R-99-64

Comp23/Nuctran user's guide

Proper version 1.1.6

Leonardo Romero
Universidad Catolica del Norte, Chile

Andrew Thompson
AEA Technology plc, UK

Luis Moreno, Ivars Neretnieks
Royal Institute of Technology, Sweden

Hans Widén, Anders Boghammar
Kemakta Konsult AB, Sweden

October 1999

SKB

Comp23/Nuctran user's guide

Proper version 1.1.6

Leonardo Romero
Universidad Catolica del Norte, Chile

Andrew Thompson
AEA Technology plc, UK

Luis Moreno, Ivars Neretnieks
Royal Institute of Technology, Sweden

Hans Widén, Anders Boghammar
Kemakta Konsult AB, Sweden

October 1999

Contents

1   Managing a problem with the model NUCTRAN

NUCTRAN is a fast multiple path model that calculates nuclide transport in the near field of a repository as occurring through a network of resistances and capacitances coupled together like an electrical circuit network. The model, which is a very coarsely discretized integrated finite difference model, was devised to be very fast and compact by embedding analytical solutions at sensitive zones. The code allows the user to simultaneously consider a multitude of pathways by which the nuclides are transported by diffusion to the flowing water in fractures surrounding the barrier system.

The nuclide dissolution may be calculated using two approaches, a solubility limit approach or a congruent dissolution approach. The conceptual model used in NUCTRAN can be represented by three bodies as shown in Fig. 1. The bodies are the source, the barrier system, and the sinks. The source is defined as a well-mixed compartment. The barrier system is the physical medium by which the nuclides migrate to reach the sinks located in the surrounding system or outside of the region considered as the barrier system. The sinks, considered as recipients where the water flows, are fully defined by a local equivalent flow rate.
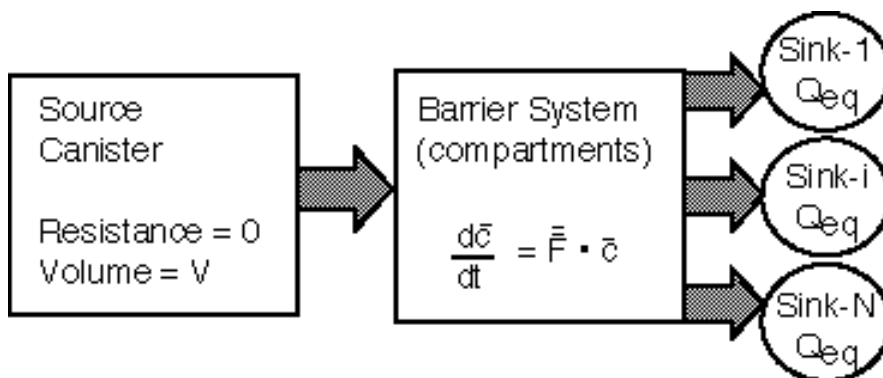

Figure 1   A schematic of the conceptual model used by NUCTRAN

The purpose of this document is to assist the user in managing problems with NUCTRAN. An overview of the theory, numerical method, and the code designed to solve the problem will be presented in the following sections. Finally, an example will be handled in detail.

2 Conceptual model

To describe the conceptual model used in NUCTRAN, we will follow the definition given by Olsson et al., [1]: "a relatively general description or a definition of the way the model is constructed. This should be separated from any specific realization or application of the conceptual model." The concepts that make up the conceptual model are: specification of the processes modelled, geometric framework, specification of the parameters, specification of the assignment of material properties and specification of boundary and initial conditions required by the model.

2.1 Processes modelled

The processes modelled are radioactive decay and ingrowth, diffusion, advection, dissolution/ precipitation and sorption. The formulation of these processes is shown for the case of a single radionuclide, but may easily be extended to a radionuclide chain. The equations presented in this section do not include the term for alpha-radiolytically-induced dissolution that is covered later in the report.

The notation used for the equations described in this report is given in Appendix 1.

2.1.1 Radionuclide transport by diffusion

Radionuclides leaking from a damaged canister spread into the backfill material surrounding the canister and then migrate through different pathways into water-bearing fractures in the rock surrounding the repository. If the backfill and other materials surrounding the canister have a low permeability, water flow is then excluded from these materials and the solute transport will be only by diffusion. The water flow is confined to the fractures in the rock. Some solutes may be sorbed on the materials surrounding the canister along the transport paths and will be retarded. Solutes may also precipitate. Radionuclides with short half-lives may then decay to an insignificant concentration.

NUCTRAN handles chain decay but for illustration purposes we describe only transport of a single nuclide below. The material balance for a single nuclide over a unit volume is described by two equations:

$$K\frac{\partial c}{\partial t} = q_d + \nabla \cdot D_e \nabla c - K\lambda c$$

$$\frac{dm}{dt} = -m\lambda - q_d$$

where D_e is the effective diffusivity, λ is the decay constant, c is the concentration in the pore water, m is the mass of nuclide as solid (as precipitate) in the unit volume, and K is a distribution coefficient. The first equation models the transport of the aqueous phase

(dissolved solid), including sorption in the pores of the material. The second equation models the amount of nuclide as solid. The two equations are coupled by a dissolution term (q_d).

The distribution coefficient K considers the nuclides dissolved in the water and sorbed in the solid, and can be described by:

$$K = \varepsilon_p + \left(1 - \varepsilon_p\right) k_d \rho$$

where ε_p is the porosity, ρ is the density of the solid material and k_d is the nuclide sorption coefficient.

2.1.1.1 Parameters required

The parameters required are as follow:
- Solid density and porosity of the materials.
- Sorption coefficient and diffusivity of each radionuclide in each of the different materials.
- Solubility limit of each radionuclide in each of the different materials.
- Nuclide inventories and half lives.

2.1.2 Radionuclide transport by diffusion and flow

In addition to the above processes, water flow could exist through some zones of the barrier system. The equation for the transport of the solid dissolved, including the sorption in the pores of the material, is modified to:

$$K \frac{\partial c}{\partial t} = q_d + \nabla \cdot D_e \nabla c - u_0 \nabla c - K \lambda c$$

where u_o is the flux (Darcy velocity) in $m^3/(m^2$ year).

2.1.2.1 Parameters required

The parameters required are as follow:
- Solid density and porosity of the materials.
- Sorption coefficient and diffusivity of each radionuclide in each of the different materials.
- Solubility limit of each radionuclide in each of the different materials.
- Nuclide inventories and half lives.
- The groundwater flux.

2.2 Geometric framework

To represent the barrier system through which the species are transported, NUCTRAN makes use of the integrated finite difference method [2] and of the concept of "compartments". The barrier system is discretized into compartments. Average properties are associated over these compartments with representative nodal points within the compartment. From the theoretical point of view the compartments may have any shape, but consist of only one material. The material balance over a compartment connected to some other compartments for a fully dissolved single species is described by:

$$V_i K_i \frac{dc_i}{dt} = \sum_{j \neq i} \left(\frac{AD_e}{d} \right)_{i,j} (c_j - c_i) - V_i K_i \lambda c_i$$

where V_i is the volume of the compartment assuming no advective transport. The left-hand side accounts for the accumulation of the species in the water and the solid by sorption. The right-hand side accounts for diffusive transport from one compartment to the adjacent compartments and the degradation rate of the species by decay.

The elements required to define the compartmentalization are the geometry of the system, dimensions of the system and the type of material. The compartments are defined by their volume, their diffusion length and cross-sectional area used by the diffusion. Conceptually, the model uses a rather straightforward compartmentalization process. This coarse compartmentalization could yield poor or even meaningless numerical results. To avoid this, analytical or semi-analytical solutions are introduced in the model in the zones where a finite difference scheme would require a fine discretization. Some of the approaches used by the model to describe the solute transport in these sensitive zones are shown below.

2.2.1 Approaches used for the model

The approaches developed at present include transport by diffusion into the flowing water, transport of solute through a small contacting area into a large volume compartment and transport of solute into a narrow slit [3]. Other approaches could be included in the code in the future.

2.2.1.1 Transport into flowing water

For compartments in contact with water flowing in fractures in the rock, the diffusive transport is determined by an equivalent flow rate Q_{eq}. This parameter is a fictitious flow rate of water that carries with it a concentration equal to that at the compartment interface. It has been derived by solving the equations for diffusional transport to the passing water by boundary layer theory [4]. This entity is obtained from:

$$Q_{eq} = q_o \cdot W \cdot \overline{\eta} = q_o \cdot W \cdot \sqrt{\frac{4 \cdot D_w \cdot t_w}{\pi}}$$

where D_w is the diffusivity in free water, W is the width of the compartment in contact with water flowing in fractures, fracture zones or damaged zones and $\bar{\eta}$ is the mean thickness of penetration into the water by diffusion from the compartment. The residence time, t_w, is the time that the water is in contact with the compartment. This time is obtained from the flux of water q_o, the flow porosity, and the length of the pathway in contact with water.

2.2.1.2 Transport into a large compartment

Species diffusing out of a small hole into a very large volume of material spread out spherically. Very near the hole, the cross-section is still of the order of the size of the hole. Further away, the cross section increases considerably as the "sphere" grows. Thus, most of the resistance to diffusion is concentrated very near the mouth of the hole. This resistance is calculated by integrating the transport rate equation:

$$N = -2\pi\, r^2\, D_e \frac{dc}{dr},$$

from a small hemisphere into a very large volume, between the limits of the sphere of radius r_{sph} and an outer radius r. Since the species spread over a large volume in the surrounding medium ($r \gg r_{sph}$), the nuclide transport rate simplifies to $N = 2\,\pi\, r_{sph}\, D_e\, \Delta c$. In the model, the real situation is approximated by using an equivalent plug. This plug of a cross-sectional area equal to the hole area has a thickness Δx given by $\Delta x = r_{hole} / \sqrt{2}$.

2.2.1.3 Transport into a narrow slit

For the diffusive transport into a narrow fracture, most of the resistance to the transport will be located nearest to the fracture because of the contraction in the cross-sectional area. The transport resistance is then approximated by a plug through which the nuclides are transported. The plug has a transport area equal to the cross-sectional area of the fracture, and a diffusion length equal to a factor times the fracture aperture. Neretnieks analytically modelled the stationary transport from the bentonite surrounding a canister for spent nuclear fuel into a fracture [5]. The procedure uses the exact solution of the steady-state two-dimensional diffusion equation for a sector of the clay barrier representing half the fracture spacing that allows symmetry conditions to be used. After some simplifications, the resistance of the plug at the mouth of the fracture is expressed as:

$$R_f = \left[\left(F_{x,0} / \delta\right)\delta / \left(D_e A_f\right)\right].$$

The factor ($F_{x,0}/\delta$) was calculated by Neretnieks for a number of fracture spacings, fracture apertures and barrier thicknesses. For fractures with an aperture varying between 10^{-4} and 10^{-3} m, and a backfill thickness of 0.30 to 0.35 m, the factor ranges between 3 and 7. It can be visualized as having a plug of clay at the mouth of the slit with a thickness of ($F_{x,0}/\delta$) times the slit aperture.

2.3 Assignment of material properties

In the standalone version of NUCTRAN, the physical parameters defining the transport in a repository are specified as constant. In the version to be used as a module in the PROPER package [6], the transport parameters, as sorption, diffusion, and equivalent flow rates may be specified as constant or as functional correlations.

2.4 Constraints and initial conditions

NUCTRAN solves an initial value problem, where the solution of the system of differential equations is straightforward once the initial conditions have been defined. The variables defining these conditions are determined by the amount of the species dissolved, the amount of the species as solid inventory in the compartments and the amount of nuclides embedded in the fuel matrix when a congruent dissolution model is used. The default initial condition is zero for all compartments, except for the compartment acting as a source where the initial condition is determined by the inventory and the solubility of the species. Additionally, with the FUELSURFACE and alpha-radiolysis models, an instantaneous release fraction (IRF) can be specified for each radionuclide.

To handle the inventory in the source, two situations are considered:

a) OWNSOL and FUELSURFACE types: nuclides are not embedded in the spent fuel matrix in the canister. They may then be handled as independent of each other with the nuclide inventory and the nuclide's own solubility as the only limitations.

b) MATRIX type: nuclides are embedded in the fuel matrix. This case is formulated considering a congruent approach for the dissolution of the matrix. Here, the nuclides are liberated from the matrix as the matrix dissolves. As the matrix consists mainly of uranium dioxide, the rate N_c at which the embedded nuclides are liberated depends on the uranium dissolution rate N_u and on the nuclide fraction in the fuel matrix. This process is formulated as:

$$N_c = N_u \frac{M_{i,matrix}}{M_{u,\,matrix}}$$

where $M_{i,matrix}$ and $M_{u,matrix}$ are the amounts (moles) of the embedded nuclide and the uranium dioxide in the fuel matrix respectively. The material balance for a nuclide in the fuel matrix is expressed as:

$$\frac{dM_{i,matrix}}{dt} = -N_c - M_{i,matrix}\,\lambda$$

The dissolution rate term (N_c) is included as a production term in the equations defining the source.

At present there are two distinct approaches to calculating N_U:

(a) the rate of dissolution of U-238 is controlled by its solubility limit and the rate at which it is removed from the first compartment,

(b) the user specifies the rate at which U-238 dissolves to represent the effect of alpha radiolysis. If this rate is lower than the rate calculated by (a), then the rate calculated by (a) will be used.

3 Numerical methods

3.1 Spatial discretization

The compartment model NUCTRAN formulates the near-field transport in terms of integrated finite differences, introducing the concept of "compartments" to define the discretization of the system. This concept is very useful when the transport is through materials with different properties and the geometry of the whole system is complex. The compartment is fully defined by its capacity and the resistances defining a two-dimensional nuclide transport. The capacity is determined by the volume and the sorption coefficient, and will also depend on the porosity and density of the material. The diffusion length(s), cross sectional area(s), and the diffusion coefficient define the resistance. The compartmentalization of the system is rather straightforward. The system to be modelled is subdivided into compartments taking into consideration the different geometric shapes and the various materials found in the system. NUCTRAN uses a coarse discretization.

3.2 Temporal discretization

To solve the system of ordinary differential equations (ODEs), NUCTRAN uses the package SDRIV2 found in Kahaner et al., [7]. This package uses the Gear multi-value (MV) method, which has an infinite stability region allowing changing step size [8]. The algorithm choosing the time-step is incorporated in SDRIV2. The user of NUCTRAN specifies points at which output is needed.

3.3 Solution methods

The ODEs expressed by a rearrangement of transport equations are solved by using standard numerical techniques for initial value problems. Here, the package SDRIV2 solves problems of the form $y^{'} = f(t, y)$ using implicit multi-value methods [7]. This package is specifically designed to solve stiff problems but is also appropriate for non-stiff problems. For the solver SDRIV2, the user's job is to write a main program, and a subroutine that evaluates the right hand sides of the system of equations. In this case, the main program is NUCTRAN. The stiff solver used by the package is based on the set of stiffly-stable formulae of Gear, which are efficient for the integration of stiff problems as long as these do not have any oscillating solution. The Gear MV method has an infinite stability region, although the method is not absolutely stable. The package also includes the Adams-Moulton MV that has a finite stability region and it is appropriate for the solution of non-stiff problems.

4 Managing the material balance

The terms determining the material balance for the compartments are: rate at which the species are transported to the adjacent compartments, degradation by decay, dissolution of the solid inventory and the accumulation of the species in the compartment. To account for the solid inventory, the physical compartment is defined by two fictitious sub-compartments - one to handle the solid inventory and the other to handle the species dissolved in the water and sorbed in the solid material. When a MATRIX dissolution model is used, an additional compartment is included for the amounts of radionuclides embedded in the fuel matrix. This situation is described in more detail later in the report.
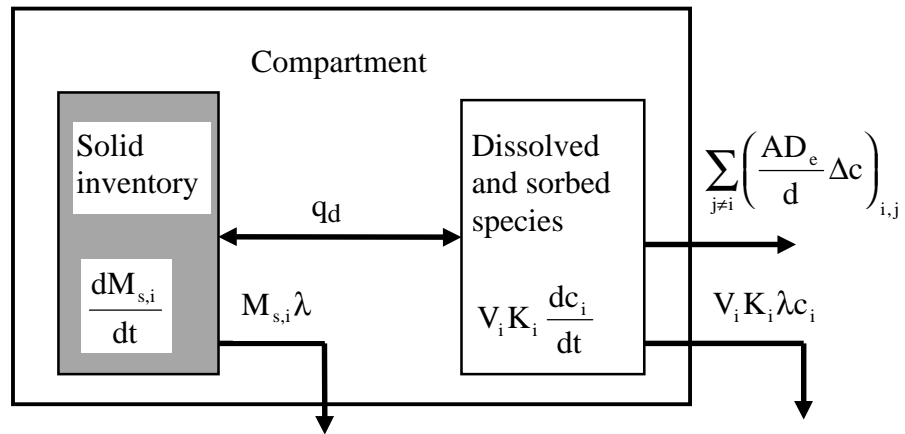
Figure 2 **A schematic view of the processes occurring for a single nuclide in a compartment. Two fictitious sub-compartments describe the compartment.**

The total amount of the species in the compartment is then the amount of the solid inventory plus the amounts of species dissolved and sorbed in the porous material. The two sub-compartments in Fig. 2 are coupled by a dissolution term. A change in the solid (inventory) mass with time is determined by the degradation of inventory by decay and the dissolution rate as follows:

$$\frac{dM_{s,i}}{dt} = - M_{s,i} \, \lambda \, - \, q_d$$

The change in the dissolved and sorbed species is determined by the dissolution of the solid inventory, degradation of the dissolved and sorbed species by decay and the rate at which the species are transported away by diffusion as follows:

$$V_i K_i \frac{dc_i}{dt} = q_d - V_i K_i c_i \lambda - \sum_j \left(\frac{AD_e}{d} \Delta c \right)_{i,j}$$

The term Δc is defined as the concentration difference between two adjacent compartments $\Delta c = c_i - c_j$. Equilibrium between the species dissolved and the solid inventory is assumed. Thus, if there is solid inventory in the compartment, $M_{s,i} > 0$, the governing equations are:

$$\frac{dc_i}{dt} = 0 \quad \text{and}$$

$$\frac{dM_{s,i}}{dt} = -M_{s,i}\lambda - V_i K_i c_s \lambda - \sum_j \left(\frac{AD_e}{d} \Delta c \right)_{i,j}$$

and if all solid inventory has been dissolved, $M_{s,i} = 0$, then:

$$\frac{dc_i}{dt} = -c_i \lambda - \frac{1}{V_i K_i} \sum_j \left(\frac{AD_e}{d} \Delta c \right)_{i,j} \quad \text{and}$$

$$\frac{dM_{s,i}}{dt} = 0$$

4.1 Handling nuclides embedded in the fuel matrix

Embedded nuclides are liberated as the predominantly U-238 fuel matrix dissolves. Embedded nuclides are assumed to dissolve congruently as the U-238 matrix dissolves. In this case, the nuclide inventory in the canister is handled in a special way. Nuclides in the canister may be found in three states: embedded in the uranium matrix, precipitated as solids and dissolved in the water volume. An additional fictitious compartment handles the nuclide fraction embedded in the matrix. The material balance for this additional compartment is expressed as:

$$\frac{dM_{i,matrix}}{dt} = -N_c - M_{i,matrix}\lambda$$

The rate at which the nuclide leaves the matrix, N_c, is related to the matrix dissolution rate and the ratio of inventory of the embedded nuclide and the nuclide identified as the fuel matrix (U-238). N_c, is determined by solving the transport equations for uranium (specified as the fuel matrix) and the embedded nuclide simultaneously.

The fraction of solid released from the fuel matrix (precipitated or dissolved) is handled by the following equations:

a) If the solubility limit is not reached, no solid precipitate is present in the canister, $M_{s,c} = 0$. Then the governing equations are:

$$\frac{dc_c}{dt} = \frac{N_c}{V_c K_c} - c_c \lambda - \frac{1}{V_c K_c} \sum_j \left(\frac{AD_e}{d} \Delta c \right)_{c,j} \qquad \text{and}$$

$$\frac{dM_{s,c}}{dt} = 0$$

b) If, on the other hand, the solubility limit is reached, the governing equations are then expressed as:

$$\frac{dc_c}{dt} = 0 \quad \text{and}$$

$$\frac{dM_{s,c}}{dt} = N_c - M_{s,c} \lambda - V_c K_c c_s \lambda - \sum_j \left(\frac{AD_e}{d} \Delta c \right)_{c,j}$$

4.2   Dissolution due to alpha radiolysis

NUCTRAN can also include a representation of spent fuel dissolution due to alpha-radiolysis [9 and 10]. The model assumes that the dissolution rate is related to the alpha-energy release of the fuel. When an alpha-radiolysis model is used with the MATRIX dissolution type, an IRF value can be specified for each embedded nuclide. This determines the fraction of the nuclide that is assumed to dissolve instantaneously.

Three different representations of the evolving alpha-energy release are included in NUCTRAN.

a) The dissolution rate of the fuel matrix occurs at a constant rate (CONSTANT type).
b) The dissolution rate is a function of the alpha-radiolysis dose rate of the fuel, and decreases with time as a result of radioactive decay (DECAY type).
c) The dissolution rate is a function of the alpha-radiolysis dose rate of the fuel, and decreases with time as a result of radioactive decay and dissolution of alpha-emitting solids from the fuel matrix (EXPLICIT type).

Note: the dissolution-controlled rate is used if it is greater than the alpha-radiolysis-controlled rate in any of the alpha-radiolysis dissolution models.

4.2.1   CONSTANT model

The CONSTANT representation assumes that the alpha-energy release from the fuel remains constant. The equation representing loss of U-238 from the fuel matrix is:

$$\frac{\partial M}{\partial t} = -K_{CON} - \lambda M$$

where:
    K_{CON} is the constant dissolution rate (mol yr^{-1});
    M is the amount of U-238 in the fuel matrix at time t (mol); and
    λ is the decay constant for U-238 (yr^{-1}).

The first term on the right-hand side of the above equation represents loss by radiolytically-induced dissolution and the second term represents loss by radioactive decay.

4.2.2   DECAY model

The DECAY representation assumes that four elements contribute to alpha radiolysis, namely Am-241, Pu-239, Pu-240 and Np-237. The alpha energy release from the fuel is related to the amount of each of these nuclides present in the fuel matrix and the energy of the alpha emissions for each of the nuclides. The energy-release rate falls with time as alpha-emitting nuclides embedded in the fuel matrix decay.

The alpha radiolysis dose rate as a function of time is given in reference [11]:

$$r_{ox} = \sum_{i=1}^{4} A_i \exp(-\frac{t \ln 2}{B_i})$$

Where:
    r_{ox} is the alpha radiolysis dose rate;
    t is the time; and
    A_i and B_i are constants for the nuclides Am-241, Np-237, Pu-239 and Pu-240:

Nuclide	A_i	B_i / years
Am-241	25.3	433
Np-237	0.04	$2.1 \cdot 10^6$
Pu-239	1.1	$2.4 \cdot 10^4$
Pu-240	2.2	6570

These constants are taken from [11] and are only applicable to a specific fuel. If a different fuel to that described in reference [11] is used, the constants will need to be modified in the code. However, it is not likely that this model will be used extensively, as the CONSTANT model provides an adequate representation of dissolution.

The rate of dissolution of U-238 from the fuel matrix due to alpha radiolysis, r_α, is given by:

$$r_\alpha = K_{DEC} r_{ox}$$

where:
    K_{DEC} is the proportionality constant and has units mol yr^{-1}/(units of A_i).

Thus the governing equation for the loss of U-238 from the fuel matrix is:

$$\frac{\partial M}{\partial t} = -K_{DEC} \sum_{i=1}^{4} A_i \exp(-\frac{t \ln 2}{B_i}) - \lambda M$$

where:
  M is the amount of U-238 in the fuel matrix at time t; and
  λ is the decay constant for U-238.

The first term on the right-hand side of the above equation represents loss by radiolytically-induced dissolution and the second term represents loss by radioactive decay.

4.2.3 EXPLICIT model

To maintain the flexibility of the program, modifications have been made to NUCTRAN to represent loss of alpha-emitting nuclides from the fuel matrix by dissolution, as well as by decay. This is achieved by including the nuclides Am-241, Pu-239, Pu-240 and Np-237 explicitly in the calculation and using the time-dependent amount of these nuclides to calculate the alpha-radiolysis dose rate. The rate of loss of U-238 from the matrix is then given by the expression:

$$\frac{\partial M}{\partial t} = -K_{EXP} \sum_{i=1}^{4} C_i M_i - \lambda M$$

where:
  K_{EXP} is the proportionality constant and has units yr^{-1}/(units of C_i).
  M_i is the amount of alpha-emitting nuclide i in the fuel matrix at time t (mol); and
  C_i is a constant for nuclide i in a given fuel:

The first term on the right-hand side of the above equation represents loss by radiolytically-induced dissolution and the second term represents loss by radioactive decay.

Nuclide	C_i
Am-241	2.85
Np-237	$8.81 \cdot 10^{-3}$
Pu-239	0.0261
Pu-240	0.0992

These constants are derived from the values in [11] and are only applicable to a specific fuel. If a different fuel to that described in reference [11] is used, the constants will need to be modified in the code. However, it is not likely that this model will be used extensively, as the CONSTANT model provides an adequate representation of dissolution.

4.3   Basic equations for radionuclide chain

The mathematical handling for a nuclide chain in series is very similar to the formulation describing the release of a single nuclide. The material balance of a nuclide chain over a compartment may be described by a set of linear differential equations. The calculations for the members in the chain are made in succession, beginning with the first member of the chain called the mother nuclide. The mother nuclide is independent of the other nuclides in the chain. The decay term for a parent is a source term for a daughter nuclide and is included as such in the calculations for the daughter. The material balance for nuclide "n" in compartment "i" is expressed by the differential equation for the solid and for the dissolved and sorbed species as follows:

$$\frac{dM_{s,i}^n}{dt} = M_{s,i}^{n-1}\,\lambda^{n-1} - M_{s,i}^n\,\lambda^n - q_d \qquad \text{and,}$$

$$V_i K_i^n \frac{dc_i^n}{dt} = q_d + V_i K_i^{n-1} c_i^{n-1}\lambda^{n-1} - V_i K_i^n c_i^n \lambda^n - \sum_j \left(\frac{AD_e}{d}\Delta c\right)_{i,j}^n$$

These equations are very similar to the governing equations for a single nuclide, except for the term of production from the parent nuclide "n-1". The initial conditions and constraints for the compartments are handled in the same way as for single nuclides.

5   Description of the NUCTRAN code

The NUCTRAN code is written in FORTRAN. There are two versions depending on the environment it works in: a standalone version and a special version to be used as a submodel of the PROPER code. The code consists mainly of three parts that are:

MAIN PROGRAM: CMP23
SOLVER: SDRI23
INPUT FILES: system.dsc, *casename*.inv, *casename*.nam and submod.lib

The main program CMP23 makes use of several subroutines that will be described later. The solver SDRI23 is based on Gear's method, which is described in the literature [7]. The INPUT files are described in detail in the next sections.

The code solves three types of specific situations:

I   Solubility limit approach. All species in the canister are available for release, independently of the structure they are part of. The only limitation in the nuclide release is the solubility of the individual species.

II   A particular case for nuclides initially located at the fuel surface. The handling of this situation is similar to I but considers that only a fraction of the total nuclide inventory is available for release.

III   Congruent approach for nuclides embedded in a fuel matrix. Since the matrix is mostly formed by uranium oxide, the escape rate for the embedded nuclides will depend on the escape rate of the uranium. Thus, to calculate the release of these nuclides, the U-238 is simultaneously run with the nuclides of interest.

There are also special uses of the code that are easy to manage. They are:

a) Addition of a transport resistance between two compartments. The code needs know only the value and where the resistance will be placed.

b) The damage in the canister wall is handled as a special compartment where the user has the alternative of choosing a growing hole or a hole of stationary size. For the situation of a growing hole, the user may chose a linear function or a step function for the growth.

c) Addition of a plug at the outlet of the source in order to approximate the mass transport between a small compartment and a large one. This is applied to the nuclide transport from the small hole in the canister wall into the bentonite outside the hole.

d) Addition of a plug situated inside the canister (source) when the backfill material is granulated. This is a special use for copper/iron canister as source, where the nuclide transport into the damage in the canister wall is approximated by a plug at the inlet of the damage. This plug has the same dimensions as the plug at the outlet of the damage. The effective diffusion coefficient (D_e) is established beforehand and is 10^{-10} m^2/s.

Note:

The dimensions of both plugs (inlet and outlet) depend on the size of the hole at the canister wall. So if the size of the hole varies, the dimensions of both plugs vary too.

5.1   Structure of the code

```
                    ┌─────────┐
                    │  START  │
                    └────┬────┘
                 ┌───────┴────────┐
                 │ CALL GSAM23    │
                 └───────┬────────┘
                 ┌───────┴────────┐      ┌──────────┐
                 │ CALL READ23    │─────▶│ HUIM23   │
                 └───────┬────────┘◀─────│ GSAM23   │
                 ┌───────┴────────┐      └──────────┘
                 │                │              ┌──────────┐
                 │ CALL SET23     │─────────────▶│ INVEN    │
                 └───────┬────────┘◀─────────────│ SETM23   │
                 ┌───────┴────────┐              │ SETS23   │
                 │                │    ┌──────────┘──────────┘
                 │ CALL FMAT23    │───▶│ FMDE23   │
                 └───────┬────────┘◀───│ FMCO23   │
                                       │ GMAT23   │
                                       │ WRIT23   │
                                       └──────────┘
```

CALL GSAM23

CALL READ23 → HUIM23 / GSAM23

CALL SET23 → INVEN / SETM23 / SETS23

CALL FMAT23 → FMDE23 / FMCO23 / GMAT23 / WRIT23

(10) → CALL SDRI23 → FSUB → FEMB23 / FBSU23 / FUEL23 / FMMA23 / GMAT23 → ALPH23

CALL SDRI23 ⇄ GFUN

CALL GETC23

CALL CONT23 → CREL23 / CSTO23 / FMMA23 / GMAT23 / CFUE23 → FMMA23 / GMAT23

ICALL = 1   YES → (10)
ICALL = 1   NO → ICALL = 2   YES → (100)
ICALL = 2   NO

CALL CONC23

ICALL = 2   YES → (100)

CALL TIME23

ICALL =1 OR ICALL = 0   YES → (10)
NO

(100) → CALL FLOW23 → PUTS

CALL PRIN23

END

5.2 Subroutines and functions used by NUCTRAN

5.2.1 Key variables used by the subroutines

AREA: variable that defines the cross-sectional area for a compartment whose size changes with time.

AGQ: partial matrix of constant coefficients.

AINV: vector storing the input amount (inventory) of the species.

EPS: requested relative accuracy in all solution components.

EWT: problem zero, i.e., the smallest physically meaningful value for the solution.

MINT: integration method flag.

FSUB: a subroutine supplied by the user.

GFUN: a real FORTRAN function supplied by the user if NROOT is not zero.

ICALL: index controlling the recall of the solver from the main program or the stopping of the program execution.

ICH: index identifying the decay chain number.

INUC: index identifying the nuclide in a chain.

IPRINT: position index in the time series.

IW: parameter used by the solver.

LW: parameter used by the solver.

LIW: parameter used by the solver.

MS: numerical parameter controlling the status of the integration.

N: the number of differential equations.

NROOT: the number of equations whose roots are desired. If NROOT is zero, the root search is not active. This option is useful for obtaining output at points that are not known in advance, but depend upon the solution.

QDURAN: dissolution rate of the fuel matrix.

QMAX: dissolution rate of the fuel matrix when an alpha-radiolysis model is used.

T: independent variable in the ODEs.

TOUT: the point (time) at which the solution is desired.

W: parameter used by the solver.

Y: vector of concentrations.

YDOT: vector of ordinary differential equations ODEs.

5.2.2 Description of key subroutines

ALPH23 (QDURAN, TIME, QMAX)

This subroutine calculates the dissolution rate of the fuel matrix (U238), QMAX, when an alpha-radiolysis model is used. QMAX is the larger of the dissolution-controlled and alpha-radiolysis-controlled dissolution rates.

CFUE23 (ICH, Y)

This subroutine dissolves the remaining solid inventory when the spent fuel or the fuel matrix is fully dissolved. CFUE23 is called by CONT23.

CONC23 (Y, IPRINT, ICALL)

This subroutine stores the output data, such as concentrations for all compartments and time at which they were calculated. The main program CMP23 calls CONC23.

CONT23 (MM, MS, IPRINT, ICALL, T, Y)

This subroutine controls the execution of the code, mainly the link between the main program and the solver. It gives information on the reasons for unexpected stops of the solver and determines the stop of the execution or the return to the solver. Before the execution returns to the solver, CONT23 determines how to continue the execution by switching variables that control the material balance and size of the damage at the canister wall. The execution control is done with help of auxiliary subroutines CFUE23, CSTO23, CREL23, FMMA23 and GMAT23. The main program CMP23 calls CONT23.

CREL23 (MS, ICALL, T, Y)

This subroutine counts the nuclides whose concentration in the canister is very small and therefore considered as negligible by NUCTRAN. For these nuclides, CREL23 makes zero the coefficients of the matrix "F" defined in the ODEs $d\overline{c}/dt = \overline{\overline{F}}\overline{c}$. CREL23 is called by CONT23.

CSTO23 (MS, ICALL, T, Y)

This subroutine switches the IFLAGs to indicate which governing equations are used to control the material balance in the compartments. This operation is only performed if a stop has been signaled by the function GFUN. CSTO23 is called by CONT23.

FBSU23 (ICH, Y, YDOT)

This subroutine calculates the RHS of the ODEs $d\overline{c}/dt = \overline{\overline{F}}\overline{c}$. If there are nuclides embedded in a fuel matrix, the RHS for the compartment representing the source (fuel matrix) is calculated by FEMB23. FBSU23 is called by FSUB.

FEMB23 (ICH, QDURAN, Y, YDOT, T)

This subroutine calculates the RHS of the ODEs $d\overline{c}/dt = \overline{\overline{F}}\overline{c}$ that define the nuclide transport in the canister for nuclides embedded in a matrix. If an alpha-radiolysis model is used, FEMB23 calls ALPH23 to calculate QMAX. For the other compartments, the RHS is evaluated by FBSU23. FEMB23 is called by FSUB.

FLOW23 (IFSTOP, IPRINT, MULTI)

This subroutine calculates the release, by molar flow rate, from the source and into the various sinks. These data are written to the PROPER database via a call to PUTS. The time series at which the molar flow rates are calculated is determined beforehand by the user. The main program CMP23 calls FLOW23.

FMAT23 (NAMES)

This subroutine evaluates the matrix of coefficients "F" defined in the ODEs $d\overline{c}/dt = \overline{\overline{F}}\overline{c}$ using FMDE23, FMCO23 and GMAT23. FMAT23, based on information defined in the INPUT file, defines the capacity and resistances to transport of all compartments to calculate the coupling resistances between compartments. The main program CMP23 calls FMAT23.

FMCO23 (ICH)

This subroutine defines the coupling resistances between compartments, including the external coupling with the flowing water in the surrounding of the barrier system. FMCO23 is called by FMAT23.

FMDE23 (ICH)

This subroutine defines the capacity and individual resistances to transport of all compartments. FMDE23 is called by FMAT23.

FMMA23 (ICH, AREA)

This subroutine calculates the matrix of coefficients "F" defined in the ODEs $d\overline{c}/dt = \overline{\overline{F}}\overline{c}$ by recalculating the capacity and the resistance to transport if there is a growing hole at the source (canister wall). It also calculates the resistance of the plugs connected to the hole. FSUB, CONT23 and CSTO23 call FMMA23.

FSUB (N, T, Y, YDOT)

This subroutine computes ydot = f(y,t), the right-hand side of the ODEs, using the subroutines FEMB23, FBSU23, FMMA23, FUEL23 and GMAT23. When this subroutine is called, the first N components of Y are intermediate approximations to the solution components. The user should not alter these values. A return from this function passes control back to SDRI23.

FUEL23 (QDURAN, Y, YDOT)

This subroutine computes the dissolution rate (QDURAN) and the right-hand side of the ODEs, ydot = f(y,t), for the fuel matrix in the canister. FUEL23 is called by FSUB.

FUNCTION GFUN (N, T, Y, IROOT)

A double precision FORTRAN function that is supplied by the user if NROOT in the argument of the subroutine SDRI23 is not 0. This function is called repeatedly by SDRI23 with different values of IROOT to obtain the value of each of the NROOT equations for which a root is desired [7]. The function is used to control the material balance in the compartments, and the growth of the hole in the canister wall.

GMAS23 (ICH, INUC, AGQ)

This subroutine builds up the matrix "AGQ" that is a partial matrix of constant coefficients defined for one nuclide as:

$$AGQ = \begin{bmatrix} -(\lambda + g_{1,1})\ldots & g_{1,j}\ldots & g_{1,n} \\ g_{i,1}\ldots & -(\lambda + g_{i,j})\ldots & g_{i,n} \\ g_{n,1}\ldots & g_{n,j}\ldots & -(\lambda + g_{n,n}) \end{bmatrix}$$

The diagonal terms include the decay constant and the boundary conditions connecting the sink to the system. The other terms account for the connection of compartments "i" and "j". The coefficients of this matrix are calculated as:

$$g_{i,j} = \frac{1}{C_i \ R_{i,j}}$$

where C_i is the capacity of the compartment and $R_{i,j}$ is the coupling resistance between compartments "i" and "j". GMAS23 is called by GMAT23.

GMAT23 (ICH)

This subroutine builds up the global matrix of coefficients "F" in the ODEs $d\overline{c} / dt = \overline{\overline{F}}\overline{c}$. The terms of this matrix "F" are calculated using subroutine GMAS23. CSTO23, FMAT23, FSUB and CONT23 call GMAT23.

PRIN23 (IPRINT, NAMES)

This subroutine prints out results of the execution such as concentrations in the various compartments, release into the various sinks and solid inventory in the canister. The results are printed at times determined by TIME23. It also prints out the variation with time of the hole size in the canister wall. The main program CMP23 calls PRIN23.

READ23 (NAMES, EPS, EWT, NLOOP, NPREPAR, LSTALON, U0, CASE)

This subroutine reads the data from the INPUT files with help of an auxiliary subroutine HUIM23 created specially to operate in the PROPER environment. The main program CMP23 calls READ23.

SDRI23 (N, T, Y, FSUB, TOUT, MS, NROOT, EPS, EWT, MINT, W, LW, IW, LIW, GFUN)

This subroutine solves the N ordinary differential equations (ODEs) of the form $dY(I)/dt = F(Y(I),t)$, given the initial conditions $Y(I) = Y(t = 0)$. The program has options to allow the solution of both stiff and non-stiff differential equations. SDRI23 is called once for each output point of T. For SDRI23, the user's job is to write a directive program NUCTRAN, a subroutine that evaluates the right-hand sides of the equations FSUB and a FORTRAN function GFUN to control the root of the variable concerning the user. The main program CMP23 calls SDRI23.

SET23 (NAMES, Y, CASE)

This subroutine initializes the execution of the code. It sets the initial conditions for the problem and the flags (IFLAG) for the material balance control in the various compartments. The main program CMP23 calls SET23.

SETM23 (ICH, TOTINV, Y)

This subroutine initializes the execution of the code when the problem to be solved is nuclides embedded in the matrix (case specified by ISPEC = 3). SETM23 is called by SET23.

SETS23 (ICH, TOTINV, Y)

This subroutine initializes the execution of the code when the problem to be solved is nuclides whose inventory is completely available to be released with the only limitation being the solubility, ISPEC = 1 or 2. SETS23 is called by SET23.

TIME23 (TINIT, ICALL)

This subroutine determines the time series at which the user wishes to obtain results. The time step is determined by a geometric progression. The main program CMP23 calls TIME23.

WRIT23 (NAMES)

This subroutine gives information on capacities and resistances of the various compartments. It also gives information on the matrix of the coefficients "F" initially executed. WRIT23 is called by FMAT23.

5.2.3   Additional subroutines

A description of these subroutines is found in the Proper Monitor User's Manual [8] and Proper Submodel Designer's Manual [6]. These subroutines are implemented and of use only in the PROPER version of the code NUCTRAN. Some of them are part of the PROPER package. "They are service routines for setting up communications with the

database, for acquiring values of sampled parameters, for handling time series and for handling data from external submodel-specific files" [6].

INVEN (FILNAM, IUNAM, FILINV, IUINV, NNUCL, NAMES, TBREAK, AINV)
This subroutine gives information on the inventory in the canister. SET23 calls INVEN.

PRELUD (FILE)
Set up communication with the internal database.

POSLUD
Closing down communication with the internal database.

GETP (INDEX)
Used to obtain sampled parameters from the internal database.

PUTS (IDD, TSERIE, VALUE)
Used to send output time series to the internal database.

To avoid conflict, service routines are used to obtain an unused unit number and the unique name of a data file when opening. They are:

| IGETUN (IDUM) | to get the unit number. |
| GETDAT (IDUM) | to get the unique name of a data file. |

and when closing the file

| PUTUN (IUDAT) | to return the unit number to the monitor. |

5.3   General description of the input requirements

The physical geometry of the system is simplified by dividing it into blocks. The blocks are numbered in ascending order. The discretization in blocks considers the geometry and the various materials by which the nuclides migrate. Not all blocks have the same geometry and orientation, so each block has its own axes of references. So far there are only two transport directions defined for the block (r-direction and z-direction).

The transport in a block is completely defined by the physical properties of the material, the nuclide transport properties and the geometrical dimensions of the block. These properties are used by the code to calculate the capacity of the block and its resistance to transport for each of the two defined transport directions. These data are the input data for the problem to be solved. The capacity and the resistance are determined by the following expressions:

$$\text{Capacity} = V\left[\varepsilon_p + \left(1 - \varepsilon_p\right)k_d\rho\right]$$

$$\text{Resistance} = \frac{x}{A\,D_e}$$

where x is the diffusion length of the block, A is the cross-sectional area of the block perpendicular to the direction of transport and D_e is the effective diffusion coefficient. The porosity and the density of the material solid are denoted by ε_p and ρ respectively, k_d is the sorption coefficient and V is the volume of the block.

As two transport directions have been defined, the code needs to know the geometrical dimensions of the block for both directions to define the transport. Such geometrical dimensions are diffusion length and cross-sectional area for each direction. The volume of the block is calculated by the code from the given dimensions for transport in the z-direction.

Any block, except the source, may be subdivided into compartments in any of the two directions named above. The block itself is a compartment if it is not subdivided. A simple subdivision is that by discretization in the z-direction the code makes the block into compartments of equal capacities. The code needs only to know the desired number of subdivisions in the z-direction NZ, so that NR = 1, which means no subdivision in the r-direction. For any other subdivision, all compartments have to be defined in the INPUT file. The compartments are numbered by the code following the z-axis for the block, while the other direction (r-axis) is kept constant.

The first block defined in the INPUT data is the source (interior of the canister) followed by the block describing the damage of the source (hole in the canister wall). For the special case of a copper/iron canister (ICAN = 2) with few zircaloy tubes damaged (less than 20 tubes), Romero et al. defined the source as two well-mixed blocks connected by an equivalent resistance [12]. So for this special situation, the user has to define the source as two blocks followed by the block that describes the hole in the canister wall. The first block is for the zircaloy tubes containing the nuclides, and the second block is for the granulated material surrounding the zircaloy tubes.

Each block may be connected to one or more than one block, except the source. At present, the source can only be connected to one compartment. Each couple of connected blocks (A and B) is specified by the user in the INPUT file. Several control numbers define the connection. Such numbers are used to define: the position of each block (A and B), the numbers of the couples of compartments involved in the connection of block A and B, the direction (z-axis or r-axis) of each block and the contribution of each block to the coupling resistance. After the connection of the two blocks is specified, the code needs to know the position of the couples of compartments involved in each connection. All this information is used to calculate the coupling resistance $R_{i,j}$:

$$R_{i,j} = \frac{R_i}{2} + \frac{R_j}{2}$$

where R_i and R_j are the individual resistances of the compartment "i" and the adjacent compartment "j" respectively.

External resistances specified in the input data may be added between two blocks. For resistances added in the form of a plug, these are codified by IPLUG = index (block number A or B). This index block is used by the code to obtain the diffusion coefficient. This plug concept is very useful when the transport is between a block of very small volume (block A) and a block of large volume (block B). The plug resistance value is:

$$\text{plug resistance} = \frac{\sqrt{A_{hole} / (2\pi)}}{D_e A_{hole}} = \frac{1}{D_e \sqrt{2\pi A_{hole}}}$$

where A_{hole} is the cross-sectional area of the block of small capacity. Suppose that for some A-B connection, IPLUG = block B number (block of large capacity). Then, the coupling resistance calculated by the code is:

$$\text{Coupling resistance } R_{ab} = \frac{R_a}{2} + \text{plug resistance}$$

The size of the block of small capacity may vary with time. If this is the case, the plug resistance will also vary. The existence of other types of resistances added to the connection are indicated by IRADD = 1, whose values are given following the description for such effects in the input file. Suppose that for some A-B connection, IRADD = 1. If the user specifies the resistance value as RADD, the calculated coupling resistance is:

$$\text{Coupling resistance } R_{ab} = \frac{R_a}{2} + \text{RADD} + \frac{R_b}{2}$$

The connections of the various sinks to the system (repository) are defined by identifying the position of the block and compartment connected to each sink. In addition, the user has to codify the direction of the nuclide transport (IRZ) and the contribution of the compartment to the coupling resistance (ICRS). For the situation of one fracture intersecting the system, a plug approximates the nuclide transport into the fracture. The dimension of this plug have to be defined by the user. For this situation ICRS = 0.

As the code for NUCTRAN exists in two different operative versions, there are two input data files. The differences between them are in the structure of the subroutines to read the input data, to process the output data, and to get information of the initial nuclide inventory. The variable definition is the same for both versions. In the standalone version of NUCTRAN, the nuclide inventory has to be given in the INPUT file. In the PROPER version of NUCTRAN, the nuclide inventory is implicitly obtained by the code; it needs only to know the names of the nuclides and the break-time (TINIT) for the canister. In the next section, the INPUT file for the PROPER version will be presented. The standalone version is described in a separate manual.

6 Simulation setup using PROPER

6.1 Overview

NUCTRAN can be run as a submodel of the PROPER package. A User Guide for the PROPER package is given in reference [8]. This section gives details specific to running NUCTRAN as a submodel of PROPER.

Section 6.2 describes the input files that are required by the PROPER package when NUCTRAN is included as a submodel.

The PROPER version of NUCTRAN uses the HYDRASTAR User Interface (HUI) [13]. HUI is a preprocessing facility incorporated into the HYDRASTAR code and the NUCTRAN code. The main objectives of the HUI are:

- Free-format input in a single input file.
- Input data validity testing.
- Input data consistency checking.

A general description of the HUI input file is given in section 6.3 of this report. A detailed description of the COMP23 input required is given in section 6.4.

Details of the output produced by the HUI interface are discussed in section 6.5.

6.2 Input files required

The following input files are required to run NUCTRAN as a submodel of PROPER:

Name	Description
system.dsc	The system description file that is used for all PROPER simulations. The input requirements for this file are described in more detail below.
casename.inv, *casename*.nam	These two files hold the information on the radionuclide inventories for one canister. The inventory files used by NUCTRAN are identical to the files used by TULLGARN [14], so these two files should preferably be links if TULLGARN is used in the same simulation. *casename* is read in from the CONTROL block of the system.dsc file.
submod.lib	Lists the modules that are used by PROPER. The module name for NUCTRAN is COMP23 and it is connected as an internal submodel [8].

6.3 HUI input

The following notational conventions are used to describe the format of the input data for this model. In general the definition follows that of INFERENS [15]. A detailed description of the input required when running the PROPER version of NUCTRAN is given in Section 6.4.

6.3.1 Typefaces

CAPITAL	Keywords.
lowercase italics	Variable name or variable text.
CAPITAL ITALICS	Object name or name of variable in the code.
<u>underlined</u>	Fixed text.
<u>bold underline</u>	Program validity check.

6.3.2 Separators

#	Beginning of comment.
{}	List of items.
[]	Enclosed list of optional items.
|	Optional items separator.
@	Zero or more occurrences of preceding expression.
+	One or more occurrences of preceding expression.
<>	Keystrokes or special characters.

6.3.3 Format specifiers

A*	Free-format string.
A*:wordlist	Free-format string; must be member of wordlist.
F*	Free-format floating point number.
I*	Free-format integer.
wordlist	List of alternative keywords [word1|...|wordN] which are allowed for the specified context.

6.3.4 Program validity check

V	Validity.
C	Consistency.
E	Existence.
W	Warning.

6.3.5 Format syntax

{input_file} =
   [system_commands]@
   [{a_block} | {comments}]+

{a_block} =
   BEGIN_BLOCK block_id
     {block_definition}
   END_BLOCK

{block_definition} =
   [{keywords} |
   {lists}|
   {definition_block} |
   {switchcommands} |
   {comments}]+

{keywords} =
   KEYWORD [I*|F*|A*|A*:wordlist|]+ [{comments}]

{lists} =
   KEYWORD
     [I*|F*|A*|A*:wordlist|]+ [{comments}]
   END_LIST

{definition_block} =
   BEGIN_DEF def_id
     [[{keywords} | {comments}]+ | [I*|F*|A*|A*:wordlist|]+]
   END_DEF

{switch_command} =
switch (switch_var)
   case VALUE1:
     {keywords}
   ...
   case VALUEn:
     {keywords}
endswitch

{comments} =
   ' #' {any_text} {end_of_line}

system_commands =
   SYSTEM KEYWORD

6.4 The system description file

6.4.1 Overview

The system description file, system.dsc, is used to specify for PROPER the module intercommunication, sampled parameters etc. The general format of the system description file is (only the parts relevant to this manual have been included):

PROPER_KEYWORDS
MACROS definitions
PARAMETERS section
MODULE definitions

each **MODULE** definition has the general format:

MODULE module (Input_Timeseries; Output_Timeseries)

INPAR section
DATA section

END module

Note: the order of the INPAR and DATA sections is not significant.

Any lines beginning with the '#'-character are handled as a comment line.

6.4.2 Parameters

The PARAMETERS section in the system description file defines the sampled parameters and their distributions. The general syntax for the PARAMETERS section is:

PARAMETERS
Parameter_name distribution_type [(arg1[,arg2[...]])]
END PARAMETERS

The order in which the parameters are passed to NUCTRAN is defined within the INPAR section within the MODULE definition. The general syntax for the INPAR section is:

INPAR
parameter_name1 [parameter_name2 ...]
END INPAR

The following parameters need to be defined in the PARAMETERS section for NUCTRAN. A detailed description of the parameters is given below.

Suggested name(s)	Description
CSYNC	Random number seed.
MULT	The number of canisters the calculation segment (normally from HYDRASTAR).
PROB	Probability for initial damage to canister.
PITFAQ, QS, CONHS, DEFFS	Corrosion data (not used by present version of the model).
CSOL	Solubility of the nuclides.
DIF	Effective diffusion coefficient in the different barrier materials for all nuclides.
KD	K_d-value in the different barrier materials for all nuclides.
IRF	Instantaneous Release Fraction (IRF) values (only required for nuclides that use a FUELSURFACE or alpha-radiolysis model).

The order that these are listed above is the order that the parameters must be specified within the INPAR section. When NUCTRAN is used as a standalone model two extra parameters are given as the first parameters. These are NCAN and UC.

Suggested name	Units	Description
NCAN	[-]	Number of canisters to be modelled (recommended value 1). (Only used for standalone version)
UC	$[m^3/m^2/yr]$	Water flow rate outside canister. (Only used for standalone version)
CSYNC	[-]	Random number seed.
MULT	[-]	Number of (assumed identical) canisters in region.
PROB	[-]	Probability of a single canister being penetrated at time of emplacement.

If NUCTRAN finds that there is a single canister that has been penetrated at time of emplacement, it ignores the other canisters and calculates the rate using one initially damaged canister. If no canister is initially damaged, no release calculations are performed in the present version.

The following four parameters have been included for future use regarding corrosion of the canister. The *must* always be specified but are not used for any calculations in the present version of the code.

Suggested name	Units	Description
PITFAQ	[-]	Not used
QS	[-]	Not used
CONHS	[-]	Not used
DEFFS	[-]	Not used

The next parameter defines the solubility of the nuclides:

Suggested name	Units	Description
CSOL	[mol/m^3]	Solubility for the nuclides, one parameter for each nuclide n.

For example, if two nuclides (A and B) are used the definition would look like:

    CSOLA CONST (2.0E-4)
    CSOLB CONST (2.0E-5)

The next parameters are repeated in the order stated below, for each material type in the model (except IRF values that are specified only once). Usually there are four materials used, *water*, *bentonite*, *sand-bentonite* and *rock*. For every material type, the data for every nuclide must be defined. Note that the order of nuclides as defined within the DATA section must be followed.

Suggested name	Units	Description
DIF *nm*	[m^2/year]	Effective diffusivity for nuclide *n* in material *m*.
KD *nm*	[m^3/kg]	K_d for nuclide *n* in material *m*.
IRF *n*	[-]	Instantaneous Release Fraction for nuclide *n*. One value for each nuclide that uses a FUELSURFACE or alpha-radiolysis model.

For example, if two nuclides (A and B) and four materials (1 to 4) are used the definition would look like:

```
# Specific to material 1
    DIFA1 CONST (0.123E0)
    DIFB1 CONST (0.123E0)
    KDA1  CONST (0.0E0)
    KDB1  CONST (0.0E0)
#
# Specific to material 2
    DIFA2 CONST (0.3154E-2)
    DIFB2 CONST (0.3154E-2)
    KDA2  CONST (3.0E0)
    KDB2  CONST (3.0E0)
#
# Specific to material 3
    DIFA3 CONST (0.3154E-2)
    DIFB3 CONST (0.3154E-2)
    KDA3  CONST (0.1E0)
    KDB3  CONST (0.1E0)
```

```
#
# Specific to material 4
    DIFA4 CONST (3.1536E-6)
    DIFB4 CONST (3.1536E-6)
    KDA4  CONST (3.0E0)
    KDB4  CONST (3.0E0)
#
# IRF values (only for FUELSURFACE or alpha-radiolysis models)
    IRFA CONST (0.1E0)
    IRFB CONST (0.1E0)
```

6.4.3   Input timeseries

As with the TULLGARN code, NUCTRAN needs to obtain the Darcy groundwater velocities outside the canister(s) as an input timeseries from another model.

There are three pseudo-timeseries that are to be supplied to NUCTRAN. The pseudo-timeseries must be specified in the order below within the MODULE definition.

Suggested name	Units	Description
U0X	$[m^3/m^2/yr]$	Darcy velocity of groundwater in first horizontal direction outside canister.
U0Y	$[m^3/m^2/yr]$	Darcy velocity of groundwater in second horizontal direction outside canister.
U0Z	$[m^3/m^2/yr]$	Vertical Darcy velocity of groundwater outside canister.

Note that the number of entries in the input time series is not equal to the sampled parameter MULTI. Normally when using HYDRASTAR to generate the input pseudo-timeseries, there is only one entry in the input pseudo-timeseries.

The input timeseries are defined in the MODULE definition in the system description file. The general syntax for the MODULE definition is:

   MODULE [mod:]sub([pm1:ts1 [ts2 ...][, pm2: ...]];[tso1 [tso2 ...]])

   END sub/mod

For example the module definition for a segment can look like:

   MODULE NUCTRAN1:COMP23 (HYDR11: TS1HYDR11; TS1COMP23)

when the Macros:

   MACRO TS1HYDR11 = UxC1 UyC1 UzC1
   MACRO TS1COMP23 = QAC1 QBC1

have been defined earlier. In this example, water flow rates are taken as input timeseries from HYDRASTAR and output mass flow for the two nuclides are sent as output timeseries to a subsequent submodel, e.g. FARF31. The unit of mass flux is moles/year or Bq/year depending on the value of the keyword RELEASE_TYPE in the CONTROL BLOCK described below.

6.4.4 Module specific input data

NUCTRAN also needs a number of non-sampled parameters specified in the module specific DATA section within the module definition. The general syntax for the DATA section is:

> **DATA**
> *GEOMETRY BLOCK*
> *CONTROL BLOCK*
> **END DATA**

The DATA section for NUCTRAN is read in free format using the HUI input format, very similar to the input structure used for HYDRASTAR in *casename*.hyd.

The input in the DATA section to the NUCTRAN submodel is supplied as module-specific data and consists of two input data blocks. The input data blocks may appear in any order. Between the input data blocks, comment lines and system commands may be inserted in any order. The lines in the input file <u>must not exceed 71 characters</u>. The program ignores multiple white space. That is two or more consecutive occurrences of white-space characters. White space is defined as <blank> or <tab>. The general format of a block is described in Section 6.3.

The *block_id* identifies the current input data block. Valid input data block names are:

GEOMETRY   <u>Required</u> block. Contains definitions of compartments, sinks, materials, etc.

CONTROL   <u>Required</u> block. General description of calculation control parameters and nuclide definitions.

The *{block_definition}* holds data for the current input data block. It is constructed from a set of keywords with optional values assigned or special keyword constructs. The input data is case-insensitive regarding keywords. Filenames input may be case-sensitive depending on computer system requirements and is always case sensitive on UNIX systems.

Sub-blocks can also be included between blocks, for example the SINK sub-block is used within the GEOMETRY block.

6.4.4.1 CONTROL block definition

The general structure/syntax of the CONTROL block is given below. A description of each keyword is given.

```
Keyword                      Variable type   Comment
BEGIN_BLOCK CONTROL
    {nuclidedef}+
    RELEASE_TYPE             A*:itype        #  ITYPE = [MOL|BQ]
    {chaindef}+
    CANISTER_TYPE            A*:ican         #  ICAN = [LEAD_FILLED| CUFE]
    TSTART                   F*              #  AGEO                          >-SMALL
    INCREMENT_FACTOR         F*              #  RAT                           >-SMALL
    NUM_STEP                 I*              #  NTERM                         >-SMALL
    CONV_EPS                 F*              #  EPS                           >-SMALL
    EWT                      F*              #  EWT                           >-SMALL
    NLOOP                    I*              #  NLOOP                         >-SMALL
    CASENAME                 A*              #  CASE
    [ADVECTION]                              #  IWFLOW
    [SPECIAL_U_ISOTOPES]                     #  ISPSOL
    [AER_CONSTANT]           F*              #  }Only one AER keyword may be used
    [AER_DECAY]              F*              #  }for each problem
    [AER_EXPLICIT]           F*              #  }K1AER
END_BLOCK
#
nuclidedef = {
BEGIN_DEF NUCLIDE
    NAME                     A*              #  NAMES, for example I129
    HALF_LIFE                F*              #  TAU                                > 0
    SOL_TYPE                 A*:ispec        #  ISPEC = [OWNSOL| FUELSURFACE |   MATRIX]
    switch (SOL_TYPE)
    case (FUELSURFACE)
    case (MATRIX and one of
    the AER_ keywords used)
    IRF                      F*              #  CIRF
    endswitch
END_DEF
}
#
    {chaindef} = {
                             CHAIN MOTHER DAUGHTER+ END_CHAIN
}
```

RELEASE_TYPE	Units in which the results will be given. It is specified as MOL for molar units and Bq for activity units.
CANISTER_TYPE	[LEAD_FILLED| CUFE]. Type of source: LEAD_FILLED for a canister with a negligible capacity for the backfill material in its interior, as in a copper/lead canister. CUFE for a canister where the capacity of its interior is considerable, as in a copper/iron canister.
TSTART	Time for the first printed results.
INCREMENT_FACTOR	Ratio between two successive times in output timeseries.
NUM_STEP	Number of printouts required. For different reasons, the code sometimes needs to take smaller time increments

	than defined by INCREMENT_FACTOR. This can often create output time series with more time steps than defined by NUM_STEP.
CONV_EPS	On input, the requested relative accuracy in the calculations of the solver. EPS = 0 is allowed. On output, the adjusted relative accuracy if the input value was too small.
EWT	Problem zero, i.e., the smallest physically meaningful value for the solution. Setting EWT to zero provides pure relative error control.
NLOOP	Number of times the solver is called if there is no satisfactory solution.
CASENAME	Name for identifying model case. Used to define the names of input and output files.
ADVECTION	Optional keyword indicating if advection should be modelled. Not fully developed.
SPECIAL_U_ISOTOPES	Optional keyword for shared solubilities. Not implemented yet.

Three different models of alpha-radiolytically-induced spent fuel dissolution are available in NUCTRAN. The models are described in Section 4 of this report. The models are selected by including a keyword-value pair in the CONTROL block of the system.dsc file; a different keyword is used to select each of the models. Only one of the keywords can be used for each simulation. If two or more of the keywords are found, an error message will be given and the code will stop.

AER_CONSTANT x	x is the value of the constant K_{CON} in: $\dfrac{\partial M}{\partial t} = -K_{CON} - \lambda M$
AER_DECAY x	x is the values of the constant K_{DEC} in: $\dfrac{\partial M}{\partial t} = -K_{DEC} \sum_{i=1}^{4} A_i \exp(-\dfrac{t \ln 2}{B_i}) - \lambda M$.
AER_EXPLICIT x	x is the values of the constant K_{EXP} in: $\dfrac{\partial M}{\partial t} = -K_{EXP} \sum_{i=1}^{4} C_i M_i - \lambda M$. If this model is used, one or more of the alpha-emitting nuclides Am-241, Pu-239, Pu-240 and Np-237 should be included in the simulation.

For each of the three alpha-radiolysis models, U-238 must be defined as SOL_TYPE MATRIX and must be in the first group defined in the CONTROL block. Any nuclides that are assumed to dissolve congruently from the fuel matrix should also be defined as SOL_TYPE MATRIX. An IRF value must be included for each nuclide that is defined as SOL_TYPE MATRIX when an alpha-radiolysis model is used. When MATRIX is used without an alpha-radiolysis model, IRF values must not be specified. The alpha radiolysis model underestimates ingrowth of U-238. This is unlikely to introduce a significant error since the amount of U-238 will usually be much greater than the amount of the U-238 parent nuclide.

NUCLIDE DEFINITION

NAME	Name of the nuclide.
HALF_LIFE	Nuclide half-life.
SOL_TYPE	[OWNSOL| FUELSURFACE| MATRIX]. Dissolution model to use within the canisters. Use OWNSOL for nuclides that are not embedded in the fuel matrix, FUELSURFACE when only a fraction of the nuclide is available for release and MATRIX when nuclides are embedded in the fuel matrix. *All nuclides in a CHAIN-definition must have the same SOL_TYPE.* When MATRIX is used for any of the nuclides, the chain that contains U-238 must be the first chain defined in the CONTROL block.
IRF	Instantaneous release fraction for this nuclide. Only used when: (i) SOL_TYPE is set to FUELSURFACE or (ii) SOL_TYPE is set to MATRIX and an alpha-radiolysis model is used. Note: for the FUELSURFACE model, the IRF specified for the first nuclide in the chain is used for all nuclides in the chain.

Summary of data required for each dissolution model

Description	SOL_TYPE [note 1]	IRF	AER keyword
Dissolution rate controlled by the solubility of each radionuclide.	OWNSOL	No	No
Dissolution rate controlled by the solubility of each radionuclide. Only a fraction of the nuclide inventory is available for release.	FUELSURFACE	Yes [note 2]	No
Radionuclides dissolve congruently with the fuel.	MATRIX [note 3]	No	No
The fuel dissolution rate is controlled by a constant rate of alpha radiolysis. Radionuclides dissolve congruently with the fuel.	MATRIX [note 3]	Yes	AER_CONSTANT
The fuel dissolution rate is controlled by a variable rate of alpha radiolysis. Decay of alpha emitters is represented in the model. Radionuclides dissolve congruently with the fuel. [note 4]	MATRIX [note 3]	Yes	AER_DECAY
The fuel dissolution rate is controlled by a variable rate of alpha radiolysis. Decay and transport of alpha emitters are represented in the model. Radionuclides dissolve congruently with the fuel. [notes 4 and 5]	MATRIX [note 3]	Yes	AER_EXPLICIT

Notes:

1. All nuclides in a chain must have the same SOL_TYPE.
2. The IRF value specified by the user for the first nuclide in a chain will be used for all nuclides in that chain when the FUELSURFACE model is used.
3. The chain containing U238 must be the first chain defined. The U238 chain must be defined as SOL_TYPE MATRIX.
4. The expression used for this model is only valid for a specific fuel. See reference [11] for more details.
5. One or more of the following nuclides must be included as SOL_TYPE MATRIX when this option is used: Am-241, Np-237, Pu-239, Pu-240.

CHAIN DEFINITION

CHAIN	Keyword indicating start of a decay chain.
MOTHER	The first modelled nuclide in a decay chain.
DAUGHTER	The nuclide(s) following in the decay chain. The list of the daughter nuclide must be given in the correct decay order for example:

For example, "CHAIN Am241 Np237 U233 Th229 END_CHAIN"

Where Am241 is modelled as the mother nuclide and Np237, U233 and Th229 are the daughter nuclides in the given order. The maximum number of nuclides in a chain is defined by parameter NUN (currently 6) and no nuclide can be defined in more than one CHAIN definition.

6.4.4.2 Limitations

A group is either a CHAIN or a single nuclide defined by BEGIN_DEF NUCLIDE but not used in any CHAIN definition. For example, defining the nuclides C14, I129 and the CHAIN:

   "CHAIN Am241 Np237 U233 Th229 END_CHAIN"

corresponds to 3 groups. C14 is one group, I129 is the second group and the CHAIN is the third group. It is very important to notice that every nuclide in a group *must* have the same SOL_TYPE.

6.4.4.3 GEOMETRY block definition

The general structure/syntax of the GEOMETRY block is given below. A description of each keyword is given. The contents of the GEOMETRY block define which materials are included in the simulation, typically water, bentonite, sand-bentonite and rock. There are also keywords for describing the canister and its properties.

The basic entities for building a model with NUCTRAN are the model block. The use of the term "block" should not be confused with the input data blocks CONTROL and GEOMETRY. The model block consists of one or more connected compartments of the same material with identical properties. For example, the bentonite barrier around the canister can be divided in to several blocks, where each block can be compartmentalized or subdivided into compartments. Every model block is given an identification number that is later used to define the connections between the model blocks.

The connections between the model blocks are also given inside the GEOMETRY block. This defines which model blocks are connected to each other and can thus exchange nuclides. The final part of the GEOMETRY block is the SINK DEFINITIONs. This section defines the locations in the model where nuclides can escape from the near-field and the flow rates that the individual sinks should have.

Keyword	Variable type		Comment	
BEGIN_BLOCK GEOMETRY				
{*materialdef*}+				
{*blockdef*}+				
{*connectiondef*}+				
{*sinkdef*}+				
TUBE_VOLUME	F*	#	VTUBE	> 0
NUM_ TUBE	I*	#	NTUBES	> 0
TUBES_DAMAGED	I*	#	NTDAM	> -1
VOID	F*	#	VHEMA	> -SMALL
GROWTH_TYPE	A*:*marea*	#	MAREA [NONE| RAMP| STEP]	
switch (MAREA)				
case RAMP or STEP:				
T_ZERO	F*	#	TINIT	> -SMALL
A_ZERO	F*	#	AZERO	> -SMALL
T_LIMIT	F*	#	TMAXAR	> -SMALL
A_LIMIT	F*	#	AMAXAR	> -SMALL
endswitch				
END_BLOCK				
#				
materialdef = {				
BEGIN_DEF MATERIAL		#	NMAT	
MATERIAL_NAME	A*:*mater*	#	MATNAM = [WATER; BENTONITE; SAND-BENTONITE; ROCK]	
DENSITY	F*	#	DENSM	> 0
POROSITY	F*	#	PORM	1> PORM > 0
END_DEF				
}				
#				
blockdef = {				
BEGIN_DEF BLOCK		#	NBLOCK	
BLOCK_NUMBER	I*	#	BLOCK	
MATERIAL_NAME	A*:*mater*	#	MATNAM= [WATER; BENTONITE;.SAND-BENTONITE; ROCK]	
NUM_Z_COMP	I*	#	NZ	
NUM_R_COMP	I*	#	NR	
[AREA_CHANGE]	A*:*nbtd*	#	NBTD = [NO| YES]	
{*discretizationdef*}+		#	Either 1 or NZ*NR times	

```
    [ADVECTION]                        #   IQFLOW
    switch (IQFLOW)
    case (TRUE):
    FLOW_RATE              F*          #   QFLOW
    INIT_CONC              F*          #   CINPUT
    INLET_NODE             I*          #   NWIN
    OUTLET_NODE            I*          #   NWOUT
    endswitch
discretizationdef = {
    BEGIN_DEF DISCRETIZATION
        Z_LENGTH           F*          #   ZBLEN                          > 0
        Z_AREA             F*          #   ZBAR                           > 0
        R_LENGTH           F*          #   RBLEN                          > 0
        R_AREA             F*          #   RBAR                           > 0
    END_DEF
END_DEF
}
#
connectiondef = {-
BEGIN_DEF CONNECTION               #   NCB
    FROM_BLOCK             I*          #   IBA                            > 0
    TO_BLOCK               I*          #   IBB                            > 0
    COUPLES                I*          #   NCON                           > 0
    FROM_DIRECTION         A*:irza     #   IRZA = [AXIAL| RADIAL]
    TO_DIRECTION           A*:irzb     #   IRZB = [AXIAL| RADIAL]
    FROM_RESISTANCE        A*:icra     #   ICRA = [FALSE| TRUE]
    TO_RESISTANCE          A*:icrb     #   ICRB = [FALSE| TRUE]
    [PLUG_POSITION]        A*:nplug    #   IPLUG, NPLUG = [NONE|
                                           FROM_BLOCK | TO_BLOCK]
    [EXTERNAL_RESIST]      A*:iradd    #   IRADD = [FALSE| TRUE]
    switch (IRADD)
    case TRUE:

    EXTERNAL_VALUE         F*          #   List of real values, one for each nuclide in the same
                                           order  as  in  BEGIN_DEF NUCLIDE  section
                                           END_LIST
    endswitch
    FROM_COMP              I*          #   IDCA
    TO_COMP                I*          #   IDCB
 END_DEF
}

sinkdef = {
BEGIN_DEF SINK                     #   NSINKS
    FROM_BLOCK             I*          #   IBS
    FROM_COMP              I*          #   ICS
    DIRECTION              A*:irz      #   IRZ = [AXIAL| RADIAL]
    PLUG_AT_INLET          A*:icrs

                                       #   ICRS = [NO| YES]
    PLUG_LENGTH            F*          #   ZRFRA
    PLUG_AREA              F*          #   PRFRA
    QEQ_FACTOR             F*          #   QFAC                           > 0.0
    QEQ_EXPONENT           F*          #   QEXP                           > 0.0
END_DEF
}
#
```

GEOMETRY DEFINITION

TUBE_VOLUME	Water volume in one zircaloy tube.
NUM_TUBE	Number of tubes in the canister (source).
TUBES_DAMAGED	Number of tubes damaged in the canister (source).
VOID	Water volume inside the canister.

Note: The volume of the source is calculated by:

Volume = TUBE_VOLUME*TUBES_DAMAGED + VOID

GROWTH_TYPE	Growth of the hole. There are three alternatives:
	    NONE = no growth of the hole.
	    RAMP = linear function for the growth.
	    STEP = step function for the growth.
	When RAMP or STEP is selected the following four additional keywords must be given:

T_ZERO	Breaching time for the canister.
A_ZERO	Initial size of the rupture.
T_LIMIT	Time at which the hole reaches, or jumps to, its maximum size.
A_LIMIT	Maximum size for the hole.

MATERIAL DEFINITION

The MATERIAL definition is repeated once for each material used in the model.

MATERIAL_NAME	Defines the material type. Available options are [WATER| BENTONITE| SAND-BENTONITE| ROCK]
DENSITY	Density of the material.
POROSITY	Porosity of the material.

BLOCK DEFINITION

The BLOCK definition is repeated once for each block used in the model.

BLOCK_NUMBER	Block identity number, a unique integer number for each model block.
MATERIAL_NAME	Define the material in the present model block.
NUM_Z_COMP	Number of subdivisions in the z-axis directions. That is, the number of compartments vertically in this model block.
NUM_R_COMP	Number of subdivisions in the r-axis. That is, the number of compartments horizontally in this model block. Usually, one or both of NUM_Z_COMP and NUM_R_COMP are equal to 1.
[AREA_CHANGE]	Optional keyword indicating the contact area of this model block will change size. Used to model increased hole size for the damage in the canister.

[ADVECTION]	Optional. Default is no advection.
[FLOW_RATE]	If ADVECTION selected, defines the flow of water through the block.
[INIT_CONC]	If ADVECTION selected, concentration in incoming water, usually zero. Can only be non-zero if only one nuclide is modelled.
[INLET_NODE]	If ADVECTION selected, node number for inlet.
[OUTLET_NODE]	If ADVECTION selected, node number for outlet.
Z_LENGTH	Diffusion length in the z-axis. The vertical length of the model block.
Z_AREA	Cross-sectional area for transport in the z-direction.
R_LENGTH	Diffusion length for transport in the r-direction. The horizontal length of the model block.
R_AREA	Cross-sectional area for transport in the r-direction.

CONNECTION DEFINITION

FROM_BLOCK	Number (BLOCK_NUMBER) of the first model block in this connection.
TO_BLOCK	Number (BLOCK_NUMBER) of the second model block in this connection.
COUPLES	Number of connections or couples of compartments involved in the connection between two blocks. FROM_COMP and TO_COMP have to be repeated COUPLES times.
FROM_DIRECTION	Direction in which the transport of the species occurs from first block in connection. [AXIAL| RADIAL].
TO_DIRECTION	Direction in which the transport of the species occurs to second block in connection. [AXIAL| RADIAL].
FROM_RESISTANCE	Set to TRUE if the resistance from the first block is to be included in the coupling resistance and to FALSE otherwise.
TO_RESISTANCE	Set to TRUE if the resistance from the second block is to be included in the coupling resistance and to FALSE otherwise.
[PLUG_POSITION]	Addition of a plug (an extra transport resistance between the model blocks) between two blocks is indicated by an index block. A particular case is a small hole connected to a large compartment. For example, the couple formed by the hole in the canister wall and the bentonite compartment outside the canister. This keyword must give the number of the block with the largest capacity associated to the plug.
[EXTERNAL_RESIST]	Set to TRUE if there exist resistances added by the user and not defined by the code [FALSE| TRUE]. Normally set to FALSE.
[EXTERNAL_VALUE]	If EXTERNAL_RESIST is set to TRUE this keyword defines the list of values (one value for each nuclide) for resistances added by the user and not defined in the system.

FROM_COMP	The number for the compartment inside the first model block that is part of the connection. This and the next keyword have to be repeated as many times as defined by COUPLES.
TO_COMP	The number for the compartment inside the second model block that is part of the connection.

SINK DEFINITION

FROM_BLOCK	Number (BLOCK_NUMBER) of the model block in contact with the sink.
FROM_COMP	Number of the compartment in contact with the sink.
DIRECTION	Direction in which the transport of the species occurs. [AXIAL| RADIAL].
PLUG_AT_INLET	Contribution to the coupling resistance R_{SW} of the compartment in contact with the sink [NO| YES].
PLUG_LENGTH	Length of the plug (extra resistance) added to the connection between IDCS and the sink.
PLUG_AREA	Cross-sectional area of the plug added to the connection between compartment and the sink.
QEQ_FACTOR	See below.
QEQ_EXPONENT	See below.

The QEQ_FACTOR and QEQ_EXPONENT values are used to calculate the equivalent groundwater flow at the sink according to the formula:

Qeq = QFactor. $q^{Qexponent}$

where *q* is the groundwater flux at the canister position obtained from HYDRASTAR.

6.4.4.4 Limitations

The maximum number of material definitions that can appear in the system.dsc file is defined by the parameter MATER in file nvol23.inc. MATER is set to 4 in the latest version of COMP23.

The maximum number of block definitions that can appear in the system.dsc file is defined by the parameter NBOX in file nvol23.inc. NBOX is set to 10 in the latest version of COMP23.

The maximum number of connection definitions that can appear in the system.dsc file is defined by the parameter NVOL*2 in file nvol23.inc. NVOL is set to 20 in the latest version of COMP23.

The maximum number of sink definitions that can appear in the system.dsc file is defined by the parameter NPATH in file nvol23.inc. NPATH is set to 7 in the latest version of COMP23.

6.4.4.5 System commands

The SYSTEM_COMMANDS may be inserted in the input data at any point outside a block. The format of a SYSTEM_COMMAND is:

SYSTEM KEYWORD

The SYSTEM keywords available in the latest version of COMP23 are:

LIST_DICTIONARY	This command toggles the flag that controls listing of a dictionary[1] and the accompanying wordlists whenever a dictionary is loaded. A subsequent LIST_DICTIONARY command will turn the listing off. This option is off by default.
LIST_ALL_DICTIONARIES	At the instant the LIST_ALL_DICTIONARIES command is encountered, all dictionaries with accompanying wordlists will be listed.
IGNORE_FATAL_ERRORS	This command toggles the IFATDO flag. If a fatal error is encountered the program will stop if the flag is off. If the flag is on the program stops after reading all input data. The flag is off by default.
SAVE_SCRATCH_FILES	This command toggles the ISAVSC switch. It allows temporary files used by the code to be retained (they are normally deleted when the code finishes executing).
IGNORE_ERRORS	This command toggles the IGNERR flag. If an error is encountered (validity, consistency, syntax or existence) in the input data the program will stop after reading all input if the flag is off. If the flag is on the program continues. The flag is off by default.
SKIP_USER_INTERFACE	This command sets IGNERR flag to 1.
WRITE_INPUT_FILE	This flag is intended for debugging purposes. If turned on the code will generate a NUCTRAN input file of the old format (via routine WOFC23). That is a FORTRAN formatted data file. The flag is off by default.
NO_SIMULATION	This command sets the INOSIM flag to 1. This is not used in the latest version of COMP23.
READ_INVENTORY_IN_EACH_LOOP	This command causes the *casename*.inv file to be read for each simulation. When this flag is not set, the file will be read only once.

---

[1]Whenever a new block is to be read, the dictionary for that block is loaded. The dictionary consists of the keywords and associated wordlists that are valid in the current block.

OUTPUT_DEBUG_FILES	This flag is intended for debugging purposes. If turned on, the code will generate four files that give the following information: a summary of the properties of the compartments, including capacitances and resistances; the release rate of each nuclide; the aqueous and solid amount of each nuclide; and, flow rate fractions for each nuclide. The flag will also cause debug information to be written to standard output.
SKIP_USER_INTERFACE	If this command is found the rest of the input data block is skipped and the code continues with the main NUCTRAN code.

6.5   HUI output

All output is sent through the routine WLIN11. This means that currently all output is sent to the standard output. A description of the output format used by HUI is given in Appendix 2.

7 Example

7.1 Description of problem

Release of U-238 and Pu-239 from the KBS-3 repository type will be calculated. A view of this repository is showed in Fig. 3, including a view of the canister and the small hole in the canister wall. These nuclides escape from a copper/iron canister through a small hole into the bentonite by diffusion. In the bentonite, they migrate through various pathways into the water flowing in the rock. As this repository has been well described in several SKB-reports, details on this repository will not be given here.
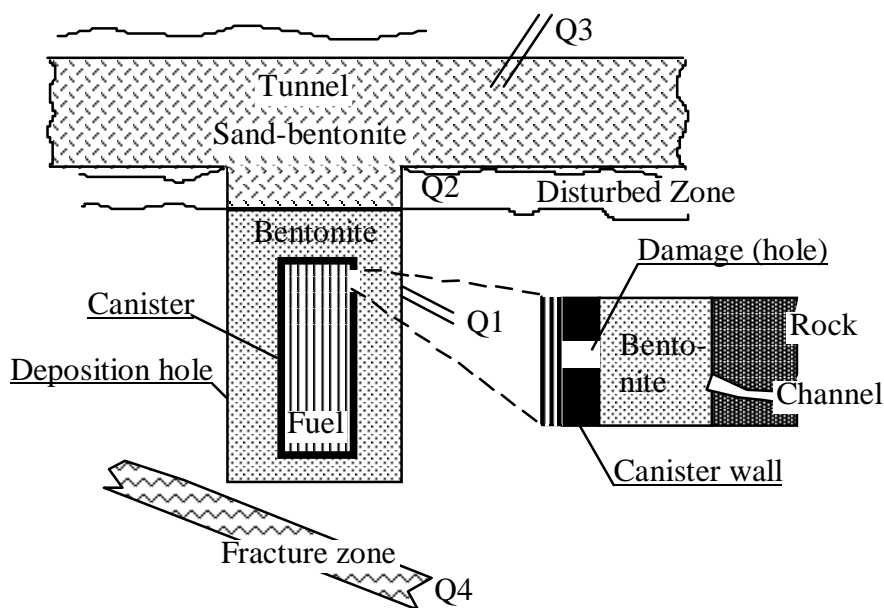
Figure 3   Schematic view of the KBS-3 repository design, showing the small hole in the canister and the location of the various escape routes.

7.2 Compartmentalization of the KBS-3 repository

The compartmentalization of the barrier system in the KBS-3 repository is made considering the geometry of the system and the materials through which the species are transported. The material, the transport properties and the dimensions of the compartment define the transport in a compartment. They are used by NUCTRAN to calculate the capacity of the compartment and the transport resistance in each transport direction. The capacity of a compartment includes the nuclide either in its water volume or sorbed in/on the solid. The volume of the compartment and the distribution coefficient K in the compartment determines this term. The diffusion coefficient, diffusion length(s) and cross-sectional area(s) of the compartment determine the transport resistance. These resistances are used to determine the nuclide flow rate between coupled compartments.

In the compartmentalization, the geometry of the repository is simplified as shown in Fig. 4 for the bentonite surrounding the canister. It is clear that errors are introduced in the solution given by NUCTRAN, but these errors are minimized by an additional subdivision into a few compartments as is the case when the transport is controlled by sensitive points such as at the small hole in the canister wall for the KBS-3 repository.
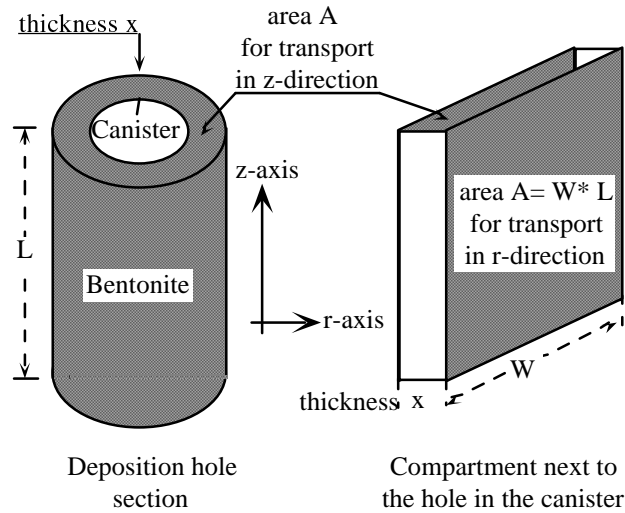
Figure 4   **A schematic view of the simplification process of the volume of bentonite surrounding the canister into a compartment. W is an average dimension (width).**

Figure 5 shows the coarse discretization of the whole repository. A finer compartmentalization next to the hole in the canister wall (bentonite surrounding the canister) may be made in order to get a better accuracy in the calculations of the early release. This finer subdivision into two to four compartments is important for the short-lived nuclides [1].
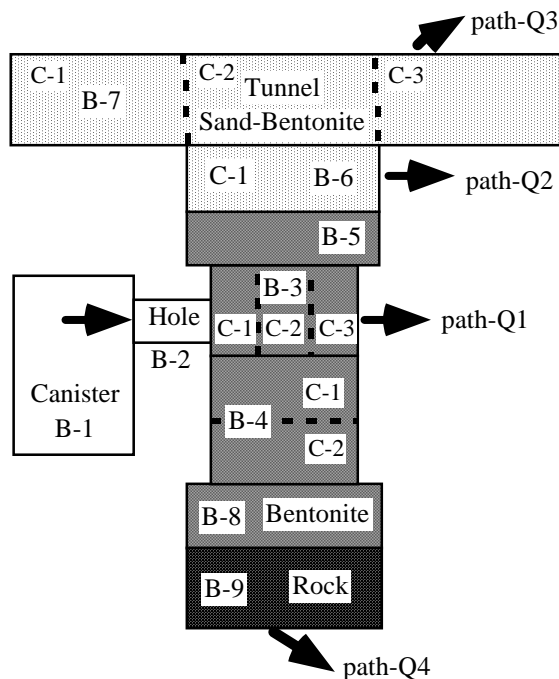
Figure 5 **The coarse compartmentalization of the KBS-3 repository, where each rectangle is a compartment and the dash lines are finer divisions of the blocks. The arrows show the various pathways followed by the species from the canister to the flowing water. (Not to scale.)**

7.3   INPUT file used for problem

The INPUT file to execute this sample problem is shown below; the results are shown in Figure 6 and 7.

This format of the INPUT file is valid for running NUCTRAN as a submodel of the PROPER package. The characteristic of this system is "to provide a module-type code package where submodels describing radionuclide transport, etc., are linked by the user and not by the programmer. The PROPER Monitor controls the execution given to the interconnected system of submodels". Only the parts that belong to NUCTRAN are included in this example. The data used in the calculations are tabulated in Table I.

System Description File system.dsc (applied to the PROPER system)

```
# Input file in the PROPER system
# U-238 and Pu-239, copper/iron canister, KBS-3.
BATCHES 1,1
SEED 930311
CPUTIME 00:15:00
VRMETHOD CMC:1
MACRO TS1HYDR11 = UxC1 UyC1 UzC1
# ---------------------------------------------------------
# PARAMETERS
# A and B... Denote nuclides;
# 1, 2, 3 , and 4....denote different materials
```

```
# S1, S2, S3, and S4....denote Qeqs for the various sinks
# Number of canisters (Standalone only)
    NCAN CONST ( 1 )
# NUMBER OF IDENTICAL CANISTERS IN THIS STREAM TUBE
    CSYNC1  UNIF (0.  1.)
    MULTIS1 COMPUTED
# Groundwater flux for NCAN canisters (standalone only)
    UC CONST( 1.0E-3 )
# PROBABILITY FOR AN INITIALLY DAMAGED CANISTER
# Probability that a canister is penetrated at emplacement
    PROB CONST(1.E-3)
    PITFAQ   CONST(1.)
    QS   CONST(1.)
    CONHS    CONST(1.)
    DEFFS    CONST(1.)
#
# Nuclide solubility, mol/m3
    CSOLA    CONST (2.0E-4)
    CSOLB    CONST (2.0E-5)
#
# Specific data to Material. Dif m2/yr; kd m3/kg
# 1 WATER
# 2 BENTONITE
# 3 BENTONITE-SAND
# 4 ROCK
#
# Specific to material 1
    DIFA1    CONST ( 0.123 )
    DIFB1    CONST ( 0.123 )
    KDA1 CONST ( 0.0 )
    KDB1 CONST ( 0.0 )
#
# Specific to material 2
    DIFA2    CONST ( 0.003154 )
    DIFB2    CONST ( 0.003154 )
    KDA2 CONST ( 3.0 )
    KDB2 CONST ( 3.0 )
#
# Specific to material 3
    DIFA3    CONST ( 0.003154 )
    DIFB3    CONST ( 0.003154 )
    KDA3 CONST ( 0.1 )
    KDB3 CONST ( 0.1 )
#
# Specific to material 4
    DIFA4    CONST ( 3.1536E-6 )
    DIFB4    CONST ( 3.1536E-6 )
    KDA4 CONST ( 3.0 )
    KDB4 CONST ( 3.0 )
#
# If any of the nuclides require IRF values,
# they should specified here:
# For example
#   IRFC14   CONST (0.5)
END PARAMETERS


# ------------------------------------------------------------
MODULE COMP23 (HYDR11: TS1HYDR11 ; UB1 UB2)
INPAR
# The parameters are numbered following the order indicated below
# For instance:DIFA1 corresponds to GETP(5)
    NCAN
    CSYNC1
    MULTIS1
    UC
```

```
              PROB
              PITFAQ      QS              CONHS           DEFFS
              CSOLA       CSOLB
              DIFA1       DIFB1           KDA1            KDB1
              DIFA2       DIFB2           KDA2            KDB2
              DIFA3       DIFB3           KDA3            KDB3
              DIFA4       DIFB4           KDA4            KDB4
#
# Any nuclides with an IRF must have one parameter for the IRF value.
# These should be the last items in the INPAR section
END INPAR
#
# DATA
#
# Input file for COMP23 with HUI
# U-238 and Pu-239, copper/iron canister, KBS-3.
#
SYSTEM WRITE_INPUT_FILE
#
BEGIN_BLOCK GEOMETRY
   VOID          1.0
   TUBE_VOLUME 4.655E-5
   NUM_TUBE                        1440
   TUBES_DAMAGED                   1440
   GROWTH_TYPE STEP
   T_ZERO        0.0
   A_ZERO        5.0E-6
   T_LIMIT       5.0E+3
   A_LIMIT                         0.1E-0
#
#    ------ Materials definition
   BEGIN_DEF MATERIAL
      MATERIAL_NAME            WATER
      DENSITY   1000.0
      POROSITY  1.0
   END_DEF
   BEGIN_DEF MATERIAL
      MATERIAL_NAME            BENTONITE
      DENSITY                  2700.0
      POROSITY  0.25
   END_DEF
   BEGIN_DEF MATERIAL
      MATERIAL_NAME            SAND-BENTONITE
      DENSITY                  2280.0
      POROSITY  0.24
   END_DEF
   BEGIN_DEF MATERIAL
      MATERIAL_NAME            ROCK
      DENSITY                  2700.0
      POROSITY  0.005
   END_DEF
#
#    ------ Blocks definition
   BEGIN_DEF BLOCK
      BLOCK_NUMBER             1
      MATERIAL_NAME            WATER
      NUM_Z_COMP               1
      NUM_R_COMP               1
      AREA_CHANGE              NO
      BEGIN_DEF DISCRETIZATION
         Z_LENGTH              0.0E+0
         Z_AREA 1.0E+0
          R_LENGTH             0.0E+0
          R_AREA               1.0E+0
      END_DEF
```

```
END_DEF
BEGIN_DEF BLOCK
   BLOCK_NUMBER          2
   NUM_Z_COMP            1
   NUM_R_COMP            1
   AREA_CHANGE           YES
   MATERIAL_NAME         WATER
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           5.0E-2
      Z_AREA             5.0E-6
      R_LENGTH           0.0E+0
      R_AREA 1.0E+0
   END_DEF
END_DEF
BEGIN_DEF BLOCK
   BLOCK_NUMBER          3
   NUM_Z_COMP            1
   NUM_R_COMP            3
   AREA_CHANGE           NO
   MATERIAL_NAME         BENTONITE
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           0.5E+0
      Z_AREA 5.13E-1
      R_LENGTH           1.17E-1
      R_AREA 1.8E+0
   END_DEF
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           0.5E+0
      Z_AREA 5.13E-1
      R_LENGTH           1.17E-1
      R_AREA 2.2E+0
   END_DEF
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           0.5E+0
      Z_AREA 5.13E-1
      R_LENGTH           1.17E-1
      R_AREA 2.6E+0
   END_DEF
END_DEF
BEGIN_DEF BLOCK
   BLOCK_NUMBER          4
   NUM_Z_COMP            2
   NUM_R_COMP            1
   AREA_CHANGE           NO
   MATERIAL_NAME         BENTONITE
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           4.333E+0
      Z_AREA 1.539E+0
      R_LENGTH           0.0E+0
      R_AREA 1.0E+0
   END_DEF
END_DEF
BEGIN_DEF BLOCK
   BLOCK_NUMBER          5
   NUM_Z_COMP            1
   NUM_R_COMP            1
   AREA_CHANGE           NO
   MATERIAL_NAME         BENTONITE
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           1.5E+0
      Z_AREA             2.405E+0
      R_LENGTH           0.0E+0
      R_AREA 1.0E+0
   END_DEF
END_DEF
```

```
BEGIN_DEF BLOCK
   BLOCK_NUMBER          6
   NUM_Z_COMP            1
   NUM_R_COMP            1
   AREA_CHANGE           NO
   MATERIAL_NAME         SAND-BENTONITE
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           1.0
      Z_AREA 2.405E+0
      R_LENGTH           0.875E+0
      R_AREA 4.95E+0
   END_DEF
END_DEF
BEGIN_DEF BLOCK
   BLOCK_NUMBER          7
   NUM_Z_COMP            3
   NUM_R_COMP            1
   AREA_CHANGE           NO
   MATERIAL_NAME         SAND-BENTONITE
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           2.125E+0
      Z_AREA             1.224E+1
      R_LENGTH           1.97E+0
      R_AREA 2.64E+01
   END_DEF
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           1.75E+0
      Z_AREA             1.224E+1
      R_LENGTH           1.97E+0
      R_AREA 2.17E+01
   END_DEF
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           2.125E+0
      Z_AREA             1.224E+1
      R_LENGTH           1.97E+0
      R_AREA 2.64E+01
   END_DEF
END_DEF
BEGIN_DEF BLOCK
   BLOCK_NUMBER          8
   NUM_Z_COMP            1
   NUM_R_COMP            1
   AREA_CHANGE           NO
   MATERIAL_NAME         BENTONITE
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           0.5E+0
      Z_AREA 2.405E+0
      R_LENGTH           0.0E+0
      R_AREA 1.0E+0
   END_DEF
END_DEF
BEGIN_DEF BLOCK
   BLOCK_NUMBER          9
   NUM_Z_COMP            1
   NUM_R_COMP            1
   AREA_CHANGE           NO
   MATERIAL_NAME         ROCK
   BEGIN_DEF DISCRETIZATION
      Z_LENGTH           3.0
      Z_AREA 2.405E+0
      R_LENGTH           0.0E+0
      R_AREA 1.0E+0
   END_DEF
END_DEF
#
```

```
#     ------ Connection definition
#
   BEGIN_DEF CONNECTION
      FROM_BLOCK               1
         TO_BLOCK              2
      COUPLES   1
      FROM_DIRECTION           AXIAL
         TO_DIRECTION          AXIAL
      FROM_RESISTANCE          FALSE
         TO_RESISTANCE         TRUE
      PLUG_POSITION            NONE
      EXTERNAL_RESIST          FALSE
      FROM_COMP 1
         TO_COMP               1
   END_DEF
   BEGIN_DEF CONNECTION
      FROM_BLOCK               2
         TO_BLOCK              3
      COUPLES   1
      FROM_DIRECTION           AXIAL
         TO_DIRECTION           RADIAL
      FROM_RESISTANCE          TRUE
         TO_RESISTANCE          FALSE
      PLUG_POSITION            TO_BLOCK
      EXTERNAL_RESIST          FALSE
      FROM_COMP 1
         TO_COMP               1
   END_DEF
   BEGIN_DEF CONNECTION
      FROM_BLOCK               3
         TO_BLOCK              4
      COUPLES   3
      FROM_DIRECTION           AXIAL
         TO_DIRECTION          AXIAL
      FROM_RESISTANCE          TRUE
         TO_RESISTANCE         TRUE
      PLUG_POSITION            NONE
      EXTERNAL_RESIST          FALSE
      FROM_COMP 1
         TO_COMP               1
      FROM_COMP 2
         TO_COMP               1
      FROM_COMP 3
         TO_COMP               1
   END_DEF
   BEGIN_DEF CONNECTION
      FROM_BLOCK               3
         TO_BLOCK              5
      COUPLES   3
      FROM_DIRECTION           AXIAL
         TO_DIRECTION          AXIAL
      FROM_RESISTANCE          TRUE
         TO_RESISTANCE         TRUE
      PLUG_POSITION            NONE
      EXTERNAL_RESIST          FALSE
      FROM_COMP 1
         TO_COMP               1
      FROM_COMP 2
         TO_COMP               1
      FROM_COMP 3
         TO_COMP               1
   END_DEF
   BEGIN_DEF CONNECTION
      FROM_BLOCK               5
         TO_BLOCK              6
```

```
         COUPLES   1
         FROM_DIRECTION          AXIAL
            TO_DIRECTION         RADIAL
         FROM_RESISTANCE         TRUE
            TO_RESISTANCE        TRUE
         PLUG_POSITION           NONE
         EXTERNAL_RESIST         FALSE
         FROM_COMP               1
            TO_COMP              1
      END_DEF
      BEGIN_DEF CONNECTION
         FROM_BLOCK              6
            TO_BLOCK             7
         COUPLES                 1
         FROM_DIRECTION          AXIAL
            TO_DIRECTION         RADIAL
         FROM_RESISTANCE         TRUE
            TO_RESISTANCE        TRUE
         PLUG_POSITION           NONE
         EXTERNAL_RESIST         FALSE
         FROM_COMP 1
            TO_COMP              2
      END_DEF
      BEGIN_DEF CONNECTION
         FROM_BLOCK              4
            TO_BLOCK             8
         COUPLES                 1
         FROM_DIRECTION          AXIAL
            TO_DIRECTION         AXIAL
         FROM_RESISTANCE         TRUE
            TO_RESISTANCE        TRUE
         PLUG_POSITION           NONE
         EXTERNAL_RESIST         FALSE
         FROM_COMP               2
            TO_COMP              1
      END_DEF
      BEGIN_DEF CONNECTION
         FROM_BLOCK              8
            TO_BLOCK             9
         COUPLES                 1
         FROM_DIRECTION          AXIAL
            TO_DIRECTION         AXIAL
         FROM_RESISTANCE         TRUE
            TO_RESISTANCE        TRUE
         PLUG_POSITION           NONE
         EXTERNAL_RESIST         FALSE
         FROM_COMP               1
            TO_COMP              1
      END_DEF
#
#   ------ Sink definition
#
   BEGIN_DEF SINK
      FROM_BLOCK                 3
      FROM_COMP 3
      DIRECTION RADIAL
      PLUG_AT_INLET              NO
      PLUG_LENGTH                5.0E-4
      PLUG_AREA                  5.5E-4
      QEQ_FACTOR                 0.25
      QEQ_EXPONENT               1.0
   END_DEF
   BEGIN_DEF SINK
      FROM_BLOCK                 6
      FROM_COMP 1
```

```
          DIRECTION                RADIAL
          PLUG_AT_INLET            YES
          PLUG_LENGTH              0.0
          PLUG_AREA 1.0
          QEQ_FACTOR               4.0
          QEQ_EXPONENT             1.0
      END_DEF
    BEGIN_DEF SINK
          FROM_BLOCK               7
          FROM_COMP 3
          DIRECTION RADIAL
          PLUG_AT_INLET            NO
          PLUG_LENGTH              5.0E-4
          PLUG_AREA 5.97E-3
          QEQ_FACTOR               0.5
          QEQ_EXPONENT             1.0
      END_DEF
    BEGIN_DEF SINK
          FROM_BLOCK               9
          FROM_COMP 1
          DIRECTION AXIAL
          PLUG_AT_INLET            YES
          PLUG_LENGTH              0.0
          PLUG_AREA 1.0
          QEQ_FACTOR               6.0
          QEQ_EXPONENT             1.0
      END_DEF
END_BLOCK
#
BEGIN_BLOCK CONTROL
    CASENAME                     sr95test
    RELEASE_TYPE                 Bq
    CANISTER_TYPE                CUFE
    TSTART       5.0
    INCREMENT_FACTOR             2.5
    NUM_STEP                     50
    CONV_EPS                     1.0E-6
    EWT                          1.0E-18
    NLOOP                        3
#  An AER_keyword would appear in this block.
    BEGIN_DEF NUCLIDE
       NAME      U238
       HALF_LIFE 4.47E9
       SOL_TYPE  OWNSOL
    END_DEF
    BEGIN_DEF NUCLIDE
       NAME                      PU239
       HALF_LIFE 2.41E4
       SOL_TYPE  OWNSOL
    END_DEF
END_BLOCK
#
END DATA

    STATISTICS
#
    END STATISTICS
#
END COMP23
```

7.4 Results

The predicted release by NUCTRAN for U-238 and Pu-239 are shown in Figs 6 and 7 respectively. These releases were calculated assuming that the only limitation for the nuclides to be released is the solubility, i.e. the solubility limit approach (OWNSOL type).

Table I Data used in the calculations of the release

Data on the KBS-3 repository design

Deposition hole	Diameter [mm]	1 750
	Length [mm]	7 833
Canister	Diameter [mm]	1 050
	Thickness [mm]	50
	Length [mm]	4 833
Tunnel	Width [mm]	3 400
	Height [mm]	4 000
Distance between canister centres [mm]		6000
Canister bottom to deposition hole bottom [mm]		500
Thickness of the disturbed zone [mm]		1 000
Deposition hole bottom to fracture zone [mm]		3 000

Hydraulic properties

Gradient [m/m]		$3 \cdot 10^{-3}$		
Fracture aperture [mm]		0.1		
	path-Q1	path-Q2	path-Q3	path-Q4
Flow porosity	$2 \cdot 10^{-5}$	10^{-3}	$1.7 \cdot 10^{-5} 10^{-3}$	10^{-3}
Conductivity [m/s]	10^{-9}	10^{-7}	10^{-9}	10^{-6}
Flux [m^3/m^2, year]	$9.5 \cdot 10^{-5}$	$9.5 \cdot 10^{-3}$	$9.5 \cdot 10^{-5}$	$9.5 \cdot 10^{-2}$
Qeq [l/year]	0.25	4	0.5	6

Material properties

	Bentonite	Sand-bentonite	Rock
Density [kg/ m^3]	2 700	2 280	2 700
Porosity [%]	25	24	0.5

Data on transport properties

Nuclide	Solubility [mol/l]	Half-life [yr]	Bentonite D_e [m^2/s]	Bentonite K_d [m^3/kg]	Sand-bentonite D_e [m^2/s]	Sand-bentonite K_d [m^3/kg]	Rock D_e [m^2/s]	Rock K_d [m^3/kg]
U-238	10^{-4}	$4.47 \cdot 10^9$	10^{-10}	3.0	10^{-10}	0.1	10^{-13}	3.0
Pu-239	10^{-5}	24 100	10^{-10}	3.0	10^{-10}	0.1	10^{-13}	3.0

Figure 6   Release of U-238 from the repository for a copper/iron canister with a suddenly rupture of the canister at 10^3 years after deposition. See Fig. 3 or 5 for the location of the various paths.
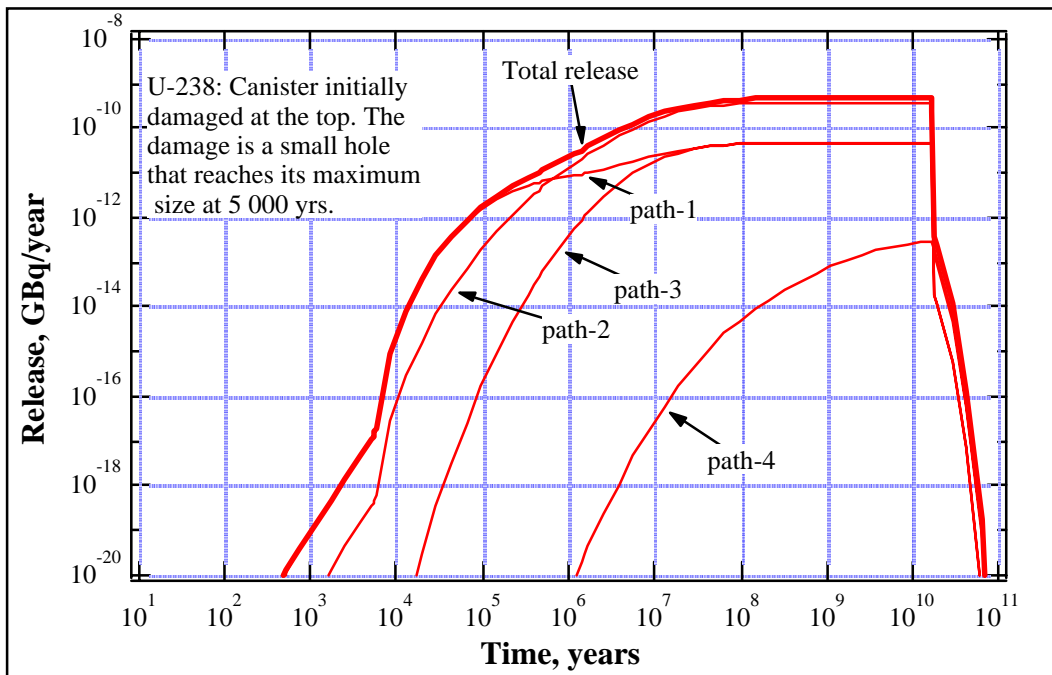

Figure 7   Release of Pu-239 from the repository for a copper/iron canister with a suddenly rupture of the canister at 10^3 years after deposition. See Fig. 4 or 5 for the location of the various paths.
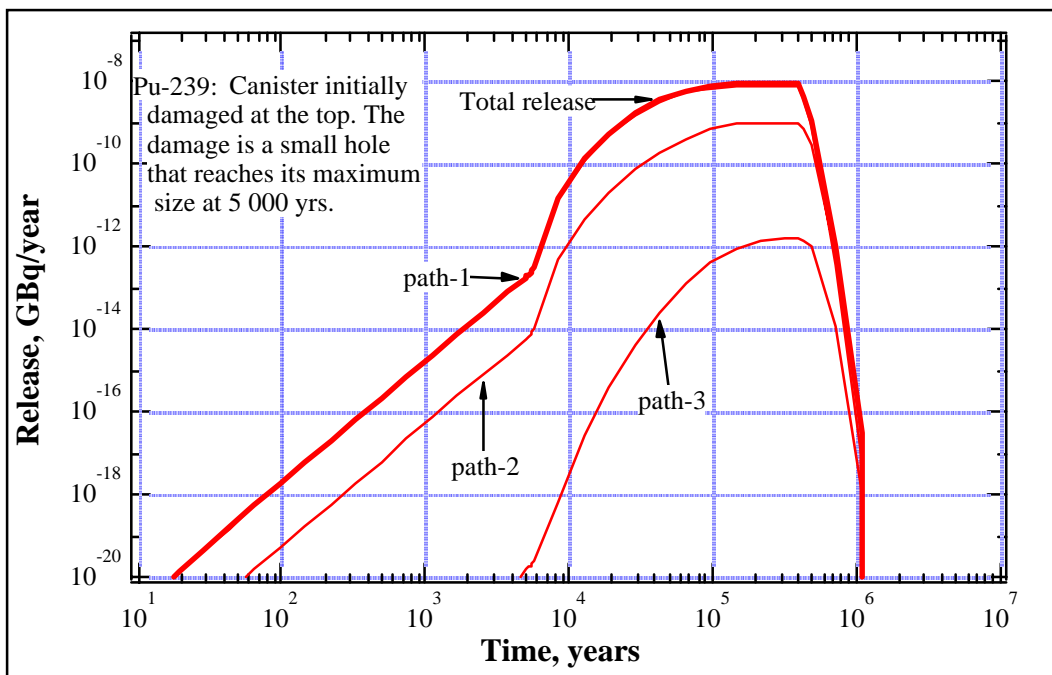
62

Notation

A	area	m^2
c	concentration	mol/m^3
C	capacity of the compartment	mol/m^3
d	diffusion length	m
D_e	effective diffusivity	$m^2/year$
D_w	diffusivity in free water	$m^2/year$
k_d	sorption coefficient	m^3/kg
K	distribution coefficient	-
L	length	m
m	mass of solid in the unit of volume	mol/m^3
M	amount of nuclide in the solid phase (inventory)	mol
N	dissolution rate or molar flow rate	mol/year
q_d	dissolution rate	mol/year
q_o or u_0	flux of water	$m^3/m^2/year$
Q_{eq}	equivalent flow rate	$m^3/year$
R	resistance to transport	$years/m^3$
r	radius	m
t	time	years
t_w	residence time	years
V	volume	m^3
W	width	m
x	diffusion length	m

Greek

ϕ	diameter	m
λ	decay constant	$year^{-1}$
δ	fracture aperture	m
ε	porosity	-
ε	flow porosity	-
$\bar{\eta}$	mean penetration thickness	m
ρ	density	kg/m^3

Subscripts

i	index compartment
j	index compartment
c	canister
s	solid inventory
matrix	fuel matrix
n	index nuclide in the chain
u	uranium

HUI output

The HUI output starts with the following lines of output:

```
===============================================================
```
HYDRASTAR User Interface - Initialize
Opened unit nn as file:
filename
```
===============================================================
```

and ends with the following lines of output:

```
===============================================================
```
HYDRASTAR User Interface Error Report
   Number of validity   errors :   nv
   Number of consistency   errors :   nc
   Number of existence   errors :   ne
   Number of syntax   errors :   nv
Input data is correct. Continuing ...
```
===============================================================
```

Output from the various input BLOCKs read are initiated by the following lines:

```
===============================================================
=     Start of block : block_identifier        =
=============================================================
```

and ended by:

```
=============================================================
=     End of block : block_identifier          =
=============================================================
```

Between these two sets there exists four types of output:
- Input line echoing.
- Input data echoing.
- Error messages.
- Definition block IO.

Definition block IO are initiated by the following lines:

```
--------------------------------------------------------------
-               Start of definition : definition_identifier        -
--------------------------------------------------------------
```

and ended by:

```
-------------------------------------------------------------------------------
-                 End of definition : definition_identifier           -
-------------------------------------------------------------------------------
```

Between these two sets there exists three types of output:
- Input line echoing.
- Input data echoing.
- Error messages.

Input data echoing and error messages are explained in the following section

Input line echoing has the form of:

Line nnn: The comment stripped left-justified valid input line

where nnn is the line number of the line read in the input file. The line numbers include blank lines and comment lines.

Input data echoing

Input data echoing takes the form:

=============== Keyword assignments are :

Keyword <KEYWORD1> assigned value(s) : value1 value2 ...
Keyword <KEYWORDn> assigned value(s) : value1 value2 ...

=============== End of Keyword assignments

The value list may continue over several lines. For keywords assigned filenames, the filenames are just listed in the order they were read separated by one blank.

Error messages
There are five types of error reports and one warning report.

Validity errors: If the values read for a specific keyword do not lie within the specified limits, the following message will be displayed:

* ******* VALIDITY error has occurred in routine ROUTINE*

*   after reading line HUINL.*
*   Check failed for CBOUN keyword <KEYWORD>.*
*   Upper bound Lower Bound   Input value*
*   u.uuuuE+ee l.lllllE+ee   r.rrrrE+ee/iiiiiiii*

* ******* End of Error*

where
- ROUTINE is the name of the routine that read the data.
- HUINL is the number of lines, including blank and comment lines, that have been read.
- CBOUN may be "upper bound in", "lower bound in" or "" (empty string)
- KEYWORD is the name of the keyword.
- Upper bound is the specified upper limit.
- Lower bound is the specified lower limit.
- REAL is the real value that has been read for the keyword.
- INT is the integer value that has been read for the keyword.

The last line is repeated for all values that have been read for the particular keyword.

Consistency errors: If a required keyword, or an optional keyword that depends on the value assigned to a required keyword, has not been found in the input file, the following message will be displayed:

*******CONSISTENCY error has occurred in routine ROUTINE after reading line HUINL.
These are:
Keyword <KEYWORD> Not set. Tok :TOKEN
Keyword <KEYWORD> Not set. Tok :TOKEN
depend on <KEYWORD=WORD> which has been set. Tok:TOKEN
Setting keyword <KEYWORD> has no effect
it depend on <KEYWORD=WORD> which has not been set.
******* End of Error

where
- ROUTINE is the name of the routine that read the data.
- HUINL is the number of lines, including blank and comment lines, that have been read.
- KEYWORD is the name of a keyword.
- TOKEN is the token value for the keyword, see "HYDRASTAR USER INTERFACE - Programmers reference".
- WORD is the word in a wordlist applicable to the keyword (see the Input Data Chapter).

Existence errors: If any of the files in a list of files specified for a specific keyword does not exist the following message will be displayed:

*******EXISTENCE error has occurred in routine ROUTINE after reading line HUINL.
Check failed for keyword <KEYWORD>.
File:(filename) Exist:YON
******* End of Error

where
- ROUTINE is the name of the routine that read the data.
- HUINL is the number of lines, including blank and comment lines, that have been read.
- KEYWORD is the name of the keyword.
- filename is the name of a file specified.
- YON may be either "YES" or "NO" (It may be YES since all files are listed if one or more files do not exist).

The last line is repeated for all files specified in the file list for the particular keyword.

Syntax errors: If a syntax error has occurred the following message will be displayed:

******* *SYNTAX error has occurred in routine ROUTINE after reading line HUINL.*

where
- ROUTINE is the name of the routine that read the data.
- HUINL is the number of lines, including blank and comment lines, that have been read.

System errors: A system error may be over-indexed arrays, etc. This type of error should not be very common to the normal user. If a system error has occurred the following message will be displayed:

******* *SYSTEM error has occurred in routine ROUTINE after reading line HUINL.*

where
- ROUTINE is the name of the routine that read the data.
- HUINL is the number of lines, including blank and comment lines, that have been read.

Warnings: A warning is a note to the user that there may something wrong in the input. A warning does not terminate the program. If a warning has occurred the following message will be displayed:

******* *WARNING in routine ROUTINE after reading line HUINL.*

where
- ROUTINE is the name of the routine that read the data.
- HUINL is the number of lines, including blank and comment lines, that have been read.

References

1. O. Olsson, G. Bäckblom, G. Gustafson, I. Rhén, R. Stanfors, and P. Wikberg, "The Structure of Conceptual Models with Application to the Äspö HRL Project," TR 94-08, Swedish Nuclear Fuel and Waste Management Company (1991).

2. T.N. Narasimhan and P.A. Witherspoon, "An Integrated Finite Difference Method for Analyzing Fluid Flow in Porous Media," Water Resour. Res., **12**, 57 (1976).

3. L. Romero, L. Moreno, and I. Neretnieks, "Fast Multiple-Path Model to Calculate Radionuclide Release from the Near Field of a Repository," Nucl. Technol., **112**, 89 (1995).

4. I. Neretnieks, "Leach Rate of High Level Waste and Spent Fuel-Limiting Rates as Determined by Backfill and Bedrock Conditions," Proc. Scientific Basis for Nuclear Waste Management V, Berlin, Germany, June 7-10. 1982, W.Lutze, Ed., p. 559 (1982).

5. I. Neretnieks, "Stationary Transport of Dissolved Species in the Backfill Surrounding a Waste Canister in Fissured Rock: Some Simple Analytical Solutions," Nucl. Technol., 72-194, 1986.

6. PROPER: "Proper Submodel Designer's Manual, Version 3.0, Edition I," Swedish Nuclear Fuel and Waste Management Company (1993).

7. D. Kahaner, C. Moler, S. Nash, "Numerical Methods and Software," Prentice Hall, Englewood Cliffs, New Jersey (1988).

8. SKB, "Proper Monitor User's Manual. Version 3.0, Edition I," Swedish Nuclear Fuel and Waste Management Company (1993).

9. A. Haworth, D.J. Ilett, A.M. Thompson and D.J. Worth, "Developments to the NUCTRAN Computer Program", (1996).

10. A. Haworth, A.M. Thompson and D.J. Worth, "Developments to the PROPER Version of NUCTRAN", (1997).

11. L. Were, P. Sellin, R. Forsyth, "Radiolytically Induced Oxidative Dissolution of Spent Nuclear Fuel", SKB Technical Report 90-08, 1990.

12. L. Romero, L. Moreno, I. Neretnieks, "Analysis of Radionuclide Release from the Canister" KAT 93/18, Dept. of Chem. Eng. and Technology, Royal Institute of Technology, Stockholm, Sweden (1993).

13. "User's guide to HYDRASTAR 1.5", February 1996.

14.   "TULLGARN User's Guide".

15.   R. Maddock, E. Hailwood, E. Rhodes, R. Muir Wood, "Direct fault dating trials at the Äspö Hard Rock Laboratory", SKB AR 93-24.