Functional estimation of the junction temperature with Monte Carlo method for electronic module cooling

Léa Penazzi, Anne Castelan, Stéphane Blanco, Cyril Caliot, Christophe Coustet, Mouna El-Hafi, Richard Fournier, Jean-Pierre Fradin

Functional estimation of the junction temperature with Monte Carlo method for electronic module cooling

Léa Penazzi[1,*]
Post-doctorate researcher
[1] Aix Marseille University
CNRS, IUSTI UMR 7343
Marseille, France
Email: leapenazzi@gmail.com

Anne Castelan[2]
Researcher
[2] Icam site de Toulouse
75 Avenue de Grande Bretagne
CS 97615 Toulouse Cedex 3, France
Email: anne.castelan@icam.fr

Stéphane Blanco[3], **Cyril Caliot**[4], **Christophe Coustet**[5],
Mouna El Hafi[6], **Richard Fournier**[3], **Jean-Pierre Fradin**[2]
[3] LAPLACE, UMR CNRS 5213, Université Paul Sabatier, Toulouse, France
[4] LMAP, UMR CNRS 5142, Université de Pau et des Pays de l'Adour, Anglet, France
[5] RAPSODEE, UMR CNRS 5302, IMT Mines Albi, Campus Jarlard, Albi, France
[6] Méso-Star, 8 rue des Pêchers, 31410 Longages

In power electronics modules, accurately estimating the junction temperature is crucial for ensuring proper cooling and preventing device failure. Recent breakthroughs on the integral formulation of thermal couplings allows the estimation of physical quantities as transfer functions with Monte Carlo algorithms. The transfer function is estimated with a single Monte Carlo algorithm in which random walks are sampled to solve the coupled thermal problem involving conduction and convection in potential complex geometries. In the context of estimating junction temperature in power electronics modules, this involves simulating the thermal behavior of the module under various operating conditions, such as different levels of power dissipation and ambient temperatures. This paper addresses the junction temperature estimation as a function depending on a decisive parameter for the electronics power components cooling: the convective heat transfer coefficient. The different steps leading to the estimation of the transfer function are presented as well as its numerical validation against a results from a reference multiphysics electronics cooling simulation software (Celsius EC Solver). The results from the transfer functions are exposed for typical configurations of power modules and in both situations, the functions demonstrates the accuracy and reliability of the approach without degrading the geometrical or multiphysics complexities of the problem. Moreover, due to the computational time gain when estimating the junction temperature for different cooling configurations, the function obtained with Monte Carlo is prove being a relevant candidate for assessing thermal peformance of power electronics modules, accounting for uncertainties and variations inherent in operating conditions.

*Corresponding author.

NOMENCLATURE

T	Temperature (K).
λ	Thermal conductivity ($\mathrm{W\,m^{-1}\,K^{-1}}$).
Ψ	Volumetric power density ($\mathrm{W\,m^{-3}}$).
T_{BF}	Solid/fluid boundary temperature (K).

1 INTRODUCTION

There are numerous numerical approaches to thermal modeling of power electronics. This diversity reflects complex tradeoffs between accuracy, computational and human resources for data management and meshing [1].

One of the most usual approaches describes thermal transfers by using networks of thermal resistances and capacities, but this is soon limited outside 1D and 2D simplified geometries. Nodal methods extend this approach to all geometries [2, 3, 4] but there is still a loss of information because the system is divided into connected parts that are considered to be isothermal. There are more advanced 2D or 3D analytical models that, while still requiring a splitting into connected sub-systems, do not make the isothermal assumption within each sub-system for which the 3D heat equation is fully solved. They are, however, restricted to systems that can be presicely described using simple geometric primitives such as parallelepipeds, spheres, or cylinders [5, 6]. Finally, there is always the option of fully discretizing the 3D geometry and computationally solving the heat equation using approaches such as finite element and finite volume methods. However, this significantly increases the time required for both meshing and computing.

The current research intends to assist engineers facing design/optimisation instances where the geometrical characteristics of a power electronic device require a comprehensive 3D numerical approach to thermal transfer but time restrictions prevent it. When addressing this question, there are two main reasons why Monte Carlo approaches are investigated, both of which are related to the handling of complex geometries:

- Today's Monte Carlo algorithms for calculating the heat transfer equation take advantage of recent advances in the computer graphics sector. First, the engineering process of meshing the volumes is avoided. Second, the computation times are little sensitive to the size and refinement of the geometrical data set (frequently not at all).
- Symbolic Monte Carlo (SMC) algorithms can be linked to the basic Monte Carlo algorithm, allowing not only the temperatures and fluxes to be evaluated, but also a set of parameters (introduced as symbols) to be stored and later used inside transfer functions. These functions evaluate the same quantities without compromising the geometry, but they are significantly faster than the Monte Carlo itself. As a result, they can be employed in control, inversion or optimisation algorithms that involve invoking the heat transfer solver repeatedly.

The second feature is highlighted here: creating transfer functions as a natural byproduct of the main Monte Carlo simulation [[Ajout refs dans ce paragraphe MC, géo complexe avec synthèse d'image].

SMC was initially developed for inversion purposes (accessing the parameters of a physical model from observations) and was not recognized as a numerical method in and of itself until very recently. As a results, other keywords, such as "*Inverse Monte Carlo*" to refer to these inversion objectives, or "*Importance Sampling*" to refer to an old Monte Carlo trick, originally meant for variance reduction purpose (convergence speed-up), that was at the heart of all early SMC developments, must be used when exploring the corresponding literature [7, 8, 9, 10, 11]. Although only importance sampling will be employed here (the approach will be closely detailed in Sec. 3), it is worth noting that more advanced techniques have been documented in [12][???les autres refs, y compris celui de RAD ???], which may be relevant for thermal modeling in power electronics. These new techniques do, in fact, solve the specific issues associated with parameters appearing inside non-linear functions invoked by the Monte Carlo algorithm. For example, the exponential laws to statistically reconstruct thermal inertia for conduction in solids are highly similar to the exponential law of Beer extinction in [12].

Such questions will not be addressed in this paper. This does not imply that only linear symbolic models are considered; there is a sharp difference between the nonlinearity of the symbolic model and the nonlinear steps of the Monte Carlo algorithm. The simulation examples will focus on how a junction's temperature varies as a function a cooling device parameter, and this dependence is fully nonlinear. However, there will be no nonlinearity in the Monte Carlo itself, as there would be if, for example, we attempted to construct a symbolic model of how the junction temperature would change as a function of the solid diffusivity in a transient simulation [13].

Even if we restrict ourselves to one of the simplest classes of SMC problems (importance sampling will suffice), the SMC literature focuses primarly on the Radiative Transfer Equation in applications such as atmospheric radiation [8], tomography for medical imaging [9], and radiative heat transfer in combustion devices [10]. To the best of authors' knowledge, this is the first attempt to extend SMC to the Monte Carlo algorithm for solving coupled heat transfer (conduction, convection and radiation).

The paper is structured as follows. In Sec. 2, we describe and validate the Monte Carlo algorithm itself, applied to an elementary 3D case of an electronic module on a baseplate with fins (heat sink) to evaluate the junction temperature of the module in steady state [6]. This Monte Carlo algorithm is modified slightly to generate a symbolic model in Sec. 3. The primary emphasis is on predicting how the convective heat transfer coefficient (or its field) affects the junction temperature. In the process of designing/selecting an air-convection cooling system,

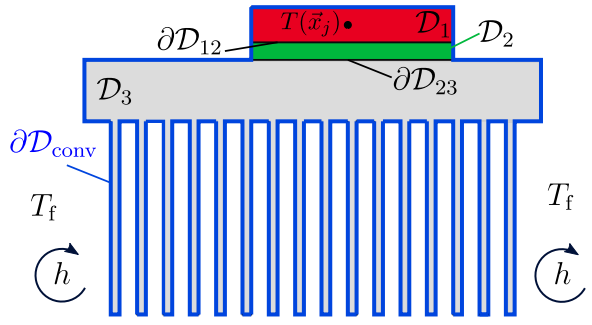


Fig. 1. 1^{st} case - Electronic module on a plate fin heat sink. Dimensions of the cooling system and further details on dimensions values are available in Appendix A.

the objective is to precisely predict the temperature evolution at the junction as a function of flow. The resulting transfer function is evaluated in Sec. 4 in terms of both precision and computation time.

Finally, conclusions and discussions regarding perspective of future work (see Sec. 5).

2 MONTE CARLO

The example of an electronic module associated with a plate-fin heat sink depicted in Fig. 1 will be used to illustrate theoretical developments.

The heat transfer model will first be expressed in very standard partial differential terms (heat equation for conduction in solids with volumetric sources and convection at the boundary) and then translated into statistical terms in order to develop a Monte Carlo algorithm evaluating a junction temperature. As Monte Carlo is rarely used to simulate thermal heat transfer in electronics, we provide exhaustive algorithmic information. This information will be required when addressing SMC. There are only successive steps leading to the sampling of a thermal path backward from the junction ot the controlled air flow (at a known temperature), with sources accumulating along the path to account for electrical energy dissipation. We will insist on the statistical steps associated with the path reaching the solid/fluid interface, whether the path is re-injected into the solid or terminates at the flow. Indeed, this is the only portion of the Monte Carlo algorithm that is dependent on the convective exchange coefficient h, and all of our SMC discussion will center on this part. Overall, it will be demonstrated that SMC imposes minimal changes on such a conducto-convective Monte Carlo algorithm: no additional random sampling is required, and all that is necessary is to store information about the locations where conducto-convective paths visited the interface.

2.1 The heat transfer model

The electronic module and the thermal interface material (TIM) are assumed to have the same thermal properties. The electronic module and TIM volumes are respectively noted $\mathcal{D}_1$ and $\mathcal{D}_2$. The corresponding thermal conductivities are λ_1 and λ_2[1]. The power density $P(\vec{x}) = P_0$ emitted by the module is uniform within $\mathcal{D}_1$. The domain corresponding to the plate fin heat sink is noted $\mathcal{D}_3$. Its thermal conductivity is λ_3 (no power density). The part of the surface of $\mathcal{D}_1$ facing the surounding fluid is noted $\partial\mathcal{D}_{\text{conv}_1}$, similarly parts of the surfaces of $\mathcal{D}_2$ and $\mathcal{D}_3$ facing the surounding fluid are noted $\partial\mathcal{D}_{\text{conv}_2}$ and $\partial\mathcal{D}_{\text{conv}_3}$, with $\partial\mathcal{D}_{\text{conv}} = \partial\mathcal{D}_{\text{conv}_1} \cup \partial\mathcal{D}_{\text{conv}_2} \cup \partial\mathcal{D}_{\text{conv}_3}$. The interface between $\mathcal{D}_1$ and $\mathcal{D}_2$ is noted $\partial\mathcal{D}_{12}$ and the interface between $\mathcal{D}_2$ and $\mathcal{D}_3$ is noted $\partial\mathcal{D}_{23}$. The temperature T_f of the surounding fluid (air) is uniform and the convective heat transfer coefficient h is also uniform[2]. The contact resistances between $\mathcal{D}_1$ and $\mathcal{D}_2$ and between $\mathcal{D}_2$ and $\mathcal{D}_3$ are neglected. We use the notation $\vec{x}$ for locations within the solids ($\mathcal{D}_1$, $\mathcal{D}_2$ or $\mathcal{D}_3$) and the notation $\vec{y}$ for surface locations ($\partial\mathcal{D}_{\text{conv}}$, $\partial\mathcal{D}_{12}$ or $\partial\mathcal{D}_{23}$). The objective is to numerically estimate the temperature $T_\text{j} \equiv T(\vec{x}_\text{j})$, at a location $\vec{x}_{\text{obs}} \equiv \vec{x}_\text{j}$ inside $\mathcal{D}_1$, representative of the junction temperature, at the stationnary regime.

The corresponding heat transfer model is the following:

$$\begin{cases} \lambda_1\,\Delta T + \Psi = 0 & \forall \vec{x} \in \mathcal{D}_1 \quad (1) \\ \Delta T = 0 & \forall \vec{x} \in \mathcal{D}_2 \cup \mathcal{D}_3 \quad (2) \\ -\lambda_1 \vec{\nabla} T.\vec{n}_1 = \lambda_2 \vec{\nabla} T.\vec{n}_2 & \forall \vec{x} \equiv \vec{y} \in \partial\mathcal{D}_{12} \quad (3) \\ -\lambda_2 \vec{\nabla} T.\vec{n}_2 = \lambda_3 \vec{\nabla} T.\vec{n}_3 & \forall \vec{x} \equiv \vec{y} \in \partial\mathcal{D}_{23} \quad (4) \\ -\lambda_\ell \vec{\nabla} T.\vec{n}_\ell = h(T - T_{\text{BF}}) & \forall \vec{x} \equiv \vec{y} \in \partial\mathcal{D}_{\text{conv}_\ell} \quad (5) \end{cases}$$

where $\vec{n_1}$ is the outward unit surface normal of $\mathcal{D}_1$ and $\vec{n_2}$ is the outward unit surface normal of $\mathcal{D}_2$ (for $\vec{y} \in \partial\mathcal{D}_{12}$, $\vec{n_2} = -\vec{n_1}$) and ℓ an index equal to either 1,2 or 3.

2.2 The equivalent statistical model

Heat equations Eqs. (1) and (2) and their boundary conditions Eqs. (3) to (5) are reformulated to get a statistical interpretation enabling a Monte Carlo approach. Such

[1] Thermal properties are independant of temperature.The addressed problem is therefore linear regarding temperature.

[2] We retain a convective heat transfer coefficient that is uniform for simplification of the analysis and the comparisons with other numerical solutions. There no theoretical limitation their and h could depend on location (different values for the plate and and the fins, higher values close to the fans, etc) as the result of detailed fluid mechanics simulations.

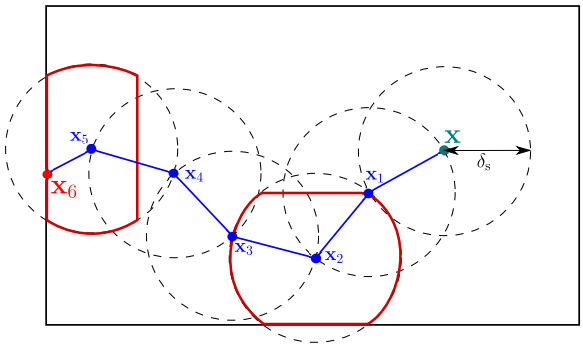


Fig. 2. Illustration of a random path sampling compatible with path-tracing acceleration techniques of computer graphics. Each sphere has the same radius δ_S and is adjusted when getting close to the domain boundary.

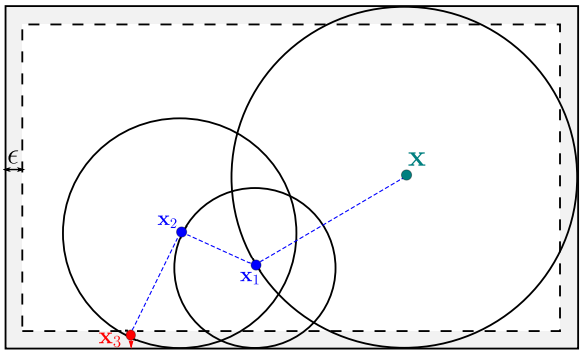


Fig. 3. Illustration of the sampling of a random path using the random walk-on-sphere method to estimate density at location $\vec{x}$. In order to end the random walk, the boundary of the domain is thickened of a small value ϵ in which the final position $\vec{x}_3$ is projected on the boundary.

translations are as old as numerical heat transfer history [14, 15, 16] and have given rise to the well-known *Walk on Sphere* Monte Carlo algorithm family [17, 18, 19]. However, the primary strength of modern Monte Carlo approaches to heat transfer is the use of the same acceleration techniques developed by the computer graphics community for the film and video game industries: the computation times become insensitive to the level of geometrical description refinement (the number of triangles in the geometrical mesh). Moreover, these techniques are currently incompatible with the deployment of Walk on Sphere algorithms in their purest form. The primary reason is that Walk on Sphere algorithms require knowledge of the largest sphere that fits the volume (the first sphere-surface intersection as the radius increases), whereas computer graphics techniques are based on line-surface intersections. We will not return to the underlying causes of this contradiction; rather, we will describe in detail one of the algorithms that can be used to circumvent it. The corresponding formalism is essentially identical to that found in the Walk on Sphere literature, with the exception that the radius of the sphere is adjusted as a function of the distance between line-surface intersections and is always chosen to be small enough to permit numerical approximations at each step [20, 21, 22, 23, 24](see Fig. 2).

The main idea is that Eqs. (1) and (2) are Laplace equations, one with a uniform source and the other without, and that the solutions to these equations satisfy the following integral constraints relating the temperature at the center of a sphere to the temperature field along the sphere: a quasi-constant source term inside a sphere with a radius of δ. The solution of Eq. (1) corresponds to Eq. (6) (the procedure is detailed in **??**) :

$$T(\vec{x}) = \int_{4\pi} \frac{1}{4\pi} T(\vec{x} + \delta\vec{u})\, d\vec{u} + \frac{\Psi\delta^2}{6\lambda_1} \qquad \forall \vec{x} \in \mathcal{D}_1 \quad (6)$$

Similarly, the solution of Eq. (2) corresponds to the following equation:

$$T(\vec{x}) = \int_{4\pi} \frac{1}{4\pi} T(\vec{x} + \delta\vec{u})\, d\vec{u} \qquad \forall \vec{x} \in \mathcal{D}_2 \quad (7)$$

The ponderation factor $\frac{1}{4\pi}$ in the integrals over the sphere signifies that these integrals may be analyzed as uniform averages across the sphere. The temperature $T(\vec{x})$ at the center of the sphere with radius δ is the uniform average of the temperatures $T(\vec{x}+\delta\vec{u})$ on the sphere, according to Eq. (7). The source term $\frac{\Psi\delta^2}{6\lambda_1}$ can be incorporated into the integral for Eq. (6) in order to obtain the temperature $T(\vec{x})$ at the sphere's center as the uniform average of $T(\vec{x}+\delta\vec{u}) + \frac{\Psi\delta^2}{6\lambda_1}$. Therefore, in order to calculate the temperature at a specific location $\vec{x}$, we can employ a Monte Carlo algorithm that samples uniformly the sphere surrounding $\vec{x}$ with a radius of δ, obtain the temperature at the sampled location, and calculate the mean (if necessary, including the source). However, in general, the temperature at the location on the sphere that was sampled is undetermined. Nevertheless, a significant strategy employed in Monte Carlo practice is the utilization of double randomisation: in situations where one Monte Carlo algorithm requires an unknown quantity that could be accounted for by another Monte Carlo algorithm, the second

Monte Carlo algorithm is simply invoked as a continuation of the first ("the expectation of an expectation is an expectation"). In this case, this means that when we reach a point on the sphere where the temperature is unknown, we simply sample a location on this second sphere and define a new sphere around it. The Walk On Sphere strategy has just been succinctly outlined: locations are sampled uniformly along spheres in succession until a known temperature is encountered (see Fig. 3). In Poisson equation, the only distinction is that the source term is accumulated at each stage.

The genuine challenge lies in the close proximity to boundaries. While the Walk On Sphere algorithm described above is exact, it requires numerical approximations when dealing with boundaries. The radius of the sphere is modified in the majority of Walk On Sphere algorithms to ensure that it is precisely tangent to the boundary. Subsequently, a location is sampled from this sphere, and its boundary membership is determined by whether its distance to the boundary is less than a predetermined numerical parameter denoted as ϵ (the "thickness" of the boundary) [17, 18, 19]. The WOS literature contains a detailed analysis of this approximation, which demonstrates that the numerical convergence associated with it is outstanding. This is one of WOS numeric's assets. Regrettably, this approach to boundary handling is incompatible with modern instruments designed to accelerate path sampling in complex geometries and inherited from the computer graphics commmunity [25] These tools were developed exclusively for line-boundary intersections (locating the nearest intersection with the boundary in a given direction) and not sphere-boundary intersections (locating the largest sphere tangent to the boundary). This is a highly active area of research in computer graphics that is currently under investigation. However, in order to continue benefiting from the insensitivity of Monte Carlo algorithms to the complexity of the geometry (same computation time regardless of the level of refinement in the geometric description), we must handle the vicinity of boundaries using only line-boundary intersections [26, 20]. In this case, we will implement the suggested solution by utilizing two line-boundary intersections at each stage [27, 28, 29, 24]. Regardless of the distance to the boundary, the approach entails sampling a location on a sphere of radius δ in an isotropically sampled direction $\vec{u}$; subsequently, we examine the intersection of the sampled location with the boundary in both the direction $\vec{u}$ and the direction $-\vec{u}$; if the distance between the intersection and the boundary in question is less than δ, then δ is modified to reflect this new value, and the sampled location becomes $\vec{x} + \delta_{\text{intersect}}\ \vec{u}$.

The first term in Eqs. (6) and (7) is derived from the classical result of harmonic functions theory, which states that the average value of T over the surface of a sphere with any radius δ is equivalent to the value of T at the sphere's center, assuming no external sources. $\vec{u}$ represents the direction unit vector in both equations.

Then, discretization is performed on the three boundary conditions corresponding to Eqs. (3) to (5) using finite differences and a backward difference scheme corresponding to the following equation (further information regarding discretization of boundary conditions can be found in **??**):

$$\vec{\nabla} T.\vec{n} \approx \frac{T(\vec{y}) - T(\vec{y} - \delta\vec{n})}{\delta} \tag{8}$$

In equations (6), (7) and (8), δ is a numerical parameter corresponding to the walking step for conduction in the solid. In equations (6) and (7), when $\vec{x}$ is in the vicinity of the boundary, δ inside the angular integral must be adjusted and numerous strategies are available in the literature. The numerical parameter denoted as δ in Eqs. (6) to (8) represents the walking step for conduction in the solid. In Eqs. (6) and (7), when $\vec{x}$ is in the vicinity of the boundary, δ within the angular integral must be modified; numerous methods for this purpose are described in the literature

By substituting the given Eq. (8) for the three boundary conditions Eqs. (3) to (5) and for any location $\vec{y}$ belonging to successively the boundaries $\partial\mathcal{D}_{12}$, $\partial\mathcal{D}_{23}$ and $\partial\mathcal{D}_{\text{conv}_l}$, one obtains respectively the following equations:

$$\begin{aligned} T(\vec{y}) = &\overbrace{\frac{\lambda_1}{\lambda_1 + \lambda_2 \frac{\delta b_1}{\delta b_2}}}^{\mathrm{P}_{12}} T(\vec{y} - \delta_{b_1}\vec{n_1}) \\ &+ \underbrace{\frac{\lambda_2}{\lambda_2 + \lambda_1 \frac{\delta b_2}{\delta b_1}}}_{1-\mathrm{P}_{12}} T(\vec{y} - \delta_{b_2}\vec{n_2}) \forall \vec{y} \in \partial\mathcal{D}_{12} \end{aligned} \tag{9}$$

$$T(\vec{y}) = \overbrace{\frac{\lambda_2}{\lambda_2 + \lambda_3 \frac{\delta b_2}{\delta b_3}}}^{\mathrm{P}_{23}} T(\vec{y} - \delta_{b_2}\vec{n_2}) + \underbrace{\frac{\lambda_3}{\lambda_3 + \lambda_2 \frac{\delta b_3}{\delta b_2}}}_{1-\mathrm{P}_{23}} T(\vec{y} - \delta_{b_3}\vec{n_3}) \forall \vec{y} \in \partial\mathcal{D}_{23} \tag{10}$$

$$T(\vec{y}) = \overbrace{\frac{h}{h + \frac{\lambda_l}{\delta b_l}}}^{\mathrm{P}_{\mathrm{conv}_l}(h)} T_f + \underbrace{\frac{\frac{\lambda_l}{\delta b_l}}{h + \frac{\lambda_l}{\delta b_l}}}_{1-\mathrm{P}_{\mathrm{conv}_l}(h)} T(\vec{y} - \delta_{b_l}\vec{n_l}) \forall l = 1,2,3; \forall \vec{y} \in \partial\mathcal{D}_{\mathrm{conv}_l} \tag{11}$$

Eqs. (9) to (11) can be given a statistical interpretation by introducing probabilities $\mathrm{P}_{12} = \frac{\lambda_1}{\lambda_1 + \lambda_2 \frac{\delta b_1}{\delta b_2}}$, $\mathrm{P}_{23} = \frac{\lambda_2}{\lambda_2 + \lambda_3 \frac{\delta b_2}{\delta b_3}}$ and $\mathrm{P}_{\mathrm{conv}_l}(h) = \frac{h}{h + \frac{\lambda_l}{\delta b_l}}$ $(\forall l \in 1,2,3)$; as well as Eqs. (6) and (7) introducing the probability density $p(\vec{u}) = \frac{1}{4\pi}$.

Those equations can be gathered using the Heaviside function $\mathcal{H}$ (taking the value 1 if the condition is satisfied, 0 otherwise), to express the complete Monte Carlo algorithm to evaluate the junction temperature $T_\mathrm{j}(\vec{x}_\mathrm{obs})$, this equation can hence be visualized represented in an algorithmic way in Fig. 4[3].

Fig. 4 can be given a statistical interpretation as the integral formulation of the junction temperature may be read as the expectation of a random variable W corresponding to the Monte Carlo weight, such as $T_\mathrm{j}(\vec{x}_\mathrm{obs}) = \mathbb{E}(W)$. Every realization w_i represents a random path that starts at a location $\vec{x}_\mathrm{obs}$ in the module's core ($\mathcal{D}_1$) and progresses through volumes $\mathcal{D}$ until it reaches the predetermined temperature T_f. A power term contribution of $\frac{\Psi L_j^2}{6\lambda_1}$ is accumulated at each step j of the walk in the volume as the path traverses the volume $\mathcal{D}_1$ corresponding to the module. The index j is a counter incremented at each step executed in the volume $\partial\mathcal{D}_1$. At the end of an i^{nth} MC realisation, a path carried out K_i paths in the volume $\mathcal{D}_1$. Hence, the Monte Carlo weight w_i corresponding to the i^{nth} realization, can be expressed as the following:

$$w_i = T_{\mathrm{BF}} + \frac{\Psi}{6\lambda_1} \sum_{j=1}^{K_i} L_j^2 \tag{12}$$

Fig. 5 illustrates the Monte Carlo algorithm diagram associated with a particular realization i of the MC weight w_i. The subsequent steps comprise the procedure of a realization:

The path starts in volume $\mathcal{D}_1$, in the module's center, where the estimation of the junction temperature is required. The current location $\vec{x}_\mathrm{c}$ is set at the probe start position $\vec{x}_0$ at the center of the volume $\mathcal{D}_1$. The path length L_j is initialized to zero and the corresponding counter j is set equal to 1. A direction $\vec{u}$ is sampled isotropically according to a probability density $p_{\vec{U}}(\vec{u}) = \frac{1}{4\pi}$ and a step δ is made in the direction $\vec{u}$. Then, a test is performed to know if the segment $[\vec{x}_\mathrm{c} - \delta\vec{u}; \vec{x}_\mathrm{c} + \delta\vec{u}]$ intersects one of the boundaries forming $\partial\mathcal{D}_1$[4]. If the segment interesects a boundary, the current discretization step δ_c is adjusted (see Fig. 2). Otherwise, the segment does not intersect any boundary and the current discretization step δ_c value is kept. The current position is updated, such as, $\vec{x}_\mathrm{c} \leftarrow \vec{x}_\mathrm{c} + \delta_\mathrm{c}\vec{u}$, j is incremented by one and L_j stores the added current path length. At each step in the module volume $\mathcal{D}_1$ informations along the path are recorded to store the contribution related to the volumetric power density Ψ.

After this, a test is performed to know if the current location $\vec{x}_\mathrm{c}$ is on a boundary $\partial\mathcal{D}_1$. If not, the randow walk in the volume goes on another conductive path is sampled in $\mathcal{D}_1$ such as previously. If the location is on a boundary, what happens next is determined by which boundary has been reached. If $\vec{x}_\mathrm{c}$ belongs to the solid/fluid boundary $\partial\mathcal{D}_{\mathrm{conv}_1}$ and a Bernoulli test is performed according to the convective probabilities $\{\mathrm{P}_{\mathrm{conv}_1}(h); 1-\mathrm{P}_{\mathrm{conv}_1}(h)\}$. Either, the path stops at this solid/fluid boundary and the Monte Carlo weight corresponding to Eq. (12) is stored or the path is reinjected in the volume $\mathcal{D}_1$ and the conductive random walk goes on in the volume. Otherwise, $\vec{x}_\mathrm{c}$ is located on the boundary $\partial\mathcal{D}_{12}$ between $\mathcal{D}_1$ and $\mathcal{D}_2$, a Bernoulli test is performed to know if the conductive path goes on in $\mathcal{D}_2$ or keeps going in $\mathcal{D}_1$ according to the probabilities $\{\mathrm{P}_{12}; 1 - \mathrm{P}_{12}\}$. At the end of this test, either the path remains in $\mathcal{D}_1$ and is reinjected along the

[3] Considering that $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$ (volume elements) and $\partial\mathcal{D} = \partial\mathcal{D}_{\mathrm{conv}_1} \cup \partial\mathcal{D}_{\mathrm{conv}_2} \cup \partial\mathcal{D}_{\mathrm{conv}_3} \cup \partial\mathcal{D}_{12} \cup \partial\mathcal{D}_{23}$ (boundary elements).

[4] The test is performed in the direction $-\vec{u}$ in order to respect the Laplacian harmonicity Eq. (6) and Eq. (7)

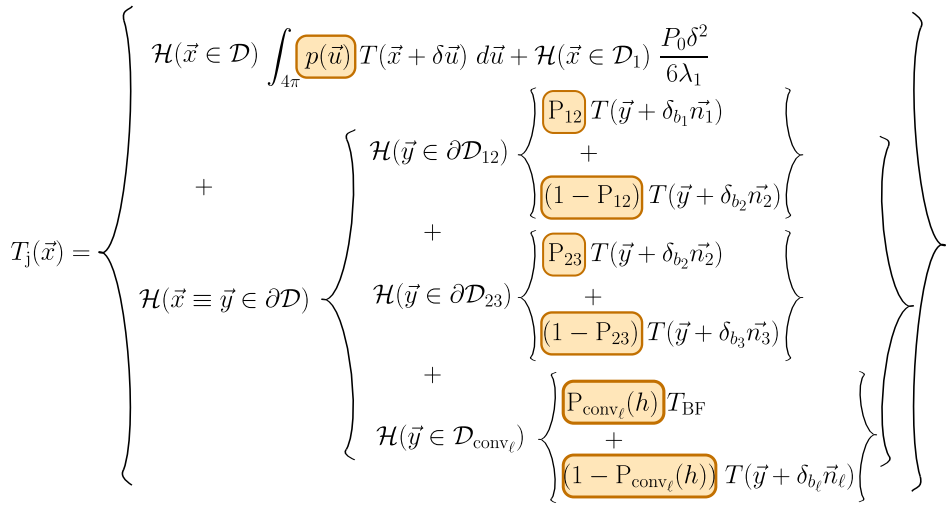


Fig. 4. Equation describing the Monte Carlo algorithm. Probabilities are enclosed in orange frames.

normal $\vec{n}_1$ at a location $\vec{x}_\mathrm{c} - \delta_{\mathrm{b}_1}\vec{n}_1$ within volume $\mathcal{D}_1$ and the conductive random walk goes on in the volume, or the path leaves $\mathcal{D}_1$ for $\mathcal{D}_2$ and the path is relocated along the normal $\vec{n}_2$ at a location $\vec{x}_\mathrm{c} - \delta_{\mathrm{b}_2}\vec{n}_2$ within $\mathcal{D}_2$ while the conductive random walk continues.

The conductive random walk is executed in the volume $\mathcal{D}_2$ in the same manner as in volume $\mathcal{D}_1$, until it reaches a boundary. The sole distinction lies in the absence of volumetric power density in $\mathcal{D}_2$, consequently, no contribution accumulates along the sampled path. When reaching a boundary, the path encounters one of the following: solid/fluid boundary $\partial\mathcal{D}_{\mathrm{conv}_2}$, solid/solid boundary $\partial\mathcal{D}_{23}$ or solid/solid boundary $\partial\mathcal{D}_{12}$. In the event that the path intersects the solid/fluid boundary, a Bernoulli test is executed in accordance to the convective probabilities $\{\mathrm{P}_{\mathrm{conv}_2}(h); 1-\mathrm{P}_{\mathrm{conv}_2}(h)\}$. In this case, either the path terminates and the Monte Carlo weight (see Eq. (12)) is retained, or it is reinjected at a location $\vec{x}_\mathrm{c} - \delta_{\mathrm{b}_2}\vec{n}_2$ in $\mathcal{D}_2$. Subsequently, the conductive random walk continues throughout the volume $\mathcal{D}_2$. When the path encounters the solid/solid boundary $\partial\mathcal{D}_{23}$, a Bernoulli test is executed to determine whether the path continues in $\mathcal{D}_2$ or continues in $\mathcal{D}_3$ based on the corresponding probability values $\{\mathrm{P}_{23}; 1-\mathrm{P}_{23}\}$. In line with prior details, the random conductive walk continues after the reinjection in the corresponding volumes $\mathcal{D}_2$ or $\mathcal{D}_3$ starting at the corresponding location, respectively, $\vec{x}_\mathrm{c} - \delta_{\mathrm{b}_2}\vec{n}_2$ or $\vec{x}_\mathrm{c} - \delta_{\mathrm{b}_3}\vec{n}_3$. When the path encounters the solid/solid boundary $\partial\mathcal{D}_{12}$, a Bernoulli test is executed to determine, on the basis of the corresponding probabilities $\{\mathrm{P}_{12}, 1-\mathrm{P}_{12}\}$, whether the path proceeds the conductive walk in $\mathcal{D}_2$ or returns to $\mathcal{D}_1$.

When the path leaves $\mathcal{D}_2$ for $\mathcal{D}_3$, in a manner analogous to that which was described in volume $\mathcal{D}_2$, a conductive random walk is executed until a boundary is encountered. This boundary is either the solid/solid boundary $\partial\mathcal{D}_{23}$ or the solid/fluid boundary $\partial\mathcal{D}_{\mathrm{conv}_3}$. Identical to before, a Bernoulli test is conducted if this boundary is $\partial\mathcal{D}_{23}$ to determine whether the path continues in $\mathcal{D}_3$ or exits $\mathcal{D}_3$ to resume the conductive random walk in $\mathcal{D}_2$. Ultimately, when the path reaches the solid/fluid boundary $\partial\mathcal{D}_{\mathrm{conv}_3}$, as determined by the Bernoulli test according to probabilities $\mathrm{P}_{\mathrm{conv}_3}(h); 1-\mathrm{P}_{\mathrm{conv}_3}(h)$: either the path terminates and the Monte Carlo weight value is retained (see Eq. (12)), or the path is reinjected into the volume at a location $\vec{x}_\mathrm{c} - \delta_{\mathrm{b}_3}\vec{n}_3$.

From a synthetic perspective, the conductive path commences at location $\vec{x}_{\mathrm{obs}}$ within the module's core, passes through the different components volumes, and ends at the solid/fluid boundary via convective exchange when encountering the temperature T_{BF}. Along the way, it accumulates a contribution denoted as $\frac{\Psi}{6\lambda_1}\sum_{j=1}^{K_i} L_j$, which is attributable to the power density Ψ present in the module. A Monte Carlo computation consists in N realizations w_i of W are performed and the mean m of these realizations corresponds to the estimation of the junction temperature $T_j(\vec{x}_{\mathrm{obs}})$. The equations that represent the temperature estimate and the corresponding statisticall errors are as follows:

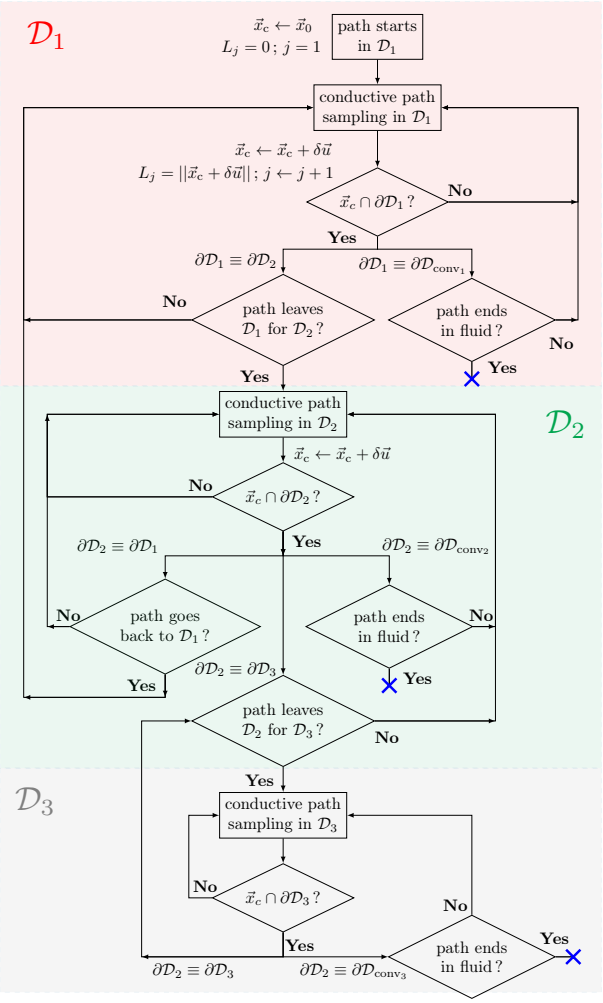


Fig. 5. Scheme describing the Monte Carlo algorithm to evaluate the junction temperature at the i^{nth} realization

$$T_j(\vec{x}_{\text{obs}}) \simeq m = \frac{1}{N}\sum_{i=1}^{N} w_i \tag{13}$$

$$s = \frac{1}{\sqrt{N}}\left[\frac{1}{N}\sum_{i=1}^{N} w_i^2(h) - \left(\frac{1}{N}\sum_{i=1}^{N} w_i\right)^2\right]^{\frac{1}{2}} \tag{14}$$

3 SYMBOLIC MONTE CARLO

3.1 The corresponding symbolic model on h

The Monte Carlo algorithm for determining the junction temperature $T_j(h|\vec{x}_{\text{obs}}, \tilde{h})$ for a given value $h = \tilde{h}$ has been defined previously. We shall now demonstrate how the storing of the information contained in the random paths is affected when h is encountered, thereby constructing the temperature as a function of the heat transfer coefficient h. The exhaustive method is described in [22].

The starting point is a rewriting of the solid/fluid boundary condition (see Eq. (5)) utilizing an importance sampling strategy, i.e., introducing a new arbitrary probability $\tilde{\mathrm{P}} = \frac{\tilde{h}}{\tilde{h}+\frac{\lambda_l}{\delta_{\mathrm{b}_l}}}$ and adjusting the Monte Carlo weight as necessary, the following modifications are made:

$$\begin{aligned} T(h|\vec{x}_{\text{obs}}, \tilde{h}) =& \tilde{\mathrm{P}}\left\{\frac{\mathrm{P}_{\text{conv}_l}(h)}{\tilde{\mathrm{P}}}\, T_{\text{BF}}\right\} \\ &+ (1-\tilde{\mathrm{P}})\left\{\frac{1-\mathrm{P}_{\text{conv}_l}(h)}{1-\tilde{\mathrm{P}}}\, T(\vec{y} - \delta_{\mathrm{b}_l}\vec{n}_l)\right\} \end{aligned} \tag{15}$$

Finally, the functional expression for the Monte Carlo weight and the temperature expression are provided:

$$w_i(h) = \left(\frac{\mathrm{P}_{\text{conv}_l}(h)}{1-\tilde{\mathrm{P}}}\right)^{k_i} \frac{\mathrm{P}_{\text{conv}_l}(h)}{\tilde{\mathrm{P}}}\, T_{\text{BF}} + \frac{\Psi}{6\,\lambda_1}\sum_{j=1}^{J_i} L_j^2 \tag{16}$$

With k_i the number of times the path reached the boundary and continued in the solid (reinjection) during one Monte Carlo realization.

Hence, the temperature estimation $T_j(\vec{x}_{\text{obs}}|h)$ as a function of h with Symbolic Monte Carlo, may be expressed as follows:

$$T_j(h|\vec{x}_{\text{obs}}, \tilde{h}) \simeq \frac{1}{N}\sum_{i=1}^{N} w_i(h) \tag{17}$$

$$s(h) = \frac{1}{\sqrt{N}}\left[\frac{1}{N}\sum_{i=1}^{N} w_i(h)^2 - \left(\frac{1}{N}\sum_{i=1}^{N} w_i(h)\right)^2\right]^{\frac{1}{2}} \tag{18}$$

4 RESULTS AND ANALYSIS

Fig. 1 depicts the configuration that was utilized to test the method. In this configuration the Monte Carlo and Symbolic Monte Carlo algorithms are used and compared to a reference thermal simulation tool used in the electronic area, 6SigmaET software. This first part of results will be discussed in Sec. 4.1.

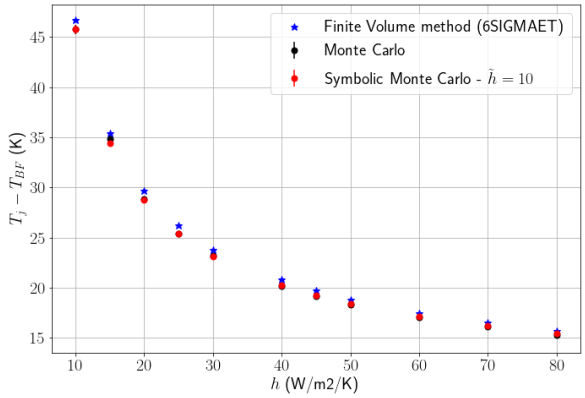


Fig. 6. Monte Carlo simulations (black rounded marks) computed each for $N = 10^4$ and 6SIGMAET simulation results (blue star marks) to evaluate the junction temperature $T_j(\vec{x}_{\text{obs}})$ according to the heat transfer coefficient h. Results from the Symbolic Monte Carlo method (red rounded marks) were computed thanks to the evaluation of the function estimated with a single calculation with $\tilde{h} = 10\ \text{W}\,\text{m}^{-2}\,\text{K}^{-1}$. Results are presented as temperature gradients $T_j(\vec{x}_{\text{obs}}) - T_{\text{BF}}$ with $T_{\text{BF}} = 295.65\text{K}$.

In the second stage, the Symbolic Monte Carlo method will be implemented in the case illustrated in Fig. 7 which is a typical example of electronic component in electric vehicles [30]. . This second case involves the forced convection cooling of four power electronics modules. Two cases will be investigated, a single-sided cooling and double-sided cooling, this would be the objectif of Sec. 4.2.

4.1 Validation of the Monte Carlo and Symbolic Monte Carlo methods

The Monte Carlo algorithm is utilized to estimate the junction temperature in the initial case involving the electronic module disposed on a plate fin heat sink (see Fig. 1), employing the heat transfer model delineated by Eqs. (1) to (3) and (5). For the purposes of numerical comparison, the Monte Carlo results and results of Finite Volume simulations generated by the 6SIGMAET software are compared for various convection coefficient h values, from $h = 10\text{W}\,\text{m}^{-2}\,\text{K}^{-1}$ up to $h = 80\text{W}\,\text{m}^{-2}\,\text{K}^{-1}$. These results are presented in Fig. 6.

Each of the Monte Carlo computation has been performed in order to output a result with a relative error value below or equal to 1%, the relative error being expressed as $s(h)/(T_j(\vec{x}_{\text{obs}}, h) - T_{\text{BF}})$. In the case of the Symbolic Monte Carlo, the single computation of the function has been performed at $\tilde{h} = 10\ \text{W}\,\text{m}^{-2}$

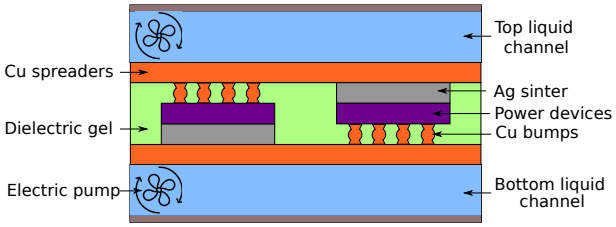


Fig. 7. Second case of a double sided component.

where the relative error is as well equal to 1%. For the other values of h, the relative error related to the computation of the junction temperature with the function never exceeds 3%. Afterwards, relative errors between the Monte Carlo and Symbolic Monte Carlo results and 6SIGMAET Finite Volume results have been computed for each value of the heat transfer coefficient and are represented in Table 1. The values of relative errors are below 4%. Most of the values are under 3% except the one at $h = 25\text{W}\,\text{m}^{-2}\,\text{K}^{-1}$, indeed at such value of the convective heat transfer coefficient, conductive paths tend to be longer within volumes and have more difficulties to terminate, in contrary with higher values of h. This phenomenom retranscripted by the conductive path reflects the reality of the physics but may cause more variance in the case of the Monte Carlo method.

h (W m^{-2} K^{-1})	$\lvert\frac{T_{\text{MC}} - T_{\text{6SIGMAET}}}{T_{\text{6SIGMAET}}}\rvert \times 10^{-2}$	$\lvert\frac{T_{\text{SMC}} - T_{\text{6SIGMAET}}}{T_{\text{6SIGMAET}}}\rvert \times 10^{-2}$
10	2.01	2.01
15	1.64	2.67
20	2.97	3.02
25	3.05	3.00
30	2.59	2.77
40	2.89	2.69
45	2.59	2.33
50	2.74	2.01
60	2.44	2.01
70	2.35	1.89
80	2.73	1.74

Table 1. Relative errors between the estimations of the junction temperature in the case of the Monte Carlo method (T_{MC}), the Symbolic Monte Carlo method (T_{SMC}) and the results from the Finite Volume method from the 6SIGMAET software (T_{SIGMAET}).

4.2 Application of the SMC method in the case of forced convection cooling

The studied configuration consists of two copper spreaders encapsulating four power modules (chips). The four chips are composed of a typical component material. Two of the four power modules (chips 1 and 4) are linked to the bottom spreader via silver sinters while copper alloy bumps ensure the mechanical resistance and connection to the top spreader. The remaining two chips (chips 2 and 3) are connected to the top spreader via copper alloy bumps and the bottom spreader via silver sinters. Only chips 1 and 3 are emitting a volumetric power density equal to $\Psi = 1.25 \times 10^{10} \mathrm{W\,m^{-3}}$ (corresponding to $250\mathrm{W}$ dissipated for each power module). Power modules, bumps and sinters are submerged in a dielectric gel made of typicall underfill. Lateral surfaces of the system are isolated, the top and bottom surfaces of the spreaders are in contact with channels filled with glycoled water cooled down by cryogeny, through which electric pumps circulates the fluid in the cylindrical cavities. The convection occuring between the the upper channel and the top surface of the top spreader is represented by a heat transfer coefficient h_{top}, whereas the convection occuring between the lower channel and the bottom surface of the bottom spreader is denoted by a heat transfer coefficient h_{bot}. For both solid/fluid interfaces, a unique boundary fluid temperature $T_{\text{BF}} = 263.15\mathrm{K}$ is considered. The aforementioned configuration is represented in a 2D front view in Fig. 7 in which power modules 1 and 2 are visible.

The four temperatures at the center of each of the four power modules, i.e. the junction temperatures, will be be estimated as a function depending on the bottom heat transfer coefficient h_{bot} with a Symbolic Monte Carlo approach as described in Sec. 3. To accomplish this, the heat transfer model under consideration closely resembles the one that was previously introduced in Sec. 2.1. Indeed, conduction at steady state with uniform volumetric power density in power modules 1 and 3 is described by Eq. (1) and stationnary conduction is described by Eq. (2) in power modules 2 and 4. No volumetric power exist in modules 2 and 4. Solid/solid interfaces between each of the components are described by equations depending on the thermal properties of the elements, similar to Eqs. (3) and (4) and solid/fluid boundary conditions for convection are represented by Eq. (5).

Two scenarios are considered, the first one will be the case of a single-side cooling in which natural convection is considered in the top channel and the power of the pump is variable is the bottom channel. The second scenario will be a double-side cooling configuration in which the water pump in the bottom channel is set a the maximum speed and the power of the pump in the top channel is changing. Both of these scenarios as well as the corresponding results are described in the following paragraphs.

Single-side cooling In the present case, h_{bot} is set equal to $10\ \mathrm{W\,m^{-2}\,K^{-1}}$ and h_{top} is variable. This value of h_{bot} corresponds to a configuration where the bottom channel is removed, the plate is at the contact of air and natural convection occurs with minimum radiation. Temperatures at the center of each module are evaluated as transfer function of the bottom heat transfer coefficient h_{top} and for the single Monte Carlo computation needed to estimate each of the four functions, the reference value of $\tilde{h}_{\text{top}}$ is set equal to $260.728\ \mathrm{W\,m^{-2}\,K^{-1}}$. The function is then used to estimate the junction temperature of each chip on an interval of h_{bot} values with a minimum value of 260.728 and a maximum value equal to 10000. For each evaluation of the transfer function, a Monte Carlo computation is made.

First, results from the Symbolic Monte Carlo are compared to results from the Finite Volume method used in the 6SIGMAET software. These results are presented in Fig. 8. Results from the Symbolic Monte Carlo method are relatively closed to the ones estimated by the Finite Volume method and show good concordance.

More closely, the relative errors $\epsilon = \frac{|T_{\text{SMC}} - T_{\text{6SIGMAET}}|}{T_{\text{6SIGMAET}}}$ estimated for each of temperature evolution of the chip according to h_{top} are plotted and represented in Fig. 9. The higher h_{top} value, the lower the relative error of the Symbolic Monte Carlo results compared to the Finite Volume method results. A maximum value of relative error may be observed for the minimum value of h_{top}, quite above 10% and the maximum error is in the case of chips 1 and 3 where a volumetric power is on. For higher values of h_{top}, the relative error values decrease and are way lower than 10%.

In order to judge of the relevance of an estimation of the temperature with the function, a good tool is the computation of the relative error such as s/m or $s/\Delta T$. In the case of a classical Monte Carlo resolution, the quantity estimation is judged relevant when this error is below 1%. In the case of the Symbolic Monte Carlo results, each of the relative errors $s/\Delta T$ are below 1%, except the temperature value estimated for $h_{\text{top}} = 10^4$ in which the relative error value is below 2%. A temperature computation with the function on the h_{top} range value is considered valid if this relative error is below 5%, but this is quite arbitrary, it can be more strict, hence in the results presented here the results are considered more than valid and acceptable regarding the relevance of the method itself.

Another major benefit of the function is the computation time gain compared to a classical Monte Carlo reso-

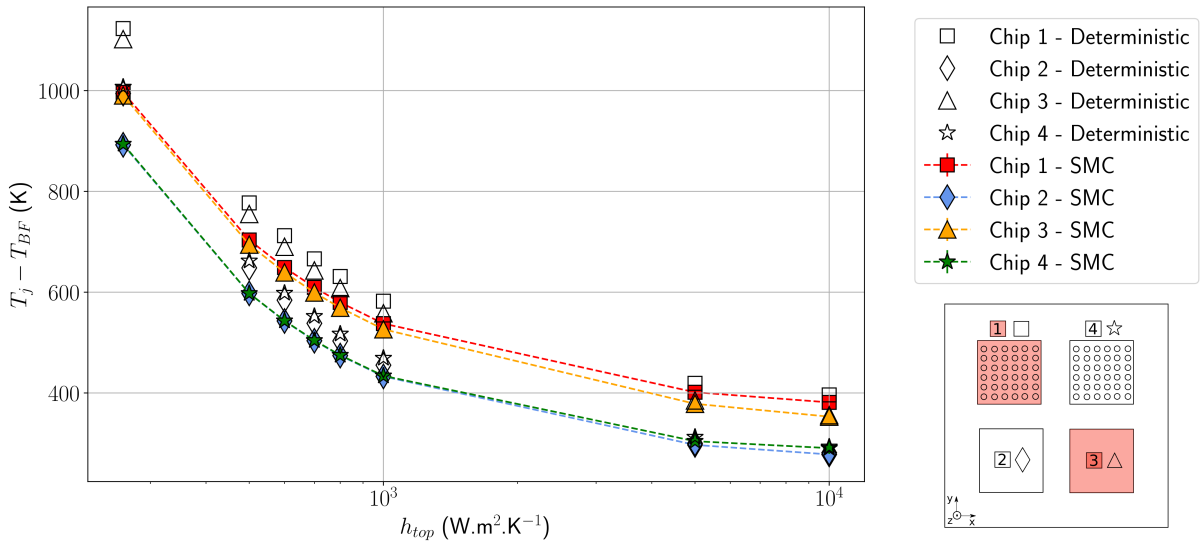


Fig. 8. Symbolic Monte Carlo simulations (coloured filled marks) were computed with the evaluation of the function. The function was estimated with a single Monte Carlo with $N = 10^5$ realizations and $\tilde{h}_{\text{top}} = 260.728\ \text{W m}^{-2}$. Results are presented as temperature gradients $T_j(\vec{x}_{\text{obs}}) - T_{\text{BF}}$ with $T_{\text{BF}} = 263.15\ \text{K}$

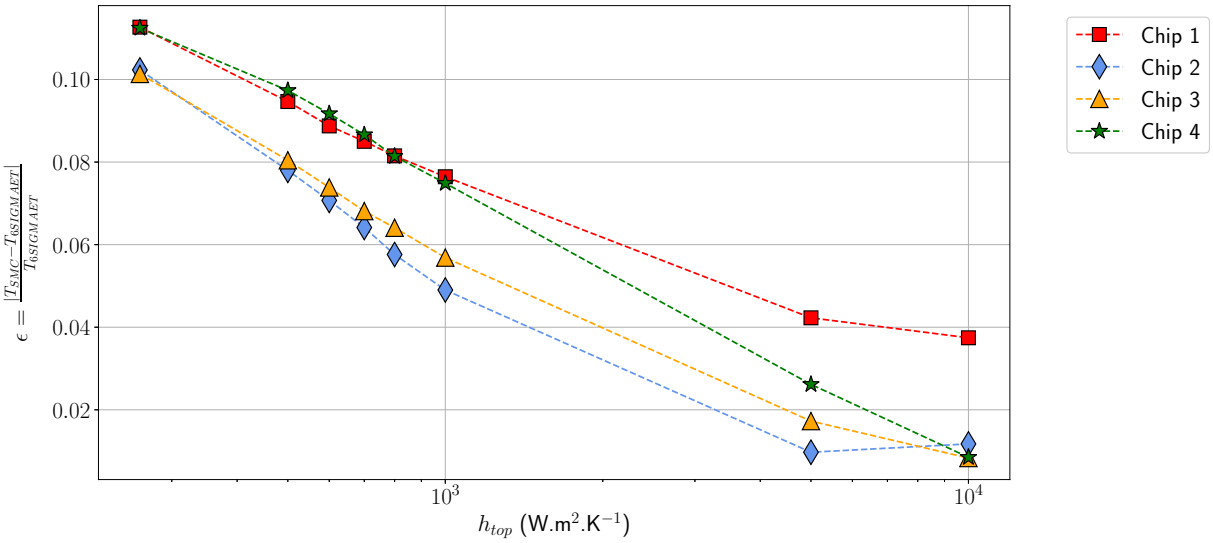


Fig. 9. Error profiles between results from 6SIGMAET simulations and results from the Symbolic Monte Carlo functions evaluations made for each chip.

lution in which the run needs to be done for each different value of h_{top}. The function needs to be computed with a single Monte Carlo computation with an incompressible time, but when this is done the function is used to evaluated the temperature. In these cases the temperature evaluation with the function is between 100 times faster for the lowest value of h_{top} and 10 times faster for the highest values (see Fig. 10).

Double-side cooling The top heat transfer coefficient h_{top} is set equal to 10000 $\text{W m}^{-2}\,\text{K}^{-1}$, corresponding to the pump at maximum speed in the liquid cooling chan-

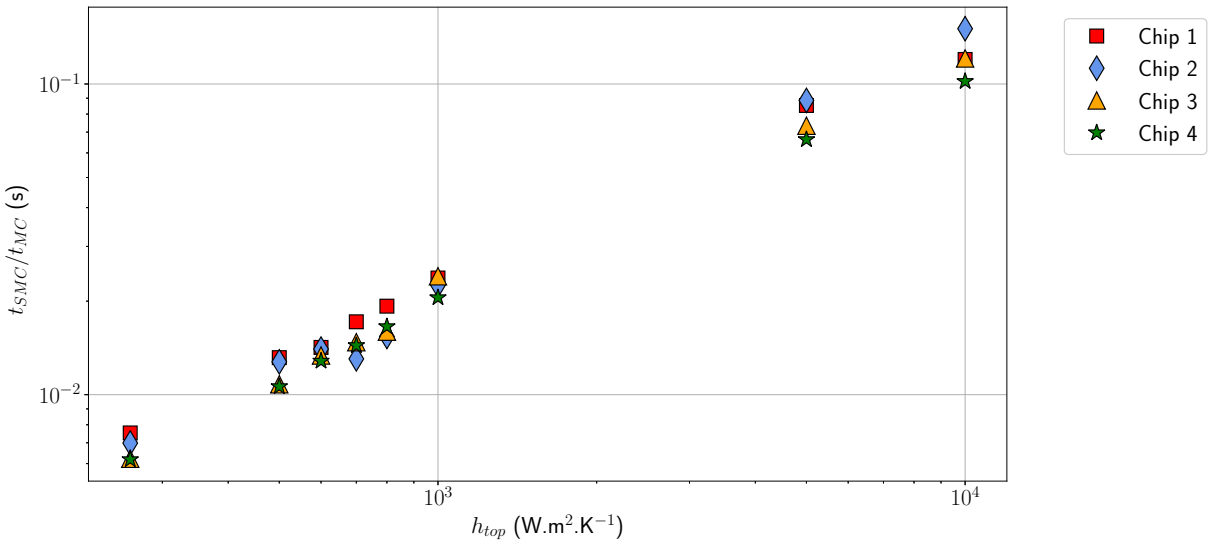


Fig. 10. Computation time ratios corresponding to $t_{\mathrm{SMC}}/t_{\mathrm{MC}}$ according to h_{top} values.

nel. For the single Monte Carlo simulation, the reference value of h_{top} is set equal to $260.728\ \mathrm{W\ m^{-2}\ K^{-1}}$ enabling the transfer function calculation. The function is then used to estimate the junction temperature of each chip on an interval of h_{top} values with a minimum of 260.728 and a maximum value equal to 10000. As previously, for each evaluation of the transfer function, a Monte Carlo computation is made.

Results with the function evaluation are compared to Finite Volume method results from the 6SIGMAET software, similarly as before. These results are presented in Fig. 11. Temperature estimations follow the right tendency when compared to 6SIGMAET results. The maximum gap is in the case of the chip 1 and 3 temperature evolution, when volumetric power are emitted.

If we look more closely to the relative error between the function from Symbolic Monte Carlo results and Finite Volume results though, as represented on Fig. 12, even in the case of chip 3, the relative error is below 4%. In the case of the other chips, most of the relative errors are below 2%. Hence, the results from the Symbolic Monte Carlo are considered in good accordance with the ones from the reference method, the Finite Volume results from 6SIGMAET.

Now, regarding the relevance of the results from the function estimated by the Symbolic Monte Carlo method in the heat transfer coefficient interval of values, let's compared the statistical relative error s/m similarly as before. The statistical relative error for the Symbolic Monte Carlo is compared to the relative error from classical Monte Carlo computations. For the same number of realizations, i.e. $N = 10^6$, classical Monte Carlo computations show a statistical relative error below 0.1% when results from the Symbolic Monte Carlo method show a statistical relative error below 1%. The temperature estimatons in the h_{top} value interval is hence considered valid regarding the relevance of the function. Moreover, statistical relative error are diminished compared to the single side cooling which may be explained by the fact that "paths tend to stop more easily so there is less variance".

Regarding the computation time gain, when compared to Monte Carlo results, a temperature evaluation with the function is 3000 times to 8000 times faster, respectively for lowest and highest values of h_{top}. Once more, compared to the single side cooling configuration, this configuration show a more profitable gain in computation time.

One last word to say that even if its trivial, when comparing the single and double side cooling, the results (see Fig and Fig) show that the lowest gradient $T_j - T_{\mathrm{BF}}$ may be achieved with double side cooling.

5 CONCLUSIONS

The objective of this article was to show the development of a Symbolic Monte Carlo algorithm in order to produce and estimate the junction temperature of a typical power module set (consisting of a module on "dissipateur à ailettes) as the function of a parameter directly expressing the cooling potential of a fluid, the heat trans-

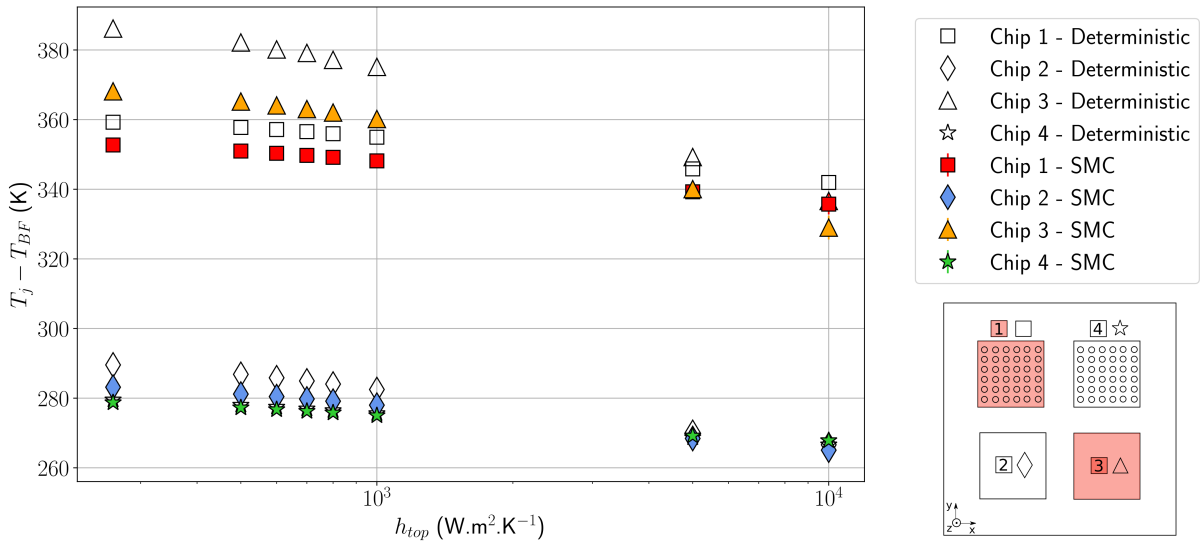


Fig. 11. Symbolic Monte Carlo simulations (coloured filled marks) were computed with the evaluation of the function. The function was estimated with a single Monte Carlo with $N = 10^6$ realizations and $\tilde{h}_{\text{top}} = 260.728\ \text{W m}^{-2}$. Results are presented as temperature gradients $T_j(\vec{x}_{\text{obs}}) - T_{\text{BF}}$ with $T_{\text{BF}} = 263.15\ \text{K}$

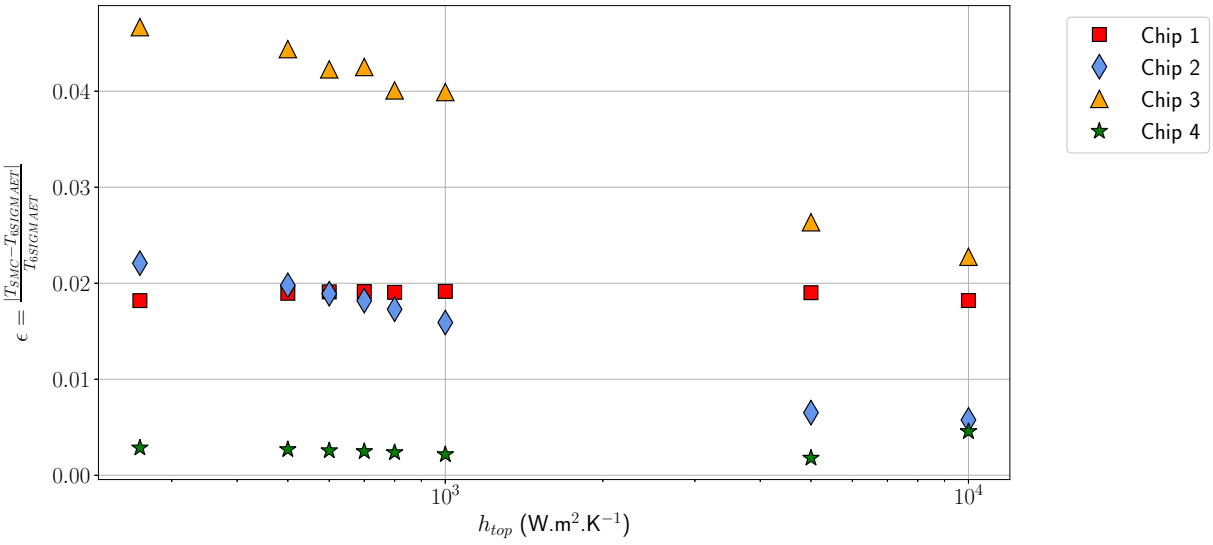


Fig. 12. Error profiles between results from 6SIGMAET simulations and results from the Symbolic Monte Carlo functions evaluations made for each chip.

fer coefficient. The implementation of this function and its comparison with reference method such as Finite Volume method used in the 6SIGMAET software has been conducted and showed good concordance. To go further and demonstrate the interest and potential of such a function in designed or optimization phases of power modules cooling systems, the Symbolic Monte Carlo algorithm to estimate the function was then implemented and deployed in a practical geometry of power modules from electric vehicles and more specifically in two extreme configurations, the single side and double side cooling of the multiples modules setup. In these cases, after a mandatory

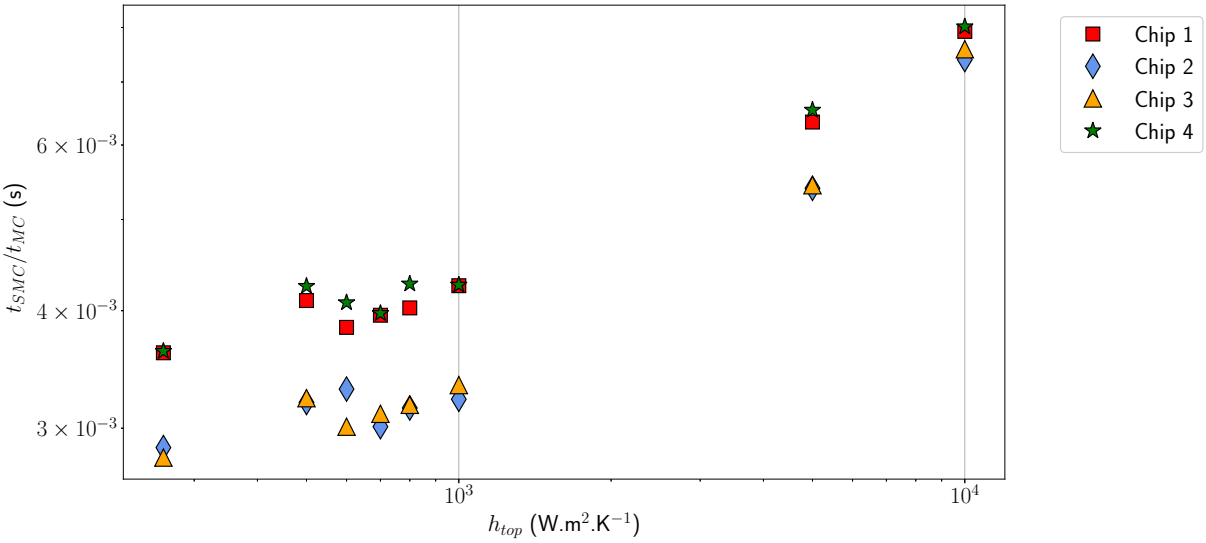


Fig. 13. Computation time ratios corresponding to $t_{\mathrm{SMC}}/t_{\mathrm{MC}}$ according to h_{top} values.

comparison with reference results from the same 6SGI-MAET software, these functions have shown their intern tools to question their relevance and know when they should not be used anymore and their interest of computation time gain when compared to classical Monte Carlo method. These two points show how to rely and the function and its interest in being used in a optimization process where the direct model need to be evaluated a large number of times.

This function proves as well its potential when choosing the right cooling configuration. Of course in this case it is nothing new, but still, suppose the researcher know the potential configurations of interest and the thermal model corresponding to it, the function can be estimated and it can be used for analysis purpose and help in a decision of a cooling configuration. Moreover, if we take a step back, with only one single Monte Carlo computation, which means with only one reference sampling of the physics, a landscape of multiple configurations may be explored, supposed it is in the range of the validity of the function. Through this function and the information contained in its coefficients, the physics is explored without compromise on the geometric complexity.

Now the perspectives for the work to follow, first it would be to explore a wider range of cooling configuratioon, go out of monophasic way of cooling and go through diphasic cooling.

ACKNOWLEDGEMENTS

Don't forget to thank everyone who needs to be thanked (especially Labex)!

REFERENCES

[1] Ledoux, C., Lefranc, P., and Larouci, C., 2013, "Pre-sizing methodology of embedded static converters using a virtual prototyping tool based on an optimisation under constraints method: comparaison of two power-sharing topologies," *IET Electrical Systems in Transportation,* **3**(1), pp. 1–9.

[2] Lee, S., Seaho Song, V. A., and Moran, K. P., 1996, "Constriction/spreading resistance model for electronics packaging," In ASME/JSME Thermal Engineering Conference, Vol. 4.

[3] Drofenik, U., and Kolar, J. W., 2006, "A thermal model of a forced-cooled heat sink for transient temperature calculations employing a circuit simulator," *IEEJ Transactions on Industry Applications,* **126**(7), pp. 841–851 Number: 7.

[4] Gammeter, C., Krismer, F., and Kolar, J. W., 2015, "Weight optimization of a cooling system composed of fan and extruded-fin heat sink," *IEEE Transactions on Industry Applications,* **51**(1), pp. 509–520.

[5] Yovanovich, M. M., Muzychka, Y. S., and Culham, J. R., 1999-10, "Spreading resistance of isoflux rectangles and strips on compound flux channels," *Journal of Thermophysics and Heat Transfer,* **13**(4), pp. 495–500.

[6] Castelan, A., Cougo, B., Dutour, S., and Meynard, T., 2019-04, "3d analytical modelling of plate fin heat sink on forced convection," *Mathematics and Computers in Simulation,* **158**, pp. 296–307.

[7] Dunn, W. L., 1981, "Inverse monte carlo analysis," *Journal of Computational Physics,* **41**(1), pp. 154–166 Number: 1.

[8] Dunn, W. L., 1983, "Inverse monte carlo solutions for radiative transfer in inhomogeneous media," *Journal of Quantitative Spectroscopy and Radiative Transfer,* **29**(1), pp. 19–26 Number: 1.

[9] Floyd, C. E., Jaszczack, R., Greer, K. L., and Coleman, R. E., 1986, "Inverse monte carlo as a unified reconstruction algorithm for ECT," *Journal of Nuclear Medicine*.

[10] Subramaniam, S., and Mengüç, M., 1991-01, "Solution of the inverse radiation problem for inhomogeneous and anisotropically scattering media using a monte carlo technique," *International Journal of Heat and Mass Transfer,* **34**(1), pp. 253–266.

[11] Dunn, W. L., and Shultis, J. K., 2009-10, "Monte carlo methods for design and analysis of radiation detectors," *Radiation Physics and Chemistry,* **78**(10), pp. 852–858.

[12] Galtier, M., Roger, M., André, F., and Delmas, A., 2017-07, "A symbolic approach for the identification of radiative properties," *Journal of Quantitative Spectroscopy and Radiative Transfer,* **196**, pp. 130–141.

[13] Sans, M., Penazzi, L., El Hafi, M., Caliot, C., Farges, O., Fournier, R., and Blanco, S., 2021, "Méthode de monte-carlo symbolique pour la caractérisation des propriétés thermophysiques : cas de la méthode flash," In SFT 2021 - 29ème congrès Français de Thermique.

[14] Kac, M., 1947, "Random Walk and the Theory of Brownian Motion," *The American Mathematical Monthly*.

[15] Kac, M., 1951, "On some connections between probability theory and differential and integral equations," In Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, pp. 189–215.

[16] Courant, R., Friedrichs, K., and Lewy, H., 1967, "On the partial difference equations of mathematical physics," *IBM Journal of Research and Development,* **11**(2).

[17] Muller, M. E., 1956, "Some Continuous Monte Carlo Methods for the Dirichlet Problem," *The Annals of Mathematical Statistics,* **27**(3), Sept., pp. 569–589.

[18] Mascagni, M., and Hwang, C.-O., 2003, "ϵ-Shell error analysis for "Walk On Spheres" algorithms," *Mathematics and Computers in Simulation,* **63**(2), June, pp. 93–104.

[19] Hwang, C.-O., Hong, S., and Kim, J., 2015, "Off-centered "Walk-on-Spheres" (WOS) algorithm," *Journal of Computational Physics,* **303**, Dec., pp. 331–335.

[20] Ibarrart, L., Caliot, C., El Hafi, M., Fournier, R., Blanco, S., Dutour, S., Dauchet, J., Tregan, J.-M., Eymet, V., and Forest, V., 2018, "COMBINED CONDUCTIVE-CONVECTIVE-RADIATIVE HEAT TRANSFER IN COMPLEX GEOMETRY USING THE MONTE CARLO METHOD : APPLICATION TO SOLAR RECEIVERS," In International Heat Transfer Conference 16, Begellhouse, pp. 8135–8142.

[21] Caliot, C., Blanco, S., Coustet, C., Hafi, M. E., Forest, V., Fournier, R., and Piaud, B., 2018, "Combined conductive-radiative heat transfer analysis in complex geometry using the Monte Carlo method," In Eurotherm Seminar 110 Computational Thermal Radiation in Participating Media - VI.

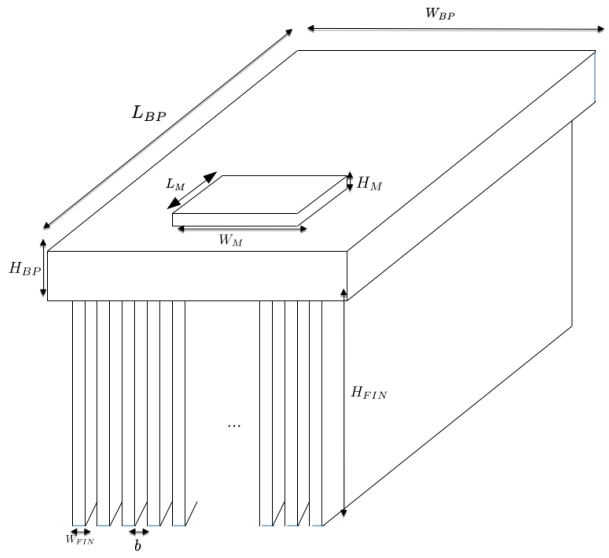
[22] Penazzi, L., Blanco, S., Caliot, C., Coustet, C., El Hafi, M., Fournier, R., Galtier, M., Ibarrart, L., and Roger, M., 2019, "Toward the use of Symbolic Monte Carlo for conduction-radiation coupling in complex geometries," In Proceeding of Proceedings of the 9th International Symposium on Radiative Transfer, RAD-19, Begellhouse, pp. 311–318.

[23] Sans, M., Farges, O., Schick, V., and Parent, G., 2022, "Solving transient coupled conductive and radiative transfers in porous media with a Monte Carlo Method: Characterization of thermal conductivity of foams using a numerical Flash Method," *International Journal of Thermal Sciences,* **179**, Sept., p. 107656.

[24] Penazzi, L., Blanco, S., Caliot, C., Coustet, C., El Hafi, M., Fournier, R., Gautrais, J., Golijanek-Jędrzejczyk, A., and Sans, M., 2024, "Path integrals formulations leading to propagator evaluation for coupled linear physics in large geometric models," *Computer Physics Communications Journal,* **294**, Jan.

[25] Pharr, M., Jakob, W., and Humphreys, G., 2016, *Physically Based Rendering: From Theory to Implementation (3rd ed.)*, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Nov.

[26] Villefranque, N., Fournier, R., Couvreux, F., Blanco, S., Cornet, C., Eymet, V., Forest, V., and Tregan, J. "A path-tracing monte carlo library for 3-d radiative transfer in highly resolved cloudy atmospheres," pp. 2449–2473.



[27] Penazzi, L., Blanco, S., Caliot, C., Coustet, C., El Hafi, M., Fournier, R., Gautrais, J., and Sans, M. "Transfer function estimation with SMC method for combined heat transfer: insensitivity to detail refinement of complex geometries," In Proceeding of Proceedings of CHT-21 ICHMT International Symposium on Advances in Computational Heat Transfer, Begellhouse, pp. 383–386.

[28] Ibarrart, L., Blanco, S., Caliot, C., Dauchet, J., Eibner, S., Hafi, M. E., Farges, O., Forest, V., Fournier, R., Konduru, R., Penazzi, L., Tregan, J.-M., and Vourc'h, T., 2022, Advection, diffusion and linear transport in a single path-sampling monte-carlo algorithm : getting insensitive to geometrical refinement Submitted to Computational Physics.

[29] Méso-Star, 2023, Stardis.

[30] Broughton, J., Smet, V., Tummala, R. R., and Joshi, Y. K., 2018, "Review of thermal packaging technologies for automotive power electronics for traction purposes," *Journal of Electronic Packaging,* **140**(4), Dec., p. 040801.

A DIMENSIONS OF THE FIRST CASE