



SOLIDUSS: Solid-state diffusion software for radiation protection

F. Ogallar Ruiz^{a,b,*}, C. Theis^a, I. Porras^b, H. Vincke^a

^a European Organization for Nuclear Research (CERN), Geneva, Switzerland

^b Universidad de Granada, Granada, Spain

ARTICLE INFO

Keywords:

Out-diffusion
Fire
Radionuclide
Source term
CERN-FLUKA
SOLIDUSS

ABSTRACT

The out-diffusion of radionuclides from activated material in case of a fire may represent a non-negligible contribution to the radiological source term of such an event. In order to assess the contribution of this phenomenon, a software package has been designed and implemented. In the present document we briefly introduce the numerical treatment used to tackle the problem prior to the explanation of the software's logic. The document ends with an exemplary simulation and a study carried out to validate the implementation of the algorithm.

The presented tool has been named SOLIDUSS, it is mainly written in C++ and uses a Monte Carlo based approach to simulate the diffusion of radioisotopes within solid materials. It is designed to run coupled with CERN-FLUKA, taking advantage of its geometry kernel to carry out diffusion calculations in arbitrarily complex geometries. The user can provide 3D temperature maps along with many other parameters that allow the program to target a wide range of different scenarios. As results SOLIDUSS provides 3D radionuclide concentration maps as well as the amount of radionuclides out-diffused from the selected materials. So far, this software has undergone numerical validation which will be discussed in this paper. Benchmarking against experimental data is currently ongoing.

1. Introduction

A fire in an activated area, such as those of the CERN (European Organization for Nuclear Research) accelerator complex, could result in the release of radionuclides to the environment. In the past the combustion of activated material was considered as the only radionuclide release mechanism due to fire in most studies. Yet, the requirement of exhaustive radiological source terms for this type of scenarios has revealed the need to evaluate the contribution of a second mechanism: the out-diffusion of radionuclides due to the high temperatures reached in the fire and surroundings (*i.e.* radionuclides being liberated from their original host material). A complete neglect of this phenomenon would probably underestimate the radiological hazard but an assumed release scenario of 100% could be grossly over-conservative. This paper explains the problem and the tool we have designed which aims at more accurately estimating the contribution of radioisotope out-diffusion to radiological source terms in case of fires in arbitrary geometries. This was done in the context of the FIRIA (Fire Induced Radiological Integrated Assessment) project (FIRIA, 2018; Gai, 2021) at CERN, which aims to develop an integrated approach to quantitatively assess potential discharges of radioactive substances induced by a fire accident in particle physics' experimental facilities. To this end, state-of-the-art tools and methods are used to study the radioactive source term, fire

dynamics, evacuation, intervention time, environmental dispersion and dose assessment.

2. Theoretical summary

In this section we aim to give a general but not exhaustive introduction to diffusion in solids that shall help us to understand and eventually overcome the aforementioned problem. Diffusion in solids is a well-known topic adequately addressed by many sources. To name but one, Ref. Mehrer (2007) by H. Mehrer is regularly consulted by the authors. Also, a gentle introduction can be found online in Ref. Föll (2019) by H. Föll.

2.1. Diffusion

From a microscopic perspective, diffusion in solids is nothing other than *Brownian motion* of atoms or molecules. They move due to their thermal energy, which makes them vibrate around their equilibrium position following a Maxwell–Boltzmann velocity distribution. Most of the hops of the vibrating atoms will find the potential well driving the atoms back to their equilibrium position, but sometimes their energy

* Corresponding author at: European Organization for Nuclear Research (CERN), Geneva, Switzerland.

E-mail address: francisco.ogallar.ruiz@cern.ch (F.O. Ruiz).

will be sufficient to overcome it, ending up in a different location. The success rate of the jumps follows an exponential dependence on the medium temperature since the probability for an atom to be found with a thermal energy above E [J] is given by $\exp[-E/(k_B T)]$, with k_B [J K⁻¹] the Boltzmann's constant and T [K] the absolute temperature. As a result, atomic migration in a solid is, in general, the result of successive jumps of fixed lengths.

From the phenomenological perspective, the equations governing diffusion are known as *Fick's laws* (Crank, 1975). These laws describe the macroscopic manifestation of the microscopic processes outlined above. Fick's laws as they will be shown here are applicable to isotropic media, which is not a major constraint for our purposes. Fick's first law relates the diffusion particle flux $\vec{J}(\vec{r}, t)$ [cm⁻² s⁻¹] and the particle concentration gradient $\nabla C(\vec{r}, t)$ [cm⁻⁴] in a point \vec{r} for a given time t as shown below:

$$\vec{J}(\vec{r}, t) = -D\nabla C(\vec{r}, t). \quad (1)$$

The factor D [cm² s⁻¹] is known as the *diffusion coefficient* and it will depend, in general, on the diffusing particles, the host material and the temperature in a way that will be detailed later. Fick's second law, also called *diffusion equation*, relates the particle concentration evolution with its gradient divergence.

$$\frac{\partial C(\vec{r}, t)}{\partial t} = D\nabla^2 C(\vec{r}, t), \quad (2)$$

Eq. (2) is the one to be solved in order to calculate the temporal evolution of our radionuclide concentrations in case of fire.

The connection between microscopic and phenomenological visions can be illustrated using a very simple random walk model as shown in Appendix A.

2.2. Diffusion coefficient

The diffusion coefficient, D , tells us how fast a given species diffuses within a host material at a certain temperature. It has dimensions of [length²/time] and is commonly expressed in [cm² s⁻¹]. Typically, for metals it is a function of temperature following the Arrhenius formula as the one below:

$$D = A \exp\left(-\frac{Q}{RT}\right), \quad (3)$$

where A [cm² s⁻¹] is the so-called *frequency factor*, Q [J mol⁻¹] is the *activation energy*, T [K] is the absolute temperature and R [J K⁻¹ mol⁻¹] is the *gas constant*.¹ The *activation parameters* A and Q depend on the host materials and the diffusing species. Also, depending on the medium of consideration, the diffusion coefficient may not follow a simple Arrhenius behaviour, but is the addition of several of them.

3. Numerical treatment

In Appendix A it is shown that random walks emulate the diffusion process under certain assumptions. This well known result suggests that we can simulate diffusion processes using stochastic methods, avoiding the need for deterministic solutions of the diffusion equation often based on the use of Finite Elements Methods that for us would pose a number of difficulties – especially considering diffusion of radionuclides produced in an arbitrary geometry following the interaction of the primary particle beam with material – such as:

- The simulated geometry used to obtain the radionuclide inventory and its spatial distribution via particle physics simulation packages such as CERN-FLUKA (Battistoni et al., 2015; Bohlen et al., 2014) would need to be recreated from scratch as the

input format is not compatible. Finite Element packages generally create their meshes from standard CAD (Computer-Aided Design) models that are usually surface-based boundary representation models, whereas particle transport codes require a full volumetric description and typically use Boolean Combinatorial Solid Geometry (CSG) to describe complex objects.

- Results are susceptible to meshing quality, boundary conditions and step sizes of the implemented solver. Therefore, the results need to be accompanied by sensitivity studies assessing the impact of these parameters. For general studies, it would require users with experience in the handling of these tools, which is not common within the Radiation Protection (RP) community.
- Simulations have to be carried out with different packages, one for obtaining the radionuclide inventory and another one for simulating its diffusion, which requires data handling and a perfect duplicate of the simulated system, making the whole process more prone to errors.

Considering all the above, we deem the development of a Monte Carlo based diffusion model internally linked to CERN-FLUKA as the best strategy to overcome these difficulties.

3.1. Wiener process

To simply use random walks to simulate each single atomic jump of the radionuclides would obviously be straightforward but very cumbersome for reasons of computational effort. At the same time, it will not be necessary since we can infer a more suitable numerical treatment to be implemented in our tool. Let us have a look at the probability distribution associated to the radionuclide's position after $n \rightarrow \infty$ atomic jumps (Theis and Vincke, 2016; Evans, 2019; Lalley, 2019). Consider an atom that could jump to the right or to the left with the same probability a distance Δx every Δt seconds. Its position after n jumps would be $X(t) = S_n \Delta x$, with $S_n = \sum_{i=1}^n \epsilon_i$ and $\epsilon_i = \pm 1$ (both values with probability 1/2). From this we can see that:

$$\mu(\epsilon_i) = 0, \quad (4a)$$

$$\sigma^2(\epsilon_i) = 1. \quad (4b)$$

where $\mu(\epsilon_i)$ is the mean of the distribution associated to ϵ_i and $\sigma^2(\epsilon_i)$ its variance. Since ϵ_i are real, independent and identically distributed variables, and $\sigma^2(\epsilon_i)$ is positive, we know that because of the *Central Limit Theorem*, the following is true for all $-\infty < a < b < +\infty$:

$$\lim_{n \rightarrow \infty} P\left(a < \frac{S_n - n\mu(\epsilon_i)}{\sqrt{n}\sigma(\epsilon_i)} < b\right) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{x^2}{2}} dx. \quad (5)$$

Consequently the position of our atom after n jumps can be expressed as follows:

$$X(t) = S_n \Delta x = \left(\frac{S_n - n\mu(\epsilon_i)}{\sqrt{n}\sigma(\epsilon_i)}\right) \sqrt{n}\sigma(\epsilon_i)\Delta x, \quad (6)$$

where we have used Eq. (4a). We can see that:

$$\begin{aligned} & \lim_{n \rightarrow \infty} P(a < X(t) < b) \\ &= \lim_{n \rightarrow \infty} P\left(\frac{a}{\sqrt{\frac{t}{\Delta t}} \Delta x} < \frac{S_n - n\mu(\epsilon_i)}{\sqrt{n}\sigma(\epsilon_i)} < \frac{b}{\sqrt{\frac{t}{\Delta t}} \Delta x}\right). \end{aligned} \quad (7)$$

If we substitute $n = t/\Delta t$ and apply Eqs. (4b) and (5):

$$\begin{aligned} & \lim_{n \rightarrow \infty} P(a < X(t) < b) \\ &= \lim_{n \rightarrow \infty} P\left(\frac{a}{\sqrt{\frac{t}{\Delta t}} \Delta x} < \frac{S_n - n\mu(\epsilon_i)}{\sqrt{n}\sigma(\epsilon_i)} < \frac{b}{\sqrt{\frac{t}{\Delta t}} \Delta x}\right) \\ &= \frac{1}{\sqrt{2\pi}} \int_{\frac{a}{\Delta x \sqrt{\frac{t}{\Delta t}}}}^{\frac{b}{\Delta x \sqrt{\frac{t}{\Delta t}}}} e^{-\frac{x^2}{2}} dx. \end{aligned} \quad (8)$$

¹ Depending on the units of Q , it is common to find Eq. (3) written in terms of k_B , since $R = N_A k_B$, with N_A the Avogadro number.

after a simple variable substitution we obtain²:

$$\lim_{\substack{n \rightarrow \infty \\ t = n\Delta t}} P(a < X(t) < b) = \frac{1}{\Delta x \sqrt{\frac{2\pi t}{\Delta t}}} \int_a^b e^{-\frac{x^2}{2t(\Delta x)^2/\Delta t}} dx. \quad (9)$$

Based on the relation $\Delta t \rightarrow 0$ and $\Delta x \rightarrow 0$, one can define (as mentioned in [Appendix A](#)) the diffusion coefficient as follows:

$$D := \lim_{\substack{\Delta x \rightarrow 0 \\ \Delta t \rightarrow 0}} \left(\frac{(\Delta x)^2}{2\Delta t} \right), \quad (10)$$

finally obtaining the expression:

$$\lim_{\substack{n \rightarrow \infty \\ t = n\Delta t \\ \Delta x \rightarrow 0, \Delta t \rightarrow 0}} P(a < X(t) < b) = \int_a^b \phi(x, t) dx, \quad (11)$$

where $\phi(x, t) = \frac{1}{\sqrt{4\pi Dt}} e^{-\frac{x^2}{4Dt}}$ is the probability density function (PDF) of a normal distribution with $\mu = 0$ and $\sigma^2 = 2Dt$. This equation demonstrates that the probability of finding an atom in a given position after a large number of jumps converges towards the result obtained from a normal distribution centred on the atom's initial position whose variance is proportional to the diffusion coefficient and the diffusion time. It is easy to prove, via simple substitution, that the aforementioned distribution is a fundamental solution of the 1-D diffusion equation.

In light of the previous derivation, we can make use of a standard *Wiener process* to simulate diffusion. This process is a real valued continuous-time stochastic process with normally distributed and independent increments and can be understood as the *scaling limit* of random walks (*Donsker's theorem* [Donsker, 1951](#)).

In practice this means that if the diffusion coefficient is known, one can sample the position of a radionuclide after a given diffusion time (big enough to accommodate a large number of atomic jumps) using a multivariate normal distribution. Of course, by doing this we would overlook the presence of boundaries during this step, which are of major importance to us because we are interested in the number of radionuclides reaching an object's surface. In order to resolve this problem, we can split the total diffusion time into a suitable number of sub-steps that are sampled in sequence. This allows for checking the position of the radionuclide after each of them, as schematically shown in [Fig. 1](#), and therefore detecting if a given radionuclide reached a surface during its trajectory through the host material.

4. SOLIDUSS: implementation and logic

The pursued aim is the creation of a simple and reliable software that could be used in the context of radiation protection studies whenever it is important to evaluate the potential out-diffusion of radionuclides from arbitrarily complex geometries in case of fire. The developed diffusion software has been named *SOLIDUSS*, which stands for SOLID-state diffUSion Software, and it is mainly written in C++. This tool takes advantage of the CERN-FLUKA geometry kernel to perform out-diffusion calculations in arbitrarily complex geometries. Using it we can obtain the amount of radionuclides out-diffused from a given *source material* into an arbitrary number of *target materials* in any area of interest of our geometry. This is done for a set of diffusion periods as defined by the user³ and, in addition, the software will also provide a 3-D map of the radionuclide concentration for each of them. It also offers the possibility of performing multiple diffusion calculations while running a single CERN-FLUKA simulation, which is especially useful if we need to assess the out-diffusion of multiple radionuclide species, or if we need to perform a sensitivity analysis

over a certain parameter. In addition, *SOLIDUSS* can make use of a 3-D map of temperatures provided by the user to track the radionuclides accordingly, allowing the user to evaluate out-diffusion based on more realistic scenarios where the temperature is not found to be the same all over the area of interest. To make this possible, *SOLIDUSS* is linked to CERN-FLUKA at two different levels:

- We make use of CERN-FLUKA user routines to trigger different actions of the diffusion software. Upon creation of a radionuclide in a region of interest (ROI) a hook into *SOLIDUSS* is activated which then takes care of the actual diffusion calculation.
- For the actual tracking *SOLIDUSS* calls back into the geometry kernel of the Monte Carlo software, which allows for reusing the identical geometry model used already for simulating the creation of radionuclides.

4.1. How to use the software

In order to use the software to simulate diffusion in addition to the standard CERN-FLUKA calculations, one must carry out the following actions:

- Create the so-called *diffusion input file* (see [Appendix B](#) for an example), which will include all the information needed to simulate diffusion according to the user preferences (radionuclide of interest, host-material, activation parameters, diffusion times, temperature maps, concentration mesh boundaries, etc.).
- Make sure the CERN-FLUKA input file includes the necessary options to simulate radioactive decays according to the user's needs.
- Compile *SOLIDUSS* and link it against CERN-FLUKA.

Once this setup is accomplished, the procedure to run the simulation is identical to that of a standard CERN-FLUKA simulation.

4.2. Further considerations

There are few things that should be kept in mind by the user:

1. We are currently always considering isotropic host materials. This is justified for our applications since diffusion is isotropic in the majority of the materials we are interested in. Extension of the methodology towards anisotropic diffusion is in principle easily possible. However, it would depend on the crystalline structure of the material which generally is not known to the user.
2. Regarding materials that are liquid at typical fire temperatures. Although the numerical treatment should still be valid for static liquids if proper diffusion parameters are provided, the modelled geometry of the originally solid objects would no longer be applicable. Also, for non-static liquids, their flow would significantly impact the out-diffusion fraction in ways that *SOLIDUSS* cannot predict.
3. It is important to stress the stochastic nature of the tool we are describing and pay attention to the number of histories simulated (amount of nuclides followed for each particular case) as it will be of extreme importance to ensure the statistical significance of the results.
4. The current model does not include any surface effect and every nuclide will be considered as out-diffused from its host material as soon as it reaches the surface. Future versions of the software will address this scenario as optional features.
5. After a given diffusion time, the total amount of radionuclides may have been significantly reduced due to decay. This is taken into account by the software, which provides the user with the fraction of radionuclides remaining after the diffusion time by means of a message printed in the output file.

² This can be proven using the change of variable $y = x \cdot \Delta x \sqrt{\frac{1}{\Delta t}}$.

³ For instance, we could be interested in studying how out-diffusion evolves during a fire, so we would need to request different diffusion times such as 10 min, 30 min, 60 min, etc.

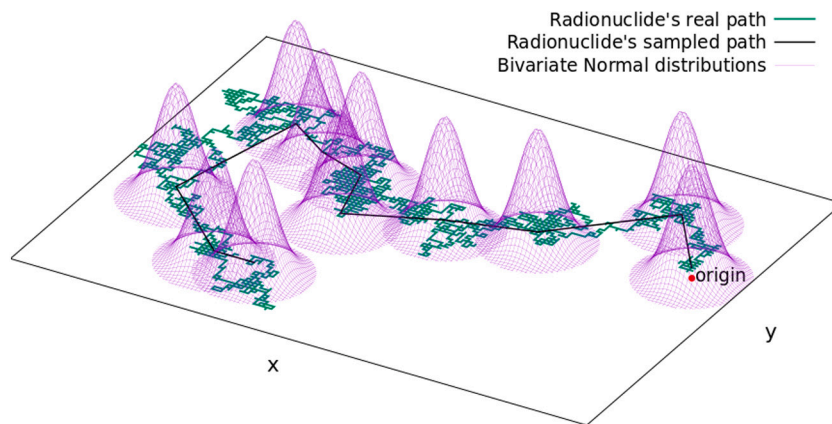


Fig. 1. Schematic example of the 2-D path followed by a radionuclide and the path that would be obtained by sampling multivariate normal distributions for a given number of time steps. Note that the variances of the normal distributions have been reduced here for illustration purposes, normally these distributions would present a significant overlap.

6. Chemical reactions are not simulated by SOLIDUSS and therefore the diffusion of potential compounds containing radioactive elements is not automatically accounted for. Nonetheless, the diffusion of this type of compounds could be simulated by providing its diffusion parameters to SOLIDUSS and requesting the simulation of a given radionuclide as if it was the compound. Note that volatile elements are simulated like any other and their higher diffusivity will be accounted for by using their diffusion coefficient.

5. Outputs post-processing: out-diffusion fraction estimation

The fraction of radionuclides out-diffused from a given object, what we call the out-diffusion fraction (ODF), will always be underestimated when calculated using simulated step lengths greater than the real atomic step lengths of the atoms in the solid lattice. This is because even if we check (sample) the position of a radionuclide often, it may happen that the radionuclide's path would reach a boundary and come back to the host material region between two consecutive position checks. If the final position is known to be inside the material, then we would wrongly consider the radionuclide as not out-diffused. Of course, this underestimation is less significant as we reduce the simulated step length approaching the atomic scale. In the limit of simulated steps of the size of real atomic steps we say that the simulation is performed with a perfect path resolution. However, this type of simulation is unpractical or even impossible for bulky objects, due to its high consumption of CPU resources, as mentioned earlier.

The discrepancy between the estimator and the accurate result is therefore influenced by the chosen size of the step lengths, which directly influences the reliability of the final result. A dedicated study showed that the ODF follows a well defined function of the time step used to sample the radioisotopes paths. The extrapolation of this function could be used to obtain a more accurate estimation of the ODF. This behaviour can be observed using a simple random walk simulation, as explained below.

Imagine a 1-D crystal lattice and a radionuclide located at the origin ($x = 0$) that can make a jump every second to the right or to the left with same probability. Let us place a boundary at a given position x_b . If at some point we find out that the radionuclide has crossed this boundary, we will consider it as out-diffused from the lattice. Now, we can let the radionuclide move during a time period t and check its position every Δt seconds. By doing this we will see whether the radionuclide is still inside the lattice or, on the contrary, it has traversed the boundary and therefore we must consider it as out-diffused. The radionuclide will keep moving unless we detect it beyond the boundary x_b during one of the checks performed every Δt s. It might very well cross x_b ($x > x_b$) and come back ($x < x_b$) within Δt s. In

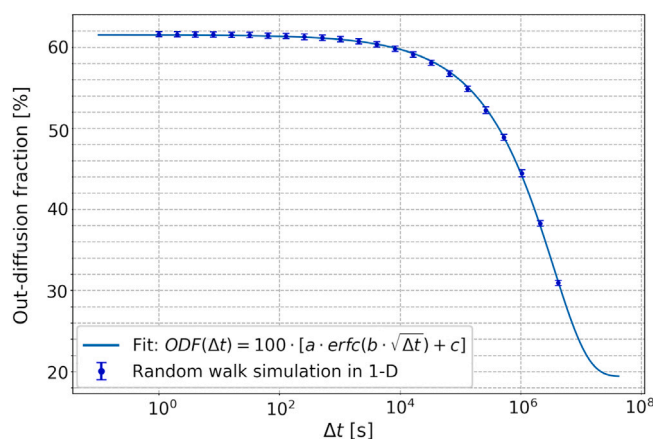


Fig. 2. Out-diffusion fraction as a function of the time interval Δt between consecutive checks of the radionuclide position for a 1-D random walk considering one jump per second and $N = 2^{20}$ jumps per radionuclide in total. A radionuclide is considered out-diffused if found beyond the position x_b . The fit of these data using Eq. (12) is also shown.

such a case, it will not be observed beyond the boundary and therefore not accounted for as out-diffused. We performed different simulations for different values of Δt in order to compare the fraction of out-diffused radionuclides that we obtain for each of them.

The results considering $N = 2^{20}$ jumps per radionuclide and a boundary placed at $x_b = \sqrt{N} \cdot \Delta x / 2^4$ (where Δx is the length of a single jump) are shown in Fig. 2, where we can see that the out-diffusion fraction we obtained does depend on how often we check the position of the radionuclides. This dependence perfectly obeys the following function:

$$ODF(\Delta t) = a \cdot \text{erfc}(b \cdot \sqrt{\Delta t}) + c, \tag{12}$$

where erfc is the complementary error function and a , b and c are the fit parameters. This is a remarkable result since it will allow us to estimate

⁴ Note that after N steps, the position of the radionuclide will follow a normal distribution centred on its initial position and with a standard deviation given by $\sigma = \sqrt{N} \cdot \Delta x$. Therefore, we have located the boundary at a distance of $\sigma/2$ from the initial position of the radionuclides. It can be shown, using the reflection principle for a Wiener process (Jacobs, 2010), that the out-diffusion fraction (as defined here) would be approximately at 61.8% of the total number of radionuclides, coinciding with our simulation result for $\Delta t \rightarrow 0$ (see Fig. 2).

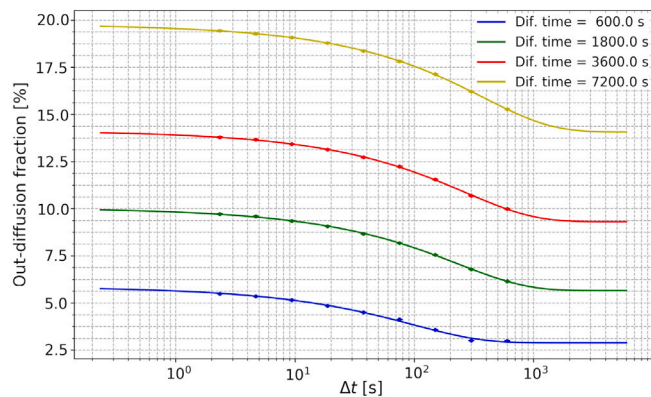


Fig. 3. Out-diffusion fraction as a function of the time step used in the tracking of the radionuclides for several diffusion times. As expected, the smaller the time step used, the larger the out-diffusion fraction we obtain, saturating as it gets closer to the real value. The curves represent fits to the data obtained using Eq. (12). The statistical uncertainties of the simulation results are also plotted, but they are too small to be visible.

the ODF for tiny time steps just by knowing it for larger ones. In other words, if we have enough data to fit the curve, we can obtain the ODF for any time step, including $\Delta t \rightarrow 0$. The same behaviour has been observed in 2-D and 3-D, as well as for non-trivial boundaries.

Similar results can be observed using SOLIDUSS. Fig. 3 shows the ODF of ^{32}P from a Fe cylinder at 1273 K as a function of the time step sizes used when sampling the radionuclides' paths. The initial distribution of radionuclides was considered as uniform within the cylinder and the results are shown for several total diffusion times, which are the duration of the material exposure to high temperatures (*i.e.* to fire). As we can see, the agreement between the fitting function [Eq. (12)] and the simulation results is excellent, which corroborates the hypothesis of using it to estimate the real ODF: the one we would obtain for $\Delta t \rightarrow 0$. Nevertheless, using a fitted function for extrapolation purposes should be done with extreme caution; in particular, one should make sure that the tendency of the fitted data in the fit region remains unchanged in the region of interest (Wolberg, 2006 pp. 96–99). We are convinced that we should expect nothing else than an asymptotic behaviour of the simulated ODF towards the real value when $\Delta t \rightarrow 0$. To test this assumption, we performed the pure random walk simulation presented earlier, in which this asymptotic tendency has been proven as true down to the minimum time step size possible, that one needed for an isotope to perform a single atomic jump. This is why we believe the use of this fit for extrapolation is justified. Once the fit parameters have been obtained, we can estimate the out-diffusion fraction as follows:

$$ODF(\Delta t \rightarrow 0) = a \cdot \operatorname{erfc}(b \cdot \sqrt{0}) + c = a + c, \quad (13)$$

since $\operatorname{erfc}(0) = 1$. One may suspect that the behaviour shown in Fig. 3 is just a consequence of the uniform initial distribution of radionuclides that we have used; or even that this behaviour would be distorted in the presence of non-uniform temperatures. In order to study these hypotheses we have performed a number of tests involving different pairs radionuclide — host material, geometries, temperature gradients and non-uniform initial distributions. The aforementioned behaviour remains present in all of them, which allow us to estimate the ODF whenever we have several estimated values for different time steps, even if these values are far from the real fraction. As a consequence, the CPU time needed to estimate an ODF is largely reduced,⁵ and we obtain a much better approximation of the real ODF. The associated confidence intervals can be estimated using the covariance matrix

⁵ The larger the time step, the smaller the number of steps and hence, the smaller the number of calculations to be performed.

resulting from the fit and scaling the obtained variance according to the critical values of the *Student's t-distribution* to account for the small number of data points involved in the fit (Wolberg, 2006 pp. 15–16).

6. Exemplary calculation

In the present section we show an exemplary case which aims at illustrating some of the software's capabilities. In particular, let us consider a warm quadrupole from the Large Hadron Collider (LHC) at CERN to investigate the out-diffusion of several radionuclides produced in its yoke and coils. We have assumed the yoke to be made of Fe, and the coils of Cu. A cross section of the quadrupole is shown in the left part of Fig. 4, together with a zoom to better observe the coils. These coils consist of rectangular Cu cells with a cylindrical cavity in their centre through which cooling water flows. An insulation layer wraps each of those cells.

We assume the temperature map provided in the upper right image of Fig. 4, with 1000 °C at the outer surface of the quadrupole and a reduction of the temperature as a function of the distance to this surface. As a consequence, we find the lowest temperatures in the inner part of the quadrupole.

In the lower right image of the same figure, we can see the initial distribution of ^{55}Fe in the magnet yoke. It was produced for this example by simulating the collision of 7 TeV protons with an LHC collimator some metres away from the quadrupole, which generates the particle showers that activate the materials in the magnet. The residual radionuclides produced in this simulation by CERN-FLUKA are internally passed to SOLIDUSS, triggering the diffusion calculation according to the user specifications. The isotope ^{55}Fe has been arbitrarily chosen, similar concentration maps are found for other species.

The resulting ODF for different radionuclides from yoke and coils is shown in Table 1. The selection of radionuclides for this example is arbitrary. One can perform these calculations for any radionuclides of interest, provided they are produced in the irradiation scheme simulated with CERN-FLUKA and the necessary diffusion parameters have been communicated to SOLIDUSS by means of the diffusion input file. In the aforementioned table it can be observed that we have divided the radionuclides out-diffusing from the Cu coils into those entering the insulation and those entering the water cooling. This can be very easily done with SOLIDUSS and may be of interest to assess the portion entering the cooling system and that one that could be released if the coil insulation is burnt during an hypothetical fire in the area. The computational time spent by SOLIDUSS to carry out the presented calculation in a 3.40 GHz Intel Core i7-6700 CPU was 46 h. Considering 4 CPUs running in parallel, the diffusion calculation could be performed in approximately 11.5 h using a desktop PC. Note that the example shown here is a very exigent one in terms of CPU time due to the quadrupole yoke, which is a very bulky object, resulting in very low ODFs. As a consequence, longer run times are needed to achieve sufficiently small uncertainties. In addition, statistical uncertainties are also affected by the number of radionuclides followed. For radioisotopes with lower production yields one will typically find ODF with larger uncertainties, two examples of this in Table 1 are ^{32}P in Fe and ^{48}V in Cu to a lesser extent.

The diffusion coefficient parameters used for this simulation has been extracted from Brandes and Brook (1992).

7. Bootstrapping test

In this section we explain the study carried out to check SOLIDUSS' implementation and the numerical treatment on which it is based. To simulate the diffusion of radionuclides, one of the most important input parameters that needs to be provided to the code is the diffusion coefficient. This quantity, which is available in the literature for a large variety of species and materials, is usually obtained experimentally (Brandes and Brook, 1992). One of the typical procedures to

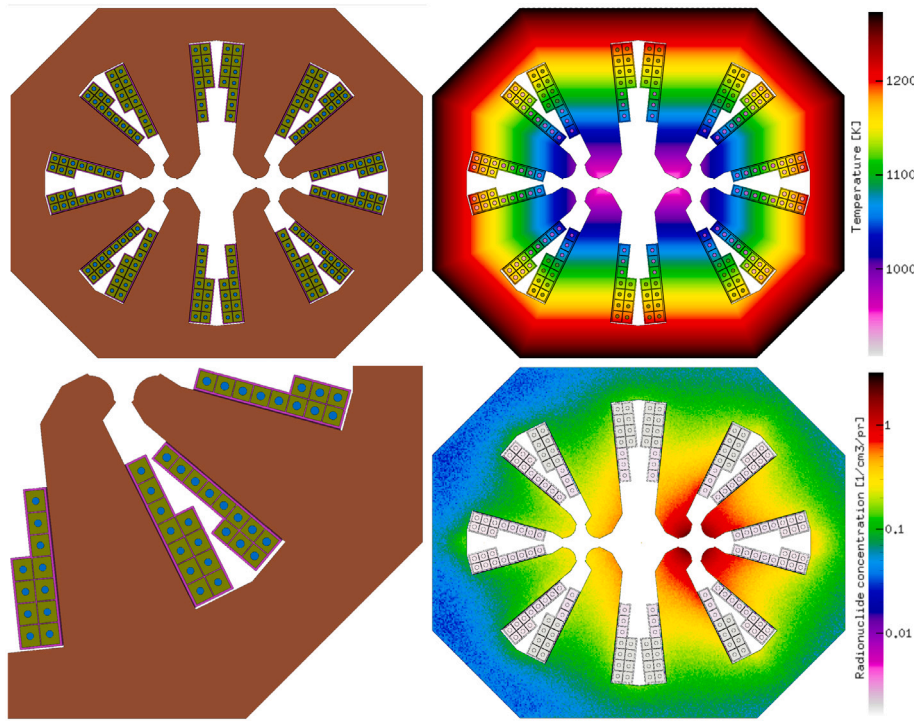


Fig. 4. Cross section of the simulated geometry of a LHC warm quadrupole (left), together with the temperature map assumed (upper right) and the concentration of ⁵⁵Fe obtained with CERN-FLUKA (lower right). The pictures have been created using Flair (Vlachoudis, 2009).

Table 1

Out-diffusion fraction for some residual nuclei generated in the quadrupole yoke and coils. Radionuclides are generated in the source material and out-diffused into the target material. We have considered two diffusion times: 30 min and 1 h. Results are provided in parts-per-million (ppm) and its uncertainties correspond to 2σ confidence intervals.

Source material	Target material	Isotope	ODF [ppm]	
			30 min	1 h
Fe	Air	⁵⁵ Fe	$(2.3 \pm 0.4) \cdot 10^1$	$(3.3 \pm 0.4) \cdot 10^1$
		⁵⁴ Mn	$(3.3 \pm 0.4) \cdot 10^1$	$(4.7 \pm 0.4) \cdot 10^1$
		⁴⁸ V	$(3.2 \pm 0.4) \cdot 10^1$	$(4.6 \pm 0.4) \cdot 10^1$
		³² P	$(6 \pm 5) \cdot 10^1$	$(9 \pm 5) \cdot 10^1$
Cu	Insulation	⁶⁵ Zn	$(2.52 \pm 0.05) \cdot 10^3$	$(3.61 \pm 0.05) \cdot 10^3$
		⁶⁴ Cu	$(1.48 \pm 0.04) \cdot 10^3$	$(2.11 \pm 0.05) \cdot 10^3$
		⁶⁰ Co	$(9.7 \pm 0.4) \cdot 10^2$	$(1.39 \pm 0.05) \cdot 10^3$
		⁵⁴ Mn	$(2.58 \pm 0.10) \cdot 10^3$	$(3.68 \pm 0.13) \cdot 10^3$
		⁴⁸ V	$(1.8 \pm 0.2) \cdot 10^3$	$(2.6 \pm 0.3) \cdot 10^3$
Cu	Cooling water	⁶⁵ Zn	$(8.5 \pm 0.4) \cdot 10^2$	$(1.23 \pm 0.05) \cdot 10^3$
		⁶⁴ Cu	$(5.0 \pm 0.3) \cdot 10^2$	$(7.1 \pm 0.3) \cdot 10^2$
		⁶⁰ Co	$(3.3 \pm 0.2) \cdot 10^2$	$(4.7 \pm 0.3) \cdot 10^2$
		⁵⁴ Mn	$(8.5 \pm 0.7) \cdot 10^2$	$(1.22 \pm 0.09) \cdot 10^3$
		⁴⁸ V	$(5.2 \pm 1.3) \cdot 10^2$	$(7.8 \pm 1.9) \cdot 10^2$

obtain the diffusion coefficient, and in particular the activation parameters A and Q , are the so-called *tracer diffusion experiments* (Mehrer, 2007 pp. 215–223). In such experiments a specific radioactive isotope – the tracer – of the element whose diffusion coefficient is to be obtained, is deposited on the material of interest. This material is then heated up to a certain temperature to promote diffusion and the resulting concentration profile of the tracer inside the material is analysed in order to extract the activation parameters. The boundary conditions of the experiment (material geometry, tracer deposition, temperature gradient, etc. are chosen in such a way that the diffusion equation can be solved analytically and hence, the shape of the tracer’s profile inside the host material is known beforehand. One can then obtain the best fit to the experimental results using the analytical solution of the diffusion equation with the diffusion coefficient D as the free parameter

of the fit. In this way, D can be determined for a given temperature. Repeating the process, several values of D can be obtained for different temperatures. Since the dependence of the diffusion coefficient with the temperature is given by Eq. (3) or similarly, one can fit the obtained diffusion coefficients using this expression with A and Q as the free parameters of the fit. Using this approach, both quantities can be derived from the experimental data.

We can emulate this procedure by replacing the diffusion experiment with a diffusion simulation. The idea is to set up a simple problem that can be solved analytically, give the diffusion coefficient parameters A and Q to SOLIDUSS and simulate the system’s evolution. The simulation will return the radionuclide’s concentration profile after a given diffusion time and we will treat these results as if they were experimental ones. Our goal would be to determine A and Q using those results and compare the obtained values with the ones we provided to SOLIDUSS in the first place. If the resulting values are compatibles with the input ones, we can be fairly confident about the correctness of SOLIDUSS’ implementation and underlying numerical simulation method.

Let us consider a 1-D scenario in which the initial concentration of radionuclides is given by:

$$C(x, 0) = \begin{cases} C_0 & \text{if } x \leq 0, \\ 0 & \text{if } x > 0, \end{cases} \quad (14)$$

with x denoting the spatial coordinate. It can be shown that the solution to the diffusion equation for this particular case is the following (Mehrer, 2007 pp. 41–42):

$$C(x, t) = \frac{C_0}{2} \operatorname{erfc}\left(\frac{x}{2\sqrt{Dt}}\right), \quad (15)$$

where D is the diffusion coefficient and t the diffusion time.

The concentration profiles of ⁴⁸V after 2 h of diffusion in Cu at different temperatures have been calculated using SOLIDUSS and reported in Fig. 5. Note that the initial concentration of radionuclides for $x \leq 0$ is $C_0 = 2 \cdot 10^{-3} \text{ cm}^{-3}$. In the same figure, we can see the best fits of Eq. (15) for the different temperatures. The free parameter in the

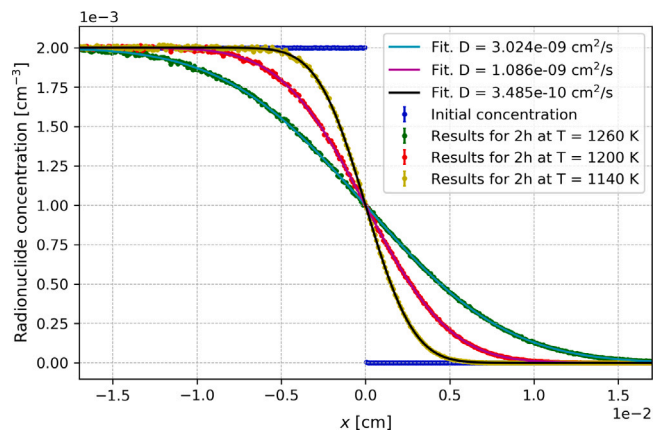


Fig. 5. Radionuclide concentration profiles returned by SOLIDUSS for ⁴⁸V in Cu as well as their best fits using Eq. (15) with *D* as the free parameter.

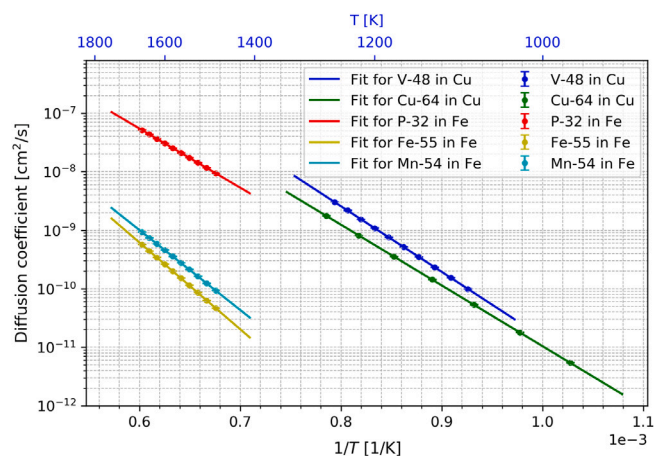


Fig. 6. Diffusion coefficients for different temperatures and different species extracted from SOLIDUSS results, together with the best fits of Eq. (3) for each pair radionuclide — host material, being *A* and *Q* the free parameters.

fits has been defined as *D*, and its best fitting values are reported in the figure's legend.

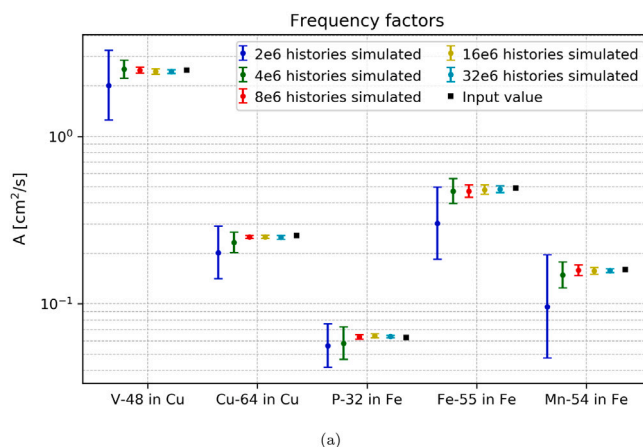
Repeating the process for further temperatures and radionuclides, we obtained the results shown in Fig. 6. There we also show the best fits of the diffusion coefficient expression for each pair radionuclide — host material.

From each of these fits, we can get *A*, *Q* and their uncertainties, which are intrinsic to the stochastic nature of our simulation and to the fitting process. We deemed it convenient to repeat the entire process considering different amounts of tracked radionuclides to verify if the obtained results converge to the input values when the number of histories increase. The extracted values are reported in Fig. 7, together with the original activation parameters provided to SOLIDUSS.

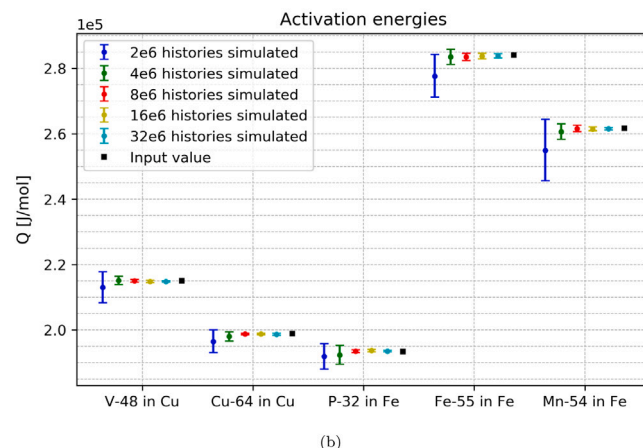
We can observe a clear convergence towards the input values in all cases, as well as a good compatibility of input and extracted values within the error bars. Therefore, we conclude that the software implementation and the numerical treatment of diffusion on which it is based work as intended to the best of our knowledge. Note that SOLIDUSS does not make explicit use of the diffusion equation, yet its results are in excellent agreement with those of the diffusion equation.

8. Conclusion

Through the previous sections we have tried to show that SOLIDUSS is conceived as a versatile and reliable tool to perform solid-state



(a)



(b)

Fig. 7. Frequency factors (a) and activation energies (b) for different radionuclides derived from SOLIDUSS simulations. The results are reported for various number of histories simulated. The original input values given to SOLIDUSS are also shown for comparison. The error bars correspond to 2σ confidence intervals.

diffusion calculations for radiation protection purposes involving arbitrary geometries, temperatures and radionuclide distributions. It can be used for overall and detailed diffusion studies and allows for multiple calculations during a single run, facilitating a complete assessment of the out-diffusion impact in complex scenarios.

The software's limitations have been pointed out along this document and the reader is encouraged to carefully read Sections 3 and 4, paying particular attention to Section 4.2, where we have identified a number of considerations that could help to understand the current scope of the software.

SOLIDUSS has been designed and built in such a way that allows for an easy introduction of future improvements and, in particular, those related to the underlying physical model.

The experimental benchmarking of this tool is currently ongoing. Preliminary results show the need to equip the software with the means to simulate the desorption of those radionuclides reaching the object's surface. A thoroughly description of the benchmarking will be provided in a future publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Random walks and the diffusion equation

The use of random walks is a very pedagogical way of showing the connection between the microscopic and the phenomenological visions. Let us outline it in a similar way as in Ref. Evans (2019) pp. 36–37, starting by a single atom that can jump a distance Δx to the left or to the right (we constrain ourselves to 1-D for simplicity) each time interval Δt , with probability $1/2$ for both directions (here we are assuming isotropic media). Let us then define $P(m, n)$ as the probability of finding the particle at a position $m\Delta x$ at a time $n\Delta t$ with $\{m = 0, \pm 1, \pm 2, \dots, \pm n; n = 0, 1, 2, \dots\}$. Then, considering $P(0, 0) = 1$ (we place the atom initially at the origin), we can see that:

$$P(m, n + 1) = \frac{1}{2}P(m - 1, n) + \frac{1}{2}P(m + 1, n). \quad (\text{A.1})$$

In plain words: the probability of being at a given position at a given time is the probability of being next to it just before $[P(m - 1, n) + P(m + 1, n)]$ multiplied by the probability of jumping to it ($1/2$). We can now add $-P(m, n)$ to both sides of the equation and multiply them by $(\Delta x)^2/\Delta t$, getting the following:

$$\frac{P(m, n + 1) - P(m, n)}{\Delta t} = \frac{(\Delta x)^2}{2\Delta t} \frac{P(m - 1, n) + P(m + 1, n) - 2P(m, n)}{(\Delta x)^2}. \quad (\text{A.2})$$

We would like to move from our discrete description to a continuous one, and this can be done by means of the *continuous limit* considering very small Δx and Δt , which means $\Delta x \rightarrow 0$ and $\Delta t \rightarrow 0$. In this case, $n\Delta t \rightarrow t$, $m\Delta x \rightarrow x$ and $P(m, n) \rightarrow \phi(x, t)$, where $\phi(x, t)$ is the continuous probability density function of finding the atom in a position x , at a time t . The definition of derivative tells us that:

$$\frac{\partial \phi(x, t)}{\partial t} = \lim_{\Delta t \rightarrow 0} \frac{\phi(x, t + \Delta t) - \phi(x, t)}{\Delta t}, \quad (\text{A.3})$$

and

$$\frac{\partial^2 \phi(x, t)}{\partial x^2} = \lim_{\Delta x \rightarrow 0} \frac{\phi(x + \Delta x, t) + \phi(x - \Delta x, t) - 2\phi(x, t)}{(\Delta x)^2}. \quad (\text{A.4})$$

Using all of this, Eq. (A.2) becomes:

$$\frac{\partial \phi(x, t)}{\partial t} = \lim_{\substack{\Delta x \rightarrow 0 \\ \Delta t \rightarrow 0}} \left(\frac{(\Delta x)^2}{2\Delta t} \right) \frac{\partial^2 \phi(x, t)}{\partial x^2}. \quad (\text{A.5})$$

The similarities between Fick's second law [Eq. (2)] and Eq. (A.5) are now clear. Identifying $\phi(x, t)$ as the particle concentration $C(x, t)$ and defining a positive constant D such as:

$$D := \lim_{\substack{\Delta x \rightarrow 0 \\ \Delta t \rightarrow 0}} \left(\frac{(\Delta x)^2}{2\Delta t} \right), \quad (\text{A.6})$$

we obtain the diffusion equation in 1-D:

$$\frac{\partial C(x, t)}{\partial t} = D \frac{\partial^2 C(x, t)}{\partial x^2}. \quad (\text{A.7})$$

In 3-D, we would obtain Eq. (2). This approach is one of the many variants that can be used to show the link between microscopic and phenomenological descriptions of diffusion.

Appendix B. Diffusion input file

The diffusion input file is a card-based text file specifying the information necessary to perform the diffusion calculations. An example consisting of two different sections can be found below. Please note that only a sample of the available options is used in this example and it is therefore not intended to show the full flexibility of the software. A brief explanation on the meaning and usage of the different entries employed is given after it.

```
* Section 1 deals with Mn-54 coming out from iron. Since the
* diffusion coefficient parameters are different for
* different temperatures ranges we need to include several
* entries in the file.
```

```
SECTION 1
RADIOISOTOPE = Mn-54
HOST_MATERIAL = IRON
DIFFUSION_TIME = 600 1800 3600 7200
GLOBAL_TEMPERATURE = 1173
DIFFUSION_COEFFICIENT = 1.49 233600 0 0 0 1043
DIFFUSION_COEFFICIENT = 0.16 261700 0 0 1043 1667
DIFFUSION_COEFFICIENT = 0.35 219800 0 0 1667 1811
CONCENTRATION_GRID = -16 -41.5 -17602 86 31.5 -17262 100 100
20
FAST_TRACKING = Yes
TEMPERATURE_MAP_FILE = ../TempFile
```

```
* Section 2 deals with Co-60 coming out from copper, but we
* only want to take into account the amount not out-diffused
* into water.
```

```
SECTION 2
RADIOISOTOPE = Co-60
HOST_MATERIAL = COPPER
DIFFUSION_TIME = 100 500 1000 2000 4000
GLOBAL_TEMPERATURE = 900
DIFFUSION_COEFFICIENT = 0.74 217200 736 312800 0 1358
CONCENTRATION_GRID = 2.84 -20.65 -17617 67.16 10.65 -17247 50
50 100
FAST_TRACKING = Yes
EXCLUDE_TARGET_MATERIALS = Water
TEMPERATURE_MAP_FILE = ../TempFile
```

- SECTION unused_variable
Example: SECTION 1
Description: we can perform multiple diffusion calculations while running a single CERN-FLUKA simulation. To do this, we must organize the diffusion parameters corresponding to different calculations in different sections. Each of these sections must start with this card.
- RADIOISOTOPE = "chemical element symbol"- "mass number"
Example: RADIOISOTOPE = Mn-54
Description: specify the radioisotope whose out-diffusion needs to be assessed.
- HOST_MATERIAL = "CERN-FLUKA material name"
Example: HOST_MATERIAL = Iron
Description: it allows us to select the material in which diffusion will take place.
- DIFFUSION_TIME = Time_1 < Time_2 Time_3 ... >
Example: DIFFUSION_TIME = 600 1800 3600 7200
Description: using this card the user can request an arbitrary number of diffusion times and the program will provide the results for each of them.
- GLOBAL_TEMPERATURE = Temperature
Example: GLOBAL_TEMPERATURE = 1173
Description: selects the temperature of the environment where diffusion need to be simulated.
- DIFFUSION_COEFFICIENT = A₁ Q₁ A₂ Q₂ T_i T_f
Example:
DIFFUSION_COEFFICIENT = 1.49 233600 0 0 0 1043
Description: card used to specify the diffusion coefficient to be used in the simulation. In particular, the user can provide up to two pairs of activation parameters (A₁, Q₁, A₂ and Q₂) for a given range of temperature [T_i, T_f].
- CONCENTRATION_GRID = MinX MinY MinZ MaxX MaxY MaxZ NrBinsX NrBinsY NrBinsZ
Example: CONCENTRATION_GRID = -16 -41.5 -17602 86 31.5 -17262 100 100 20
Description: we can define our area of interest specifying the structural characteristics of the radioisotope concentration grid that will be generated by the program using this card. MinX, MinY and MinZ are the minimum coordinates in the X, Y and Z

axes respectively, while $MaxX$, $MaxY$ and $MaxZ$ are the maximum ones. The parameters $NrBinsX$, $NrBinsY$ and $NrBinsZ$ represent the number of bins into which the delimited volume will be divided in each direction.

- `FAST_TRACKING = "Yes/No"`

Example: `FAST_TRACKING = Yes`

Description: if selected, we will make use of an optimization procedure aimed to reduce the computation time of the radioisotope tracking whenever possible.

- `TEMPERATURE_MAP_FILE = "path to file"`

Example:

`TEMPERATURE_MAP_FILE = ../temperatures_1.dat`

Description: we can provide a temperature map by means of an external file. This card enable this option and provide the file path.

- `EXCLUDE_TARGET_MATERIALS = "Material1" < "Material2" ... >`

Example: `EXCLUDE_TARGET_MATERIALS = Water`

Description: this card allows us to exclude an arbitrary number of materials from a list of target materials. In the example provided, the program results will tell us the amount of radioisotopes out-diffused from the host material into any material but Water.

References

- Battistoni, G., Boehlen, T., Cerutti, F., Chin, P.W., Esposito, L.S., Fassò, A., Ferrari, A., Lechner, A., Empl, A., Mairani, A., Mereghetti, A., Garcia Ortega, P., Ranft, J., Roesler, S., Sala, P.R., Vlachoudis, V., Smirnov, G., 2015. Overview of the FLUKA code. *Ann. Nucl. Energy* 82, 10–18. <http://dx.doi.org/10.1016/j.anucene.2014.11.007>.
- Bohlen, T.T., Cerutti, F., Chin, M.P.W., Fassò, A., Ferrari, A., Ortega, P.G., Mairani, A., Sala, P.R., Smirnov, G., Vlachoudis, V., 2014. The FLUKA code: Developments and challenges for high energy and medical applications. *Nucl. Data Sheets* 120, 210–214. <http://dx.doi.org/10.1016/j.nds.2014.07.049>.
- Brandes, E.A., Brook, G.B. (Eds.), 1992. *Smithells Metals Reference Book*, seventh ed. Reed Educational and Professional Publishing Ltd.
- Crank, J., 1975. *The Mathematics of Diffusion*, second ed. Oxford Science Publications.
- Donsker, M.D., 1951. An invariance principle for certain probability limit theorems. *Mem. Amer. Math. Soc.* 6.
- Evans, L.C., 2019. An introduction to stochastic differential equations. Version 1.2 [online]. <http://ft-sipil.unila.ac.id/dbooks/AN%20INTRODUCTION%20TO%20STOCHASTIC%20DIFFERENTIAL%20EQUATIONS%20VERSION%201.2.pdf> (accessed September 2019).
- FIRIA project [online]. <https://hse.cern/content/firia>.
- Föll, H., 2019. Diffusion in Iron [online]. https://www.tf.uni-kiel.de/matwis/amat/iss/kap_5/illustr/s5_2_3.html (accessed January 2019).
- Gai, G., 2021. FIRIA Methodology. Tech. Rep., CERN, [EDMS 2426715](https://cds.cern.ch/record/2426715).
- Jacobs, K., 2010. *Stochastic Processes for Physicists*. Cambridge University Press, pp. 57–59. <http://dx.doi.org/10.1017/CBO9780511815980>.
- Lalley, S., 2019. Brownian motion [online]. <https://galton.uchicago.edu/~lalley/Courses/313/BrownianMotionCurrent.pdf> (accessed September 2019).
- Mehrer, H., 2007. *Diffusion in Solids*. Springer, Berlin Heidelberg New York, <http://dx.doi.org/10.1007/978-3-540-71488-0>.
- Theis, C., Vincke, H., 2016. Development of a computational model for the out-diffusion of radioisotopes from metals. Tech. rep., CERN, [EDMS1720724](https://cds.cern.ch/record/2720724).
- Vlachoudis, V., 2009. FLAIR: A powerful but user friendly graphical interface for FLUKA. In: *Proc. Int. Conf. on Mathematics, Computational Methods & Reactor Physics (M&C 2009)*. Saratoga Springs.
- Wolberg, J., 2006. *Data Analysis using the Method of Least Squares*. Springer, Berlin Heidelberg New York.