A New Energy Conservation Scheme for the Numeric Study of the Heat Transfer in Profile Extrusion Calibration

Filipe Marques, Stéphane Clain, Gaspar Machado, Bruno Martins, Miguel João Nóbrega

A New Energy Conservation Scheme for the Numeric Study of the Heat Transfer in Profile Extrusion Calibration

Filipe Marques[a], Stéphane Clain[a,c], Gaspar J. Machado[a], Bruno Martins[b], João M. Nóbrega[b]

[a] *Centre of Mathematics, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal*
e-mail: a62055@alunos.uminho.pt,{clain,gjm}@math.uminho.pt
[b] *Institute for Polymers and Composites/I3N, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal*
e-mail: brunodiogomartins22@gmail.com,mnobrega@dep.uminho.pt
[c] *Institut de Mathématiques de Toulouse, Université Paul Sabatier, 31062 Toulouse, France*

Abstract

Thermoplastic extrusion processes use a dry calibration/cooling system, composed by one or several calibrators in series. One of the major difficulties in the modelling is to prescribe adequate values for the heat transfer coefficients between the polymer and the cooler or the air. We present an optimization procedure coupled with a finite volume method to evaluate such coefficients from experimental data. The tecnhique is based in a new conservative second-order finite volume scheme using a cell-to-vertex interpolation to solve the thermal problem involving a specific method to take the discontinuities (temperature, velocity, and thermal conductivity) into account. We have also performed a sensitive analysis of the different parameters of the problem to evaluate the variation of the thermal transfer coefficient with respect to parameters such as velocity or inflow temperature.

Keywords: Interface heat transfer coefficient, Extrusion, Calibration, Finite volume method, Cellvertex interpolation

1. Introduction

Extrusion plays an important role in the thermoplastics industry due to its wide range of applications. Any extrusion line aims to increase the rate of production and the quality of the produced profiles. However, these two goals are antagonist since the increase of production speed leads to a decreasing of product quality. The low thermal diffusivity of polymers is one of the major responsibles for that behavior since thermal gradients during the cooling stage develop internal stresses which diminish the performance on duty. Consequently, the calibration/cooling stage has a crucial role in the entire extrusion process since it controls the production rate. The cooling process should be faster to solidify the outer layers and, at the same time, to provide a uniform temperature as far as possible to minimize the internal thermal induced stresses.

In order to provide a correct evaluation of the heat transfers, one has to develop a numerical model that fully describes the cooler, and particular attention should be paid on

the boundary and interface conditions. The present study consider an aluminium calibrator in contact with the polymer (Polystyrene) where the main goal is the determination of an equivalent heat transfer coefficient h_{int} at the interface between the calibrator and the polymer in function of data collected by an experimental set-up. Such a coefficient highly depends on several conditions such as the contact resistance at the interface, the roughness of the material as it is shown in Figure 1 while temperature is assumed to be discontinuous at the interface to take into account the large gradient at a microscopic level. Due to the sensitivity of the cooling process with respect to the thermal transfer

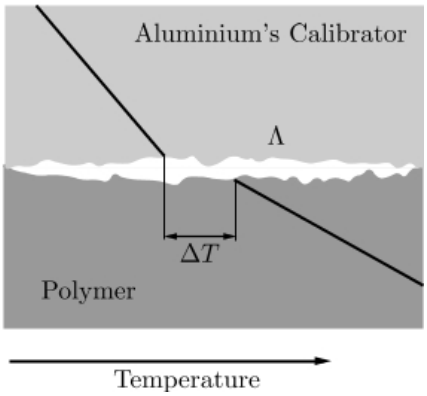

Figure 1: Interface between the calibrator and the polymer.

coefficient, an accurate model and numerical scheme are mandatory to provide approximations leading to a correct evaluation of the transfer coefficient [1] that ranges between 10 and 10000 W/(m^{2}K) according to its location in the calibration system [4].

The strategy relies on the numerical determination of the value of h_{int} that matches the values of temperature for numerical and experimental results. To solve the thermal problem we will consider a finite volume scheme, which guarantees the local conservation of the energy and enable to treat situtations with discontinuous temperature. Although the finite element method is still the most popular method for convection diffusion problem, second-order finite volume schemes become more attractive. Simplicity, versatility, mostly independent of the cell shape, easy to code, many practical problems in physics and engineering are now discretized with de FV method on unstructured meshes and recent progress enables to consider a wide range of applications for two or three dimensional geometries. In [2], a generic second-order finite volume scheme for the convection-diffusion-reaction equation based on the cell to vertex technology turns to be very efficient and robust and the method was experimented for non-homogeneous and anisotropic problems [3]. The specificity of this work consists in extending the method for the discontinuous situations, both for the solution and the coefficients.

2. Model and optimization

2.1. The thermal problem

A two-dimensional convection diffusion model of the calibrator together with a portion of polymer is considered. Figure 2 displays the geometry of the cooler where Ω_c and Ω_p stands for the sub-domains of the calibrator and the polymer (the subscripts c and p refer to the calibrator and polymer, respectively) while the boundary are defined as:

- calibrator: $\Gamma_c = \Gamma_{sup} \cup \Gamma_{lat} \cup \Lambda$,
- polymer: $\Gamma_p = \Gamma_{air} \cup \Gamma_{in} \cup \Gamma_{out} \cup \Lambda$.

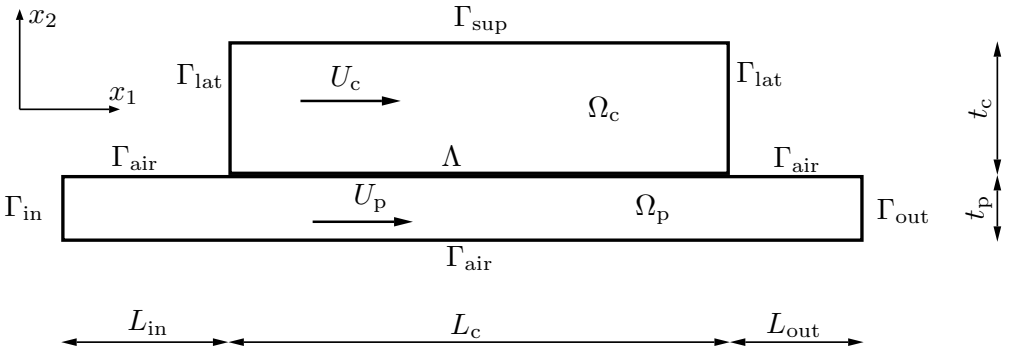

Figure 2: Two-dimensional model used to describe the heat transfer between the calibrator and the polymer.

The energy conservation equations for the two domains write

$$\nabla \cdot (C_\alpha U_\alpha T_\alpha - k_\alpha \nabla T_\alpha) = f_\alpha \quad \text{in } \Omega_\alpha \quad \alpha \in \{c, p\}, \tag{1}$$

where k is the thermal conductivity, $C = \rho c_p$ is the volumetric heat capacity with specific heat, c_p, and density, ρ, T is the temperature distribution, U is the velocity vector, and f is the source term which, in this case, is null in either domain. For the particular problem we are dealing with, we assume the referential associated to Ω_c to be fixe and $U_c = \binom{0}{0}$ whereas $U_p = \binom{u}{0}$ with u the polymer velocity.

The heat transfer between the calibrator and the polymer is defined as a thermal contact resistance which originates a discontinuity in temperature and the flux of energy across the interface is assumed to be linear with respect to the temperature differential. Energy conservation then writes

$$k_c \frac{\partial T_c}{\partial n_c} = -k_p \frac{\partial T_p}{\partial n_p} = h_{int}(T_p - T_c) \quad \text{on } \Lambda, \tag{2}$$

where n denotes the unit vector normal to the boundary outward to the domain.

We prescribe a polynomial temperature distribution given by eq. (3) for the upper surface of the calibrator which depends on the variable x_1.Real values a_1, a_2, a_3, a_4, a_5

and a_6 are the polynomial coefficients and will be set experimentaly using thermocouples embedded in the calibrator block

$$T_{\text{sup}} = a_1 x_1^5 + a_2 x_1^4 + a_3 x_1^3 + a_4 x_1^2 + a_5 x_1 + a_6 \quad \text{on } \Gamma_{\text{sup}} \tag{3}$$

Once the lateral calibrator sides, direct contact with the air, natural convection is assumed and we have

$$-k_{\text{c}} \frac{\partial T_{\text{c}}}{\partial n_{\text{c}}} = h_{\text{air}}(T_{\text{c}} - T_{\text{air}}) \quad \text{on } \Gamma_{\text{lat}}, \tag{4}$$

where T_{air} is the temperature of the air in the room assumed to be constant and h_{air} is the convection heat transfer coefficient of the air.

As regards to the polymer, the bottom and the upper surfaces is also in direct contact with the air and natural convection is assumed:

$$-k_{\text{p}} \frac{\partial T_{\text{p}}}{\partial n_{\text{p}}} = h_{\text{air}}(T_{\text{p}} - T_{\text{air}}) \quad \text{on } \Gamma_{\text{air}}. \tag{5}$$

The left boundary of the polymer, where it is feeded to the system, is defined by a constant prescribed temperature, as

$$T = T_{\text{in}} \quad \text{on } \Gamma_{\text{in}}. \tag{6}$$

Lastly, on the right side, the polymer flows out flux is prescribed by

$$k_{\text{p}} \frac{\partial T_{\text{p}}}{\partial n_{\text{p}}} = g_{\text{N}} \quad \text{on } \Gamma_{\text{out}}. \tag{7}$$

In our application we shall assume the adiabatic condition, $g_{\text{N}} = 0$.

3. Numerical scheme

To provide a generic expression for the discretization of the numerical scheme, we consider the governing equation given by (1) on an open bounded polygonal domain Ω of $\mathbb{R}^2$ with boundary Γ (we skip the index α for the sake of simplicity). We seek the temperature distribution $T \equiv T(x_1, x_2)$ as a solution of the steady-state convection-diffusion equation

$$\nabla \cdot (VT - k\nabla T) = f \quad \text{in } \Omega, \tag{8}$$

where VT is the convective term with $V = CU$, $k\nabla T$ is the diffusive term and f is the the source term. All the parameters in the equation were already described in section 2. Boundaries, Γ_{c} and Γ_{p} are partitioned into four subsets Γ_{D}, Γ_{N}, Γ_{R} and Λ where we prescribe different types of boundary conditions, namely:

- Dirichlet: $T = T_{\text{D}}$, on Γ_{D} [Eq. (3) and (6)];
- Neumann: $-k\nabla T \cdot n = g_{\text{N}}$, on Γ_{N} [Eq. (7)];
- Transference (Robin): $(VT - k\nabla T) \cdot n = h(T - T_{\text{R}})$, on Γ_{R} [Eq. (4) and (5)].
- Interface: $k_{\text{c}} \nabla T_{\text{c}} \cdot n_{\text{c}} = -k_{\text{p}} \nabla T_{\text{p}} \cdot n_{\text{p}} = h(T_{\text{p}} - T_{\text{c}})$, on Λ [Eq. (2)]

where $T_{\text{D}} \equiv T_{\text{D}}(x_1, x_2)$, $g_{\text{N}} \equiv g_{\text{N}}(x_1, x_2)$, and $T_{\text{R}} \equiv T_{\text{R}}(x_1, x_2)$ are given regular functions.

3.1. Mesh

We denote by $\mathcal{T}$ a mesh consisting of I non-overlapping convex polygonal cells c_i, $i = 1, \ldots, I$, and K vertices v_k, $k = 1, \ldots, K$. We highlight that $\mathcal{T}$ is constituted by two sub-meshes $\mathcal{T}_\mathrm{c}$ and $\mathcal{T}_\mathrm{p}$ for the subdomains Ω_c and Ω_p, respectively and are conformed with Λ such that if $\mathring{e} \cap \Lambda \neq \varnothing$ then $e \subset \Lambda$. We adopt the following conventions (see Figure 3) we detail hereafter:

- the mesh $\mathcal{T}_\mathrm{c}$ consists of I_c non-overlapping convex polygonal cells c_{i_c}, $i_\mathrm{c} = 1, \ldots, I_\mathrm{c}$, and K_c vertices v_{k_c}, $k_\mathrm{c} = 1, \ldots, K_\mathrm{c}$;
- the mesh $\mathcal{T}_\mathrm{p}$ consists of I_p non-overlapping convex polygonal cells c_{i_p}, $i_\mathrm{p} = I_\mathrm{c} + 1, \ldots, I_\mathrm{c} + I_\mathrm{p}$, and K_p vertices v_{k_p}, $k_\mathrm{p} = K_\mathrm{c} + 1, \ldots, K_\mathrm{c} + K_\mathrm{p}$;
- for any cell c_i, ∂c_i represents its boundary and $|c_i|$ its area; we denote by m_i the mass centre of c_i;
- two cells c_i and c_j share a common edge e_{ij} whose length is $|e_{ij}|$ and the midpoint is m_{ij}; n_{ij} is the unit normal vector to e_{ij} outward to c_i, *i.e.* $n_{ij} = -n_{ji}$; if an edge of c_i belongs to the boundary Γ, we replace the index j by D, N, R, or Λ if e_{ij} belongs to Γ_D, Γ_N, Γ_R, or Λ, respectively;
- for any cell c_i belonging to Ω_c or Ω_p we associate the index set $\nu(i_\mathrm{c}) \subset \{1, \ldots, I_\mathrm{c}\} \cup \{\mathrm{D}, \mathrm{N}, \mathrm{R}, \Lambda\}$ or $\nu(i_\mathrm{p}) \subset \{I_\mathrm{c} + 1, \ldots, I_\mathrm{c} + I_\mathrm{p}\} \cup \{\mathrm{D}, \mathrm{N}, \mathrm{R}, \Lambda\}$, respectively, such that $j \in \nu(i)$ if e_{ij} is a common edge of c_i and c_j or with the boundary Γ_j if $j = \{\mathrm{D}, \mathrm{N}, \mathrm{R}\}$ or with the interface Λ;
- for any vertex v_k belonging to Ω_c or Ω_p we associate the index set $\mu(k_\mathrm{c}) \subset \{1, \ldots, I_\mathrm{c}\}$ or $\mu(k_\mathrm{p}) \subset \{I_\mathrm{c} + 1, \ldots, I_\mathrm{c} + I_\mathrm{p}\}$, respectively, such that $i \in \mu(k)$ if v_k is a vertex belonging to the cell c_i.

Remark 3.1. If v_k is a vertex at the intersection of Γ_D and Γ_N, Γ_D and Γ_R, or Γ_D and Λ, we assume that v_k belongs to Γ_D and will be treated as a Dirichlet point.

Remark 3.2. On the interface Λ, each node considers two vertices, v_{k_c} and v_{k_p}, which share the same position but they are evaluated separately.

3.2. Generic finite volume scheme

To provide the finite volume scheme, equation (8) is integrated over cell c_i

$$\int_{c_i} \nabla \cdot (VT - k\nabla T) \mathrm{d}x = \int_{c_i} f \mathrm{d}x, \tag{9}$$

and applying the divergence theorem we get an integral over the surface

$$\int_{\partial c_i} (VT - k\nabla T) \cdot n \mathrm{d}s = \int_{c_i} f \mathrm{d}x, \tag{10}$$

and, then,

$$\sum_{j \in \nu(i)} \int_{e_{ij}} (V \cdot n_{ij} T - k\nabla T \cdot n_{ij}) \, \mathrm{d}s - \int_{c_i} f \mathrm{d}x = 0. \tag{11}$$

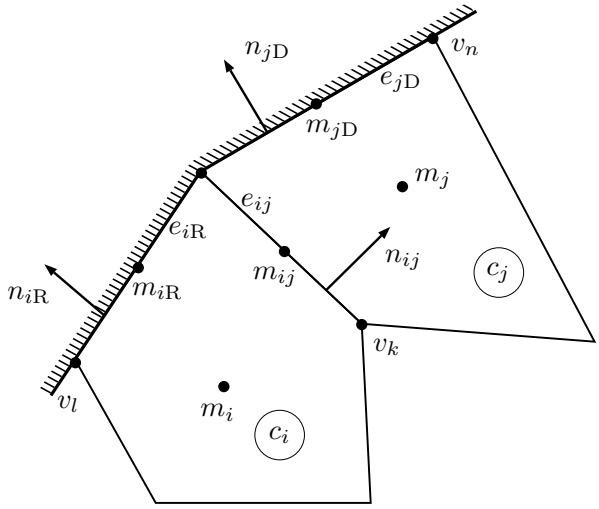

Figure 3: Mesh notation.

The numerical integration of equation (11), based on a quadrature method, introduces consistancy second-order errors and provides the following approximation

$$\sum_{j\in\nu(i)} |e_{ij}| \left(V(m_{ij})\cdot n_{ij}T(m_{ij}) - k(m_{ij})\nabla T(m_{ij})\cdot n_{ij}\right) - |c_i|f_i \approx 0. \tag{12}$$

Let T_i be an approximation of T at m_i. We gather all the approximations in two vectors $\mathbf{T}_{\mathrm{c}} = (T_i)_{i=1,\dots,I_{\mathrm{c}}}$ and $\mathbf{T}_{\mathrm{p}} = (T_i)_{i=I_{\mathrm{c}}+1,\dots,I_{\mathrm{c}}+I_{\mathrm{p}}}$ and also consider the global vector $\mathbf{T} = (\mathbf{T}_{\mathrm{c}}, \mathbf{T}_{\mathrm{p}})^T$. On the other hand, we define the residual as a function of the vector $\mathbf{T}$

$$\mathcal{G}_i = \sum_{j\in\nu(i)} |e_{ij}|\mathcal{F}_{ij} - |c_i|f_i, \tag{13}$$

where $\mathcal{F}_{ij}$ is an approximation of the convective and diffusive fluxes through the edge e_{ij} presented in equation (12), and $f_i = f(m_i)$ is a second-order approximation of the mean value of f over c_i.

3.3. Vertices interpolation

One of the keys of the method is the computation of diffusive flux across the edges. To this end, we use both approximations at the cell points and at the vertices to provide a good approximation of the temperature gradient. Since the unknowns are only located in the cells, we have to introduce a mechanism linking the cell values with the vertex values and provides an accurate evaluation of the vertices temperature based on the cells temperature. Let $\theta_{k_{\mathrm{c}}}$ and $\theta_{k_{\mathrm{p}}}$ be an approximation of T at vertex $v_{k_{\mathrm{c}}}$ and $v_{k_{\mathrm{p}}}$, respectively, and let us gather all the approximations in two vectors $\mathbf{\Theta}_{\mathrm{c}} = (\theta_{k_{\mathrm{c}}})_{k_{\mathrm{c}}=1,\dots,K_{\mathrm{c}}}$ and $\mathbf{\Theta}_{\mathrm{p}} = (\theta_{k_{\mathrm{p}}})_{k_{\mathrm{p}}=K_{\mathrm{c}}+1,\dots,K_{\mathrm{c}}+K_{\mathrm{p}}}$ while $\mathbf{\Theta} = (\mathbf{\Theta}_{\mathrm{c}}, \mathbf{\Theta}_{\mathrm{p}})^T$ stands for the global vector.

We introduce the generic linear mapping $\mathbf{T} \to \mathbf{\Theta}(\mathbf{T})$

$$\theta_k = \sum_{i\in\mu(k)} \beta_{ki} T_i \tag{14}$$

where β_{ki} are the interpolation coefficients for vertex k and we recall that $\mu(k)$ is the index set of the cells having the vertex v_k. For $v_k \in \Lambda$, one has to considertwo different temperatures, θ_{k_c} and θ_{k_p}, whether the temperature is associated to the calibrator or to the polymer. Two different stencils for the same location are introduced to estimate the temperature where we use the node v_{k_c} for the informations belonging to Ω_c, and the node v_{k_p} for the informations coming from Ω_p (see Figure 4).

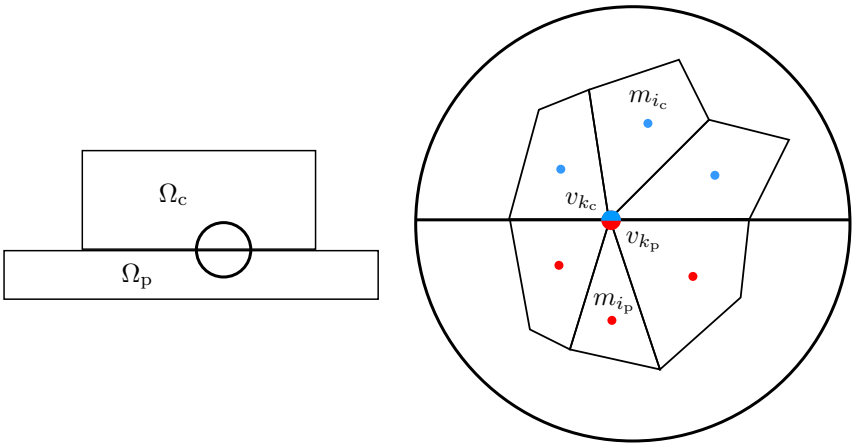

Figure 4: Schematic representation of the stencil used to interpolate the temperature at vertex on Λ, for point in the calibrator (blue) and in the polymer (red).

Consistency of the mapping with linear functions implies that coefficients β_{ki} have to fulfill the following restrictions

$$\sum_{i \in \mu(k)} \beta_{ki} = 1, \tag{15}$$

$$\sum_{i \in \mu(k)} \beta_{ki} m_i = v_k, \tag{16}$$

except the case when the vertex v_k belongs to a boundary where the temperature is prescribed (Dirichlet condition)

$$\theta_k = T_D(v_k) \quad \text{on } \Gamma_D. \tag{17}$$

For the Neumann condition, vertex interpolation can be preformed in two way; we introduce ghost cells where the temperature is evaluated using the Neumann condition; or we consider a stencil for the interpolation gathering cells around the the vertex. We refer to [2] for further details of the method.

3.4. Polynomial reconstructions

Based on the data stored in the two vectors $\mathbf{T}$ and $\mathbf{\Theta}$, we carry out the computation of several linear functions for the flux evaluations. For each cell c_i (resp.edge e_{ij}), we associate an index set S_i (resp. S_{ij}) of verticesand we consider linear functions on the cells and edges based on the information of vectors $\mathbf{T}$ and the $\mathbf{\Theta}$ associated to the appropriated stencils.

3.4.1. Polynomial reconstructions in the cells

For each cell c_i we define the affine function

$$T_i(x_1, x_2) = T_i + \mathcal{C}_{i,1}\,(x_1 - m_{i,1}) + \mathcal{C}_{i,2}\,(x_2 - m_{i,2})\,, \tag{18}$$

where $\mathcal{C}_{i,1}$ and $\mathcal{C}_{i,2}$ are the coefficients to be determined. For a given stencil S_i we consider the quadratic functional

$$\widetilde{E}_i(\mathcal{C}_{i,1}, \mathcal{C}_{i,2}) = \sum_{k \in S_i} \left(T_i(v_k) - \theta_k\right)^2 \tag{19}$$

and we denote by $\widetilde{\mathcal{C}}_{i,1}$ and $\widetilde{\mathcal{C}}_{i,2}$ the unique coefficients that minimize the quadratic functional (19) with $\widetilde{T}_i(x_1, x_2)$ the associated polynomial function which corresponds to the best approximation in the least squares sense of the data of the stencil.

3.4.2. Polynomial reconstructions at inner edges and dirichlet boundary

For a given inner edge e_{ij}, we must define two different polynomials, one with information of c_i, and another with c_j. The polynomial of degree 1 is given by

$$T_{ij}(x_1, x_2) = T_i + \mathcal{C}_{ij,1}(x_1 - m_{i,1}) + \mathcal{C}_{ij,2}(x_2 - m_{i,2}), \tag{20}$$

where $\mathcal{C}_{ij,1}$ and $\mathcal{C}_{ij,2}$ are the coefficients to be determined. We denote by $\widetilde{\mathcal{C}}_{ij,1}$ and $\widetilde{\mathcal{C}}_{ij,2}$ the unique coefficients such that the associated polynomial function $\widetilde{T}_{ij}(x_1, x_2)$ interpolates T_i, defined at m_i, and θ_k, defined at $v_k, k \in S_{ij}$. Notice that we also define the polynomial $\widetilde{T}_{ji}(x_1, x_2)$ using the reference cell point m_j and the associated value T_j. For an edge $e_{i\mathrm{D}} \subset \Gamma_D$, we proceed in the same way to provide polynomial $\widetilde{T}_{i\mathrm{D}}(x_1, x_2)$.

3.5. Second-order scheme

Based on local affine reconstruction for T based on vectors $\mathbf{T}$ and $\mathbf{\Theta}$ we compute the numerical approximations $\mathcal{F}_{ij}$ we detail in the following subsections.

3.5.1. Numerical fluxes

For an inner edge e_{ij}, there are two different polynomials $\widetilde{T}_{ij}$ and $\widetilde{T}_{ji}$ and define the resultant polynomial

$$\breve{T}_{ij} = \breve{T}_{ji} = \sigma_{ij}\widetilde{T}_{ij} + \sigma_{ji}\widetilde{T}_{ji}, \tag{21}$$

where $\sigma_{ij} = \frac{|c_i|}{|c_i|+|c_j|}$ and $\sigma_{ji} = \frac{|c_j|}{|c_i|+|c_j|}$. Notice that interface corresponding to discontinuous diffusion coefficients or discontinuous function requires a specific reconstruction. Therefore, five situations are identified:

- for an inner edge e_{ij}, the numerical flux at the midpoint m_{ij} writes

$$\mathcal{F}_{ij} = \left[V(m_{ij}) \cdot n_{ij}\right]^+ \widetilde{T}_i(m_{ij}) + \left[V(m_{ij}) \cdot n_{ij}\right]^- \widetilde{T}_j(m_{ij}) - k(m_{ij})\nabla\breve{T}_{ij}(m_{ij}) \cdot n_{ij};$$

- for a Dirichlet boundary edge $e_{i\mathrm{D}}$, the numerical flux at the midpoint $m_{i\mathrm{D}}$ writes

$$\mathcal{F}_{i\mathrm{D}} = \left[V(m_{i\mathrm{D}}) \cdot n_{i\mathrm{D}}\right]^+ \widetilde{T}_i(m_{i\mathrm{D}}) + \left[V(m_{i\mathrm{D}}) \cdot n_{i\mathrm{D}}\right]^- T_{\mathrm{D}}(m_{i\mathrm{D}}) - k(m_{i\mathrm{D}})\nabla\widetilde{T}_{i\mathrm{D}}(m_{i\mathrm{D}}) \cdot n_{i\mathrm{D}};$$

- for a Neumann boundary edge $e_{i\mathrm{N}}$, the numerical flux at the midpoint $m_{i\mathrm{N}}$ writes

$$\mathcal{F}_{i\mathrm{P}} = V(m_{i\mathrm{N}}) \cdot n_{i\mathrm{N}} \widetilde{T}_i(m_{i\mathrm{N}}) + g_{\mathrm{N}}(m_{i\mathrm{N}});$$

- for a transference (Robin) boundary edge $e_{i\mathrm{R}}$, the numerical flux at the midpoint $m_{i\mathrm{R}}$ writes

$$\mathcal{F}_{i\mathrm{R}} = h\left(\widetilde{T}_i(m_{i\mathrm{R}}) - T_{\mathrm{R}}(m_{i\mathrm{R}})\right);$$

- for an interface edge e we have to distinguish two cases regarding to the side where the flux come from. If we consider the calibrator side such $e = e_{i_{\mathrm{c}}\Lambda}$, the numerical flux at the midpoint $m_{i_{\mathrm{c}}\Lambda}$ writes

$$\mathcal{F}_{i_{\mathrm{c}}\Lambda} = h\left(\widetilde{T}_{i_{\mathrm{c}}}(m_{i_{\mathrm{c}}\Lambda}) - \widetilde{T}_{i_{\mathrm{p}}}(m_{i_{\mathrm{p}}\Lambda})\right);$$

  and, for the polymer side, a similar expression can be derived;

with the notations $[\Phi]^+ = \max(0, \Phi)$ and $[\Phi]^- = \min(0, \Phi)$, $\Phi \in \mathbb{R}$.

3.5.2. Residual scheme

Since $\mathcal{F}_{ij}$, and f_i linearly depend on vector $\mathbf{T}$, we define the affine operator $\mathbf{T} \to \mathcal{G}_i(\mathbf{T})$ for each cell c_i, $i = 1, \ldots, I$, as

$$\mathcal{G}_i(\mathbf{T}) = \sum_{j \in \nu(i)} |e_{ij}| \mathcal{F}_{ij}(\mathbf{T}) - |c_i| f_i. \tag{22}$$

Gathering all the components $\mathcal{G}_i(\mathbf{T})$ of the residual in vector $\mathcal{G}(\mathbf{T})$, we seek the solution vector $\mathbf{T}^\star$ such that $\mathcal{G}(\mathbf{T}^\star) = 0_I$. We obtain a matrix-free scheme and the affine problem is solved by applying a GMRES procedure as explained in [5].

4. Numerical tests

In this section, there will be presented several numerical tests from manufactured solutions in order to assess the robustness and accuracy of the implemented numerical scheme. Firstly, smooth solutions were tested for pure diffusion and convection-diffusion problems with low and high Péclet number, and different types of boundary conditions. Heat transfer problems with discontinuous diffusion coefficients is then tackled in the context of convection-diffusion problem. On the other hand, preconditioning the operator $\mathcal{G}$ is of paramount importance, since it provokes the decreasing of computational effort, and, in some cases, enabling the convergence of the method. Taking into account that we use a matrix-free scheme, the idea is to consider a simple approximated matrix that we will use to design the preconditioning matrix. In this sense, it was used the Patankar-like matrix which is discussed in [5].

To measure the accuracy of the obtained solution for a single mesh, two error estimators are introduced

$$E_1(\mathcal{T}) = \sum_{i=1}^{I} \frac{|c_i||T_i^\star - T_i|}{|\Omega|}, \quad E_\infty(\mathcal{T}) = \max_{i=1}^{I} |T_i^\star - T_i|.$$

The first norm gives the average evaluation of the error in all the domain, while the second norm indicates the highest local error. These error estimators can also be applied to the values interpolated on the vertices of the mesh. To study the convergence order, there are considered successive finer unstructured triangular Delaunay meshes. The convergence order between two different meshes $\mathcal{T}_1$ and $\mathcal{T}_2$ with I_1 and I_2 cells, respectively, is also analyzed recurring to the following expression

$$O(\mathcal{T}_1, \mathcal{T}_2) = 2\frac{|\log(|E(\mathcal{T}_1)/E(\mathcal{T}_2))}{|\log(I_1/I_2)|}. \tag{23}$$

4.1. Smooth solutions case

4.1.1. Pure diffusive problems with Dirichlet boundary conditions

We consider pure diffusive problem in the unit square with homogeneous Dirichlet boundary conditions where diffusion coefficient is set to $k = 1$. Figure 5 presents the normalized solution $T(x, y) = \sin(2\pi x)\sin(2\pi y)$ solution of the diffusion problem with the right-hand side $f(x, y) = 8\pi^2 \sin(2\pi x)\sin(2\pi y)$. Table 1 shows the errors and order of convergences. Second-order is achieved and the preconditioning matrix has no influence in the accuracy of the results but significantly affect the computational time.

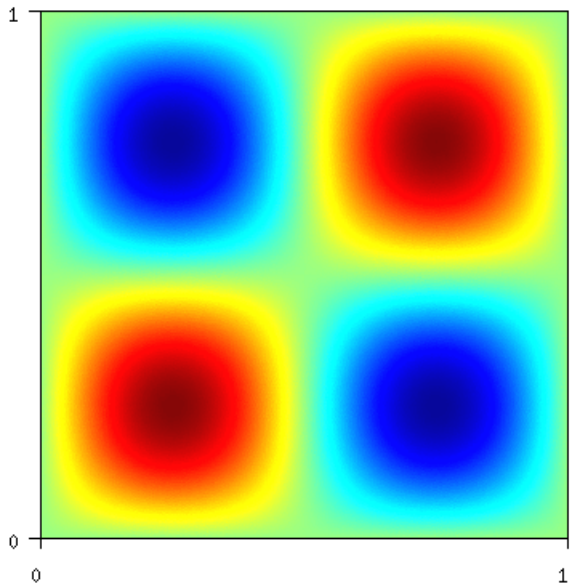

Figure 5: Representation of $T(x, y) = \sin(2\pi x)\sin(2\pi y)$

We now perform tests on a smooth solution with nonhomogeneous Dirichlet boundary condition. Function $T(x, y) = \cos(2\pi x)\cos(2\pi y)$ is the solution of the diffusion problem with the right-hand side $f(x, y) = 8\pi^2 \cos(2\pi x)\cos(2\pi y)$ displayed in Figure 6. Table 2 reports that we get a second-order of convergence in the non-homogeneous case.

Table 1: Errors and convergence rates for $T(x, y) = \sin(2\pi x)\sin(2\pi y)$ with Dirichlet condition for all the boundary

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1054	102	0.38	1.53e-03	NA	6.04e-03	NA	5.10e-03	NA	1.87e-02	NA
	4262	190	3.24	4.25e-04	1.84	2.10e-03	1.51	1.21e-03	2.06	4.43e-03	2.06
	16778	352	29.19	1.15e-04	1.91	5.36e-04	1.99	3.06e-04	2.01	1.25e-03	1.85
	67862	650	344.59	2.79e-05	2.02	1.44e-04	1.88	7.61e-05	1.99	3.28e-04	1.92
Pat	1054	47	0.18	1.53e-03	NA	6.04e-03	NA	5.10e-03	NA	1.87e-02	NA
	4262	92	1.54	4.25e-04	1.84	2.10e-03	1.51	1.21e-03	2.06	4.43e-03	2.06
	16778	181	14.47	1.15e-04	1.91	5.37e-04	1.99	3.06e-04	2.01	1.25e-03	1.84
	67862	356	139.73	2.79e-05	2.02	1.44e-04	1.88	7.61e-05	1.99	3.28e-04	1.92

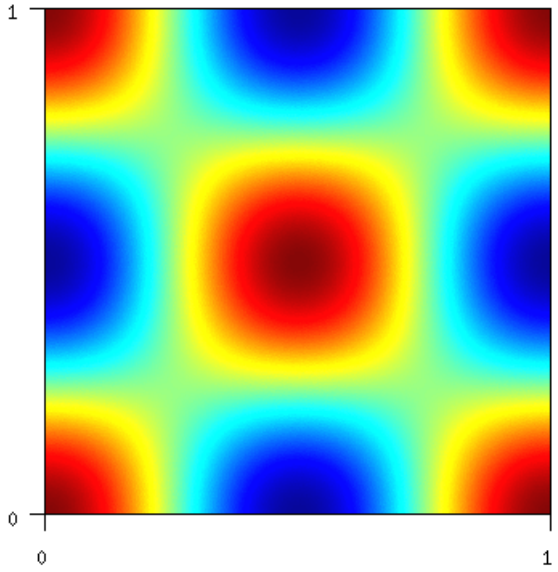

Figure 6: Representation of $T(x, y) = \cos(2\pi x)\cos(2\pi y)$

4.1.2. Pure Diffusive Problems with Neumann Boundary Conditions

In the present test, we prescribe Dirichlet boundary conditions for the upper and lower borders while Neumann boundary conditions are given on the lateral edges. Two different methods to evaluate the temparature at the node corresponding to a Neumann condition are assessed: with or without ghost cells; the first one including the information of the Neumann condition. Tables 4 and 3 present the results with and without ghost cells, respectively and observe that the introduction of ghost cells for the Neumann vertices interpolation has little influence in the accuracy of the approximations.

In the previous case, the results for a homogeneous Dirichlet condition were presented, while the following results express the case nonhomogeneous Dirichlet condition. For the solution presented in Fig. 6, Neumann boundary conditions were considered and the

Table 2: Errors and convergence rates for $T(x,y) = \cos(2\pi x)\cos(2\pi y)$ with Dirichlet condition for all the boundary

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1054	106	0.38	3.76e-03	NA	1.01e-02	NA	6.52e-03	NA	1.98e-02	NA
	4262	196	3.32	9.89e-04	1.91	2.71e-03	1.89	1.81e-03	1.83	6.91e-03	1.51
	16778	308	24.57	2.62e-04	1.94	8.18e-04	1.75	4.87e-04	1.92	1.85e-03	1.92
	67862	601	270.22	5.98e-05	2.12	2.01e-04	2.01	1.19e-04	2.02	4.86e-04	1.91
Pat	1054	59	0.21	3.76e-03	NA	1.01e-02	NA	6.52e-03	NA	1.98e-02	NA
	4262	111	1.80	9.89e-04	1.91	2.71e-03	1.89	1.81e-03	1.83	6.91e-03	1.51
	16778	218	16.70	2.62e-04	1.94	8.18e-04	1.75	4.87e-04	1.92	1.85e-03	1.92
	67862	394	153.50	5.98e-05	2.12	2.02e-04	2.00	1.19e-04	2.02	4.86e-04	1.91

Table 3: Errors and convergence rates for $T(x,y) = \sin(2\pi x)\sin(2\pi y)$ with Neumann boundary conditions for the lateral boundary (without ghost cells)

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1054	161	0.61	1.67e-03	NA	6.45e-03	NA	5.16e-03	NA	1.90e-02	NA
	4260	298	5.86	4.16e-04	1.99	2.08e-03	1.62	1.24e-03	2.04	4.45e-03	2.08
	16806	555	61.39	1.20e-04	1.81	5.21e-04	2.02	3.01e-04	2.06	1.24e-03	1.86
	67814	834	464.16	2.82e-05	2.08	1.49e-04	1.80	7.60e-05	1.98	3.37e-04	1.87
Pat	1054	87	0.33	1.67e-03	NA	6.45e-03	NA	5.16e-03	NA	1.90e-02	NA
	4260	171	3.16	4.16e-04	1.99	2.08e-03	1.62	1.24e-03	2.04	4.45e-03	2.08
	16806	329	28.53	1.20e-04	1.81	5.21e-04	2.02	3.01e-04	2.06	1.24e-03	1.86
	67814	654	372.37	2.82e-05	2.08	1.49e-04	1.79	7.60e-05	1.98	3.37e-04	1.87

Table 4: Errors and convergence rates for $T(x,y) = \sin(2\pi x)\sin(2\pi y)$ with Neumann boundary conditions for the lateral boundary (with ghost cells)

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1054	160	0.67	1.62e-03	NA	6.38e-03	NA	5.13e-03	NA	1.91e-02	NA
	4260	299	5.77	4.14e-04	1.95	2.08e-03	1.61	1.23e-03	2.04	4.45e-03	2.08
	16806	522	58.03	1.16e-04	1.85	5.16e-04	2.03	3.04e-04	2.04	1.25e-03	1.85
	67814	870	488.05	2.78e-05	2.05	1.48e-04	1.79	7.62e-05	1.98	3.38e-04	1.87
Pat	1054	87	0.40	1.62e-03	NA	6.38e-03	NA	5.13e-03	NA	1.91e-02	NA
	4260	171	3.82	4.14e-04	1.95	2.08e-03	1.61	1.23e-03	2.04	4.45e-03	2.08
	16806	329	38.06	1.16e-04	1.85	5.16e-04	2.03	3.04e-04	2.04	1.25e-03	1.85
	67814	653	390.36	2.78e-05	2.05	1.49e-04	1.78	7.62e-05	1.98	3.38e-04	1.87

errors and convergence rates are displayed in Tables 5 and 6 without and with the using of ghost cells for vertex interpolation, respectively. These results show that using ghost cells we are able to eliminate a local higher error for the vertices which affect the convergence of E_∞. It should be referred that the using preconditioning matrix implies

an higher computational effort unlike the previous examples.

Table 5: Errors and convergence rates for $T(x,y) = \cos(2\pi x)\cos(2\pi y)$ with Neumann boundary conditions for the lateral boundary (without ghost cells)

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1054	142	0.53	2.43e-03	NA	1.00e-02	NA	5.77e-03	NA	2.11e-02	NA
	4260	254	4.54	6.67e-04	1.85	2.44e-03	2.03	1.55e-03	1.88	6.73e-03	1.64
	16806	274	22.20	1.69e-04	2.00	6.74e-04	1.87	4.02e-04	1.97	3.37e-03	1.01
	67814	388	181.55	4.13e-05	2.02	1.78e-04	1.91	1.00e-04	1.99	9.72e-04	1.78
Pat	1054	87	0.43	2.43e-03	NA	1.00e-02	NA	5.77e-03	NA	2.11e-02	NA
	4260	162	3.83	6.67e-04	1.85	2.44e-03	2.02	1.55e-03	1.88	6.73e-03	1.64
	16806	302	32.57	1.69e-04	2.00	6.74e-04	1.87	4.02e-04	1.97	3.37e-03	1.01
	67814	564	338.78	4.15e-05	2.01	1.79e-04	1.91	1.00e-04	1.99	9.67e-04	1.79

Table 6: Errors and convergence rates for $T(x,y) = \cos(2\pi x)\cos(2\pi y)$ with Neumann boundary conditions for the lateral boundary (with ghost cells)

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1054	142	0.57	2.37e-03	NA	1.00e-02	NA	5.81e-03	NA	2.41e-02	NA
	4260	253	4.73	6.66e-04	1.82	2.44e-03	2.03	1.56e-03	1.89	6.73e-03	1.83
	16806	265	21.59	1.69e-04	2.00	6.74e-04	1.87	4.02e-04	1.97	1.84e-03	1.89
	67814	386	152.54	4.13e-05	2.02	1.78e-04	1.91	1.00e-04	1.99	5.00e-04	1.87
Pat	1054	87	0.46	2.37e-03	NA	1.00e-02	NA	5.81e-03	NA	2.41e-02	NA
	4260	162	3.84	6.66e-04	1.82	2.44e-03	2.03	1.56e-03	1.89	6.73e-03	1.83
	16806	302	32.63	1.69e-04	2.00	6.74e-04	1.87	4.02e-04	1.97	1.84e-03	1.89
	67814	564	333.25	4.14e-05	2.01	1.79e-04	1.91	1.00e-04	1.99	5.00e-04	1.87

4.1.3. Convection-Diffusion Problem

We now deal with the full convection diffusion process and assess the capability of the scheme to compute an accurate solution for low and large Péclet numbers. Assuming constant diffusion coefficient $k = 1$ and velocity $V = (u, v)$, we consider a solution of the form $T(x,y) = C\alpha(x)\beta(y)$, $C \in \mathbb{R}$ and

$$\alpha(x) = \frac{1}{u}\left(x - \frac{e^{ux}-1}{e^u - 1}\right), \quad \beta(y) = \frac{1}{v}\left(y - \frac{e^{vy}-1}{e^v - 1}\right)$$

with homogeneous Dirichlet boundary condition while the source term writes $f(x,y) = C\left(\alpha(x) + \beta(y)\right)$.

We take $V = (3,3)$ for the low Péclet number case, and $V = (10,10)$ for the large Péclet number case and C is calculated in such a way that the solution is normalized, that is $C = 80$ for low Péclet (Figure 7), and $C = 223$ for large Péclet (Figure 8).

Tables 7 and 8 give the errors and convergence rates for the low and large Péclet, respectively. Second-order of convergence for different Péclet numbers is obtained. No

oscilations are reported while the maximum principle is preserved, *i.e.* the solution ranges between 0 and 1, which highlight the robustness of the method.

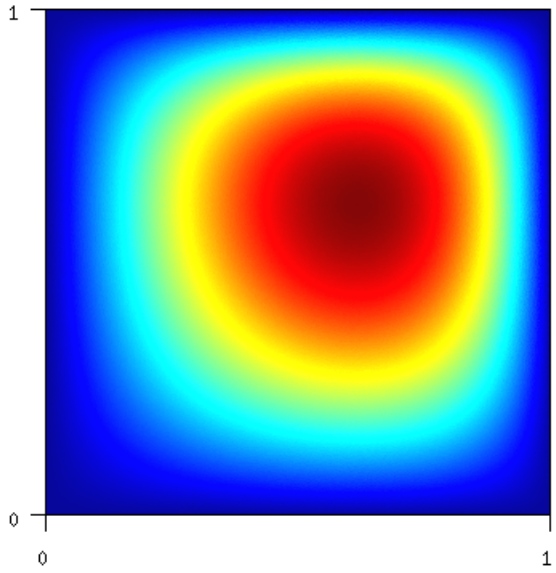

Figure 7: Representation of low Péclet function

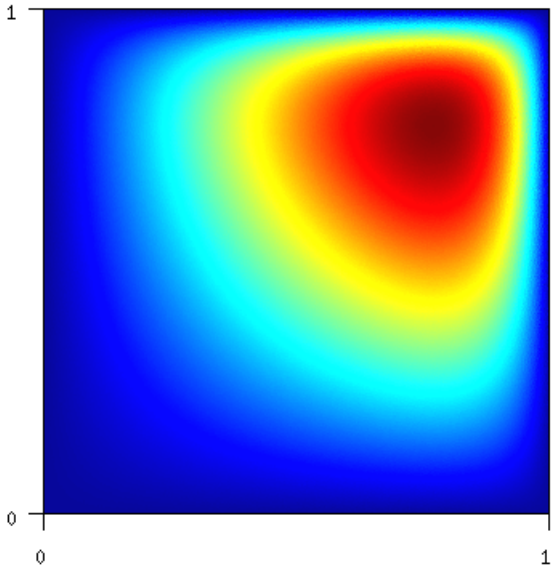

Figure 8: Representation of large Péclet function

Table 7: Errors and convergence rates for low Péclet number case

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1054	131	0.59	5.30e-04	NA	3.82e-03	NA	1.93e-03	NA	6.62e-03	NA
	4262	261	5.23	1.43e-04	1.88	1.14e-03	1.73	4.65e-04	2.03	1.53e-03	2.10
	16778	485	48.70	3.69e-05	1.98	3.19e-04	1.86	1.18e-04	2.00	4.18e-04	1.89
	67862	1116	701.17	9.27e-06	1.98	8.83e-05	1.84	2.90e-05	2.01	1.10e-04	1.91
Pat	1054	62	0.28	5.30e-04	NA	3.82e-03	NA	1.93e-03	NA	6.62e-03	NA
	4262	127	2.96	1.43e-04	1.88	1.14e-03	1.73	4.65e-04	2.03	1.53e-03	2.10
	16778	255	28.41	3.69e-05	1.98	3.19e-04	1.86	1.18e-04	2.00	4.18e-04	1.89
	67862	509	286.67	9.27e-06	1.98	8.83e-05	1.84	2.90e-05	2.01	1.10e-04	1.91

Table 8: Errors and convergence rates for large Péclet number case

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1054	121	0.55	3.65e-04	NA	4.79e-03	NA	1.14e-03	NA	9.61e-03	NA
	4262	237	4.87	1.03e-04	1.81	1.61e-03	1.56	2.98e-04	1.92	2.53e-03	1.91
	16778	454	45.71	2.91e-05	1.85	4.91e-04	1.73	8.11e-05	1.90	6.86e-04	1.90
	67862	1024	640.49	7.64e-06	1.91	1.36e-04	1.83	2.06e-05	1.96	1.99e-04	1.77
Pat	1054	59	0.26	3.65e-04	NA	4.79e-03	NA	1.14e-03	NA	9.61e-03	NA
	4262	115	2.28	1.03e-04	1.81	1.61e-03	1.56	2.98e-04	1.92	2.53e-03	1.91
	16778	229	20.00	2.91e-05	1.85	4.91e-04	1.73	8.11e-05	1.90	6.86e-04	1.90
	67862	448	227.00	7.64e-06	1.91	1.36e-04	1.83	2.06e-05	1.96	1.99e-04	1.77

4.2. Study of Problems with Interface Discontinuity

The section is dedicated to the critical part of the problem. Discontinuities of material property or discontinuity of the solution require a specific treatment to preserve the optimal order. Such a point in crucial since a degradation of the flux evaluation across the interface Γ leads to a rough evaluation of the heat transfer. We present several numerical tests to prove that the new technique correctly handle such discontinuities and provide accurate approximations.

4.2.1. Pure Diffusive Case

We consider two domains $\Omega_1 =]0,1[\times]0,\frac{1}{2}[$ and $\Omega_2 =]0,1[\times]\frac{1}{2},1[$ which share a common interface $\Lambda =]0,1[\times\{\frac{1}{2}\}$ where a linear heat transfer condition is prescribed

$$k_1 \nabla T_1 \cdot n_\Lambda = k_2 \nabla T_2 \cdot n_\Lambda = h(T_2 - T_1) \tag{24}$$

with k_1 and k_2 the constant diffusion coefficients. We first tackle the case where the convective contribution is null for both domains. Adiabatic condition $k\nabla T \cdot n = 0$ is assumed on the lateral side while we set constant Dirichlet conditions $T = \mathrm{T}_1$ on $y = 0$ and $T = \mathrm{T}_2$ on $y = 1$. The exact solution is constituted of two functions,

$$T_1(x,y) = \frac{A}{k_1}\left(\frac{1}{\pi}\right)^2 \sin(\pi y) + \frac{a}{k_1} y + \mathrm{T}_1, \quad T_2(x,y) = \frac{A}{k_2}\left(\frac{1}{\pi}\right)^2 \sin(\pi y) + \frac{a}{k_2}(y-1) + \mathrm{T}_2,$$

with

$$a = \frac{h}{1 + \frac{h}{2k_1} + \frac{h}{2k_2}} \left(A \left(\frac{1}{\pi} \right)^2 \left(\frac{1}{k_2} - \frac{1}{k_1} \right) + \mathrm{T}_2 - \mathrm{T}_1 \right)$$

with A a free parameter associated to the source term $f(x, y) = A \sin(\pi y)$.

The simulations of this problem were carried out with $A = 10$, $k_1 = 1$, $k_2 = 0.1$, $\mathrm{T}_1 = 1$, $\mathrm{T}_2 = 10$ and $h = 1$ while the solution is displayed in Figure 9. The errors and convergence rates are presented in Tables 9 and 10 for reconstruction of Neumann vertices without or with recurring to ghost cells, respectively. It should be highlighted the paramount importance of using a preconditioning matrix in this case. Beyond it decreases the computational time, it enables the convergence of some solutions. The main difference of using ghost cells for vertex reconstruction is recovering second-order of convergence for the error E_∞ at the vertices. Thus, the scheme showed an effective second order of convergence for the presented problem.

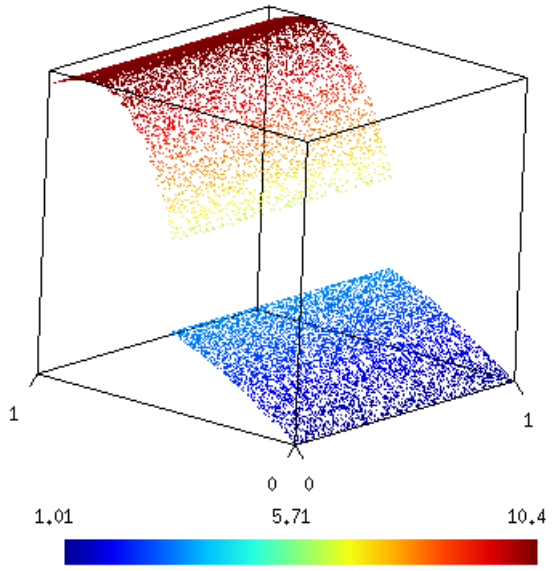

Figure 9: Solution for pure diffusive problem with discontinuous coefficients

4.2.2. Convection-Diffusion

We considered a problem where both velocity and diffusion coefficient are discontinuous across the interface Λ. where the continuity of normal velocity is required for sake of conservativeness. In the present test, null velocity is set in domain Ω_2, and a constant velocity $V_1 = (u, 0)$ is used in domain Ω_1. The exact solution is given by

$$T_1(x, y) = A_1 e^{\frac{u}{k_1} x} y + a_1 y + \mathrm{T}_1, \quad T_2(x, y) = e^{\frac{u}{k_1} x} y \left[A_2 (y - 1) + B_2 (y - 1)^2 \right] + a_2 (y - 1) + \mathrm{T}_2.$$

Conservation of the flux at the interface yields

$$k_1 A_1 = k_2 A_2 - k_2 B_2, \quad k_1 a_1 = k_2 a_2,$$

Table 9: Errors and convergence rates for pure diffusive problem with discontinuous coefficients (without ghost cells)

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1014	318	4.68	6.83e-03	NA	2.80e-02	NA	1.23e-02	NA	7.01e-02	NA
	4126	572	34.92	1.24e-03	2.43	6.58e-03	2.07	2.72e-03	2.15	2.14e-02	1.69
	17020	866	226.49	3.30e-04	1.87	1.82e-03	1.81	6.88e-04	1.94	1.14e-02	0.88
	67620	4002	4617.96	1.62e+00	12.32	3.92e+00	11.13	1.62e+00	11.26	3.92e+00	8.47
Pat	1014	92	0.96	6.83e-03	NA	2.80e-02	NA	1.23e-02	NA	7.01e-02	NA
	4126	169	7.22	1.24e-03	2.43	6.58e-03	2.07	2.72e-03	2.15	2.14e-02	1.69
	17020	306	57.57	3.29e-04	1.88	1.81e-03	1.83	6.87e-04	1.94	1.14e-02	0.89
	67620	526	526.70	8.55e-05	1.95	5.08e-04	1.84	1.79e-04	1.95	2.85e-03	2.01

Table 10: Errors and convergence rates for pure diffusive problem with discontinuous coefficients (with ghost cells)

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1014	319	4.07	6.99e-03	NA	2.81e-02	NA	1.20e-02	NA	5.38e-02	NA
	4126	569	28.08	1.28e-03	2.42	6.60e-03	2.06	2.69e-03	2.13	1.29e-02	2.03
	17020	878	196.36	3.27e-04	1.93	1.81e-03	1.83	6.76e-04	1.95	3.42e-03	1.87
	67620	4002	4512.20	1.66e+00	12.37	4.03e+00	11.18	1.66e+00	11.31	4.03e+00	10.25
Pat	1014	93	1.06	6.99e-03	NA	2.81e-02	NA	1.20e-02	NA	5.38e-02	NA
	4126	169	7.55	1.28e-03	2.42	6.60e-03	2.06	2.69e-03	2.13	1.29e-02	2.03
	17020	306	58.91	3.27e-04	1.93	1.80e-03	1.84	6.77e-04	1.95	3.41e-03	1.88
	67620	526	474.35	8.58e-05	1.94	5.08e-04	1.83	1.78e-04	1.93	1.02e-03	1.76

while the linear heat transfer condition (24) on Λ provides

$$4k_1A_1 = h\left(B_2 - 2\left(A_1 + A_2\right)\right), \quad 2k_1a_1 = h\left(2\left(\mathrm{T}_2 - \mathrm{T}_1\right) - \left(a_1 + a_2\right)\right).$$

After some algebraic manipulation, we get

$$A_2 = -A_1\left(2 + \frac{k_1}{k_2} + 4\frac{k_1}{h}\right), \quad B_2 = \frac{k_2A_2 - k_1A_1}{k_2}, \quad a_1 = \frac{2(\mathrm{T}_2 - \mathrm{T}_1)}{1 + \frac{k_1}{k_2} + 2\frac{k_1}{h}}, \quad a_2 = \frac{k_1}{k_2}a_1,$$

where A_1 is a free parameter which can be defined arbitrarily. From this exact solution, we derive the source term $f_1(x,y) = 0$ in Ω_1 and $f_2(x,y) = -k_2e^{\frac{u}{k_1}x}\left[2B_2 + \left(\frac{u}{k_1}\right)^2(y-1)(A_2 + B_2(y-1))\right]$ in Ω_2. For sake of simplicity, it is used the exact solution as a Dirichlet condition on the boundary.

On the following simulations, it was considered $A_1 = 1$, $k_1 = 0.18$, $k_2 = 23$, $u = 1$, $\mathrm{T}_1 = 1$, $\mathrm{T}_2 = 10$ and $h = 1$. With these parameters, we obtain the solution presented in Figure 10.

Table 11 shows that we obtain an effective second-order convergence. As in the previous case, no oscillations are reported even in the vicinity of the Γ which demonstrates the capability of the scheme to handle such discontinuities.

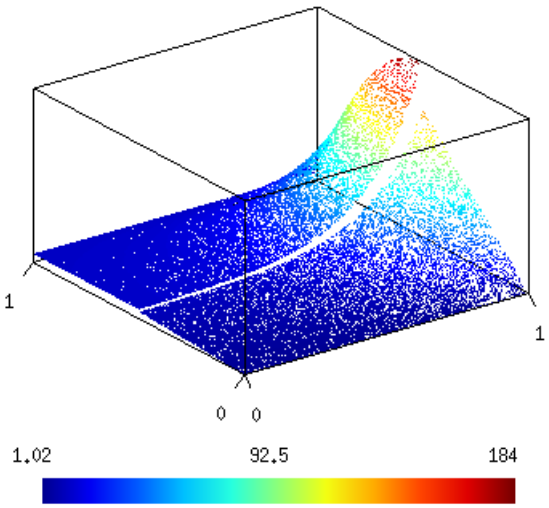

Figure 10: Solution for convection-diffusion problem with discontinuous coefficients

Table 11: Errors and convergence rates for convection-diffusion problem with discontinuous coefficients

	I	Iter	Time [s]	Cells E_1	O_1	E_∞	O_∞	Vertices E_1	O_1	E_∞	O_∞
Id	1014	760	8.38	1.38e-01	NA	1.08e+00	NA	8.66e-02	NA	1.46e+00	NA
	4126	1621	80.49	3.01e-02	2.17	3.28e-01	1.70	2.19e-02	1.96	1.69e+00	0.21
	17026	4002	1063.93	2.20e+00	6.06	1.17e+01	5.05	2.19e+00	6.50	1.17e+01	2.73
	67718	4001	4840.22	5.07e+00	1.21	4.20e+01	1.85	5.07e+00	1.21	4.19e+01	1.85
Pat	1014	73	1.00	1.38e-01	NA	1.08e+00	NA	8.65e-02	NA	1.46e+00	NA
	4126	140	7.87	3.10e-02	2.13	3.28e-01	1.70	2.13e-02	2.00	1.69e+00	0.21
	17026	264	64.59	7.37e-03	2.03	8.81e-02	1.85	4.91e-03	2.07	1.56e-01	3.37
	67718	503	612.25	1.88e-03	1.98	2.22e-02	1.99	1.26e-03	1.97	4.41e-02	1.83

5. Application to the Polymer/Calibrator Heat Transfer

We consider a polymer ribbon cooling by a unique calibrator. Figure 11 displays the quadrangular mesh geometry, and N_p, N_c and N_h denotes the number of elements on polymer and calibrator vertical direction and the number of elements on horizontal direction on calibrator, respectively. The sizes of the model are defined according to Figure 2 with $L_\mathrm{in} = 0.17$ m, $L_\mathrm{c} = 0.3$ m, $L_\mathrm{out} = 0.13$ m, $t_\mathrm{p} = 1.45 \times 10^{-3}$ m, and $t_\mathrm{c} = 0.012$ m.

Numerical simulations are carried out with $k_\mathrm{p} = 0.17$ W/(m K), $k_\mathrm{c} = 123$ W/(m K), $\rho_\mathrm{p} = 1040$ kg/m^3, $c_{p_\mathrm{p}} = 2050$ J/(kg K), $U_\mathrm{p} = (0.0175, 0)$ m/s, $U_\mathrm{c} = (0, 0)$, $h_\mathrm{air} = 50$ W/(m^{2}K), $T_\mathrm{air} = 18°$C and $T_\mathrm{in} = 229.5°$C. The temperature distribution on Γ_sup was approximated by a polynomial based on the temperature measured on seven points with thermocouples. The polynomial approximation is represented on Figure 12 along with

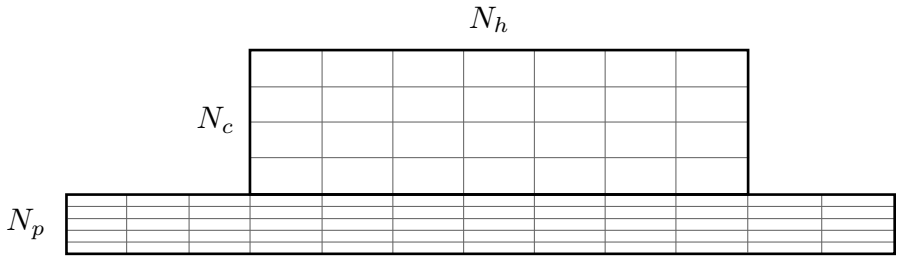

Figure 11: Schematic mesh for the polymer/calibrator model.

the experimental data, and it is given by

$$T_{\text{sup}} = 26399x_1^5 - 29434x_1^4 + 11403x_1^3 - 1848.9x_1^2 + 100.99x_1 + 39.581 \quad \text{on } \Gamma_{\text{sup}} \tag{25}$$

5.1. Mesh analysis

To obtain an accurate solution with a reduced computational time, one has to size the mesh we require to obtain a satisfactory solution. To this end, several tests were performed to check the sensibility of the approximation with respect to mesh. We assume that $h = 500$ W/(m^{2}K) and we experiment different mesh parameters. To visualize the impact of the mesh, we plot the temperature along the polymer ribbon since it corresponds to the critical location where the heat transfer acts.

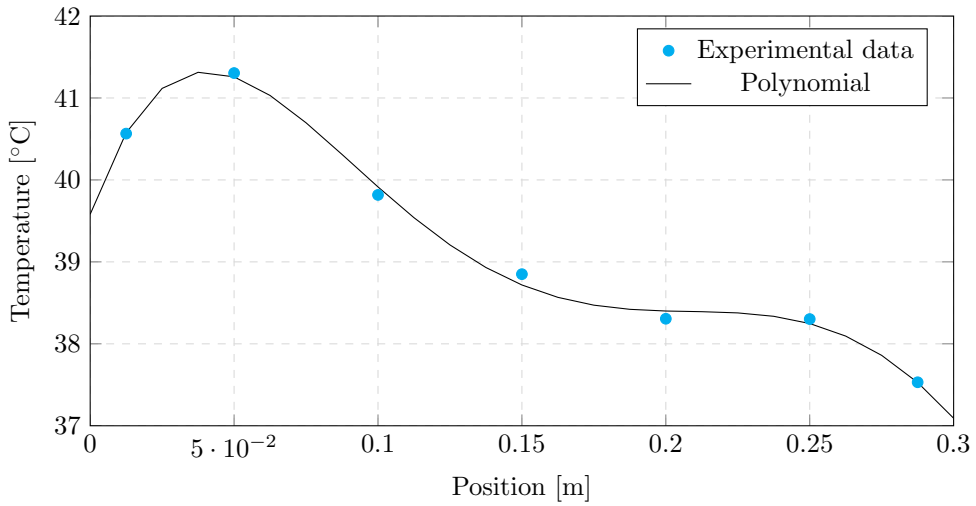

Figure 12: Polynomial approximation of temperature on the calibrator

Figure 13 shows the influence of the number of elements on the horizontal direction. All the curves present a small oscillation at the beginning and the ending of the contact interface but the default is mitigated with the increasing in the number of elements.

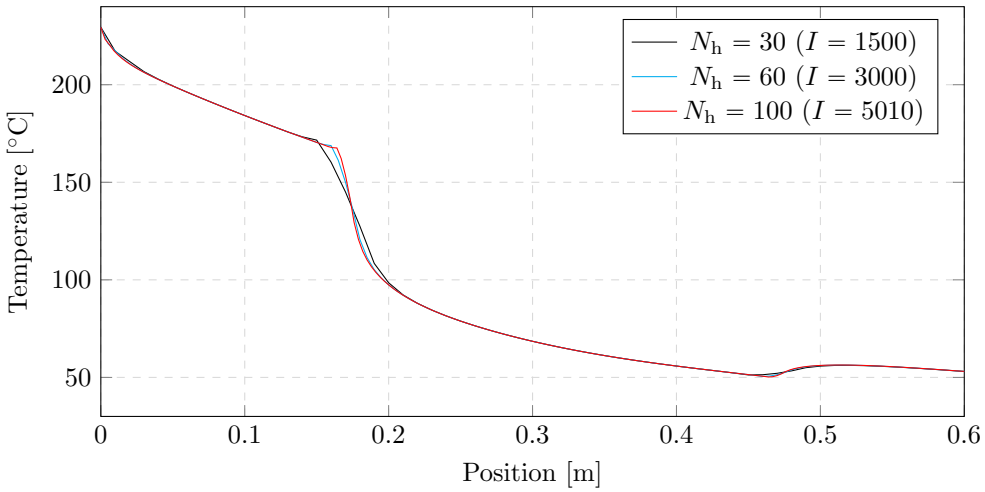

Figure 13: Comparison between meshes with different number of elements on the horizontal direction ($N_c = 30$ and $N_p = 10$)

We test the sensitivity with respect to the number of elements on the vertical direction in the polymer. Figure 14 shows that with the increasing of number of elements the solution converges to limit solution.

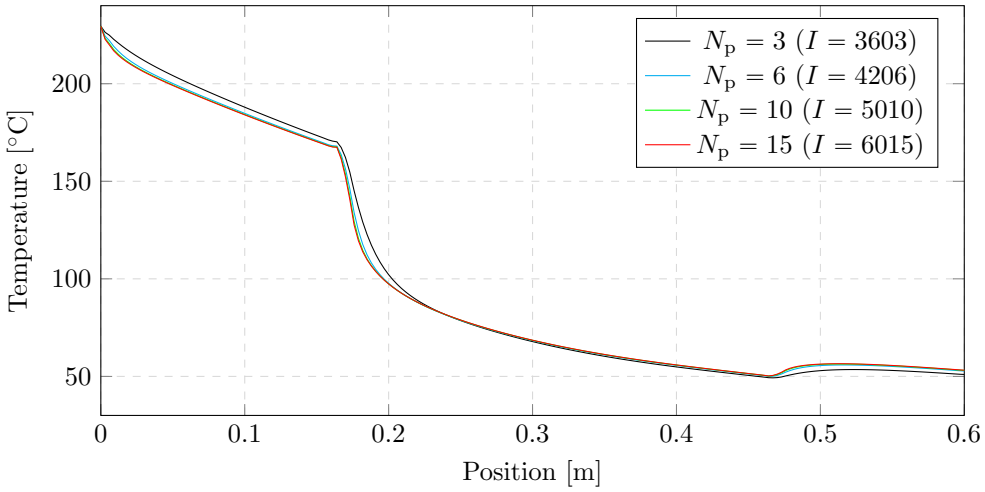

Figure 14: Comparison between meshes with different number of elements on polymer vertical direction ($N_c = 30$ and $N_h = 100$)

The influence of the number of elements on the vertical direction of the calibrator was also analyzed, and the results are shown on Figure 15. We conclude that this last parameter has a small influence on the temperature figure hence a coarse mesh can be

used for the calibrator.

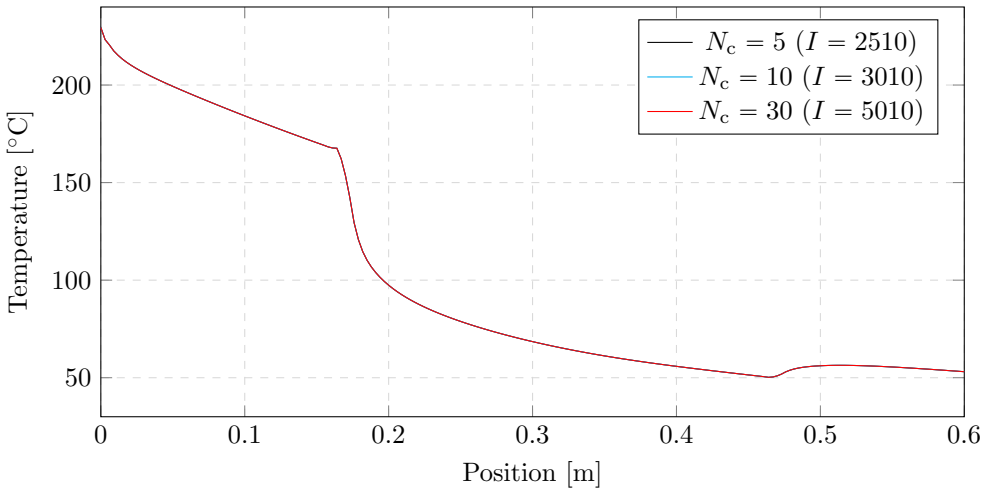

Figure 15: Comparison between meshes with different number of elements on calibrator vertical direction ($N_p = 10$ and $N_h = 100$)

5.2. Optimization

In most of the application, the h_{int} value is unknown and should be determined with experimental data. To do so, we consider the function $h \to T^h$ where we substitute (2) by

$$k_c \frac{\partial T_c^h}{\partial n_c} = -k_p \frac{\partial T_p^h}{\partial n_p} = h(T_p^h - T_c^h). \tag{26}$$

To compute an approximation of h_{int}, the least squares method is carried out where the optimal solution is found through the minimization of functional

$$F(h) = \sum_{\ell} \frac{1}{2} \left[T_p^h(q_\ell) - T_{p_\ell} \right]^2. \tag{27}$$

Here, $T_p^h(q_\ell)$ is the temperature of the polymer on the point ℓ located in q_ℓ obtained by the numerical simulation while T_{p_ℓ} is the temperature of the polymer on the point ℓ measured experimentally.

The nonlinear problem is solved with the Newton-Raphson iterative method where the condition $\frac{dF}{dh} = 0$ should be verified. Hence the updated value for heat transfer coefficient is calculated as it follows

$$h^{n+1} = h^n - \frac{F'(h^n)}{F''(h^n)}, \tag{28}$$

using the following approximations for the first and second derivatives

$$F'(h^n) \approx \frac{F(h^n + \varepsilon) - F(h^n - \varepsilon)}{2\varepsilon}, \quad F''(h^n) \approx \frac{F(h^n + \varepsilon) + F(h^n - \varepsilon) - 2F(h^n)}{\varepsilon^2}$$

with $\varepsilon > 0$ a parameter adjusted according to the simulation.

A determination of the heat transfer coefficient has been performed using the temperature measured on the polymer surface after the contact interface.

The Newton-Raphson method converges in few steps and gives $h = 390.5$ W/(m^{2}K) while temparature of at the upper side of the ribbon is presented on Figure 16.

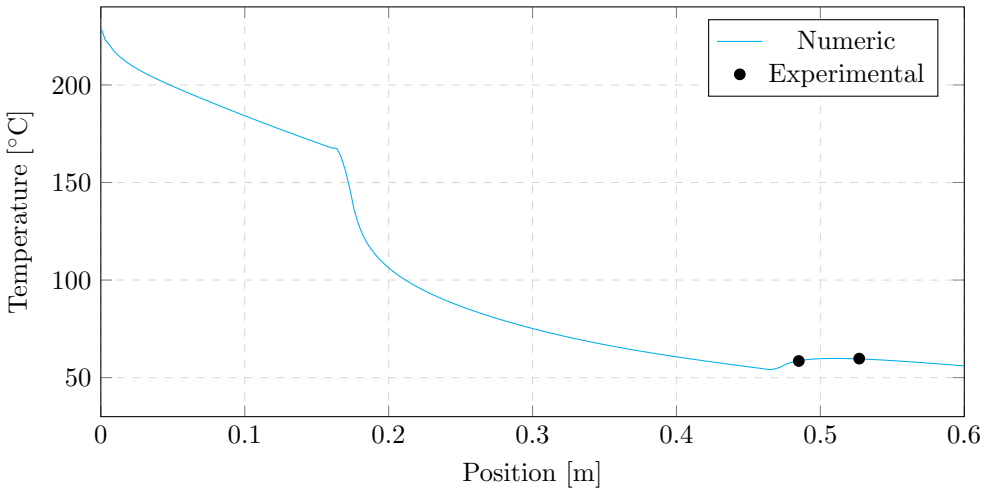

Figure 16: Comparison between numeric and experimental data ($h = 390.5$ W/(m^{2}K))

In table 12, there are presented the temperature measured experimentally which were the input data in the optimization process, and the temperature calculated from the numeric approximation in the same places. From its analysis, we can conclude that the approximation reveals a good fit since the difference between both temperatures is relatively small.

Table 12: Comparison between the temperature from experimental measurements and the numeric approximation.

Coordinate [m]	Experimental [°C]	Numeric [°C]	Error [%]
0.485	58.53	58.69	0.27
0.527	59.75	59.57	0.30

5.3. Parameters Sensibility

The presented results are directly related to the input data for the simulation which has a certain error associated. Bearing that in mind, there is of paramount importance to analyze the sensibility of results to the variation of some input data.

The analyzed parameters were divided into two groups, namely, the parameters that have a direct influence on the energy of the system, as the velocity of the polymer and input temperature, u and T_{in}, respectively. The remaining parameters are the convection heat transfer coefficient of the air and the room temperature, h_{air} and T_{air}, respectively.

5.3.1. Direct Parameters

The influence of small variations of velocity and input temperature on the results for heat transfer coefficient are displayed in Figures 17 and 18. We can see that the velocity of the polymer has a huge influence on the heat transfer coefficient but its dependence is nonlinear. Although, the input temperature seems to have a linear impact.

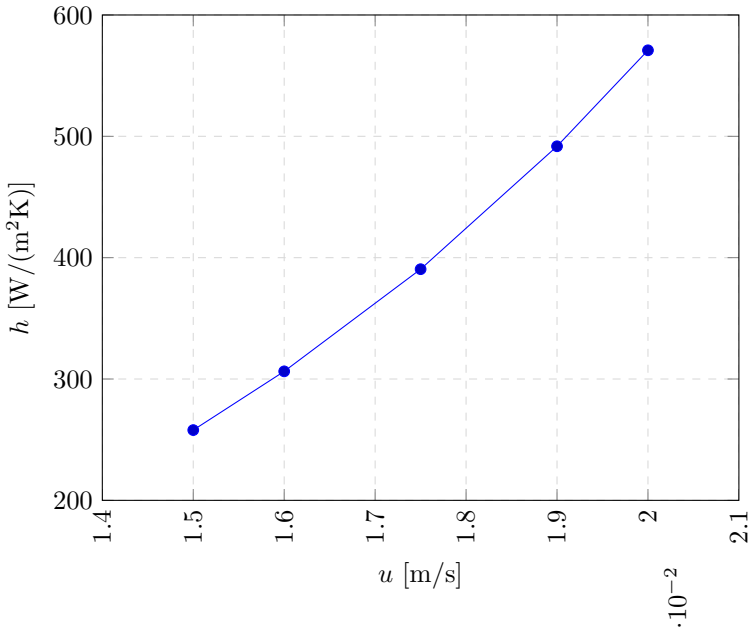

Figure 17: Sensibility of h to u $\left(\frac{\mathrm{d}h}{\mathrm{d}u} = 6.18 \times 10^4\right)$

5.3.2. Indirect Parameters

There were also tested the convection heat transfer coefficient of the air and the room temperature. In equations (4) and (5), we considered the same convection coefficient for the heat transfer with calibrator and with the polymer. However, for this analysis, there are considered two different coefficients, namely $h_{\mathrm{air,c}}$ and $h_{\mathrm{air,p}}$.

From the analysis of Figures 19 and 20, we can see that the convection coefficient for the calibrator has almost no influence in the results. In contrast to that, the convection coefficient for the polymer produces significant variations on the heat transfer coefficient.

The room temperature influence in the heat transfer coefficient is shown in Figure 21. This parameter present a linear relation and significant impact in the results.

6. Conclusion

We have presented a new second-order finite volume method for discontinuous solution and discontinuous material properties. The scheme is based in a cell-to-vertex reconstruction, being the vertex values computed via linear combinations of the closest

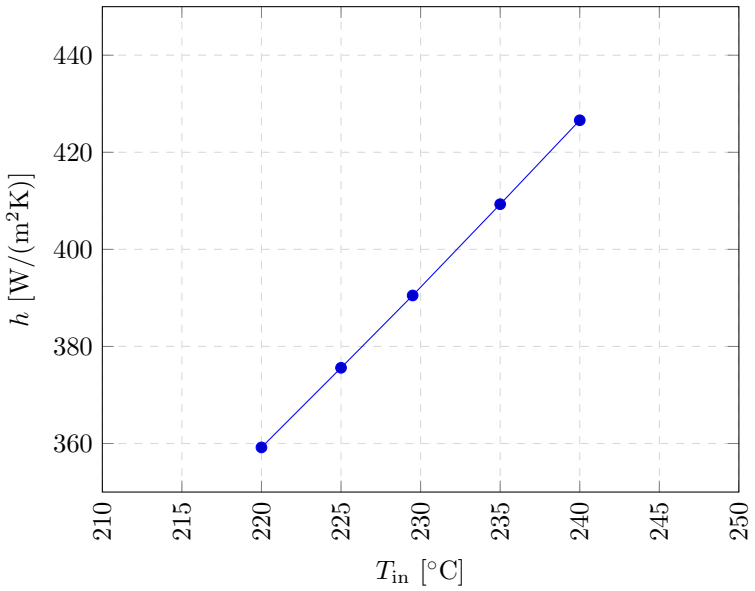

Figure 18: Sensibility of h to T_{in} $\left(\frac{\mathrm{d}h}{\mathrm{d}T_{\text{in}}} = 3.37\right)$

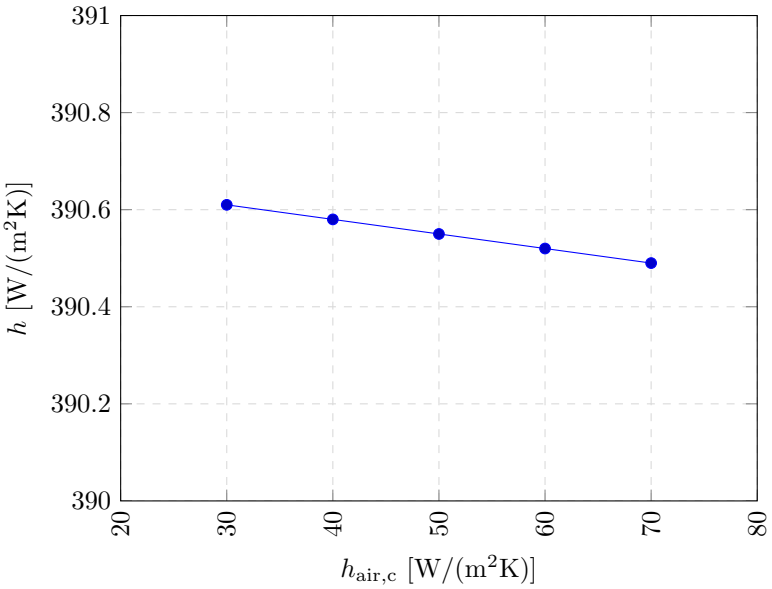

Figure 19: Sensibility of h to $h_{\text{air,c}}$ $\left(\frac{\mathrm{d}h}{\mathrm{d}h_{\text{air,c}}} = -3 \times 10^{-3}\right)$

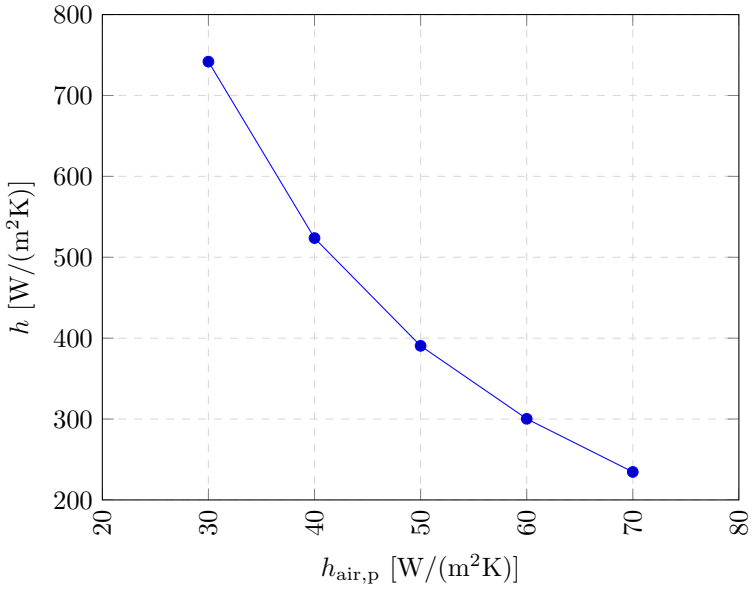

Figure 20: Sensibility of h to $h_{\mathrm{air,p}}$ $\left(\frac{\mathrm{d}h}{\mathrm{d}h_{\mathrm{air,p}}} = -11.2\right)$

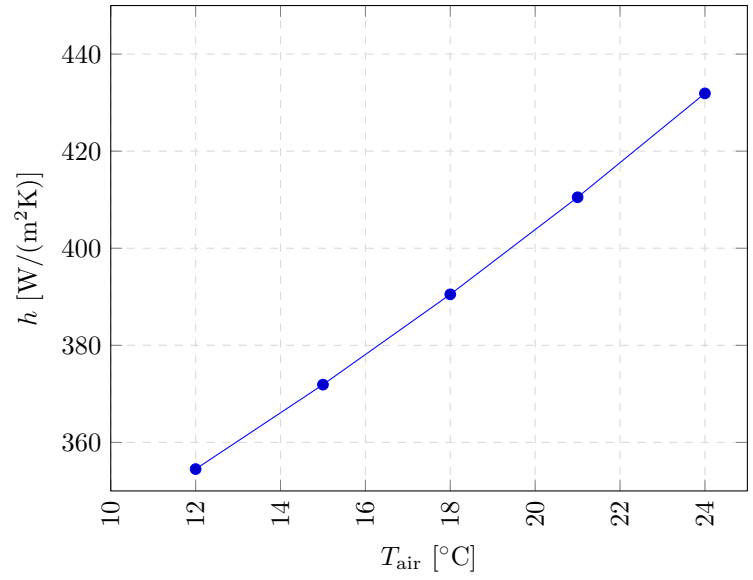

Figure 21: Sensibility of h to T_{air} $\left(\frac{\mathrm{d}h}{\mathrm{d}T_{\mathrm{air}}} = 6.43\right)$

cell values where the coefficients are determined by a functional minimization. The ob-

tained results demonstrates the robustness and effectiveness of the procedure. Based on the solver, an optimization algorithm has been developed to compute the heat transfer coefficient at the interface between the calibrator and the polymer. Influence of the input data such as the velocity, the energy transfer to air or the environmental temperature has been assessed very accurately to highlight the most impactful parameter.

Acknowledgements

This research was financed by FEDER Funds through Programa Operacional Factores de Competitividade — COMPETE and by Portuguese Funds through FCT — Fundação para a Ciência e a Tecnologia, within the Projects PEst-OE/MAT/UI0013/2014 and PTDC/MAT/121185/2010. The second author was also financed by project FCT-ANR/MAT-NAN/0122/2012.

References

[1] W. Uffrecht, B.Heinschke A. Gnther, V. Caspary, S. Odenbach, Measurement of heat transfer coefficients at up to 25,500g ? A sensor test at a rotating free disk with complex telemetric instrumentation, International Journal of Thermal Sciences, in press (2015).

[2] R. Costa, S. Clain, G. J. Machado, New cell-vertex reconstruction for finite volume scheme: Application to the convection-diffusion-reaction equation, Computers and Mathematics with Applications 68 (2014) 1229–1249.

[3] R. Costa, S. Clain e G.J. Machado, Finite Volume Scheme Based on Cell-Vertex Reconstructions for Anisotropic Diffusion Problems with Discontinuous Coefficients, 14th International Conference in Computational Science and Its Applications - ICCSA 2014, Guimares, Portugal, 30 de junho a 3 de julho de 2014, Lecture Notes in Computer Science, 8579 (2014) 87–102.

[4] O.S. Carneiro, J.M. Nbrega, A.R. Mota, C. Silva, Prototype and methodology for the characterization of the polymer-calibrator interface heat transfer coefficient, Polymer Testing 32 (6) (2013) 1154-1161.

[5] S. Clain, G. J. Machado, J. M. Nóbrega, R. M. S. Pereira, A sixth-order finite volume method for the convection-diffusion problem with discontinuous coefficients, Computer Methods in Applied Mechanics and Engineering 267 (2013) 43–64.

[6] S. Diot, R. Loubère, S. Clain, The MOOD method in the three-dimensional case: Very-High-Order Finite Volume Method for Hyperbolic Systems, International Journal for Numerical Methods in Fluids 73 (4) (2013) 362–392.

[7] A. Hidalgo, M. Dumbser, ADER Schemes for Nonlinear Systems of Stiff Advection-Diffusion-Reaction Equations, J. Sci. Comput. 48 (1-3) (2011) 173–189.

[8] L. Ivan, C.P.T. Groth, High-order solution-adaptative central essentially non-oscillatory (CENO) method for viscous flows, AIAA Paper 2011–367 (2011).

[9] C. Ollivier-Gooch, High-Order ENO schemes for unstructured Meshes based on least-squares reconstruction, AIAA Paper 97–0540 (1997).

[10] C. Ollivier-Gooch, M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation, Journal of Computational Physics Archive 181 (2) (2002) 729–752.