Vasculature optimization of actively-cooled materials

Gonçalo Ferreira Nunes Gouveia Valente
goncalo.valente@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

June 2016

Abstract

Associated with an eagerness to explore every possibility from humankind imagination came the field of engineering. In engineering applications where heat transfer processes are present arise vascular actively-cooled materials, which provide motivation for this thesis. The goal of this work is to implement the tools required to model and optimize conjugated heat transfer problems (heat disperses through the domain not only *via.* heat conduction but also convection). In order to obtain an implementation that is independent of the configuration of the cooling channels (important for the optimization process), an innovative solution for the discretization method is in order, which can be found in the Interface-enriched Finite Element Method. The equations that define heat transfer in the two phases of vascular materials are different. In the convection dominated regions, the standard finite element formulations are ineffective and it is necessary to find a different solution, being the Streamline Upwind/Petrov-Galerkin techique the chosen one. Being developed in a work group at *TU Delft*, all implementation aspects are integrated in *hybrida*, a computational tool developed by said group. Following this procedure made it possible to match results with verification tests for the discussed methods and also to optimize vascular geometries by minimizing their maximum domain temperature. Using the approach and results described above, it has been possible conclude that the developed work resulted in the correct implementation of each method and also that their combination is effective and efficient modeling and optimizing conjugated heat transfer problems defined by vascular actively-cooled domains.

Keywords: Interface-enriched Finite Element Method, Streamline Upwind/Petrov-Galerkin, Heat Convection, Optimization.

1. Introduction

As the the years are going by, the engineering inventions are getting more advanced and complex, with an outburst of new applications. With the appearance of advanced applications (some unimaginable a few years ago), a need for new materials that have certain characteristics has been created.

One of the ways used within the field of Materials Science and Engineering to design or improve materials is to employ natures solutions to deal with a certain problem. The materials that arise from this process can be called bio-mimetic and the chosen solutions being discussed here have already been time tested by nature through evolution and improved if successful. One of the concepts most easily associated with bio-mimetic materials for someone with an engineering background is the man-made honeycomb structures (inspired by natural structures such as beehives and bone, they have the goal of minimizing the amount of used material to reach minimal weight and minimal material cost).

As is indicated by the document's title, the topic of interest here is the design and modeling of actively-cooled vascular materials (which mimic the circulatory system found in plants and animals, using a coolant as the circulating fluid). The term active-cooling simply clarifies that the structures being considered use of an external force to drive the coolant in the channels, unlike natural forces such as gravity (refer to [1] for further detail). This vascular approach to cooling (with a fluid circulating in the channels) redistributes heat in the medium which, according to [2], reduces its maximum temperature, making it suitable high temperature applications (one example where the use of actively-cooled materials is being studied is the internal walls of airplanes' engines combustors).

The interest in this topic comes down to not only study the influence of the cooling channels' configuration in the maximum temperature of a certain domain but also to optimize said configuration to minimize that temperature.

The work has at its core the development of numerical and discretizaton tools that can, effectively and efficiently, help with the design and optimization of $2D$ bio-mimetic vascular materials with the objective of active-cooling by solving the conjugate heat transfer problem (heat conduction and convection). The implementation of these computational tools may not be very straight-forward due to the large number of design variables and ”special” numerical and discretization techniques (in addition to different material properties, the convection in the coolant must be considered, which means that there is a change not only in the material properties but also in the governing equations) but the potential applications of the materials being discussed provide motivation to explore it.

The development of the computational tools mentioned in the last paragraph that provide the motivation and results for this work is integrated in *hybrida*, a new in-house finite element package being developed by the *SOM* workgroup at *TU Delft*'s *3mE* department.

2. Theoretical Background

The objective of this section is to conceptually describe the theoretical concepts relevant to the study of actively-cooled materials. The fundamental concepts of Heat Transfer are presented in Subsection 2.1 and Subsection 2.2 briefly sums some mathematical concepts of Solid Mechanics needed for the implementation verification in Section 3.

2.1. Background in Heat Transfer

The theoretical concepts of Heat Transfer to describe here can all be introduced by Figure 1.

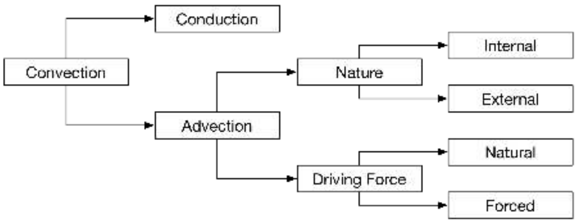


Figure 1: Concept of heat convection.

Heat is always transferred *via* heat convection which, when dealing with a solid phase, is limited to the phenomenon of conduction (process of transferring energy from particles at a greater temperature to particles at a lower temperature due to the interaction between them). When dealing with a fluid phase, the phenomenon of heat advection (process of transferring energy that relies on the macroscopic fluid motion) is no longer negligible.

Considering the vascular actively-cooled materials, the flow within the channels can be categorized as both internal and forced (external driving force).

The heat transfer process is described by the heat equation, which is simpler if advection is negligible. In this heat equation, the possible spatial boundary conditions (BCs) can be Dirichlet BCs (temperature of edge fixed), Neumann BCs (fixed edge heat flux) or Robin BCs (related to the existence of heat convection in the edge, not used in this work).

2.2. Solid Mechanics Fundamentals

Everything that needs to be described as far as Solid Mechanics goes in this work can be deduced from the equilibrium equation for a $3D$ body made of a linear elastic material (from [3])

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0\,, \tag{1}$$

where $\boldsymbol{\sigma}$ is the stress along the domain and $\mathbf{b}$ is the body force. This equation states that the sum of the point loads (left term) with the body force (right term) is null, which means that the body is in equilibrium.

The only equation that is going to be needed in Section 3 is the $1D$ solution to the equilibrium in Equation (1). The generic model problem for this $1D$ scenario is presented in Figure 2.

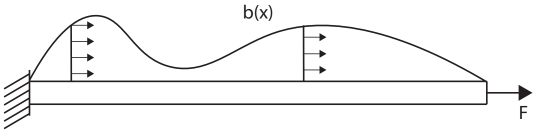


Figure 2: $1D$ linear elastic model problem.

Taking the x direction component of Equation (1) and considering the constitutive equation for linear isotropic materials, also known as Hooke's law ($\sigma = \varepsilon E$), and a material with constant Young's modulus (E), it is possible to obtain the $1D$ solution (comes down to integration and enforcing BCs)

$$Eu(x) + \int\int b(x)\,dx = C_1 x + C_2\,. \tag{2}$$

3. Discretization Method - IGFEM

The Finite Element Method (FEM) is the main tool behind obtaining the solution to the partial differential equations that describe a given problem *via* computer simulation and has at its foundation a discretization method that divides the problems domain into smaller elements. Because the main goal of this work is to model domains where there are discontinuities in the form of different materials and phases and also due to the fact that it is very expensive to discretize the whole problem for each modification in the geometry resultant from the optimization process, the choice of discretization technique has to be made “outside the box”.

3.1. Finite Element Method Fundamentals

Using just a few lines of text to describe the two main aspects of FEM (see [3]): it divides the problem's domain into smaller elements (discretization) that together form the finite element mesh and uses approximated functions of the governing equations, that depend on the type of problem being analyzed, to create the elemental matrices called stiffness matrices (that contain the coefficients for one element's equations). Then, those elemental matrices are assembled into one global stiffness matrix, resulting in a system of equations that can be solved by enforcing the boundary conditions and that yields the distribution of a main variable. If more results are needed, they can be obtained in an optional phase of post processing.

3.2. Requirements

To get accurate solutions using standard FEMs is only possible using a conforming mesh. In the case of domains where there are at least two materials, this requirement can only be addressed using a mesh whose elements boundaries are perfectly aligned with the interface.

This conforming mesh would be relatively cheap and straight-forward to implement for simple cases like the one in Figure 3, where the domain is composed of a square plate made out of two solid materials with a straight vertical interface. However, generating a conforming mesh could prove itself to be a complex and computationally heavy or even an impossible task when dealing with complex structures like vascular materials where a phase change may result in different material properties and governing equations.

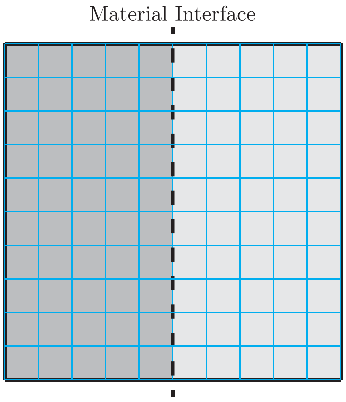


Figure 3: Example of simple conforming mesh.

3.3. Interface-enriched FEM

The answer chosen to deal with the requirements in the previous subsection is the Interface-enriched Generalized Finite Element Method (IGFEM), which was first in 2010 by [2] and is a state-of-the art improvement on the standard Generalized Finite Element Method (GFEM). GFEMs are direct extensions of the standard FEM and combine the desirable features of it with meshless methods. This leads to the possibility of obtaining an accurate solution of engineering problems using nonconforming meshes by applying *a priori* knowledge of the solution field in the numerical approximation.

Using GFEMs, the approximated solutions are obtained using

$$u^h = \underbrace{\sum_{i=1}^{n} N_i^p u_i}_{std.\ FEM} + \underbrace{\sum_{j=1}^{n_{en}} s_j \psi_j \alpha_j}_{enriched} \,, \tag{3}$$

where u^h is the approximated solution for the main variable, N_i^p is a set of n Lagrangian shape functions evaluated in the original element (of the nonconforming mesh), u_i represents the main variable of interest (in the case of this document, displacement or temperature) at the node labeled i in the mesh being used and s_j is an additional scaling factor that imposes that a well-conditioned stiffness matrix is constructed. Finally, ψ_j and α_j are a set of enrichment functions and the generalized degree of freedom associated with the j^{th} interface node created from the intersection point of the material interface with element edges, respectively.

To obtain the the enrichment functions, the implementation of IGFEM must first subdivide the elements that are cut by the phase interface (parent elements) into smaller triangular elements (children or integration elements). The phase interface should be included in the new elements' edges. The reason for the children or integration elements to be triangular is simply to simplify the computation of the enrichment functions and the only requirement for these elements is that their boundaries must conform to the discontinuity edges or surfaces of the domain (the aspect ratio of the children elements does not affect the accuracy of the solution).

The enrichment functions for a certain degree of freedom in the IGFEM formulation are obtained by evaluating the standard shape functions in the child element. This means that, after discretizing the parent element into smaller children elements it is only necessary to combine the standard shape functions for triangular elements evaluated at the parent and children elements functions in order to obtain the the complete approximation.

3.4. Implementation

The implementation of the IGFEM solver can be subdivided into two "aspects". The role of the computational functions that form the first "aspect" is to perform geometric operations. The mentioned geometric operations are related to modifying a non-conforming finite element by creating children elements that result from the intersection of the original mesh elements with the material interfaces and are performed by a module introduced to the code called *geometric engine*.

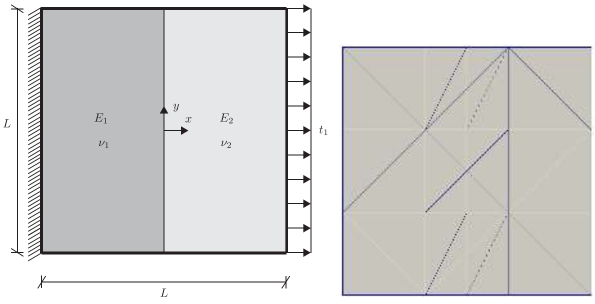
The second "aspect" of this implementation of

IGFEM is the elemental assembly. The new functionality needed to deal with the added complexity of computing enrichment functions is integrated in the standard routines and is used only in the presence of integration elements.

3.5. Verification and Applications

To have confidence in any computational results requires some sort of validation or verification and this implementation of IGFEM is no exception. In order to perform this verification, a couple of test cases have been inspired by [4] and its results reproduced using the developed implementation.

The first test case consists of the $2D$ bi-material square shaped domain represented in Figure 4(a). It has a side length of $L = 2\,m$ and is split by a material interface at $x = 0\,m$. As for the physical properties, material 1 has ($E_1 = 10\,Pa$, $\nu_1 = 0$) and material 2 ($E_2 = 1\,Pa$, $\nu_1 = 0$). The boundary conditions applied are a clamped left extremity and a constant traction per unit length applied at the right extremity. A constant body force is also being applied to the domain of the problem $b_1 = 2\,N/m^2$. The effect of using the developed implementation of IGFEM is showed with the sample mesh in Figure 4(b), which didn't have the vertical interface and smaller integration elements before it.



(a) Domain (b) Sample mesh (coarse)

Figure 4: Patch test verification problem.

Using a null Poisson's ration for both materials means that, under uniaxial traction and these boundary conditions, the deformation only happens in the x direction. Thus, the analytic solutions for the displacement and its derivative can be approximated by the $1D$ solutions and obtained from Equation (2) by applying the correct BCs and shifting the referential. Using these analytic solutions and computing the problem's solution for a range of different mesh refinements, it is possible to obtain the convergence plot in Figure 5.

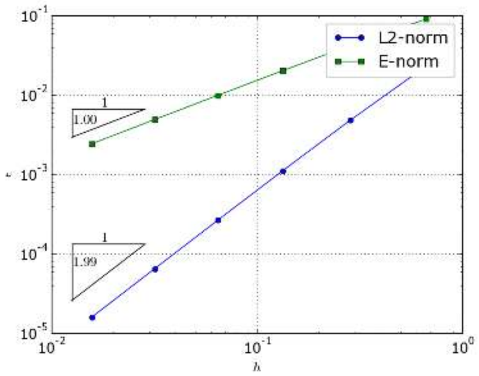
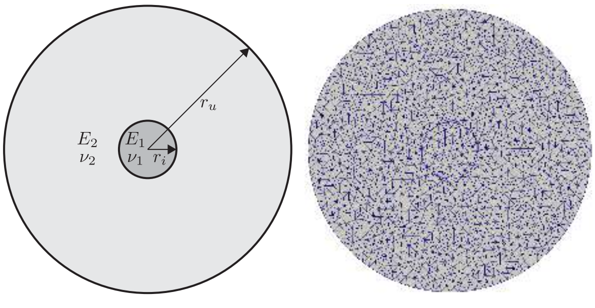


Figure 5: Convergence plot for patch test.

The obtained convergence rates are not only equal to the ones found in [4] but also the same as the optimal convergence rates for the standard FEM with a conforming mesh, which would not be possible for the nonconforming meshes that have been used with IGFEM.

The next example is based on the classical Eshelby inclusion problem and consists of the circular $2D$ domain with a concentric circular inclusion made out of a different material presented in Figure 6(a). For this convergence analysis, the outer radius chosen is $r_u = 2\,m$ and the radius of the inclusion is $r_i = 0.4\,m$. On the other hand, the inclusion is made out of material 1, with ($E_1 = 1\,Pa$, $\nu_1 = 0.25$), and the matrix out of material 2, with ($E_2 = 10\,Pa$, $\nu_2 = 0.3$). The effect of using the developed implementation of IGFEM is showed with the sample mesh in Figure 6(b), which didn't have the circular inclusion in the middle and smaller integration elements before it.



(a) Domain (b) Sample mesh (fine)

Figure 6: Eshelby inclusion verification problem.

This time, obtaining the exact solution and its derivative is not so simple, but it can be obtained in cylindrical coordinates from [4] and then converted to Cartesian coordinates and derived using a symbolic calculus tool such as *Mathematica*. Using these analytic solutions and computing the problem's solution for a range of different mesh refinements, it is possible to obtain the convergence plot in Figure 7.

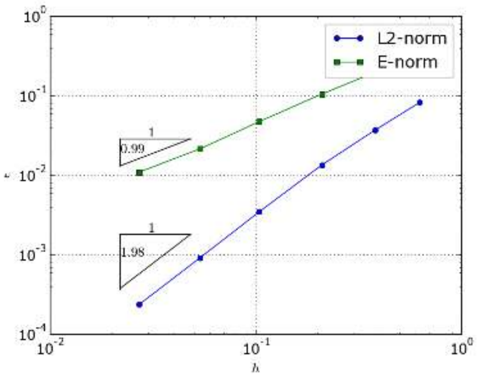


Figure 7: Convergence plot for inclusion problem.

There is not much to comment in this convergence plot, since the result is identical to what has been obtained previous one, optimal convergence rates. This result for both examples proves a correct implementation of the method, which means that the burden of creating a mesh that conforms to a possibly intricate geometry or of creating a new mesh for each optimization step can be avoided.

Finally, and to show some preliminary steps in the way of modeling vascular materials, the *geometric engine* has been used on the mesh in a nonconforming structured FE mesh to create a $2D$ rectangular plate with a vascular sinusoidal channel intersecting it, which is presented in Figure 8.

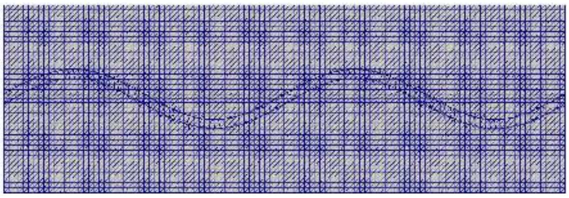


Figure 8: Sample mesh for vascular material.

3.6. Enrichment Functions Scaling

Despite the feature of not needing good quality finite elements as children elements, initial formulations of IGFEM had a problem in stiffness matrix conditioning. The answer to this problem is scaling the method's enrichment functions, which is shown in Figure 9. This operation does not affect the solution in any way, as it is accounted for in both creating the enrichment functions and in a postprocessing step.

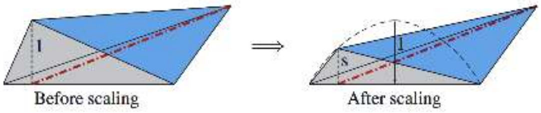


Figure 9: Enrichment function scaling from [2].

The relevant results to present are obtained for three of different scaling factors

$$s_j = 1\,, \tag{4}$$

$$s_j = \frac{min(d_1,\ d_2)}{d_1 + d_2}\,, \tag{5}$$

$$s_j = \frac{min(d_1,\ d_2)}{max(d_1,\ d_2)}\,, \tag{6}$$

where d_1 and d_2 are the distances from the interface to the previous and following original mesh nodes in the element connectivity data structure, respectively. The three scaling functions in Equations (4) through (6) correspond to a unitary scaling factor (standard) and the scaling functions from [2] and [5], which correspond to the initial IGFEM formulation (original scaling) and an hierarchic adaptive formulation (modified scaling). The impact of each scaling in the stiffness matrix conditioning is presented in Figure 10.

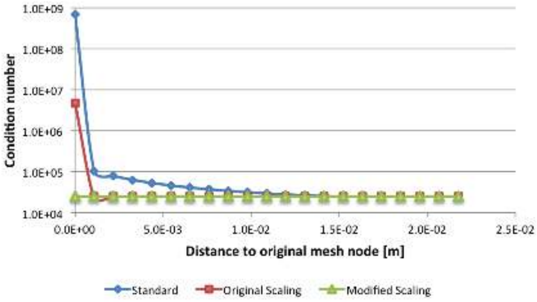


Figure 10: Condition number.

As can be easily concluded from Figure 10, the IGFEM clearly presents an ill-conditioning problem that is only slightly alleviated with the original enrichment function scaling in Equation (5) but completly fixed with the one in Equation (6).

4. Weighted Residual Formulation - SUPG

Having selected and discussed the choice of discretization method, it is now necessary to look into the choice of weighted residual formulation.

4.1. Requirements

The "best approximation" property from the standard Galerkin method is lost for convection dominated heat transfer processes, as the matrix associated to the convective term is non-symmetric, which means the solution field can be rendered useless due to the appearance of spurious oscillations, also called “wiggles” in the world of computational fluid dynamics (CFD), as is exemplified in Figure 12 for $1D$ convection.

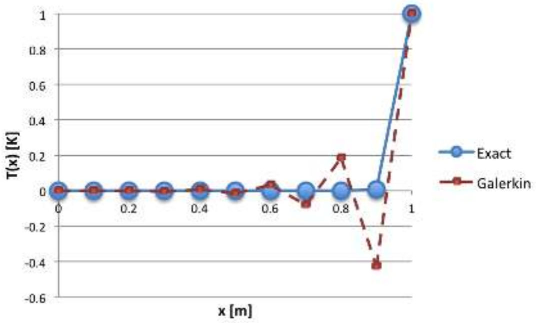


Figure 11: Analytic and Galerkin temperature distribution solution in a bar.

Taking this into account, it is necessary to find a formulation that alleviates or eliminates the appearance of “wiggles”.

4.2. Streamline Upwind/Petrov-Galerkin Method

The chosen alternative has first been presented in [6] and is the Streamline Upwind/Petrov-Galerkin (SUPG) method, which stabilizes the solution for problems that have highly convective effects present by modifying the weight function, which is done adding diffusion only in the flow direction.

In order to explain SUPG, it is necessary to present a generic conjugate heat transfer problem statement. Figure 12 represents a vascular actively-cooled material on a $2D$ domain $\Omega \subset \mathbb{R}^2$ that has two non-intersecting regions, the solid phase (Ω_s) and the fluid phase (Ω_f), which is where the the coolant flows. It is also possible to find the designations Γ_T, Γ_q and Γ_h (not used in this work), that are the boundaries to which each type of BC is applied, the first being Dirichlet, the second Neumann and third Robin boundaries.

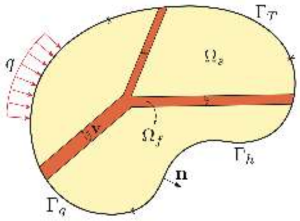


Figure 12: Domain heat transfer problem.

The strong form of the heat equation to obtain the temperature distribution field in this domain is *find the temperature field* $T : \Omega \Rightarrow \mathbb{R}$ *such that*

$$\begin{aligned} -\nabla \cdot (\mathbf{k}\nabla T) &= f \text{ in } \Omega_s \\ -\nabla \cdot (\mathbf{k}\nabla T) + \rho c_p \mathbf{v} \cdot \nabla T &= f \text{ in } \Omega_f \\ T &= \overline{T} \text{ on } \Gamma_T \\ \mathbf{k}\nabla \cdot \mathbf{n} &= q \text{ on } \Gamma_q \, . \end{aligned} \tag{7}$$

The presentation of the method’s formulation starts with defining all the relevant variables. The amplitude of the “wiggles” and the stabilization parameter for SUPG depend on the grid Péclet number. This dimensionless number is defined as

$$Pe_k = \frac{\text{rate of advection}}{\text{rate of conduction}} = \frac{|\mathbf{v_k}| h_k}{2k_k} \, , \tag{8}$$

where $|\mathbf{v_k}|$ is the norm of the vector quantity flow velocity, h_k is the element length in the flow direction and k_k is the thermal conductivity of the material in the flow direction.

The thermal conductivity of a material is actually a tensor. Consequently, the thermal conductivity in the direction of the flow can be computed as

$$k_k = \frac{||\mathbf{k} \cdot \mathbf{v_k}||}{||\mathbf{v_k}||} \, . \tag{9}$$

The formula for the stabilization factor is not element type independent and can be computed, for $2D$ elements ([7]), using

$$\tau = \frac{h_k}{2|\mathbf{v_k}|} \left(coth(Pe_k) - \frac{1}{Pe_k} \right) . \tag{10}$$

It is important to note that the stabilization is dependent on the flow velocity, it may not be the same for all elements composed of a fluid material or even constant inside the same element if the velocity profile is not constant. In reality, the flow velocity in the channels radially varies.

The SUPG method employs a modification of the weight functions used to obtain the FEM approximated solution, which means that it can be expressed as an additional stabilization term in the weak form of the problem as in

$$\begin{aligned} &\underbrace{\int_\Omega \nabla w^h \cdot \mathbf{k}\nabla u^h d\Omega + \int_{\Omega_f} w^h \rho c_p \mathbf{v} \cdot \nabla u^h d\Omega}_{\text{Standard convection}} + \\ &+ \underbrace{\int_{\Omega_f} \tau(\mathbf{v} \cdot \nabla w^h)(\mathbf{v} \cdot \nabla u^h) d\Omega}_{\text{Stabilization term}} = \int_{\Gamma_q} w^h \overline{q} d\Gamma \, . \end{aligned} \tag{11}$$

Concluding this section, the original *hybrida* code lacked the implementation of the second and third terms in Equation (11) in order to be able to effectively model the vascular cooled materials that are subject for this thesis.

4.3. Verification

Once again to ensure a good implementation, it is necessary to do verification. For this process, two examples inspired by [6] are used.

The first test case consists of a simple $1D$ bar of length $L = 1\,m$ made out of a material with a thermal conductivity of $k = 10^{-6}\,W/m \cdot K$ with the Dirichlet BCs of $T = 0K \quad @\, x = 0\,m$ and $T = 1K \quad @\, x = 1\,m$. The Galerkin solution for this problem has already been presented in Figure 11 (along with the analytic solution that derives from the heat equation and is $T(x) = \frac{1-e^{Pe\,x/L}}{1-e^{Pe}}$), while the result obtained using the implementation of SUPG is provided by Figure 13.

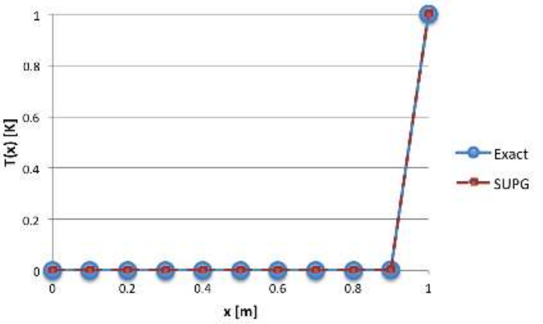
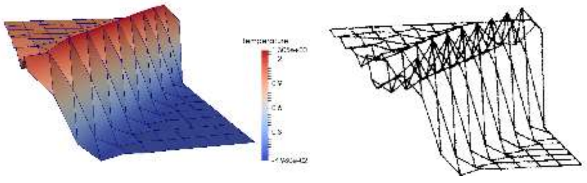


Figure 13: Analytic and Galerkin temperature distribution solution in a bar.

With a result that is identical to the analytic solution, the implementation of SUPG is proved to eliminate "wiggles" for this $1D$ convection problem.

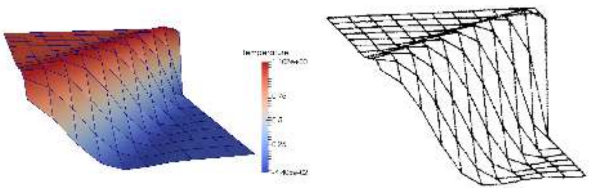
The next test case is a classical convection skew to the mesh problem whose details can be found in [6]. The results for both the Galerkin formulation and SUPG formulation from the implementation and the literature source can be found in Figures 14 and 15. Other skew angles have also been tested and similar results have been obtained



(a) Domain (b) Sample mesh (coarse)

Figure 14: Patch test verification problem.

Combining the somewhat qualitative conclusions that can found here (using the mesh outline as reference, the solutions appear to be similar) with the indisputable results obtained for the $1D$ bar, it is possible to claim with confidence that, from what is observed with these examples, the implementation of the method is correct.



(a) Domain (b) Sample mesh (coarse)

Figure 15: Patch test verification problem.

5. Test Case Modeling Results

It is in this part of the work that the two numerical methods discussed thus far are combined, showcasing the power of this union by modeling two test cases involving actively-cooled vascular material blocks.

5.1. CPU Cooler Test Case

Inspired by real life data, the first test case modeled using the developed stable thermal solver is pictured in Figure 16 and the remaining data expressed in Equation (12)

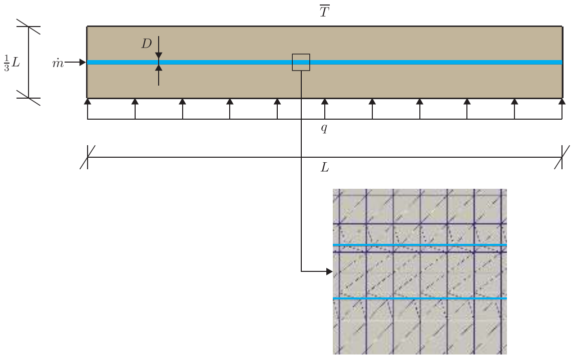


Figure 16: CPU cooler test case domain.

$$\begin{aligned}
&L = 45\,mm\\
&D = 0.8\,mm\\
&\dot{m} = 10\,g/min\\
&T_{inlet} = \overline{T} = 20\,C\\
&q = 150\,W\ (\text{uniform heat flux})\,,
\end{aligned} \tag{12}$$

where $\dot{m}$ is the mass flow rate that enters the cooling channel.

Despite being represented as a straight line, the cooling channel is modeled considering a sinusoidal centerline (the amplitude of the sine and wave length that are in this case null are going to be optimized in Section 7) with a parabolic Hagen-Poiseuille velocity profile given by

$$v = 1.5\frac{\dot{m}}{\rho\,A}\left(1 - \frac{2\,r}{D}\right)^2\,, \tag{13}$$

where r is the distance to the channel's centerline.

To ensure good numerical accuracy, a set of differently refined meshes have been used to obtain the maximum domain temperature. A convergence study graph is presented in Figure 17.

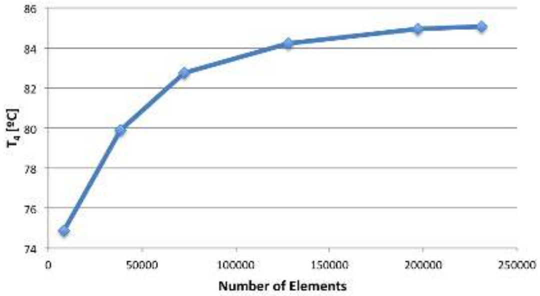


Figure 17: Mesh convergence study.

Given the time limitations, the mesh chosen to obtain the results and later optimization is a mesh with 128134 FEs, which presents a relative error of 1% to 55% of the computing time of the converged result mesh.

It is also interesting to look into the efficiency of the developed implementation, which can be done analyzing Table 1.

$NELEM$	T_{total} $[s]$	T_{solid} $[s]$	$\frac{\Delta t}{T_{\text{total}}}$ [%]
8290	13	7	43
38540	43	28	34
72591	96	63	35
128134	163	114	30
197127	255	176	31
231177	294	220	25

Table 1: Computation times *vs.* number of FEs.

Finally, the temperature field for the actively-cooled CPU cooler is presented in Figure 18.

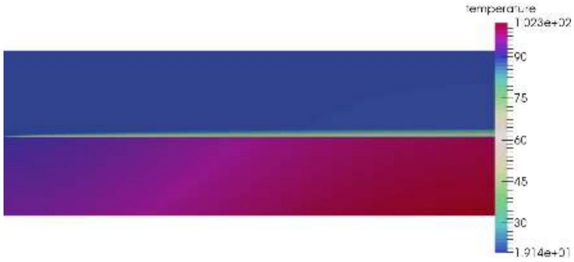


Figure 18: Mesh convergence study.

The baseline geometry characterized by the straight interface channel results in a maximum domain temperature of 102.3°C which is convenient, as it is clearly above the physical integrity threshold of a CPU (leaving room for improvement with optimization).

5.2. Engine Cylinder Test Case

In order to provide further numerical applications of the developed stable thermal solver, yet another test case has been devised. It mimics a car engine cylinder and it also proves that this implementation is capable of handling multiple channels. Because the analysis and results are similar to what has been presented for the CPU cooler and due to the space restrictions of this document, it is not presented here.

6. Optimization Fundamentals

Being one of the goals of this work to combine the stable IGFEM thermal solver with an optimization tool, this section briefly describes the fundamentals needed for this specific case.

6.1. Problem Formulation

The general form of a constrained optimization problem, which is the process of minimizing a function with respect to some variables in the presence of constraints on those variables, is

$$\begin{aligned} \text{Minimize: } & f(\mathbf{x},\mathbf{b}) \\ \text{Subject to: } & x_L < \mathbf{x} < x_U \\ & A\,\mathbf{x} \leq \mathbf{b} \text{ (Linear)} \\ & h(\mathbf{x}) \leq 0 \text{ (Non-Linear)}\,, \end{aligned} \tag{14}$$

where $f(\mathbf{x},\mathbf{b})$ is the objective function, the function that undergoes the optimization process and the equations under the category "Subject to:" are the constraints. The first equation under this category represents the bounds for the variables $\mathbf{x}$. It is also important to mention that constraints are not limited to linear relations, they can also be non-linear, as exemplified by the last two inequalities.

6.2. Optimization Tool

The process of optimization implemented for this work relies on an optimization package already implemented in a module of the programming language *Python*, which can be found under the *SciPy* library, and is called *optimize.minimize*, which does minimization of a scalar function of one or more variables.

6.3. Optimization Method

The *optimize.minimize* optimization module from the *SciPy* contains a large array of optimization methods that can be chosen by the user. Given it is the most well rounded in terms of the advantages and disadvantages, the chosen optimization method is the Sequential Least SQuares Programming (SLSQP) iterative method.

7. Test Case Optimization Results

The goal of this section is to demonstrate the possibilities that result from combining the efficient thermal solver that has been implemented with the standard *SciPy* optimization method choice discussed in Section 6 by analyzing a sample vascular actively-cooled example.

7.1. Optimization Problem Definition

As stated in Section 6, it is necessary to formulate the optimization problem before presenting its results. Using the CPU cooler test case modeled in Section 5, the objective function is the maximum domain temperature, the bounds are the second and third expressions in Equation (15) and one additional constraint is defined in the last expression

$$\begin{aligned} \text{Minimize: } & max[T(\mathbf{a}, \mathbf{n_{sw}})] \\ \text{Subject to: } & 2 < \mathbf{a} < 4\, [mm] \\ & 2 < \mathbf{n_{sw}} < 4 \\ & \Delta p \leq 1.5 \times \Delta p_{\text{straight interface}}\, , \end{aligned} \tag{15}$$

where Δp and $\Delta p_{\text{straight interface}}$ are the pressure drop in the channel due to viscous effects for the selected interface and straight interface, respectively.

While the choice of bounds for the design variables is related to geometrical restrictions introduced by the modeling strategy (sinusoidal centerline), the introduction of the pressure drop constraint is what makes this problem relevant, as the solution would otherwise be the upper limits of the design variables for an increased cooling capacity. Taking into account that in the parabolic Hagen-Poisseuile velocity profile pressure drop due to viscous effects equation (from [8]) all remaining variables are constant, the pressure drop ratio constraint in Equation (15) is equivalent to

$$\Delta p \propto s \Rightarrow \Delta p \leq 1.5 \times \Delta p_{\text{s.i.}} \Leftrightarrow s \leq 1.5 \times L \tag{16}$$

7.2. Enhanced Solution

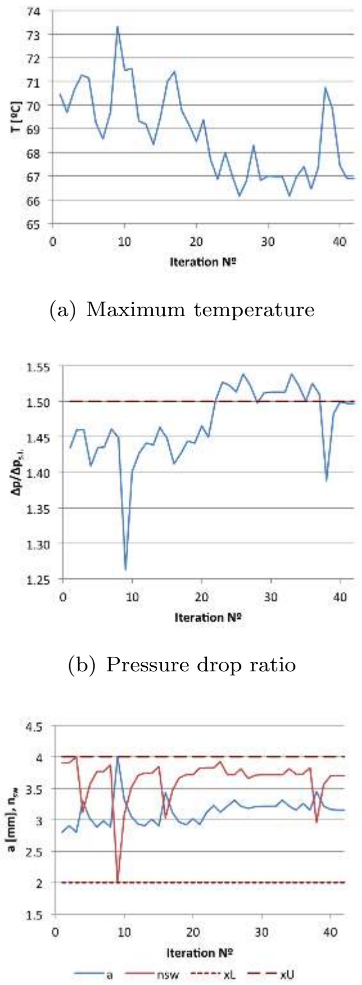
For a sinusoidal channel, the meshes that result from the IGFEM discretization are as presented in Figure 8. Table 2 presents the results of the relevant channel configurations, S.C. for straight channel, x_L for the the lower bounds of the design variables, where there is still room for improvement, U. for unconstrained, where the pressure ratio constraint is not active and O.D. for the optimized design.

$a[mm]$	n_{sw}	$max[T][°C]$	$\Delta p/\Delta p_{\text{s.i.}}$	Rmks.
0	0	102.3	1	S.C.
2	2	84.9	1.1	x_L
4	4	59.4	1.8	U.
3.15	3.7	66.9	1.5	O.D.

Table 2: Test case optimization results.

The combination of the chosen and implemented numerical techniques with the SLSQP optimization method from the *SciPy* library allowed this geometry to be optimized in just $1h40$ using a personal system running an *Intel Core i7* processor clocked at $2.2GHz$ with $16GB$ of memory and on *OS X Yosemite* despite the forty two iterations that the optimizer needed to obtain the result. A special emphasis must be given to IGFEM in problems such as the one being analyzed here, as it is a very interesting method that efficiently deals with the discontinuities and without which an optimization process to this problem would be much more user time consuming and cumbersome, with a constant need for remeshing.

The evolution of some relevant parameters with the iteration number can be analyzed in Figure 19. It is noticeable that there are maximum temperatures below the optimized result, but those correspond to geometries that violate the pressure ratio constraint.



(a) Maximum temperature

(b) Pressure drop ratio

(c) Design variables

Figure 19: Evolution of relevant optimization data with iteration number.

Finally, the temperature field solution for the optimized geometry, characterized by a channel amplitude of 3.15 mm and 3.7 sinusoidal waves in the horizontal direction of the domain, is represented in Figure 20.

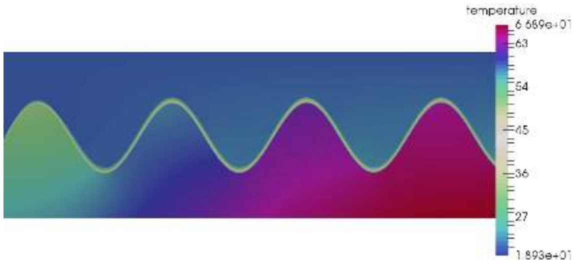


Figure 20: Optimized design.

With this optimization process, it has been possible to reduce the maximum temperature of a CPU cooler from a temperature of over 100°C, which is above integrity threshold to 66.9°C, an adequate CPU temperature under load.

8. Conclusions

If using the already implemented functionality of the computational tool simplified implementation, the conformity requirement that comes from using a preexisting tool means that the achievements required not only the understanding of the code's structure but also a lot of design thinking.

The basic formulation of a state-of-the-art discretization technique described by the Interface-enriched Generalized Finite Element Method (IGFEM) has successfully been implemented and verified for efficient mesh-independent modeling of $2D$ problems with discontinuous gradient fields. Some improvements on the original method were also implemented, namely *via* the modification of the enrichment function scaling.

Presenting oscillatory solution fields for convection dominated problems (characteristic of the methods that use a Galerkin formulation of convection), the weighted residual formulation previously implemented was inadequate. As a solution for this problem, the Streamline Upwind Petrov-Galerkin (SUPG) method has been implemented and verified for heat transfer processes where heat convection is not negligible, stabilizing the solution and reducing the amplitude of fictitious oscillations.

Combining both the IGFEM and SUPG in an efficient and stable thermal solver allowed to solve conjugate heat transfer problems of vascular actively-cooled materials. Combining this solver with a standard *SciPy* optimizer facilitated a straightforward and efficient optimization of a sample conjugate heat transfer problem of a vascular actively-cooled material.

With the goal of a continuous evolution of *hybrida* in the subject of this thesis, it could be interesting to study the implementation of specially developed optimization routines, to generalize the implemented methods to $3D$ and to use the developed tools in real optimization problems with well defined bounds, goals and constraints.

References

[1] T. L. Bergman, A. S. Lavine, F. P. Incropera, and D. P. Dewitt. *Fundamentals of Heat and Mass Transfer.* John Wiley and Sons, 7^{th} edition, 2011. ISBN:978-0470-50197-9.

[2] S. Soghrati, A. M. Aragón, C. A. Duarte, and P. H. Geubelle. An interface-enriched generalized FEM for problems with discontinuous gradient fields. *International Journal of Numerical Methods in Engineering*, 89:991–1008, August 2011. doi:10.1002/nme.3273.

[3] J.N. Reddy. *An Introduction to the Finite Element Method.* McGraw-Hill, 3^{rd} edition, 2006. ISBN: 0071244735.

[4] A.C. Ramos, A. M. Aragón, S. Soghrati, P. H. Geubelle, and J.F. Molinari. A new formulation for imposing Dirichlet boundary conditions on non-matching meshes. *International Journal of Numerical Methods in Engineering*, 2015. doi: 10.1002/nme.4898.

[5] S. Soghrati, C.A. Duarte, and P:H. Geubelle. An adaptive interface-enriched generalized FEM for the treatment of problems with curved interfaces. *International Journal for Numerical Methods in Engineering*, 102:13521370, January 2015. doi: 10.1002/nme.4860.

[6] A.N. Brooks and T.J.R Hughes. Stramline Upwind/Petrov-Galerkin formulations for convection dominated flows with particular enphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982. doi: 0045-7825/82/0000-0000/02.75.

[7] S. Soghrati. *An Interface-enriched Generalized Finite Element Method for the design of actively-cooled microvascular composites.* PhD thesis, University of Illinois at Urbana-Champaign, 2013.

[8] F.M. White. *Fluid Mechanics.* McGraw-Hill, 4^{th} edition, December 1998. ISBN: 007069673.