9 Stagnant air at 1 bar:

$$\lambda(T) = 5.14E10^{-6}(T + 273)^{0.7}$$

10 Thermal shield (information from HRB to /16/):

$$\lambda(T) = (48.4 + 0.0205T)/100$$

11 Stainless steel (V2A) (information from THYSSEN to [16]):

$$\lambda(T) = 0.1517 + 0.133\left(\frac{T}{1000}\right)$$

for –100 °C ≤ T ≤ 750 °C.

12 An older form of the pebble bed conductivity [33]:

$$\lambda(T) = 1.1538E10^{-6}(T + 100)^{1.6622}$$

for T ≥ 250 °C.

13 Steatite:

$$\lambda(T) = 0.221 + 0.058125\left(\frac{T}{1000}\right).$$

14 Graphite pebble (4 cm diameter) bed conductivity according to experiments by Robold [31]:

$$\lambda(T, \lambda_0) = \lambda_0\left(0.0162 + 0.1131\left(\frac{T-390}{1000}\right)\right)$$

for 90 °C ≤ T ≤ 680 °C.

15 Armed concrete (information from Altes (FZJ) to [16]):

$$\lambda(T, \lambda_0) = \lambda_0\left(0.0186 + 0.0118\left(\frac{T-273}{1000}\right)\right)$$

for T ≤ 550 °C.

16 Conductivity of a prismatic block HTR core with axial and radial different values:

$$\lambda_Z(T) = \left(0.003963676\left(\frac{T}{1000}\right)^4 - 0.01619826\left(\frac{T}{1000}\right)^3 + 0.01246728\left(\frac{T}{1000}\right)^2 + 0.12432555\left(\frac{T}{1000}\right) + 0.1690167\right)$$

and

$$\lambda_R(T) = \left(0.007475742\left(\frac{T}{1000}\right)^4 - 0.06353831\left(\frac{T}{1000}\right)^3 + 0.1164280\left(\frac{T}{1000}\right)^2 + 0.07005595\left(\frac{T}{1000}\right) + 0.05880077\right).$$

17 Effective conductivity of carbon and graphite felts in vacuum:

$$\lambda(T) = 0.01163\begin{cases}(3.0E10^{-5}T + 0.05) & : \quad 0^oC \le T \le 1000^oC \\ (1.167E10^{-4}T - 0.06) & : \quad 1000^oC \le T \le 1600^oC \\ (3.5E10^{-4}T - 0.41) & : \quad 1600^oC \le T \le 4000^oC. \end{cases}$$

18 Effective conductivity of carbon and graphite felts in argon or nitrogen atmosphere:

$$\lambda(T) = 0.01163.\begin{cases} (8.0E10^{-5}T + 0.07) & : \quad 0^oC \le T \le 1000^oC \\ (2.75E10^{-4}T - 0.14) & : \quad 1000^oC \le T \le 1400^oC \\ (6.5E10^{-4}T - 0.67) & : \quad 1400^oC \le T \le 1800^oC \\ (1.067E10^{-3}T - 1.4) & : \quad 1800^oC \le T \le 4000^oC. \end{cases}$$

19 Roller bearing steel (100RC6) (private information from Barthels (FZJ) to [16]):

$$\lambda(T) = 2.79914\left(\frac{T}{1000}\right)^4 - 1.92371\left(\frac{T}{1000}\right)^3 - 0.127508\left(\frac{T}{1000}\right)^2 + 0.0540229\left(\frac{T}{1000}\right) + 0.41702$$

for 0 °C≤ T ≤550 °C.

20 Stagnant nitrogen:

$$\lambda(T) = 4.373E10^{-6}(T + 273)^{0.715}.$$

21 Kaowool mats in air according to [34]:

$$\lambda(T) = 2.43191E10^{-9}T^2 + 8.42352E10^{-7}T + 3.55708E10^{-4}$$

for 50 °C ≤ T ≤ 600 °C.

22 HAW glass:

$$\lambda(T) = 0.00927 + 0.0317106\left(\frac{T}{1000}\right) - 0.112195\left(\frac{T}{1000}\right)^2 + 0.148244\left(\frac{T}{1000}\right)^3$$

for T ≤ 550 °C.

23 Graphite pebble (6 cm diameter) bed: This rule was derived from assumptions which allow a better description of new experiments done by Barthels [35]

$$\lambda(T) = \begin{cases} 2.5E10^{-6}T^{1.545} + 0.015 & : \quad 100^oC \le T \le 1300^oC \\ 2.0E10^{-5}(T - 135)^{1.287} & : \quad 1300^oC \le T \le 2500^oC. \end{cases}$$

The rules **30** to **39** describe the conductivity of insulating materials as used in the experiments SANA 1 [36] and NACOK [37].

30 Fire light bricks OFL54, (produces information Fa. Dr. Otto):

$$\lambda(T) = \left(0.24 - 0.048\left(\frac{T}{1000}\right) + 0.116\left(\frac{T}{1000}\right)^2 + 0.018\left(\frac{T}{1000}\right)^3 \right) / 100$$

for 0 °C ≤ T ≤1200 °C.

Note: As given here, +0.116 for the third term is correct. In the TINTE source (XLAMT) the wrong -0.116 should be corrected!

31 CERACHEM-Blanket K9 MODUL, Fa. GOSSLER ([36] from producers information):

$$\lambda(T) = \left(0.0985 - 0.0202\left(\frac{T}{1000}\right) + 0.0041\left(\frac{T}{1000}\right)^2 + 0.0001\left(\frac{T}{1000}\right)^3 \right) / 100$$

for 0 °C ≤ T ≤ 1650 °C.

32 Fire light bricks RI 30 B, 'SAVDIE Fuerfest GMBH' ([36] from producers information):

$$\lambda(T) = \left(0.396 + 0.0264\left(\frac{T}{1000}\right) - 0.0025\left(\frac{T}{1000}\right)^2 + 0.0001\left(\frac{T}{1000}\right)^3 \right) / 100$$

for 0 °C≤ T≤ 1650 °C.

33 CERAFORM 1000 /CERABOARD 100, Fa. GOSSLER ([36] from producers information):

$$\lambda(T) = \left(0.047 + 0.0071\left(\frac{T}{1000}\right) + 0.0005\left(\frac{T}{1000}\right)^2 + 0.00007\left(\frac{T}{1000}\right)^3 \right) / 100$$

for 0 °C ≤ T ≤ 1430 °C.

34 THERMOAIL 1100, Fa. GOSSLER ([36] from producers information):

$$\lambda(T) = \left(0.0803 + 0.004\left(\frac{T}{1000}\right) + 0.00007\left(\frac{T}{1000}\right)^2 \right) / 100$$

for 0 °C ≤ T≤ 1100 °C.

35 THERMOSIL 1000, Fa. GOSSLER ([36] from producers information):

$$\lambda(T) = \left(0.0041 + 0.0045\left(\frac{T}{1000}\right) + 0.0011\left(\frac{T}{1000}\right)^2 + 0.000055\left(\frac{T}{1000}\right)^3 \right) / 100$$

for 0 °C ≤ T ≤ 1000 °C.

36 GOSSLEROC GMP 200, Fa. GOSSLER ([36] from producers information):

$$\lambda(T) = \left(0.058 + 0.0001\left(\frac{T}{1000}\right) + 0.004\left(\frac{T}{1000}\right)^2 \right) / 100$$

for 0 °C ≤ T ≤ 800 °C.

37 UNIFELT QUALITAET FT 13 U, Fa. MORGEN ([36] from producers information)

$$\lambda(T) = \left(0.0529 + 0.0087\left(\frac{T}{1000}\right) 0.0016\left(\frac{T}{1000}\right)^2 \right) / 100$$

for 0 °C ≤ T ≤ 1250 °C

38 CERAFELT, density 192Kg/m**3, Fa. JOHNS-MANVILLE, (producers information):

$$\lambda(T) = \left(0.0461 + 0.0039\left(\frac{T}{1000}\right) + 0.0011\left(\frac{T}{1000}\right)^2 \right) / 100$$

for 0 °C ≤ T ≤ 1260 °C.

39 Fire light bricks SILCAREF 140-08 (resp. 140-87B) Fa. SILCE mbH (producers information):

$$\lambda(T) = \left(0.2241 + 0.307\left(\frac{T}{1000}\right) + 0.27\left(\frac{T}{1000}\right)^2 + 0.12\left(\frac{T}{1000}\right)^3 \right) / 100$$

for 0 °C ≤ T ≤ 1400 °C.

For investigations of the **AVR** /3/ the rules **40** to **44** have been compiled.

41 The conductivity of the reflector graphite used in the AVR [38] regarding the dose dependency. It is compiled in [39] and [40]. The conductivity of the un-irradiated graphite was assumed as 1.05 W/(cm K).

42 describes the conductivity of carbon bricks [41]:

$$\lambda(T) = 0.039\left(1 + 0.768\left(\frac{T}{1000}\right) - 0.492\left(\frac{T}{1000}\right)^2 + 0.079\left(\frac{T}{1000}\right)^3 \right)$$

This rule gives significantly lower values than rule **8**. It may be used for sensitivity investigation as lower boundary.

43 Rule 43 was used when simulating the azimuthal AVR built-ins (the AVR-noses) in two-dimensional R-Z geometry (see [1], section 7.1.5.). Modelling these 'noses' containing the control rods as a volume conserving cylindrical layer cannot conserve their linear dimensions. The azimuthal thickness of the noses (25 cm) becomes a 10 cm thick cylindrical layer and its surface is enlarged by a factor 2.5. Fixing the λ would enlarge the heat flow into it by the factor 6.25 (1/2.5 times the distance and 2.5 times the surface). This and the next rule **44** are compensating this by an anisotropic conductivity: The radial component is reduced by 1/6.25 using rule **41** for the graphite while **44** is using rule **7**, instead.

For TINTE investigations two-dimensional tables of conductivities with respect to temperature and dose are used in subroutine TILTAB. The items of the first of these tables are predefined with conductivities measured by Binkele using A4-graphite /19/. From his measurements those with an irradiation temperature of 950°C are selected. Except the graphite emission property ϵ, the other parameters describing the bed (predefined by sphere diameter 6 cm, filling factor 0.61, gas pressure 1 bar and flattening parameter 0.0165) as well as the conductivity table can be overwritten for specific investigations

Changing the emission properties demands changes in TLTAB. It is better to use the rules **70** to **71** instead. A first call to TILTAB initiates the calculation of global pebble bed conductivities at all definition points. When calling TILTAB later, the actual values are found by interpolation. Outside the definition limits, constant extrapolation is done.

In the rules **53** to **55** the gaps between the spheres are supposed to be filled with He and in **63** to **65** with N_2 for calculating effective conductivities.

52 gives λ values for the graphite spheres.

53 This rule returns the global conductivity of the bed filled with He according to a theory from Robold [31].

54 This rule returns the global conductivity of the bed filled with He according to a theory from Zehner-Schlünder /42/ (see also /23/, section De).

55 The difference between the two last rules is: The first gives higher values for higher temperatures and the second for lower ones. New experiments from Barthels [35] can be interpreted best, when the larger conductivity of both theories is taken. This maximum is returned by this rule (He-filling of gaps).

56 Metallic uranium according [43]:

$$\lambda(T) = 0.009(24.43 + 0.0327T).$$

57 Concrete according to [30], p14, Tab. 3.3:

$$\lambda(T) = 10^{-2}.\begin{cases} 1.746 & : \quad 0^o C \le T \le 100^o C \\ (1.8637 - 1.1771E10^{-3}T) & : \quad 100^o C \le T \le 800^o C \\ (2.4607 - 2.686E10^{-3}T & \\ +\,9.5333E10^{-7}T) & : \quad 800^o C \le T \le 1300^o C \\ 0.58 & : \quad 1300^o C \le T. \end{cases}$$

58 Ideal conductivity of a pre-stressed concrete vessel according [30], p24, Tab. 4.1:

$$\lambda(T) = 10^{-2}.\begin{cases} (2.050 - 0.8750E10^{-4}T) & : \quad 0^o C \le T \le 100^o C \\ (2.178 - 0.1369E10^{-2}T) & : \quad 100^o C \le T \le 800^o C \\ (2.771 - 0.3044E10^{-2}T & \\ +\,0.1073E10^{-5}T^2) & : \quad 800^o C \le T \le 1300^o C \\ 0.627 & : \quad 1300^o C \le T. \end{cases}$$

63 This rule returns the global conductivity of the bed filled with N_2 according to a theory from Robold [31].

64 This rule returns the global conductivity of the bed filled with N_2 according to a theory from Zehner-Schluender [42] (see also[23], section De).

65 Similar to rule **55** this rule returns the maximum of the last two rules for the bed conductivity (now with **N_2** filling).

The following rules **70** to **76** give the effective heat conductivity rule of the material properties, the input values for this material are read in the block starting with 'TZ%' or 'TZ*'. Similar to the previous rules **52** to **55** and **63** to **65** the generated conductivities are stored in tables.

70 This rule interpolates the λ values for the sphere material from a table.

The table is generated during the input process either from any previously described rule or by direct given values for arbitrary temperature points.

71 This rule returns the global conductivity of the bed filled with He according to a theory from Robold [31].

72 This rule returns the global conductivity of the bed filled with He according to a theory from Zehner-Schlünder [42] (see also[23], section De).

73 Similar to **55** the maximum of **71** and **72** is taken (He –filling of gaps).

74 This rule returns the global conductivity of the bed filled with N_2 according to a theory from Robold [31].

75 This rule returns the global conductivity of the bed filled with N_2 according to a theory from Zehner-Schlünder [42] (see also[23], section De).

76 Similar to **73** this rule gives the maximum of both theories (here with **N_2**-filling of the gaps).

13. APPENDIX E: LINEAR ROD MOTION MODEL (ROMO)

13.1 INTRODUCTION

NOTE: The information contained in this Appendix is taken directly from a more detailed report issued by the ISR at FZJ /50/ This report contains extensive samples and results describing the use of the ROMO model, and should be consulted when the ROMO option is selected by the user.

Control rods and small absorber spheres are described in the reactor dynamics code TINTE /1/ and /2/) according to the model of equivalent neutron poisoning in annular regions /49/. Due to the 2-D r-z geometry used in TINTE, the rods cannot be modelled explicitly as 3-D components. They are simulated by axially subdivided, homogenised annuli at the correct radial positions of the neutron absorbers. The effective poisoning is derived from detailed neutron transport calculations. In each of the axial regions a constant amount of absorber is used. The principal applicability of this model has been previously proven by comparison with 3-D equilibrium neutron diffusion codes /50/.

The rod (bank) movement can be modelled approximately in TINTE by changing the neutron poison concentration in that region which is momentarily associated with the rod tips, by using a (linear) concentration time ramp feature of the code. (Refer to the input description on Nuclear Ramps, Section 8.2.1) A certain disadvantage of this procedure is caused by the fact that a linear change in time of the poison concentration does not lead to a linear change of reactivity nor does it reflect correctly the effect of a movement of the rod bank with constant velocity. To achieve a proper correspondence here, although still in relatively coarse time steps, sophisticated stepwise ramp movements had to be applied in the past.

Now a simply to handle method for continuous rod movement between any two axial positions has been developed, the so-called ‘Linear Rod Motion Model’ (ROMO).

This model performs automatically the simulation of a uniform control rod motion by

1. using a special exponential interpolation algorithm for the poison concentration in the currently last (i.e.: lowest if inserting rods in downward direction) active rod region (which may change during rod movement) and
2. providing the correct stepwise filling or depleting by poison of all vertically adjacent rod regions, depending on the requested rod movement.

Thus the user has to define only both end positions of the control rods and the desired time-span of the motion, to conduct a rod bank movement computation.

13.2 METHOD FOR THE LINEAR ROD MOTION SIMULATION

Only the *basic formulas* of ROMO for the correspondence of absorber concentrations and partial rod insertion lengths in a single rod region (the region where the rod tips are placed) will be shown in this description.

Calculation of the variable absorber concentration c^* as a function of the virtual insertion length l of the rods in an annular region with partially inserted rods :

$$c^*(l) = \frac{\ln\left(1-\frac{l}{L}S\right)}{\ln(1-S)} * c \ , \quad 0 \le l \le L$$

c absorber concentration in the rod region when rod bank is fully inserted in this region

c*(l) absorber concentration in the rod region when rod bank is partially inserted in this region over the length l , according to the linear rod motion model

l insertion length of the rod bank in the rod region

L length (axial height) of the rod region

S interpolation factor of the special exponential interpolation scheme of the linear rod motion model (see values rblkns(*) in 16.5.2), also called “absorber blackness value” because it characterizes the neutron absorber poison strength; it is always: 0 < S < 1 .

Calculation of the virtual rod insertion depth l from the absorber concentration ratio c*/c in an annular region with partially inserted rods :

$$\frac{l}{L} = \frac{1-(1-S)^{c^*/c}}{S} \ , \quad 0 \le c^*(l) \le c$$

(The variables are the same as explained above).

Thus the virtual length is expressed as an exponential function of the absorber concentration.

13.3 A REACTIVITY CONTROLLER PROGRAM WITH AUTOMATIC ROD MOVEMENTS

This section shall be added in a later revision of the report, in order to describe the theoretical and practical details of the automatic controller program already implemented.

13.4 IMPLEMENTATION AND PROGRAM STRUCTURE OF THE ROMO CODE

The rod motion features have been added to the TINTE code by ten new subroutines. Some changes at the already existing subroutines had to be done, too, in order to join the new parts with the old ones and to extend the pertinent input and output statements.

The ROMO subroutines are characterised by the following main functions:

- TiRomo: initialises variables of the rod motion model
- RomoIn: executes the new TINTE special command options ROMO and ROMOx and reads in the general input for the rod motion model
- RodRmp: interprets the input for the ramps of the rod motion model
- RodMot: executes virtual “linear rod motion ramps” by means of exponentially interpolated poison concentration changes in the overlay regions of the rod rings and changes the overlay cross-section concentrations accordingly in each TINTE nuclear time step
- RomoCT: co-ordinates auxiliary time step functions for the execution of the reactivity ramps with automatic control of the virtual rod bank movements, calls the subroutines RomoCP and RodTim (see below) in each time step

- RomoCP: determines important controller parameters of the reactivity feedback of the core state and of the current control rod efficiency from an analysis of the preceding control time steps
- RodTim: determines the next required control time point during reactivity ramps with automatic rod motion control as well as adjustments of the nuclear and temperature time step lengths of TINTE for the rod control when necessary
- RodCtl: executes control of or controlled ramps of the external reactivity by automatic virtual rod movements and changes the overlay cross-section concentrations accordingly in the nuclear time steps
- RodDep: calculates virtual rod insertion depths in the context of the linear rod motion model, to be used at various code places and for output purposes
- ExPoRa: optional and alternative nuclear concentration ramp routine using linear or exponential interpolation for individual overlay regions; it provides the new TINTE special command options EXPO and EXPOx; this feature can be used independently of the ROMO model, but is not described here in detail.

13.5 INPUT DESCRIPTION FOR THE ROD MOTION FEATURE IN TINTE

The TINTE Input Description (see Section 5) is now extended by the following four input data sets. These are **optional** additional inputs to the classic TINTE input which only need to be supplied if one actually wants to use the new ROMO control features in the TINTE problem case to be computed.

13.5.1 Using TINTE Overlay Compositions with ROMO

In TINTE the so called overlay compositions can be used to simulate various control rod bank insertion states. That is: diluted neutron absorber poison concentrations in special annular “rod ring regions” can be defined in addition to their rod-free composition cross-sections, at the end of the .tn3 file inputs. This is described in Section 5.7. ROMO uses these same overlay inputs in the following manner:

The last (and only the last!) overlay composition number and concentration on each overlay region input entry is considered to be used and adjusted as the rod absorber material by all of the ROMO actions, if this overlay region is later defined (see next paragraph, “card 1”) as belonging to ROMO ’s rod displacement range. This last composition concentration is also accepted by ROMO as starting input when calculating the original rod insertion position (the “rod insertion depth”) before any user directed changes through rod motion “ramps” are executed.

ROMO itself will vary the last composition concentrations between the values of zero and one when it becomes active to do a rod motion or a rod control action, thereby assuming that a rod bank is fully out or in for this region with these limit values and that any partial rod insertion in a region has to be simulated by determining (in an optimised but non-linear dependency) an intermediate poison concentration equivalent in reactivity.

As usual with TINTE, any changes in the last composition concentration of an overlay region – whether by a “nuclear ramp” or by ROMO executing a virtual rod motion – are automatically compensated through a corresponding change in the first or basic composition concentration. By the way, nuclear concentration ramps and ROMO actions for rod movements may be used concurrently, if so desired.

It is however important to understand that ROMO 's view of the rod movement simulation does per definition only comprise the **last** composition entries of the overlay region cards. Therefore, any absorber material composition present in a .tn3 file will not be recognised by ROMO when it is placed before the last composition entry of an overlay region.

13.5.2 Definition of the Rod Motion Model to be Used

Prior to apply any of the ROMO features for conducting reactor transients, a certain rod bank configuration and the rod moving area have to be defined by the user. In addition, some optional input parameters for the controller program and some information with respect to poison strength and its interpolation for optimum linear rod motion simulations may be specified with the same general input data part of ROMO.

The general input for definition of the ROMO rod motion model has been integrated in TINTE by means of a new control option "**ROMO**" in the frame of the special program control commands **"-11"** in the transient control input (see Section 8). The special program control command –11 is generally input to TINTE by an input card for values of ZEITS, ZEITZ, WERTZ, IORT, IART where ZEITS = **-11** and ZEITZ, WERTZ, IORT, IART = 0.

The following new option is now available among the special program control commands "-11" (and belongs to the first group of program options to set parameters for a modification in the calculation).:

> **ROMO –** This option may be used to define or re-define a ROMO rod motion model together with its constant general input data, i.e. mainly the set and sequence of rod regions which shall be used as ROMO 's rod insertion area and their relevant properties. This input itself does not start any rod motion or control action. This is done via the input 16.5.3 and 16.5.4 below. Only the **current rod insertion depth** of the reactor model will be computed according to the ROMO rod regions and their current overlay composition concentrations which have been input from the .tn3 file (or from the restart file), and have possibly been changed by previous nuclear ramp inputs (and previous ROMO actions with any preceding ROMO model definitions, too). The current rod insertion depth ("Stabtiefe" in German), as well as the minimum and maximum rod insertion positions are reported in the console and print files output. The current insertion depth will also be written to the scalar transient functions plot data file (the ptr file).

The ROMO command needs at least one parameter (nnrg), but can optionally have up to ten parameters on the same input card. The ROMO input parameters are listed in Table 32.

Table 32: ROMO Input Parameters

Parameter	Description
nnrg	number of overlay regions which shall belong to ROMO 's rod insertion area; this value is mandatory. The following values are optional. If omitted or input as zero they will be set to the default values of the program indicated below. They are only used when the automatic controller is switched on (see 5.4).
rspeed	constant rod motion speed (absolute value) of the reactor design in cm/s (default: 1). The ROMO controller adheres to this fixed value and will never move the rods with any other speed during its autonomous rod control actions (even if it knows well in a certain case that faster moving would yield a better result).

Parameter	Description
shrircmi	minimum TINTE time step size in s which ROMO is allowed to use during automatic rod control (default: 1). In connection with the constant rod speed, the size of this value naturally limits the possible performance of the controller, since the rods must either be moved or kept fixed during any TINTE time step. One chooses this value in a compromise between a perfect control efficiency (with very small shrircmi) and avoiding of too small and too many time steps in a transient run (with larger shrircmi).
rtbb	reactivity dead band half width ("Totbandbreite") for the controller in Nile or % (default: 0.05, i.e.: 50 mNile). You may choose any other positive value or even zero. But remember that the controller can only act more precise in reality if you allow him appropriately smaller time steps, too (see the value shrircmi).
dkdrd	an initial guess for the rod efficiency (delta k-eff / delta rod depth) at the starting insertion position for the controller in Nile/cm or %/cm or 1/m (default: -0.01). This means the incremental change of the reactor reactivity per change of rod bank insertion length. Since the reactivity normally decreases with the rod insertion, this value is usually negative. The value of dkdrd is not needed as an accurate input data, since the ROMO controller will estimate this item more precisely and even dependent on the actual rod positions soon after starting its activity and wants only a crude initial guess. One may therefore also substitute an average or integral value for the true position dependent differential value, for example : the reactivity change from the state "rods completely out" (out of the area defined for the ROMO model) to a state "rods somewhere in" divided by the corresponding insertion length.
itst (1:5)	(five reserved items for testing purposes).

After the ROMO control option card, some inputs to define the actual rod motion model to be used are submitted on three additional cards which are read in format-free. The cards are:

- **Card 1: lrrg(1), lrrg(2), ..., lrrg(nnrg**)

These are the region numbers of the nnrg overlay regions (in the .tn3 file) which shall define the desired total rod insertion area (i.e. axial range) for the current ROMO model. The numbers must be given as negative numbers, as defined in the .tn3 file. Their sequence defines the order and direction in which ROMO has to insert the rod bank, starting with lrrg(1) and ending with lrrg(nnrg). This order can be defined here independent of the geometric positions of those regions in the TINTE material zones layout. The user has thus the possibility to define the starting and ending points of the possible rod motions and also of the automatic rod control independently for the ROMO model definition (e.g. for up- or downward rod or other absorber region insertions, withdrawals, etc.).

There is a special option available to facilitate the input of this region number sequence: a zero can be used to default the respective region number according to the continuation of the last preceding input numbers as an arithmetic series. For example: with nnrg = 10, the input of only the number –2 (with explicit input or defaulting of 9 zeros) will result in the region series: -2, -3, ..., -11; and the input of –1 –3 4*0 -24 -20 will yield the sequence: -1, -3, -5, -7, -9, -11, -24, -20, -16, -12 .

- **Card 2: rlen(1), rlen(2), ..., rlen(nnrg**)

These are the rod length values (in cm) which ROMO shall attribute to the nnrg rod regions when calculating the total rod insertion lengths. They can be (but need not to be) equal to the axial heights of the respective regions in the TINTE r/z-geometry reactor model.

- **Card 3: rblkns(1), rblkns(2), ..., rblkns(nnrg**)

  These are the exponential interpolation parameters to be applied by ROMO when deriving part lengths for the rod regions where the rod tips actually move from absorber concentrations and vice versa. Their values must be (strictly) between 0 and 1. They are usually determined by trial and error from a previously done series of runs using ROMO with several test numbers for the rblkns(*) in order to optimize them for a smooth S-curve over the whole rod insertion area. With the usual rod representation models for HTR ‘s by equivalent diluted absorbers, one can expect to have at least a reasonable starting model with values of about 0.6 to 0.8. Since these fit parameters of the linear rod motion model ROMO are strongly related to the effects of absorber strength, self-shielding and poison saturation (which sometimes are expressed via the notion of “blackness” of “grey” neutron absorbers), we may also call the item rblkns(i) the “absorber blackness value” (“Absorber-Schwärze-Wert”) for the i-th ROMO region.

The inputs given on the ROMO option card (Table 32) and the three parameter input cards are checked by the code for correctness as much as possible. If the input is done interactively from the console (and not from a .tn1 file) the user gets prompted for the right inputs for cards 1 to 3, and when any erroneous item is entered, the code will report it, and give the chance to correct it at once by repeating the respective input card.

Note that the parameters defined stay valid until they are explicitly reset again, either by switching off the ROMO option (via a –11 command option input of “ROMO” for example) or by defining a new rod motion model for the reactor via using the ROMO option again with new inputs.

These input data are not recorded in the TINTE restart files. Therefore, to continue a case via a subsequent run with ROMO remaining active over the restart, these inputs must be read in again for the new run. As usual with TINTE ‘s control command –11 inputs, after the ROMO option input any further special command option may be read in. A blank card signals the end of special control command –11.

13.5.3 Rod Motion Ramps

The activation of the rod motion features is done via two new nuclear ramps, similar to the already existing nuclear ramps defined on global nuclear parameters (e.g. for k-eff and reactor total power). The first new ramp is described in this section, the second one in Section 16.5.4. Note that this input belongs to the transient control inputs “TT%” or “TT*” of the TINTE input description (see Section 8.2.1).

The first of the two new ramps is designed to permit the user to conduct rod bank movements via a virtual motion, with constant velocity between any tip positions (within the previously defined ROMO rod insertion area). In contrast to the classic linear concentration ramps of TINTE, the new rod motion model yields a better approximation of rod movements linearly in time. There is no need to deal explicitly with many defined ramps to fill or deplete the individual rod regions in the right sequence, since this is now managed automatically by ROMO if the desired end position is known. It starts the movement from the current rod position (please refer to the explanation of the current rod insertion depth at the beginning of the ROMO option input description in Section 16.5.2).

The five input items of TINTE ‘s transient control input have the following meaning for the rod motion ramps (Table 33):

Table 33: Rod Motion Ramp Parameters

Parameter	Description
ZEITS	The point in time where the rod movement starts (in seconds), counted from the beginning of the TINTE transient case. (For ZEITS = 0 or ZEITS < 0 refer to the TINTE input description).
ZEITZ	The point in time where the rod movement has finished (in seconds).
WERTZ	The corresponding end position of the rod insertion at the end time point (in cm), counted from the beginning of the first rod region, and calculated with respect to the rod length values rlen(*) specified in the general ROMO inputs in 16.5.2.
IORT	Always has a value of **0** here.
IART	Always has a value of **4** here (this number indicates a *rod motion* ramp among the global nuclear ramps).

Rod motion ramps can only be used after the general inputs of rod model definition in 16.5.2 have been done. These input data are not recorded in the TINTE restart files. Therefore, to continue a not yet finished rod motion ramp via a subsequent run, the ramp inputs must be read in again at the restart run.

13.5.4 Reactivity Control and Reactivity Ramps with Automatic Rod Movements

The second new global nuclear ramp is devoted to the automatic reactor control via rod movements. The reason why this program control command has been implemented in the form of a “nuclear ramp” input deserves an explanation. At first, this makes the input simple and compliant to the other TINTE transient control commands. Secondly, it allows the user not only to have the rod controller hold the reactivity constant (most often zero), but also to increase or decrease the external reactivity (demand) in the form of a linear ramp from the current level to any arbitrary value by using the rods.

Examples of these are: evaluating the theoretical effects of counterbalancing some assumed reactor feedback or perturbations by means of true spatial rod movements, and the switching of the reactor state from a virtual global reactivity control to a state with explicit rod control or vice versa.

In most of the practical cases, one will probably use this ramp input to switch on the controlling of the reactivity by rods (and from reactivity zero to zero, i.e. with no change in external reactivity: see next paragraph).

The real target variable for the automatic controller is of course not the global reactivity of the reactor (which is undefined in a dynamic computation), but the “external reactivity” which is required to keep the reactor at that total power level which the user has specified (via the total power ramp input).

In TINTE, the calculation of this “external reactivity” is a well implemented feature for transients conducted with the boundary condition of a constant total reactor power or a power ramp. This existing algorithm has been used when implementing the ROMO controller. When the rod controller is activated, the “external reactivity” means (only) the additional reactivity which has to be added to the reactivity control by the automatic rod movements. This explains why the external reactivity as target function of the rod controller should be zero in a usual reactor control transient case.

Therefore, the input the user has to provide for the rod controller to become active is a time ramp definition with a target value for the desired external reactivity (the net reactivity not balanced by the rods). The controller manages then all the required rod movements up and down during this ramp completely autonomously, following the reactivity changes caused by temperature, Xenon redistribution or other reactivity changes caused by additional TINTE control commands specified by the user.

The controller starts the reactivity control ramp from the current external reactivity (which will be almost zero if the total power ramp is started at about the same time). The usual five input items of TINTE ‘s transient control input have the following meaning for the reactivity control ramps (Table 34):

Table 34: Automatic Reactivity Control Ramp Parameters

Parameter	Description
ZEITS	The point in time where the reactivity ramp and the automated rod control movement starts (in seconds), counted from the beginning of the TINTE transient case. (For special actions with ZEITS = 0 or ZEITS < 0 refer to the general TINTE input description part, section 8.1, table 20).
ZEITZ	The point in time where the reactivity ramp has finished (in seconds). Please note, that although the reactivity *ramp* ends here, the controller will continue with the control moving the rods up and down to keep the final reactivity value (WERTZ) constant forever or until explicitly stopped (by starting a rod *motion ramp* (IART = 4) or a new reactivity control ramp or by terminating the total power control or by switching off the ROMO option). This is quite similar to the function of the other three global nuclear ramps of TINTE, for example the total power is kept constant at the final total power ramp value after this ramp has reached its end point.
WERTZ	The corresponding final external reactivity value at time point ZEITZ (in % or Nile). In many of the practical cases, this will probably be the value zero, as mentioned above.
IORT = 0	Always has a value of **0** here.
IART = 5	Always has a value of **5** here (this number indicates a reactivity control ramp among the global nuclear ramps).

Reactivity control ramps can only be used after the general inputs of rod model definition in 16.5.2 have been done. In addition, they can only be used after a “global nuclear ramp” for the total power has been initiated and run for **at least one time step** (only then ROMO gets information about the external reactivity it needs to adjust its control). Note that this required power “ramp” may also be a constant function with a value equal to the current total power (e.g. a formal power ramp from 1.0 to 1.0).

These input data are not recorded in the TINTE restart files. Therefore, to continue a case via a subsequent run with the ROMO control remaining active over the restart, the ramp input must be read in again for the new calculation (even if the ending time point of the ramp had already passed at the transient time of the restart).

13.6 ROMO OUTPUT DESCRIPTION

13.6.1 Console and Printer File Outputs

- Detailed reporting of the input processing for the new ROMO inputs during the time-dependent parts of the console and printer file outputs.
- Addition of a column “RODP (CM)”, the rod position or insertion length in centimetre, in the time monitor part of the console and printer file outputs, whenever useful (i.e. during transient periods when the rod insertion changes).
- Information about the ROMO controller performance are obtained by inspecting the column “REACT (MNILE)” in the time monitor part of the console and printer file outputs (the column is present in this output only during transient times when using the total reactor power boundary condition). This value is the external reactivity in mNile calculated by TINTE. During the rod control by ROMO it is the residual reactivity which is not compensated for by the automatic controller (due to the user defined reactivity dead band width for the controlling, mainly, or because of insufficient control margins available to the controller, sometimes).
- Warnings in the time monitor part of the console and printer file outputs if the movements of rods reaches the limits set by the defined rod motion model, while executing a user defined rod motion ramp or while fulfilling the reactivity control requirements through automatic rod movements .
- A table called “ROD-POS (M)”, that gives the rod position in meter as function of time, in the printer file part “Transient History” (Transienten-Verlauf) at the end of a case when terminating a TINTE transient case via the transient command “-3”.

13.6.2 Plot Data File for the Scalar Time Functions (the .ptr file)

Scalar data is saved for graphical processing in the .ptr binary data file of the TINTE output. This unformatted data file currently contains 45 time functions with one function value for each temperature time step, and with the implementation of ROMO into TINTE, a 46-th function has been added: the current rod insertion depth in meter (“Stabtiefe/m”).

Note that with the current implementation, the controller calculates the rod positions for all the nuclear time steps, which are often smaller than the temperature time steps. The resulting rod insertion and external reactivity time functions are listed in the console and print output, but not as precisely in the .ptr output file since all values are recorded there only at the end of temperature time steps. This has the drawback that plotting the rod depth and the external reactivity time functions from the ptr file data does not show all the finest details of the rod control. (To overcome this, the user could try to adjust the temperature time step limits close to the nuclear time step sizes, by using the special control commands –5 to –8 of TINTE, if the plot data output for the exact reactivity and rod position functions is desired in some special case).

The auxiliary plot interface routines of TINTE (batch script “TPInt” and FORTRAN program ptrplot.exe, both yet undocumented, but easily useable due to a self-explaining interactive dialog guidance for all inputs) have also been extended to support the new ROMO output function “Stabtiefe”, too, when doing the graphical data processing of the .ptr file. The external reactivity (“RHO-E/MNILE”) was already present on the .ptr file and in the TPInt processing before the ROMO feature was added.

13.7 EXAMPLES

The following 3 sample case input files are intended for the introduction into some typical ROMO applications. The ROMO supplement /50/ lists a total of 5 examples, and the user is urged to work through the detail.

13.7.1 Rod Movements

In the "NZ*" input block of the .tn3 file, 17 overlay regions with region numbers -1 to -17 are defined which serve as a vertical rod bank moving area. The overlay composition input lines look as follows:

```
 -1   145   1.    141
 -2   145   1.    142
 -3   145   1.    143
 -4   145   1.    143
 -5   145   1.    143
 -6   145   0.9   143
 -7   145   0.    143
 -8   145   0.    143
 -9   146   0.    143
-10   146   0.    143
-11   147   0.    143
-12   147   0.    143
-13   148   0.    143
-14   148   0.    143
-15   149   0.    143
-16   150   0.    143
-17   150   0.    143
```

This indicates that the rod bank is fully inserted in regions -1 to -5 (relative concentration factor of 1) and partly inserted in region -6 (with 90 % of absorber poison of the region -6 when rods are fully inserted there), and *capable* of moving between region -1 to region -17. The cross-section composition numbers of the rod absorber material are 141 to 143, which must be listed as the last number in the input line. Note that in the examples below, the *actual* rod model defined by the user for these runs does only comprise the regions –2 to –14, i.e.: a smaller set of active rod regions and thus a smaller rod moving area.

The 3 examples given here demonstrate the application of the first feature of ROMO, i.e. the conducting of a virtual linear rod movement by using the new global TINTE ramp type 4. The transient cases will be started from the stationary full power operation state of the core which is computed first. The time control is then executed according to the following .tn1 files, which contain all the specific ROMO input lines.

Case TWCR-ROMO

This example is a TWCR (total withdrawal of the control rods) transient simulation with "explicitly" withdrawing the rods linearly in time by the aid of a ROMO rod motion ramp (in contrast to performing only a k-effective ramp with a fictitious global poison). These are the lines of the TINTE .tn1 input file for this transient computation:

```
TT* HTR 400 MW, TWCR slow (1cm/s) simulation, w ROMO rod movement.   16.9.02 7La

# In the following ROMO model input, 13 active rod regions are defined,
# with given heights and with "absorber blackness values" of 0.5 to 0.7.
# (Note: the input data "-3 -4 .... -14" can be replaced by zeros or omitted.)

# Col.  10    .   20    .   30    .   40    .   50    .   60    .   70    .   80

   -11                                  ROMO Input: def. of the rod motion model
ROMO      13
    -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14
   90.   8*51.   4*102
 3*0.7   5*0.6   5*0.5

    -1                                  Steady-state full power operation

   300                                  5 min stationary operation, before ramp
   300    -2.2                              At 5' +   0 s:  p2d & prt outputs

# Start of the TWCR transient

   300                                  ROMO rod motion ramp starts at 0:05 h
     0   590.7  +0.0   0   4            Rod insertion: 290.7 -> 0 cm in 290.7 s
   301                                      + 1 s: start step for fast transient

     0    -5    30                        Max. temperature time step size

   590.7  -2.4                            At 5' + 290.7s: rtn & p2d outputs
   620    -2                              At 5' + 320 s:  rtn/p2d/prt outputs
   680    -2.4                            At 5' + 380 s:  rtn & p2d outputs
   900    -2                              At 5' + 600 s:  rtn/p2d/prt outputs
  1200    -2.4                            At 5' + 900 s:  rtn & p2d outputs
  2100    -2.4                            At 5' +1800 s:  rtn & p2d outputs

    -3                                  END of transient simul; history printing
```

This transient case starts from the equilibrium operation as mentioned above, and the rod withdrawal with a constant speed (of the virtual motion modelled using TINTE) of about 1 cm/s is executed by means of a single rod motion ramp. In a previous run, the virtual rod insertion depth of the normal operation had been calculated by ROMO as 290.7 cm, according to the general ROMO model input shown in the .tn1 file and the equivalent rod part length which resulted from the relative poison concentration 0.9 in region -6 (see overlay composition definitions above). The computation ends after half an hour of effective TWCR reactor transient time.

Case RM3-3093-blk06 ("S-curves")

This example is a study to elaborate the rod bank reactivity worth at various insertion positions for the sample design (yielding the so called "S-curve"), by applying three consecutive "linear" rod movements and counterbalancing their effects by a virtual global external reactivity addition. The insertion ramps from 2.907 m (normal pos.) to 0 m (lower edge of top reflector), then to about 9 m depth and back to 2.907 m are easily directed by means of three ROMO rod movement ramp inputs. These are the lines of the TINTE input file .tn1 for the transient computation:

```
TT* HTR 400 MW, 3 Rod Moves "3093" + rho-e, P=100%, v=1cm/s, S=0.6.  26.9.02 7La

# Col.  10    .   20    .   30    .   40    .   50    .   60    .   70    .   80

# In the following ROMO model input, 13 active rod regions are defined,
# with given heights and with "absorber blackness value" of 0.6 for each region.
# (Note: the input data "-4 -5 .... -14" were defaulted on the line after "ROMO".)

   -11                                  ROMO Input: def. of rod motion model
ROMO      13
    -2    -3
   90.   8*51.   4*102
 13*0.60

    -1                                  Steady-state full power operation comp.

   300                                  5 min stationary operation, before ramp
   300    -2.2                              At 5 +  0 min:  p2d & prt outputs

# Start of the transient part.
# A constant total power "ramp" of arbitrary time period defines an 100 % power
# output as the boundary condition for the external reactivity to be computed
# during the whole transient time period of the case.

   300   305     1.0   0   2            Total power "ramp": 100 -> 100 % power
                                        - Aim: continued rho-external adjusting!

# Rod motion no. 1:  Withdrawal from normal op. position to top reflector edge

   309.3                                ROMO rod motion ramp start: 0:05:09.3 h
     0   600.   +0.0   0   4            Rod insertion: 290.7 -> 0 cm in 290.7 s
   311.3                                    + 2 s: start step for rod transient

     0    -5    15                          Max. temperature time step size

   600    -2.4                              At 5 +  5 min:  rtn & p2d outputs

# Pause of 5 min  (Dynamic Xenon and temperature feedback continuing)

   900    -2.4                              At 5 + 10 min:  rtn & p2d outputs

# Rod motion no. 2:  Insertion down to 906 cm depth from reflector edge

   900                                  ROMO rod motion ramp start: 0:15:00 h
     0  1806.  906.    0   4            Rod insertion: 0 -> 906 cm in 906 s
   902                                      + 2 s: start step for rod transient

  1810    -2.4                              At 5 +~25 min:  rtn & p2d outputs

# Pause of about 5 min  (Dynamic Xenon and temperature feedback continuing)

  2100    -2.4                              At 5 + 30 min:  rtn & p2d outputs

# Rod motion no. 3:  Withdrawal from 906 cm depth to normal operating position


2100                                  ROMO rod motion ramp start: 0:35:00 h
     0  2715.3 290.7   0   4            Rod insertion: 906 -> 290.7 in 615.3 s
  2102                                      + 2 s: start step for rod transient

  2720    -2.4                              At 5 +~35 min:  rtn & p2d outputs

 -3                                   END of transient simul; history printing
```

13.7.2 Automated Rod Control

This example demonstrate the application of the second feature of ROMO, i.e. the automatic control of the reactor's reactivity by means of suitable rod bank movements up and down and for any desired time span, by using the new global TINTE ramp type 5. The example is a load following transient with a 100 to 40 % reactor power output change within 6 min and the computation of the subsequent Xenon transient up to 72 h after the load change, with both periods completely controlled by only the automatic controller implemented in ROMO using autonomously the appropriate movements of the rod bank.

The sample input shown here also provides TINTE instructions for generating of enough 2-d array outputs to allow afterwards a detailed graphical evaluation of the space-time behaviour of the major physical core functions (like neutron flux, power density, fuel and moderator temperatures, iodine and xenon concentrations), in order to study possible power redistribution effects due to the xenon dynamics.

```
TT* LF40-RomoRC: HTR 302, LF 100-40% w rod ctl 72h;  P,MF,p: 6 min;  24.5.02 7La

# In the following ROMO model input, 11 active rod regions are defined,
# with given heights and with "absorber blackness values" of 0.5 to 0.7.
# The optional controller parameters are left at default values, e.g.: RTBB=0.05
# (Note: input data "-3 -4 ... -12" are defaulted from zeros on line after "ROMO")

# Col.  10    .   20    .   30    .   40    .   50    .   60    .   70    .   80

   -11                                  ROMO Input: def. of the rod motion model
ROMO      11     9*0               [nrrg  rspeed shrircmi rtbb dkdrd itst(1:5)]
    -2  10*0                                                          [lrrg]
  73.5 10*82.                                                         [rlen]
 3*0.7   4*0.6   4*0.5                                              [rblkns]

    -1                                  Steady-state full power operation

   300                                  5 min stationary operation, before ramp
   300    -2.2                              At 5 +  0 min:  p2d & prt outputs

# Start of the load follow transient

   300                                  Load change ramps start at 0:05 h
     0   660     0.4   0   2            Total power ramp:   100 -> 40%, in 6 min
     0   660    57.52 54   3            Mass flow ramp:   143.8 -> 57 kg/s = 40%
     0   660    45.93 55   5            Press. balance rmp 83.5 -> 46 bar  =~55%
   302                                      + 2 s: start step for LF transient

# A constant reactivity "ramp" of arbitrary time period (!) defines an external
# reactivity of 0 as the target function for the ROMO rod bank controller
# during the whole subsequent transient time period of the case (ramp type: 5).
# Note that we have to start the controller at least one time step after the
# beginning of the total power ramp!  (300 -> 302)

   302                                  ROMO reactivity control start: 0:05:02 h
     0   312     0.00  0   5            rho_external: ~0 -> 0 Nile in 10 & >10 s
   304                                      + 2 s: start step for rod control

   660    -2.4                              At 5 +  6 min:  rtn & p2d outputs


1200                                  At 0:20 h:
     0    -6    10                       Min. temperature time step size
     0    -8    10                       Min. nuclear time step size  (!)

  1800    -2.4                           At 5 + 25 min:  rtn & p2d outputs

  3600    -2.4                           At 1 h:  rtn & p2d outputs
```

14. APPENDIX F: MODELLING AIR AND WATER INGRESS TRANSIENTS

TINTE is capable of calculating the chemical reactions and graphite corrosion due to ingress of air or water into the reactor. To turn on the chemical calculations it is necessary to define flags 6 and 7 (REAG and REAC) as T in the third card of the TN3 input file (Table 6).

It is recommended (for practical reasons) to modify the 1-D flow network that is described in Table 7 (see section 5.5) for the calculations that involve chemical reactions. The following is an ordinary 1-D network example that works well for the TINTE calculations that does not involve chemical reactions:

```
260  -3  -2  0   2800.0    300.0   -200.0  0.0  0.0  0.0
```

In the above example -2 and -3 are the coolant inlet and outlet locations in the TZ block (see section 5.5). These components are defined in the TM (see section 5.6) block as follows:

```
COMP2     2  4   504.5     0  2.868e-02  4  1.0         2  0.6740
                500.0        0.0010       33.5   0     0     0   0  192.7
COMP3     3  4   848.8     0  1.862e-02  4  1.0         1  0.6667
                896.7        0.0010       41.4   0     0      0  -87.9
```

In the example above coolant inlet mass flow rate is defined with Component 2 (192.7 kg/s). The outlet pressure is defined with Component 3 (-87.9 bar). Hence, for the example above Component 2 is the gas source and Component 3 is the gas sink.

Component 260 is not reflected in the mesh grid (TZ block). It is a dummy component that is used to define the flow network. It is defined in the TM block as follows:

```
COMP260 260 13   650.0
                 698.4        0.9990        1.0      1.0e+20
```

For the same TZ block, the 1-D network is modified for the corrosion calculations as presented below:

```
260  261  262  0  300. 75.  0.
263  261  -2  0  7853  200. -200.
264  -3  262  0  7853. 500. 0.
```

In the 1-D network above components -2 and -3 are again the coolant inlet and outlet locations in the TZ block. Definitions of the flow network components in the TM block are as follows:

```
COMP2     2  4   504.5     0  2.868e-02  4  1.0         2  0.6740
                500.0        0.0010       33.5   0     0      0
COMP3     3  4   848.8     0  1.862e-02  4  1.0         1  0.6667
                896.7        0.0010       41.4   0     0      0
DUMMY1  260 13   600.
                 695.       .0001   100.     1d20
SOURCE  261  4   504.1     0  2.868e-02  4  1.0         2   .6740
                500.0         .0010       33.5  0     0      0   0  192.7
SINK    262  4   847.1     0  1.857e-02  4  1.0         1   .6667
                896.7         .0010       41.4  0     0      0  -87.9
DUMMY2  263 13   300.
                 490.   .0001 100.  0d20
DUMMY3  264 13   300.
                 900.   .01   100.  0d20
```

In this 1-D network the gas source is Component 261 and the sink is Component 262. Component 263 is used to link the source to the mesh grid of TZ block (to component 2). Component 264 is used to link the sink to the mesh grid of TZ block (to component 3). Component 260 is used to define the 1-D network between inlet and the outlet plenums.

Air or water ingress can be initiated anytime during the transient. The following TN1 input file is an example for the air ingress which takes place after a Depressurized Loss Of Forced Cooling (DLOFC) event which is followed by a reactor scram. The air ingress takes place after the scram.

```
 1.   9.  0.000  261  3              Inlet Mass flow
 1.   9.  0.987  262  5              Outlet Pressure

 10.                                 Scram
 11.  12.  1.0  -7  1                     Mat#7  First overlay = 100%
 11.  12.  1.0  -8  1                     Mat#8  First overlay = 100%
 11.  12.  1.0  -9  1                     Mat#9  First overlay = 100%
 11.  12.  1.0  -10 1                     Mat#10 First overlay = 100%
 11.  12.  1.0  -11 1                     Mat#11 First overlay = 100%
 11.  12.  1.0  -12 1                     Mat#12 First overlay = 100%
 11.  12.  1.0  -13 1                     Mat#13 First overlay = 100%
 11.  12.  1.0  -14 1                     Mat#14 First overlay = 100%
 11.  12.  1.0  -15 1                     Mat#15 First overlay = 100%
 11.  12.  1.0  -16 1                     Mat#16 First overlay = 100%
 11.  12.  1.0  -17 1                     Mat#17 First overlay = 100%
 11.  12.  1.0  -18 1                     Mat#18 First overlay = 100%
 11.  12.  1.0  -19 1                     Mat#19 First overlay = 100%
 11.  12.  1.0  -20 1                     Mat#20 First overlay = 100%
 11.  12.  1.0  -21 1                     Mat#21 First overlay = 100%
 11.  12.  1.0  -22 1                     Mat#22 First overlay = 100%
 11.  12.  1.0  -23 1                     Mat#23 First overlay = 100%
 11.  12.  1.0  -24 1                     Mat#24 First overlay = 100%
 11.  12.  1.0  -25 1                     Mat#25 First overlay = 100%
 11.  12.  1.0  -26 1                     Mat#26 First overlay = 100%
 11.  12.  1.0  -27 1                     Mat#27 First overlay = 100%


20.  25.  0 262 12                  new Source Mix at 262: outlet break
0.0  0.78 0.20 0 0 0.02 0           air

20.  25.  0 261 12                  new Source Mix at 261
0.0  0.78 0.20 0 0 0.02 0           air

20.  25. 0.0 261 2                  equalize pressure between 261 and 262

20.  25.  27. 262 1                 Ramp air inlet temp. at old outlet
                                    (now inlet) down to 27°C

30.  120.  208.0e-3 262 3           208 g/s air ingress at old outlet
                                    (now inlet)
```

The above TN1 file is consistent with the previous TN3 example. In this transient the DLOFC takes place between t = 1 s and t = 9 s. The reactor scram takes place over 1 s, from t = 11 s to t = 12 s.

The new gas mixture is defined at both source and sink between t = 20 s and t = 25 s. The partial pressures are used to define air's gas combination. The gas combination of air for this example is as follows: 0.78% N_2, 0.20% O_2 and 0.02% H_2O. In the same time interval the air inlet temperature is defined as 27°C and the pressure between the sink and source is equalized.

This is an outlet pipe break transient and air enters to the reactor from the outlet plenum. The mass flow rate of the air is specified as 208 g/s. The air ingress from the old sink component (Comp262) takes place between t = 30 s and t = 120 s.

Important Corrosion Modelling Considerations:

In TINTE, corrosion calculations are done in a mesh cell ***only if a gas and graphite*** are present in that mesh cell. If the graphite fraction of an outer wall of a mesh cell is zero, then TINTE does not recognize it for corrosion calculations.

It implies for the central cooling channel, in the centre reflector, that its graphite **wall will not be corroded**, if the cooling channel diameter is set equal to the real channel diameter and is only defined with gas contents. An oxygen concentration in the channel results only from this incorrect modelling! If you want to corrode the walls, you must enlarge the mesh dimension a little bit and homogenise the graphite part thus included into the channel mesh contents. Don't forget correcting the solid fraction FFK from nearly zero to the appropriated value and replace the solid material number IRHOC from zero to 201 (graphite) because the heat capacity rule at this item marks the material as graphite for corrosion. The amount of this enlargement depends on how much graphite you like to corrode or how long you like to investigate the corrosion. (If solid parts become small and are falling downward, corrosion calculation makes no longer sense as TINTE cannot describe this type of material movement!) But don't choose it too small, because if all the homogenized graphite is consumed, the corrosion will stop in that mesh - and oxygen is again passing through it. Also, if the homogenization is done with a rather small amount of graphite, it should not be necessary to regard it in the nuclear cross sections. Although this is possible, it would become complicated, as the channel is modelled as a cavity there for neutronic calculations.

The same holds true for the gap between the reflector and the core barrel. Corrosion can only be calculated, if some graphite is homogenized into the gap, e.g.: by changing the mesh radius of the graphite-gap boundary. (As the gas flow is very small there, during corrosion calculations, a 1 mm mesh or a few mm will be enough). For the non-central cooling channels such a procedure is not necessary, as the needed gas - graphite mixture results from homogenising azimuthally in those mesh cells which contain the cooling channels.

Another problem is the upper cavity above the core. For corrosion investigations we often did homogenize up to 1 cm of the top reflector into the cavity. But this is not necessary here, as usually no oxygen reaches this area if all the paths from the ingress source to this region (especially the central channel and the gap at the core barrel) are modelled properly.

15. APPENDIX G: CASE STUDY EXAMPLES

This Appendix contains a few examples of special TINTE cases. Note that some sections of the input have been reduced to save space.

15.1 PURE THERMAL HYDRAULIC MODELING

The partial .tn3 input deck shown below is an example of a pure 2D heat conduction problem, without any convective heat transfer. Note the use of the ST block input flags, and the lack of any neutronics inputs. This example also does not contain a heat source or sink, or any gas interactions.

```
ST* Egg cooker 2d heat transfer problem. R=12 cm, H=24 cm, conductivity=1.667 W/cmK,
rho_cp= 83.83 J/cm3K
FFFTTFFT   0.0    0    0.0      2.0     5.0       .5
.00002        .0001   .01      .001      .2
TTTTTTTT   2


GM%
     0.0 0 1     1.0 0 1     2.0 0 1    3.0  0 1    4.0  0 1    5.0  0 1
     6.0 0 1     7.0 0 1     8.0 0 1    9.0  0 1   10.0  0 1   11.0  0 1
    12.0 0 1    13.0 1 1    15.0
     0.0 1 1     1.0 1 1     2.0 0 1    3.0  0 1    4.0  0 1    5.0  0 1
     6.0 0 1     7.0 0 1     8.0 0 1    9.0  0 1   10.0  0 1   11.0  0 1
    12.0 0 1    13.0 0 1    14.0 0 1   15.0  0 1   16.0  0 1   17.0  0 1
    18.0 0 1    19.0 0 1    20.0 0 1   21.0  0 1   22.0  0 1   23.0  0 1
    24.0 0 1    25.0


TZ*
1   1   1   1   1   1   1   1   1   1   1   1   4    4
1   1   1   1   1   1   1   1   1   1   1   1   4    4
1   1   1   1   1   1   1   1   1   1   1   1   4    4
1   1   1   1   1   1   1   1   1   1   1   1   4    4
1   1   1   1   1   1   1   1   1   1   1   1   4    4
1   1   1   1   1   1   1   1   1   1   1   1   4    4
4   4   4   4   4   4   4   4   4   4   4   4   4    4


TM*
Egg       1  0   500.0     0   83.83    0    1.667
Boundary  4  0  -1500.0    0    0.0     0    0.0   0   0   0.0   0.0   1.0E+08
```

A less dramatic example of the same problem is shown below, now with a “dummy” gas source and sink added.

```
ST* Egg cooker 2d heat transfer problem. R=12 cm, H=24 cm, conductivity=1.667 W/cmK,
rho_cp= 83.83 J/cm3K

FFFTTTTT   0.0    0    0.0      2.0     5.0       .5

.00002        .0001   .01      .001      .2

TTTTTTTT   2


GM%
     0.0 0 1     1.0 0 1     2.0 0 1    3.0  0 1    4.0  0 1    5.0  0 1
     6.0 0 1     7.0 0 1     8.0 0 1    9.0  0 1   10.0  0 1   11.0  0 1
    12.0 0 1    13.0 1 1    15.0
     0.0 1 1     1.0 1 1     2.0 0 1    3.0  0 1    4.0  0 1    5.0  0 1
     6.0 0 1     7.0 0 1     8.0 0 1    9.0  0 1   10.0  0 1   11.0  0 1
    12.0 0 1    13.0 0 1    14.0 0 1   15.0  0 1   16.0  0 1   17.0  0 1
    18.0 0 1    19.0 0 1    20.0 0 1   21.0  0 1   22.0  0 1   23.0  0 1
    24.0 0 1    25.0


TZ*
1   1   1   1   1   1   1   1   1   1   1   1   4   -2
1   1   1   1   1   1   1   1   1   1   1   1   4   -3
1   1   1   1   1   1   1   1   1   1   1   1   4    4
1   1   1   1   1   1   1   1   1   1   1   1   4    4
1   1   1   1   1   1   1   1   1   1   1   1   4    4
1   1   1   1   1   1   1   1   1   1   1   1   4    4
4   4   4   4   4   4   4   4   4   4   4   4   4    4
5   -3  -2  0   1.0   100.0   -100.0


TM*
Egg       1  0   500.0     0   83.83    0    1.667
Sink     -2  3   500.0     0    1.0     0    1.0   0   0   0
                 500.0          0.61    6.0  0.0   0   0   10.0
Source   -3  3   500.0     0    1.0     0    1.0   0   0   0
                 500.0          0.61    6.0  0.0   0   0   0   1.0E-20
Boundary  4  0  -1500.0    0    0.0     0    0.0   0   0   0.0   0.0   1.0E+08
Dummy     5 13   500.0     11   0.01    11   0.01
                 500.0          0.01    1.0  1.0E+20


1  0
```

15.2 PURE NEUTRONICS MODELING

The partial Dodd benchmark .tn3 input deck shown below is an example of a pure neutronics problem, without any temperature or gas interactions. Note that a “dummy” TZ block is still needed, although no temperature calculations are done.

```
ST* Dodd benchmark
TTTTTTFF   1.0    0    0.0      2.0     5.0       .05
1.e-8          1.e-7    .1      .1      .1
TTTTTFF    1

GM%
     0.0 2 9  13.089 2 9  26.179 2 9  39.268 2 9  52.358 2 9  65.447 2 9
 157.074 2 9 170.163 2 9 183.253 2 9 196.342 2 9 209.432 2 9  222.521
     0.0 2 9  18.745 2 9  37.490 2 9  56.236 2 9  74.981 2 9  93.726 2 9
 112.471 2 9 131.216 2 9 149.962 2 9 168.707 2 9 187.452 2 9 206.197 2 9
 337.414 2 9 356.159 2 9 374.904 2 9 393.649 2 9 412.394 2 9 431.140 2 9
 449.885 2 9 468.630 2 9 487.375 3 9 506.120 3 9 524.866

TZ*
1  1  1   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1

TM*
DUMMY      1  0    20.  7  1.0     3  1.0     0   0.    0.0   0.0

NZ*
1  1  1   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
2  2  2   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
-1 -1 -1  3  3  3  3  3  3  4  4  4  5  5  5  9  9  9
-3 -3 -3 -3 -3 -3 -3 -3 -3  4  4  4  5  5  5  9  9  9
-3 -3 -3 -3 -3 -3 -3 -3 -3  4  4  4  5  5  5  9  9  9
2  2  2   2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
1  1  1   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
-1 3  1.0   3
-2 6  1.0   6
-3 3  1.0   3

18*1 0
28*1 0
```

The structure of the NQ block in the .tn4 file of this problem (shown below) is also of importance here: just the reference set of macroscopic cross section data is retained, since no polynomial dependant XS data was needed. The open line (zero values) and zero fission

fractions should also be noted. A full “dummy” set of decay heat values (all zero) is also needed since TINTE always require decay heat input for nuclear calculations.

```
NQ%: Dodd benchmark
  1          1 3.1199E-01 2.0000E-03 0.0000E+00 0.0000E+00 0.0000E+00 2.6000E-02
  1          2 1.0400E-00 3.3000E-03 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

   0.000000.000000.000000.000000.000000.000000.000000.000000.000000.000000.00000
  2          1 2.4701-01  1.0000E-05 0.0000E+00 0.0000E+00 0.0000E+00 1.2000E-02
  2          2 3.8300E-01 1.9000E-02 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

1.0        9.         1.000       1.
 1.0       -1.       -0.1             10       3E5     3.5E6
   1  1 0.00000E+00 0.00000E+000.000000.000000.000000.000000.000000.000000.00000
   2  1 0.00000E+00 0.00000E+000.000000.000000.000000.000000.000000.000000.00000
```

REFERENCES

	Document Title	Preparer/Author	Document Number	Revision or Date of Issue
[1]	Das Zweidimensionale Reaktordynamikprogramm TINTE. Teil 1: Grundlagen und Lösungsverfahren.	H Gerwin	Jül-2167	November 1987
[2]	Das Zweidimensionale Reaktordynamikprogramm TINTE. Teil 2: Anwendungbeispiele	H Gerwin	Jül-2266	February 1989
[3]	Planning and Performance of Dynamic Experiments with the AVR Power Plant.	M. Wimmers and P. Pohl	Nucl.Sci. Eng. 97, 53-57	1987
[4]	Analysis of Reactivity and Temperature Transient Experiments at the AVR High- Temperature Reactor.	W. Scherer and H. Gerwin, T. Kindt and W. Patscher	Nucl. Sci. Eng., 97, 58-63	1987
[5]	Computational Simulation of Transient Processes in the Primary Circuit of a Pebble Bed HTGR.	H. Gerwin and W. Scherer	International Reactor Physics Conference, Jackson Hole, Wyoming, 18-22 September.	1988, p. IV-433
[6]	Zur Validierung des Reaktordynamikprogrammsystems TINTE: Graphit-Korrosion mit Luft, Vergleichsrechnungen zum Experiment VELUNA.	H. Gerwin and W. Scherer	Interner Bericht KFA-ISR-IB-8/92	June 1992
[7]	Berechnung der Nachzerfallsleistung der Kernbrennstoffe von Hochtemperaturreaktoren mit kugelförmigen Brennelementen. Beiblatt 1 zu DIN 25485 Dokumentation und Erläuterungen.	Unknown	DIN 25485	May 1990
[8]	Fission-Energy Release for 16 Fissioning Nuclides.	R. Sher:	EPRI-NP-1771	Unknown
[9]	ENDF-102 Data Formats and Procedures for the Evaluated Nuclear Data File.	D. Garber, C. Dunford, and S.Pearlstein	ENDF, BNL-CS-50496	October 1975
[10]	Review of delayed neutron branching ratios	R. J. Thuttle	INDC(NDS)-107/G+SPECIAL, p 29	1979
[11]	Delayed Neutrons Form Fission. A compilation and evaluation of experimental data.	L. Tomlinson	AERE-6993	1972
[12]	Neutron Cross Sections Volume I, Resonance Parameters.	S. F. Mughabghab and D. I. Garber	BNL 325, Third Edition , Volume 1	June 1973
[13]	Karlsruher Nuklidkarte.	W. Seelmann-Eggebert, G. Pfennig, H. Münzel and H. Klewe-Nebenius	6. Auflage, Kernforschungszentrum Karlsruhe GmbH	1995

Document Title	Preparer/Author	Document Number	Revision or Date of Issue
[14] A Leakage Iterative Method For Solving The Three Dimensional Diffusion Equation	Y. Naito, M.Mackave, K.Shibuya	Nucl. Sci. Eng., 58, 1975	1975
[15] Räumliche Xenon-Schwingungen in Hochtemperaturreaktoren	A. Lauer	Jül-850	May 1972
[16] Zur Sicherheitskonzeption Des Hochtemperaturreaktors Mit Natürlicher Wärmeableitung Aus Dem Kern Im Störfall.	K. Petersen	Jül-1872	October 1983
[17] Berechnungsmethoden Und Analysen Zum Dynamischen Verhalten Von Kraftwerksanlagen Mit Hochtemperaturreaktor	J. Banaschek	Jül-1841	August 1983
[18] Thermo-Fluid-Dynamik, Korrosion Und Reaktivitätseffekte Bei Wasser- Und Luftein- Trittsstörfällen In Hochtemperaturreaktoren.	M. Sokcic-Kostic	Jül-2437	February 1991
[19] (a) Wärmeleitfähigkeit Und Spez. El. Widerstand Neutronenbestrahlter Graphite In Abhängigkeit Von Der Temperatur. (b) Ein Verfahren Zur Bestimmung Der Wärmeleitfähigkeit Von Neutronenbestrahlten Graphiten Bei Tempereturen Zwischen 50 Und 1000 °C	(a) L. Binkele, O. Hartleib, W. Biergans (b) L. Binkele	(a) Interner Bericht der KFA IRW-IB-13/75 (b) Jül-1096-RW.	1975 1974
[20] Ermittlung von Stoffdaten der Case O_2, N_2, CO,CO2, He, H_2O, H_2, CH_4, CH_3OH und CH_5O H und der Mischungen aus diesen Gasen.	N. Kostka, K. Hammeke	Jül-2112	1987
[21] Sicherheitstechnische Regel des KTA : Auslegung Der Reaktorkerne Von Gasgekühlten Hochtemperaturreaktoren. Teil 3: Reibungsdruckverlust in Kugelhaufen.	Unknown	KTA 3102.3	Unknown
[22] Sicherheitstechnische Regel des KTA : Auslegung Der Reaktorkerne Von Gasgekühlten Hochtemperaturreaktoren. Teil 2: Wärmeübertragung im Kugelhaufen.	Unknown	KTA 3102.2	Unknown
[23] Vdi-Wärmeatlas: Berechnungsblätter fur den Wärmeübergang. Vierte neubearbeitete und erweiterte Auflage.	Unknown	Unknown	1984
[24] (a) Treatment of the Upper Cavity in a Pebble-Bed High-Temperature Gas-Cooled Reactor by Diffusion Theory. (b) Ein Diffusionstheoretisches Simulationsverfahren Zur Behandlung Des Oberen Hohlraumes Im Kugelhaufen-HTR.	H. Gerwin, W. Scherer	(a) Nucl. Sci. Eng. 97, 9-19. (b) Jül-1599.	1987 1979

Document Title	Preparer/Author	Document Number	Revision or Date of Issue
[25] (a) MUPO. An IBM-7090 Programme to Calculate Neutron Spectra and Multigroup Constants. (b) Rechenprogramm MUPO-4-D	(a) J .Schlosser (b) H. J. Neef	(a) Dragon-Report 172. (b) MUPO-4-D KFA-IRE-70-15.	1963 1970
[26] Preparation of the TINTE Nuclear Data Base.	H. Gerwin, W. Scherer	FZJ Research Centre Jülich, Report in Preparation.	
[27] (a) V.S.O.P.-Computer Code System for Reactor Physics and Fuel Cycle Simulation. (b) V.S.O.P.('94) Computer Code System for Reactor Physics and Fuel Cycle Simulation Input Manual and Comments. (c) V.S.O.P.(97) Computer Code System for Reactor Physics and Fuel Cycle Simulation Input Manual and Comments. (d) V.S.O.P.(99) for WINDOWS and UNIX Computer Code System for Reactor Physics and Fuel Cycle Simulation: Input Manual and Comments.	(a) E. Teuchert, U. Hansen, K. A. Haas (b) E. Teuchert et al (c) H. J. Rütten et al (d) H. J. Rütten et al.	(a) Jül-1649 (b) Jül-2897 (c) Jül-3522 (d) Jül-3820.	1980 1994 1998 2000
[28] HTR-2000: Programmcode zur rechnerischen Betriebsbegleitung von HTR 's.	F. Thomas	Jül-2261	January 1989
[29] AVR Versuchskraftwerk mit Kugelhaufenreaktor in Jülich.	Unknown	Atomwirtschaft p 218-271.	1966
[30] Spannbetonbehalter Unter Hypothetischen Störfallbelastlingen: Abschlussbericht Forderungsvorhaben RS 447, Anhangband 1 ZSP 1.	Unknown	KERNFORSCHUNGSANLAGE JÜLICH GmbH, Institut für nukleare Sicherheitsforschung	January 1984
[31] Wärmetranspart Im Inneren Und In Der Randzone Von Kugelschüttungen.	K. Robold	Jül-1796-RW	July 1982
[32] Das Temperaturverhalten Eines HTR Nach Ausfall Der Gesamten Kühlung.	J. Lukaszewisz	Jül-1112-RG	October 1974
[33] Untersuchungen zum Wärmetransport in Kugelschüttungen	H. Barthels, G. Breitbach	KFA-Bericht IB-5-78	June 1978
[34] Wärmeleitfähigkeit von Faserstoffen. Möglichkeiten der Berechnung.	P. Broeckerhoff	Jül-992-RB	1973
(a) Experimentelle Untersuchungen zur Thermohydraulik in Kugelschuttungen im Vergleich mit dem Rechenprogramm Thermix-2D. Die Untersuchung des dispersiven Wiirmetransportes am Beispiel einer Kaltgasstrahne. [35] (b) Die Effektive Wärmeleitfähigkeit In Kugelschüttungen Unter Besonderer Berücksichtigung Des Hochtemperatur- Reaktors	(a) M. Schürenkrämer, H. Barthels (b) H. Barthels, M. Schürenkrämer	(a) Jül-1839-RW. (b) Jül-1893-RW.	1983 1984

	Document Title	Preparer/Author	Document Number	Revision or Date of Issue
[36]	Experimente Zur Selbsttätigen Abfuhr Der Nachwärme Bei Hochtemperatur-Reaktoren Planung, Vorbereitung Und Ergebnisse.	M. Lang	Jül-3012	January 1995
[37]	Experimente zum Wärme- und Stoffübergang infolge Naturkonvektion beim Luftein-bruchsstorfall am Hochtemperaturreaktor.	T. Schaaf	Jül-3620	January 1999
[38]	Zum Bestrahlungsverhalten des Reflektorgaphits im AVR, ARB/ AMT der Firma Bigri Elektrographit GmbH.	W. Delle	Technische Notiz IRW- TN-43/83	April 1983
[39]	Formalismus Zur Berechnug Der Dosis- Und Temperaturabhängigen Wärmeleitfähigkeit Der AVR-Reflektorgraphits. H5-X1 Ber/DR.Waw vom 4.6.1984	Bergerfurt, U. Wawrzik	AVR Aktennotiz H5-X1 Ber/ DR.Waw	June 1984
[40]	Wärmeleitfähigkeit von Reflektorgraphit und Kohlesteinisolierung für den AVR.	Bergerfurt, S. Burger	HTA-5 AVR Aktennotiz Ber/Bur/Ga.	February 1987
[41]	Untersuchungen zum Sicherheitstechnischen Potential des Hochtemperaturreaktors bei Ausfall der NW A.	K. W. Otto	Dissertation an der RWTH-Aachen	February 1978
[42]	Einfiuss der Wärmestrahlung und des Druckes auf den Wärmetransport in nichtdurchstromten Schüttungen.	P. Zehner, E. U. Schlünder	Chem.-Ing. Techn., 44.Jahrg. / Nr. 23.	1972 / Nr. 23
[43]	Cooling of the ISIS Target.	W. J. Curtis	Rutherford Appelton Lab RAL-86-014.	February 1986
[44]	Modelldatensatz fuer den Reflectorgraphit der Hugelhaufenreaktoren	J Budke, G Haag, W Hammer, A Mittenbuehler, MF O'Connor, K Peterson	Jül-1414.	February 1984
[45]	TINTE Model of PBMR with Core Nominal Thermal Power 400MW.	A Matev	PP350-018285-3220	A
[46]	Data and Boundary Conditions to be used in VSOP and TINTE PBMR 300 MW Reactor Models.	F Reitsma	PP360-022028-4590	A
[47]	Determination of Equivalent Cross Sections for Representation of Control Rod Regions in Diffusion Calculations.	W Scherer, HJ Neef	Jül-1311.	1978
[48]	Zur Darstellung von Absorberstaben in Kugelhaufenreaktoren	W Scherer	KFA-IRE-IB-21/80	1980
[49]	The Two-Dimensional Reactor Dynamics Programme TINTE. Part 3: Programme Architecture and User's Manual.	H Gerwin, W Scherer	Forschungszentrum Juelich / ISR	June 2001, Draft

	Document Title	Preparer/Author	Document Number	Revision or Date of Issue
[50]	TINTE User manual Supplement: The Rod Motion Extension “ROMO”.	A Lauer	Forschungszentrum Juelich / ISR-IB-1/2004	February 2004
[51]	TINTE Nuclear Calculation Theory Description Report	I D Clifford	052322	C
[52]	TINTE Software Installation Plan	G Strydom	029870	2
[53]	TINTE Software Installation Report	G Strydom	030440	2
[54]	PBMR Coupled Neutronics/Thermal Hydraulics Transient Benchmark: The PBMR-400 Core Design Benchmark Definition	F Reitsma	NEA/NSC/DOC (2007)	Draft-v07

Jül-4294
April 2009
ISSN 0944-2952