



# USER'S MANUAL

V1.2

**DISCLAIMER****FreeHyTE – Heat HTTE**

Hybrid-Trefftz Temperature Finite Elements for Steady-State Heat Conduction

Copyright © 2016, CERIS, ICIST, Instituto Superior Técnico, Universidade de Lisboa

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

## ACKNOWLEDGEMENTS

### **FreeHyTE – Heat HTTE**

Hybrid-Trefftz Temperature Finite Elements for Steady-State Heat Conduction

Copyright © 2016, CERIS, ICIST, Instituto Superior Técnico, Universidade de Lisboa

The following people (listed alphabetically) contributed, in one way or another, to the development of this software:

Ana Coutinho, Antero Sequeira, Belen Palácios, Carol Correia, Cornelius Cismasiu, Cristina Silva, Daniel Pereira, Ildi Cismasiu, Ionut Moldovan<sup>1</sup>, João Freitas, Lucian Radu, Michael Klanner, Rita Geraldès, Vasco Silva.

The development of this software was partially funded by Fundação para a Ciência e a Tecnologia of Portugal through Grant SFRH/BPD/87317/2012.

---

<sup>1</sup> Correspondence to dragos.moldovan@tecnico.ulisboa.pt

## TABLE OF CONTENTS

DISCLAIMER.....	2
ACKNOWLEDGEMENTS.....	3
TABLE OF CONTENTS .....	4
1. INTRODUCTION .....	6
1.1. FreeHyTE .....	6
1.2. TREFFTZ FINITE ELEMENTS IN FreeHyTE – HEAT HTTE .....	7
1.3. ABOUT THIS MANUAL.....	8
2. STRUCTURES AND MODELS.....	9
2.1. MATHEMATICAL MODEL.....	9
2.2. APPROXIMATE SOLUTIONS.....	10
3. HYBRID-TREFFTZ FINITE ELEMENTS.....	12
3.1. CONVENTIONAL vs. HYBRID-TREFFTZ TEMPERATURE FINITE ELEMENTS... ..	13
3.1.1. Enforcement of the governing equations .....	13
3.1.2. Approximation functions.....	14
3.1.3. Finite element solving system.....	15
3.1.4. Improvement of solutions.....	16
3.2. ADVANTAGES AND DRAWBACKS OF HYBRID-TREFFTZ FINITE ELEMENTS ..	17
3.2.1. Advantages of hybrid-Trefftz finite elements.....	17
3.2.2. Drawbacks of hybrid-Trefftz finite elements.....	17
4. STRUCTURE DEFINITION IN FreeHyTE – HEAT HTTE.....	18
4.1. INTRODUCTION.....	18
4.2. SYSTEMS OF REFERENCE .....	19
4.3. DESCRIPTION OF THE SAMPLE STRUCTURES.....	21
4.3.1. The solid structure .....	21
4.3.2. The hollow structure .....	22
4.4. GEOMETRY AND MESHING .....	23
4.4.1. Regular mesh generator .....	23
4.4.2. Non-regular mesh generator .....	25
4.5. BASIS REFINEMENT .....	30

4.5.1.	Strategies for basis refinement .....	30
4.5.2.	Manual basis refinement .....	31
4.5.3.	Automatic basis refinement – the basics .....	33
4.5.4.	Automatic basis refinement – selection criteria .....	34
4.5.5.	Automatic basis refinement – stopping criteria.....	35
4.5.6.	Automatic basis refinement – user input.....	37
4.6.	BOUNDARY CONDITIONS.....	38
4.6.1.	General definition.....	38
4.6.2.	Boundary conditions for the solid and hollow bodies .....	39
5.	GRAPHICAL USER INTERFACE .....	40
5.1.	INTRODUCTION.....	40
5.2.	GUI 1: STRUCTURAL AND ALGORITHMIC DEFINITIONS .....	42
5.2.1.	General features of the interface.....	42
5.2.2.	Saving and loading .....	43
5.2.3.	Data input in GUI 1.....	44
5.2.4.	GUI 1 data for the solid and hollow bodies .....	45
5.3.	GUI 2: DEFINITION OF THE BOUNDARY TYPES .....	47
5.3.1.	Structure visualization zone.....	47
5.3.2.	Definition of the external boundary types .....	48
5.4.	GUI 3: DEFINITION OF THE BOUNDARY CONDITIONS .....	49
5.5.	GUI 4: AUTOMATIC REFINEMENT.....	51
5.6.	VERIFICATION GUI .....	53
5.7.	POST-PROCESSING.....	55
5.7.1.	The solid body results .....	55
5.7.2.	The hollow body results.....	56
5.7.3.	The hollow body results – adaptive basis refinement.....	58
6.	ADVANCED STRUCTURAL DEFINITION .....	61
6.1.	INTRODUCTION.....	61
6.2.	LOCALIZED P-REFINEMENT.....	62
6.3.	DEFINITION OF DIFFERENT MATERIALS .....	64

## 1. INTRODUCTION

### 1.1. FreeHyTE

**FreeHyTE** is a collection of finite element solvers for elliptic, parabolic and hyperbolic initial boundary value problems using hybrid and hybrid-Trefftz finite elements.

The **FreeHyTE** computational platform is developed at the CERIS Research Centre, Instituto Superior Técnico, University of Lisbon. The development structure relies heavily on the constant engagement of Senior Year Master of Science students to perform the bulk of the coding duties for the various modules of **FreeHyTE**. Consequently, each module of the platform is developed and deployed separately, although they all share the same workflow, data structures, computational procedures and I/O sequences.

Each module of **FreeHyTE** is released under the GNU Public Licence and is free software. You are welcome to use it, improve it, and expand it, provided you only release improvements and/or expansions under the GNU Public Licence.

**FreeHyTE** stemmed from the gradual realization that vast amounts of code developed in higher education are simply lost when the main developer moves out, rendering most of the subjacent research virtually unreproducible; and that a change in the research paradigm was required to preserve and disseminate the results of our work. **FreeHyTE** is an attempt at the standardization of data structures, processing routines and numerical procedures that are pervasive in hybrid finite element formulations, aiming to secure a platform for the diverse developments to be plugged in with minimal coding necessities.

To the best of our knowledge, **FreeHyTE** is, as of 2016, the only user-friendly and publicly available software employing hybrid and hybrid-Trefftz finite elements. We take pride in placing the (considerable) advantages of these formulations at the fingertips of users and researchers around the world.

Unfortunately, however, research is conducted nowadays under a ‘Publish or Perish’ vision that verges, at times, on insanity. We are no exception. Consequently, **FreeHyTE** is more focused on providing a simple testing ground for new ideas rather than the best possible experience for the user (though we still think it’s fairly simple to use). While we try not to be exceedingly sloppy as we code, we cannot afford to invest heavily into the ultimate optimization of the execution times, memory allocation or bug-proofing. You are welcome to help us improve the code and we’ll give you credit if you do.

## 1.2. TREFFTZ FINITE ELEMENTS IN FreeHyTE – HEAT HTTE

**FreeHyTE – Heat HTTE** uses the temperature model of the hybrid-Trefftz finite elements for the solution of steady-state heat conduction problems in two dimensions.

Unlike conforming temperature (conventional) finite elements implemented in the vast majority of commercial codes, hybrid-Trefftz temperature finite elements (HTTE) use independent approximations of the temperature in the domains of the elements and of the normal heat flux on the Dirichlet and interior boundaries (hence the *hybrid* label of the elements). Moreover, the domain temperature bases are constructed using approximation (shape) functions that satisfy exactly all domain equations (hence the *Trefftz* label of the elements). Such functions are thus tailored for each particular problem that is being solved and embody relevant physical information about the model.

As a direct consequence, hybrid-Trefftz models endorse the use of *extremely large* finite elements as compared to the conventional models and are able to handle problems involving high solution gradients and discontinuous solution fields without cumbersome local mesh refinements. Awkward topologies and gross mesh distortions are also efficiently handled by hybrid-Trefftz elements.

Users acquainted to conventional finite elements must be familiar with their node-wise approximation functions, typically linear or bi-linear in shape, with having the nodal temperatures as degrees of freedom, and with the mesh refinement as a mean to improve the finite element solution. Conversely, hybrid-Trefftz finite elements move away from the nodes as the pivotal finite element concept to offer user more flexibility in choosing arbitrary, high-order approximations for the unknown fields. This means that, while mesh refinement remains a valid option for improving the quality of the Trefftz finite element solution, the same effect can now be obtained by simply incrementing the order of the approximation functions instead. Definition of distinct orders of refinement for each finite element and essential (i.e. exterior Dirichlet and interior) boundary is also affordable. A systematic analysis of the differences between conventional and hybrid-Trefftz finite elements from a user's standpoint is presented in Section 3.1.

**FreeHyTE – Heat HTTE** offers you all these advantages and flexibility, and features a Graphical User Interface (GUI) to endorse a seamless structure definition, with minimal effort. We hope you enjoy it and contribute to it in the near future.

### 1.3. ABOUT THIS MANUAL

This manual is aimed at presenting when and how **FreeHyTE – Heat HTTE** module should be used. Consequently, the manual follows a strictly a need-to-know, user-oriented perspective and should provide enough information for the needs of beginner and advanced users alike.

Chapter 2 focuses on *when* should you use **FreeHyTE – Heat HTTE**. It presents the main simplifying hypotheses under which the program operates and helps identifying the practical situations where it should be used.

Chapter 3 explains just as much about the hybrid-Trefftz temperature elements (HTTE) as a user should know in order to safely run the module. As most finite element users are acquainted to conventional finite elements, the text stresses the similarities and differences between conventional and hybrid-Trefftz formulations.

Chapter 4 introduces the steps a user should take for a complete structural definition using HTTE elements and explains the conventional positive directions of the involved referentials.

Chapter 5 focuses on *how* **FreeHyTE – Heat HTTE** should be used. It presents the Graphical User Interface (GUI) and employs the practical example of a structure with (and without) a circular hole to illustrate its use.

Chapter 6 is dedicated to more advanced structural definition options which require localized changes in the `InputProc.m` file of the code itself.

This manual complements the **FreeHyTE – Heat HTTE Installation Manual**, which introduces the various module deployment options and the respective installation steps.



**Please note that the module was tested in Matlab versions 2012b, 2013a and 2015a.** Kindly let us know if you experience difficulties in using it under other versions of Matlab.

## 2. STRUCTURES AND MODELS

The modelling process works in two steps. The first step deals with the definition of the mathematical model, where the phenomenon is described using some set of algebraic and differential equations. In the second step, approximate solutions of these equations are found using some numerical method implemented in a software. Both steps induce errors to the solution and it is very important to understand that *just because you get a result, it doesn't mean that it's the right result*. It is the task of the computational analyst to assess the magnitude of the error and to decide whether the results are good enough for his/her practical purposes.

### 2.1. MATHEMATICAL MODEL

The behaviour of solids exposed to intense heat is a highly complex phenomenon which cannot be fully captured by any mathematical mean. Therefore, simplifying assumptions must be made in order to reduce its complexity to a manageable level.

The following simplifying assumptions were considered for the derivation of the mathematical models implemented in **FreeHyTE – Heat HTTE**:

- the material behaviour is **physically linear**, meaning that the heat flux and the gradient of the temperature are related by a constant conductivity coefficient;
- the material is **piecewise homogeneous and uniform**, meaning that its thermal properties are identical in all points of some arbitrary regions of the structure. Consequently, insertions of one material into another can be modelled, but smooth transitions of the material's properties, taking place over some transition regions, cannot;
- the interfaces between different materials have **negligible contact resistance**, meaning that no temperature drop occurs at such boundaries;
- the material is **isotropic**, meaning that its mechanical characteristics are the same in all directions.

Based on these assumptions, the governing mathematical model expresses the *thermal energy balance* of a differential element in the domain of the heated body and the linear relationship between the heat flux and the temperature gradient, that is, the *Fourier law*. For the full definition of the problem, the temperatures and normal heat fluxes applied to the boundaries of the body must be specified by the user.

Since the objective of the heat conduction modelling is to predict the real behaviour of the heated body, working with an inadequate mathematical model will compromise the results no matter how refined the computational model is. Therefore, the results obtained with **FreeHyTE – Heat HTTE** are only as good as the mathematical model allows them to be.

## 2.2. APPROXIMATE SOLUTIONS

A sufficiently accurate mathematical model is only half way to practically usable results since the equations governing the heat conduction are only analytically solvable in a few, very simple cases. For the majority of practical situations, approximate solutions must be found instead.

In this context, **FreeHyTE – Heat HTTE** is a hybrid-Trefftz finite element solver of the differential equations describing the heat conduction through a solid body. Upon completion of the analysis, **FreeHyTE – Heat HTTE** should be able to predict the temperature and heat flux fields that occur at any arbitrary point of the body.

Since it *approximates* the solution of the governing equations, **FreeHyTE – Heat HTTE** is a source of errors that adds to those caused by the simplifying assumptions of the mathematical model. The errors induced by the computational model should be controlled through a process known as the *refinement* of the model, as discussed in Section 4.5. The refinement can be controlled either by the analyst (Section 4.5.2) or by an automatic, p-adaptive algorithm (Sections 4.5.3 to 4.5.6). However, it is important to note that a high quality finite element solution only goes as far as the mathematical model allows it to go. If the latter does not meaningfully reproduce the reality, the solution is unusable no matter how refined the finite element model is.

Besides the simplifying assumptions presented in Section 2.1, the following limitations specific to the **FreeHyTE – Heat HTTE** module are in place:

- the structure has a *plane* behaviour. This does not necessarily mean that the structure must be plane, but that its behaviour can be fully captured by knowing the states of temperature and heat flux in a single plane. This situation is typical to structural elements subjected to a uniform temperature along their length, but not along their width and height. Fully three-dimensional hybrid-Trefftz finite elements must be formulated anew since Trefftz bases are problem-dependent, thus the extension is not trivial. Provided enough help is secured, the extension will be attempted in the future, but it's not on our short-term agenda;
- The internal heat generated in the body can only be constant in a given finite element (but can be different from one element to another). The extension to non-

constant heat sources is not on our short-term agenda, but may be secured as a particular case of the initial condition-dependent source terms in transient applications, which should be pursued soon;

- no Robin type (i.e. convection) boundary conditions can be applied in the current version of **FreeHyTE – Heat HTTE**. Only pure Dirichlet (i.e. temperature) and Neumann (i.e. heat flux) boundary conditions can be defined. Adding Robin boundary conditions is, however, in our short-term plans.

### 3. HYBRID-TREFFTZ FINITE ELEMENTS

A brief perspective over some key features of hybrid-Trefftz finite elements is presented in this chapter. The objective is to introduce the reader to the main differences between handling conventional (conforming temperature) and hybrid-Trefftz finite elements from a user's standpoint. It is assumed that the reader is acquainted with the basics of the former.

The presentation is structured in two parts. A comparison between conventional and hybrid-Trefftz finite elements is made in Section 3.1. Based on the features described there, the advantages and drawbacks of using hybrid-Trefftz finite elements to solve plane heat conduction problems are described in Section 3.2.

### 3.1. CONVENTIONAL vs. HYBRID-TREFFTZ TEMPERATURE FINITE ELEMENTS

The comparison between conventional and hybrid-Trefftz temperature finite elements focuses the way these formulations enforce the domain and boundary equations, the features of the approximation functions, the nature of the solving system, and the handling of the mesh and basis refinements. It is important to note that many (and indeed diverse) hybrid-Trefftz finite element formulations exist, so the features described next refer strictly to the formulation implemented in **FreeHyTE – Heat HTTE**.

#### 3.1.1. Enforcement of the governing equations

To recover the solution of heat conduction problems, two domain and two boundary equations are solved, namely the thermal energy balance and the Fourier law in the domain of each element, and the Dirichlet (temperature compatibility) and Neumann (flux balance) equations on the exterior boundaries. As exact solutions cannot be found, in general, some of these equations are enforced in an approximate (average, or weak) form, and some in an exact (strong) form. The enforcement procedure strongly influences the behaviour of the finite element formulation so it is important to understand how it is handled.

*Conventional finite elements.* Conventional finite elements are strictly compatible. This means that they satisfy exactly the *Fourier law* in the domain of the element and the *Dirichlet* boundary conditions on the exterior Dirichlet and interior (inter-element) boundaries. The domain energy balance equation is enforced weakly and the Neumann boundary condition is typically enforced at the nodes of the mesh. As a consequence, the temperature field is typically approximated with considerably superior precision as compared to the heat flux field.

*Hybrid-Trefftz temperature finite elements.* Hybrid-Trefftz temperature elements satisfy strongly *all domain* equations. The Dirichlet and Neumann boundary conditions are enforced weakly on the exterior boundaries of the structure. The Dirichlet boundary conditions are also enforced weakly on the interior boundaries of the mesh. The Neumann boundary conditions are not enforced at all on the interior boundaries. Since temperature and heat flux boundary conditions are enforced in the same way (except for the interior boundaries), the predictions of the respective fields are much more balanced in terms of quality than in the case of conventional elements, although some slight bias towards the temperature field is still present.

### 3.1.2. Approximation functions

*Conventional finite elements.* Nodes are the pivotal concept in the definition of the approximation functions in conventional finite elements. Nodal temperatures are the main unknowns (degrees of freedom) of the elements and all fields in all points of the structure are completely determined by their values. The number and localization of the nodes also determine the expressions of the (polynomial) approximation functions for the temperature field in the domain of the element. Consequently, the redefinition of the nodes requires the recalculation of all approximation functions. In order to ensure the strong compliance with the Dirichlet boundary conditions, the nodes of adjacent elements need to be connected (i.e. the mesh must be conforming), so the insertion of additional nodes in one element generally requires the redefinition of all elements of the mesh. The approximation functions are intrinsically able to recover exactly a constant temperature field, through the partition of unity property, meaning that the temperature field will converge to the exact solution as the elements grow smaller (i.e. the mesh gets more refined).

*Hybrid-Trefftz temperature finite elements.* Hybrid-Trefftz temperature elements approximate not only the **temperature field in the domain of the element**, but also the **heat flux field on its essential** (i.e. exterior Dirichlet and interior) **boundaries**. Both approximations abandon the concept of nodes altogether. The nodal temperatures are no longer the main unknowns of the problem and the redefinition of the nodes does not call for any redefinition of the approximation functions. Since nodes lose their significance, hybrid-Trefftz meshes need not be conforming. All approximation bases are *hierarchical*, meaning that the addition of a new approximation function does not require the redefinition of those previously present in the basis. Bounded to satisfy all domain equations, the temperature approximation functions embody relevant physical information on the phenomenon they model. They are tailored for each problem being solved and account for most super-convergent features of the hybrid-Trefftz elements, as discussed in Section 3.2. However, they are generally more difficult to integrate than their conventional element counterparts, especially when their order is high. Still on the drawback side, the unknowns of the discretized problem (i.e. the weights of the approximation bases) do not have any physical meaning, being generalized temperatures and heat fluxes instead. The approximation functions are intrinsically able to recover exactly a constant temperature field, through the inclusion of the explicit unit monomial in the domain approximation basis, meaning that the temperature field will converge to the exact solution as the elements grow smaller (i.e. the mesh gets more refined).

### 3.1.3. Finite element solving system

*Conventional finite elements.* The assemblage of the solving system is based on the enforcement of the nodal energy balance condition. As a consequence, the solving system is sparse and symmetric, but the nodal temperature variables are shared by all finite elements that converge in the respective node. Summation of conductivity matrices of neighbouring elements occurs on the main diagonal of the system. The system is always *kinematically indetermined* (meaning that all nodal temperatures cannot be recovered using only the Fourier law and the temperature continuity boundary condition), provided there is at least a single node with a free temperature. The system is not *singular* provided the temperature in at least one point is enforced and generally not *ill-conditioned*.

*Hybrid-Trefftz temperature finite elements.* The solving system of hybrid-Trefftz finite elements is assembled with no summation of conductivity matrices. The generalized temperature variables are thus element-specific and not shared between neighbouring elements. While this trait enables localized refinements of the approximation bases (see Section 3.1.4), it also means that the solving systems are larger than those of the conventional elements for the same number of finite elements and the same order of the domain temperature bases. On the other hand, since very large Trefftz elements are affordable, the solving systems generally result smaller, in practice, than the conventional ones (see Sections 5.7.1 and 5.7.2). Sparseness and symmetry properties of the conventional systems are preserved.

The solving system of hybrid-Trefftz finite elements is *not* always *kinematically indetermined*. Instead, the user must secure the kinematic indeterminacy through an adequate choice of the orders of approximation bases in the finite elements and on their essential (Dirichlet and interior) boundaries, as discussed in Section 4.5.2. The system is not *singular* provided the temperature is specified on at least a boundary, but can be more *ill-conditioned* than its conventional counterpart. Ill-conditioning is controlled in **FreeHyTE – Heat HTTE** by analysing the singular values of the matrix of coefficients and using pre-conditioners. Should the system still be ill-conditioned after the pre-conditioner is applied, a special-purpose, pseudo-inverse solver is used to obtain the solution (and a warning is issued), but the results may be affected.

### 3.1.4. Improvement of solutions

Finite element solutions can be improved in two ways: by reducing the size of the finite elements ( $h$ -refinement) and by increasing the order of the approximation bases ( $p$ -refinement). Combinations of the two may also be used. Both strategies are available for conventional and hybrid-Trefftz elements, as explained below.

*Conventional finite elements.* **Mesh refinement is the main mean of improving the solution** in conventional finite elements (and indeed the *only* mean in some commercial software). Mesh refinement can be localized (e.g. in zones where larger solution gradients are expected), as long as it secures the conformity of the mesh, meaning that the nodes of adjacent elements need to be connected. Since the domain bases are not hierarchical, the  $p$ -refinement of conventional elements requires the insertion of new nodes and the re-computation of all approximation functions. Moreover, the addition of new nodes and the mesh conformity requirement renders localized  $p$ -refinement virtually impossible – refining a single element requires all elements to be refined as well. Such drawbacks hinder the adoption of the  $p$ -refinement as the main solution improvement strategy and limits, in practice, the domain bases to low-order polynomials.

*Hybrid-Trefftz displacement finite elements.* **Basis refinement is the main mean of improving the solution** in hybrid-Trefftz finite elements. Due to the physical information built in the domain bases, *the elements are super-convergent under  $p$ -refinement*. The hierarchical approximation bases mean that the addition of new functions does not call for the re-calculation of the previous ones. The removal of the node as the key concept in hybrid-Trefftz finite elements and the uncoupling of the elemental conductivity matrices in the solving system enable distinct definitions for the orders of approximation on each finite element and essential boundary of the mesh (i.e. localized  $p$ -refinement is affordable). For these reasons, hybrid-Trefftz bases are typically of higher order as compared to the conventional elements. Exceedingly high orders of the finite element bases may, however, cause numerical instability due to the integration and system solution errors, a situation that should be avoided (see Section 5.7.2). The alternative mesh refinement may be slightly less convergent than for conventional finite elements, but can be performed without securing the conformity of the mesh since the nodes of adjacent elements do not need to match.

### 3.2. ADVANTAGES AND DRAWBACKS OF HYBRID-TREFFTZ FINITE ELEMENTS

The advantages and drawbacks of hybrid-Trefftz finite elements as compared to conventional finite elements are listed below, following a user's perspective. Algorithmic advantages (like the fact that no domain integration is needed to compute the stiffness matrices) are not treated.

A practical guide on how to exploit the advantages while avoiding the pitfalls is presented in Chapter 4.

#### 3.2.1. Advantages of hybrid-Trefftz finite elements

- better balance between the quality of the temperatures and heat flux results;
- shape functions that embed the physics of the problem;
- very large finite elements are affordable. Leading dimensions can be 10-30 times larger than those of conventional elements;
- very high convergence under p-refinement. High-order approximation bases are mainstream;
- total flexibility in choosing the orders of the approximation bases in each element and on each essential (Dirichlet and interior) boundary. Localized p-refinement is affordable;
- results are insensitive to high solution gradients and gross mesh distortions;
- mesh conformity is irrelevant.

#### 3.2.2. Drawbacks of hybrid-Trefftz finite elements

- slightly lower convergence under h-refinement;
- the flexibility in choosing different orders of approximation in the elements and on the essential boundaries is a mixed blessing as it can be tough to handle for inexperienced users (this drawback is avoided in **FreeHyTE – Heat HTTE** by using an automatic p-refinement algorithm, see Sections 4.5.3 to 4.5.6);
- users need to ensure the kinematic indeterminacy of the solving system (this drawback is avoided in **FreeHyTE – Heat HTTE** by using the same algorithm);
- solving system is prone to ill-conditioning, especially if the orders of approximation bases are too high.

## 4. STRUCTURE DEFINITION IN FreeHyTE – HEAT HTTE

### 4.1. INTRODUCTION

This chapter is a hands-on guide to the structure definition in **FreeHyTE – Heat HTTE**. Its main focus is to guide the reader through the decision making process, from the choice of the most appropriate mesh generator for the given problem, the mesh definition, the refinement of the approximation bases (either manual or automatic), and the definition of the boundary conditions.



**FreeHyTE – Heat HTTE** features a **regular mesh generator**, recommended for **rectangular structures** (generates **rectangular elements**) and a **non-regular mesh generator**, recommended for **non-rectangular structures, or structures with holes, wedges or other geometric non-regularities** (generates **triangular elements**).

These concepts are applied to the solution of two problems, one featuring a solid body and the other a body with a circular hole at its geometrical centre. The first structure is analysed using two regular meshes with rectangular elements, while for the second structure, the meshes feature triangular elements and are obtained using an automatic generator built into Matlab. For both cases, one mesh is coarse and the other fine. The examples are chosen to illustrate the balance between the orders of h- and p-refinements and to give a quantitative perspective over the acceptable sizes of the elements and levels of basis refinement.

After the models' definitions are completed, the results obtained with **FreeHyTE – Heat HTTE** are presented and compared, along with the respective runtimes (see Section 5.7). The comparison illustrates the effect of the refinement choices on the points that matter most for the user: the quality of the solutions and the runtime.

To keep the presentation as light as possible, the description of the Graphical User Interface used for the definition of the models in **FreeHyTE – Heat HTTE** is saved for Chapter 5.

## 4.2. SYSTEMS OF REFERENCE

Before moving on to the presentation of the two examples, it is important to understand the definitions and positive directions of the referentials used for the descriptions of the structure's geometry and boundary conditions.



**Structure's geometry** is defined in a **global referential**. The **boundary conditions** are defined in the **boundary normal – tangential referential**. Both referentials are Cartesian.

To illustrate this principle, Figure 1 presents a structural domain with nine edges. The geometry of the domain must be specified in the global referential ( $XY$ ). The origin of the global referential is arbitrary when the non-regular mesh generator is used for the structural definition, and corresponds to the lower-left corner of the rectangular structure when using the regular mesh generator.

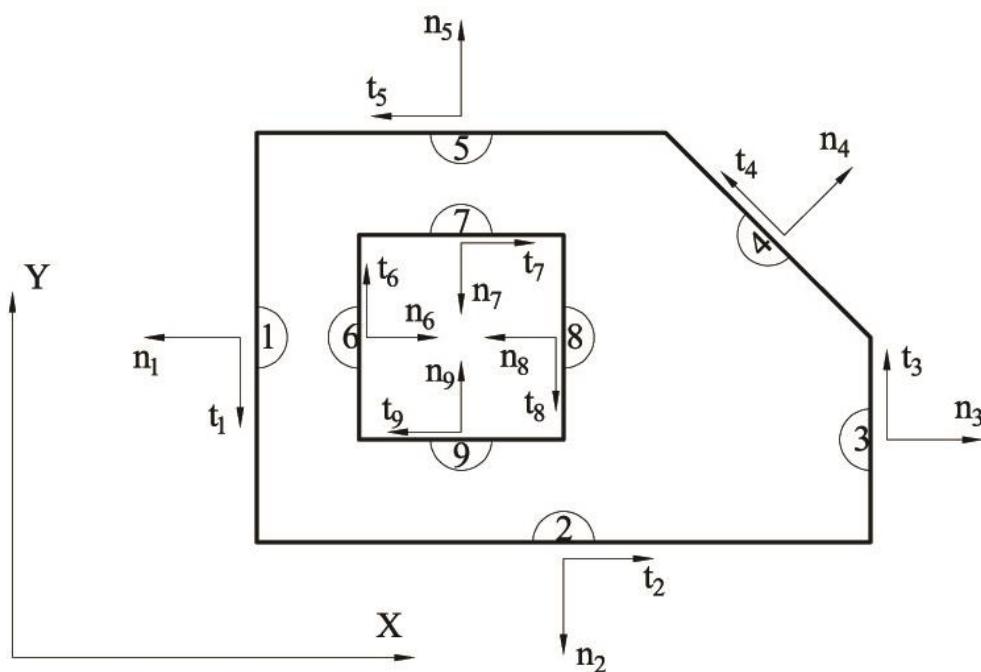


Figure 1. Positive directions of global and side referentials

The conventional positive directions of the boundary normal – tangential referential, are as follows:



- the **boundary normal axis ( $n$ )** is oriented **outwards from the element**;
- the **boundary tangential axes ( $t$ )** are oriented such as to **encircle the structure in a counter clockwise direction on the outside and in a clockwise direction on the interior openings**.

When defining normal heat fluxes on the Neumann boundaries of the mesh, their signs must be tuned according to the orientation of the boundary normal axis ( $n$ ). Please note, therefore, that the definition of the boundary conditions is **not** performed in the global Cartesian referential.

### 4.3. DESCRIPTION OF THE SAMPLE STRUCTURES

As previously mentioned, the structure definition process is presented here using two practical examples, namely two rectangular bodies without and with a circular hole in their geometric centre. The two structures are presented in this section.

#### 4.3.1. The solid structure

Consider the rectangular body presented in Figure 2, made of an isotropic and homogeneous material with a thermal conductivity  $k = 100 \text{ [W/(mK)]}$  and an internal heat generation of  $Q = 10.0 \text{ [W/m}^3\text{]}$ . The body is subjected to constant temperature fields of  $100 \text{ [K]}$  on its left ( $X = 0$ ) and right ( $X = 6$ ) boundaries. On the upper ( $Y = 3$ ) boundary of the body, a linearly distributed heat flux is applied in the negative normal direction (see Section 4.2), with the maximum intensity of  $60 \text{ [W/m}^2\text{]}$  at  $(X, Y) = (6, 3)$ .

Finally, a null normal heat flux field is specified on the lower ( $Y = 0$ ) boundary.

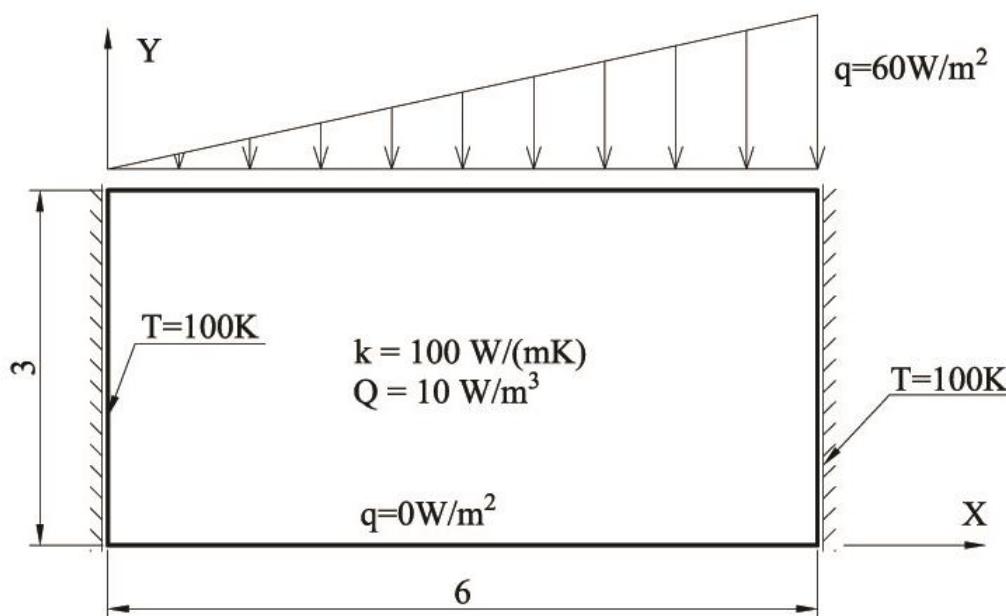


Figure 2. Solid body structure

The objective of the analysis is to determine the temperature and heat flux fields for the above structure using **FreeHyTE – Heat HTTE**. It is noted that a heat flux discontinuity in the vertical direction is expected to occur at point  $(X,Y)=(6,3)$ .

#### 4.3.2. The hollow structure

The second sample structure, presented in Figure 3, is identical to the solid structure presented in Section 4.3.1, except for an embedded circular hole with a diameter  $\phi=1[m]$  and the centre at  $(X,Y)=(3,1.5)$ , on which a constant temperature of  $100 [K]$  is specified.

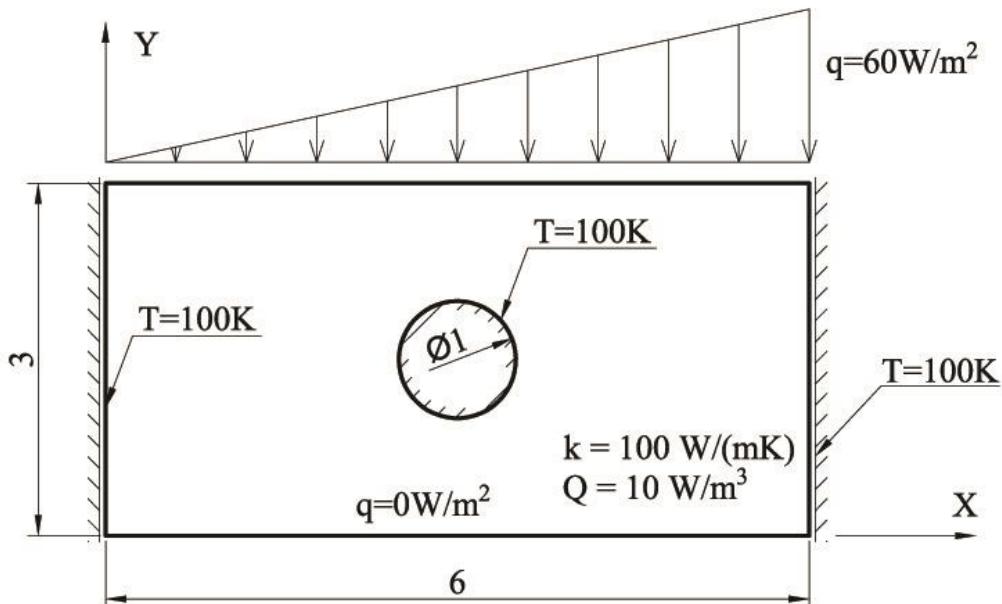


Figure 3. Hollow beam structure

The objective of the analysis performed with **FreeHyTE – Heat HTTE** is also the same.

#### 4.4. GEOMETRY AND MESHING

The definitions of the geometry and finite element mesh depend on the regularity of the structure and the mesh generator used for its discretization. Two mesh generators are currently implemented in **FreeHyTE – Heat HTTE**, namely a regular mesh generator and a non-regular mesh generator.



- the **regular mesh generator** can be used **only** for solid, **rectangular structures**;
- the **non-regular mesh generator** can be used for **any structural geometry**.

While the non-regular mesh generator can be used for rectangular structures, adopting the regular generator ensures a faster and simpler structural definition, with no risk of mesh distortion.



- the **regular mesh generator** produces **rectangular finite elements of uniform size**;
- the **non-regular mesh generator** produces **triangular finite elements**.

The usage of the two mesh generators for the definition of the bodies presented in Sections 4.3.1 and 4.3.2 is described next.

##### 4.4.1. Regular mesh generator

The regular mesh generator is used to divide a rectangular structure in an array of uniformly sized, rectangular finite elements. The geometry of the structure is defined by simply specifying its dimensions,  $D_x$  and  $D_y$ , in X and Y directions (Figure 4). The **origin of the global referential** is taken by default **in the lower left corner of the body** and cannot be changed.

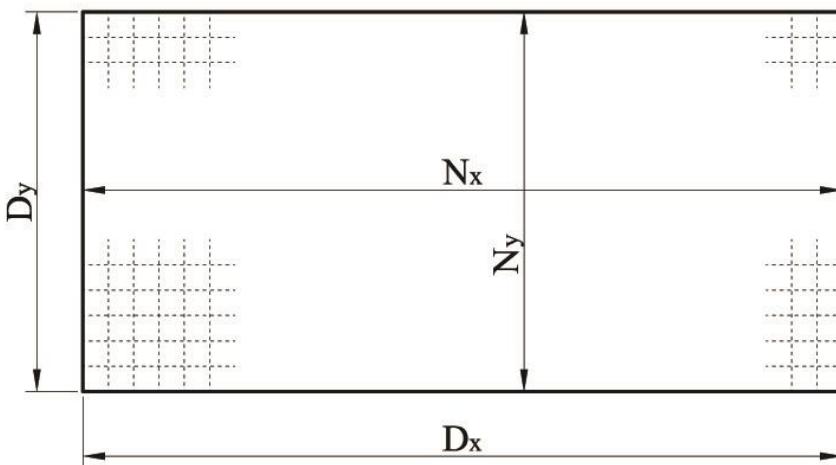


Figure 4. Regular mesh definition of a rectangular structure

Likewise, for the definition of the mesh it suffices to specify the number of finite elements in  $X$  and  $Y$  directions,  $N_x$  and  $N_y$ .



As a general rule, the **leading dimension of the hybrid-Trefftz elements** can be taken **10 to 30 times larger than** that of **conventional elements** without compromising the quality of the results.

The solid body described in Section 4.3.1 is discretized using a fine mesh with 18 finite elements (Figure 5a) and a coarse mesh with a single finite element (Figure 5b).

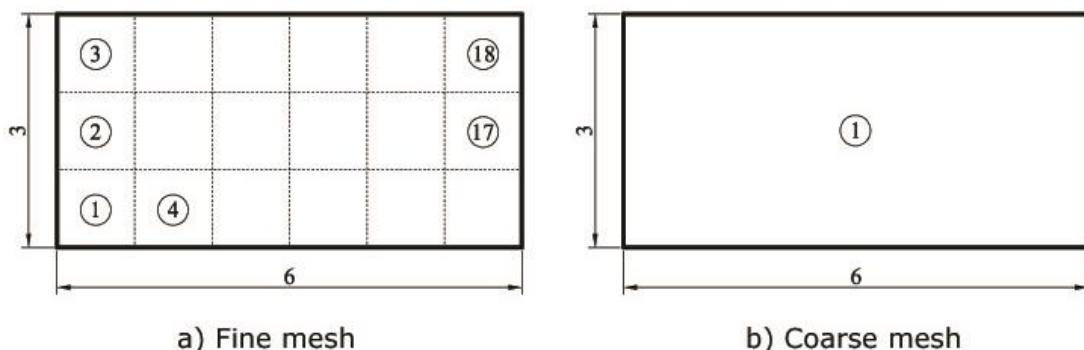


Figure 5. Meshes used for the solid structure

#### 4.4.2. Non-regular mesh generator

The non-regular mesh generation protocol in **FreeHyTE – Heat HTTE** is based on geometry definition and meshing procedures built in Matlab. The option to use native Matlab functions is justified by their efficiency, user-friendliness and increased odds of being compatible with upcoming versions.

There are essentially two ways to create a geometry in Matlab: **programmatically** (code-based) or using the **Graphical User Interface** (GUI) `pdetool`.

Code-based geometry definition enables the creation of virtually any shape, but you do not see the geometry as you create it and need to code a geometry function. The procedures are covered, among other places, [here](#), but their description falls outside of the scope of this manual.

The GUI `pdetool` is arguably the best way to create the geometry at the starting level. It features a simple click-and-drag interface and you get to see the geometry as you create it. Basic shapes as rectangles, ellipses and polygons are used as building blocks and you cannot create geometries that are not a combination of such shapes. However, keep in mind that the mesh generator is limited to triangular elements, so creating an elaborate, curved shape geometry just to triangulate it for meshing is probably not worthwhile.

A manual describing the use of `pdetool` for the geometry creation and meshing is available at this [link](#). Here, we shall only cover the basic steps needed to generate and mesh the hollow body presented in Section 4.3.2.

*Geometry definition using pdetool.* The `pdetool` command is **automatically invoked** during the execution of **FreeHyTE – Heat HTTE** (see Section 5.2.3). The command opens the GUI dialog presented in Figure 6. The geometry definition consists of the following steps:

- *draw the simple shapes that compose the geometry:* Rectangular, elliptical and polyline shapes are accessible from the buttons indicated in the upper ribbon (Figure 6). For our particular case, these shapes are a rectangle, constituting the solid part of the body, and a circle for the inner opening. Each shape receives an automatic label (*E1* and *R1*), as visible in Figure 7;
- *correction of the constituent shapes:* At this stage, it is unlikely that the shapes have the correct sizes and locations. Double click on each shape and specify its exact dimensions and positioning (Figure 8). Change the size of the window from *Options->Axes Limits* to fully visualize your structure;

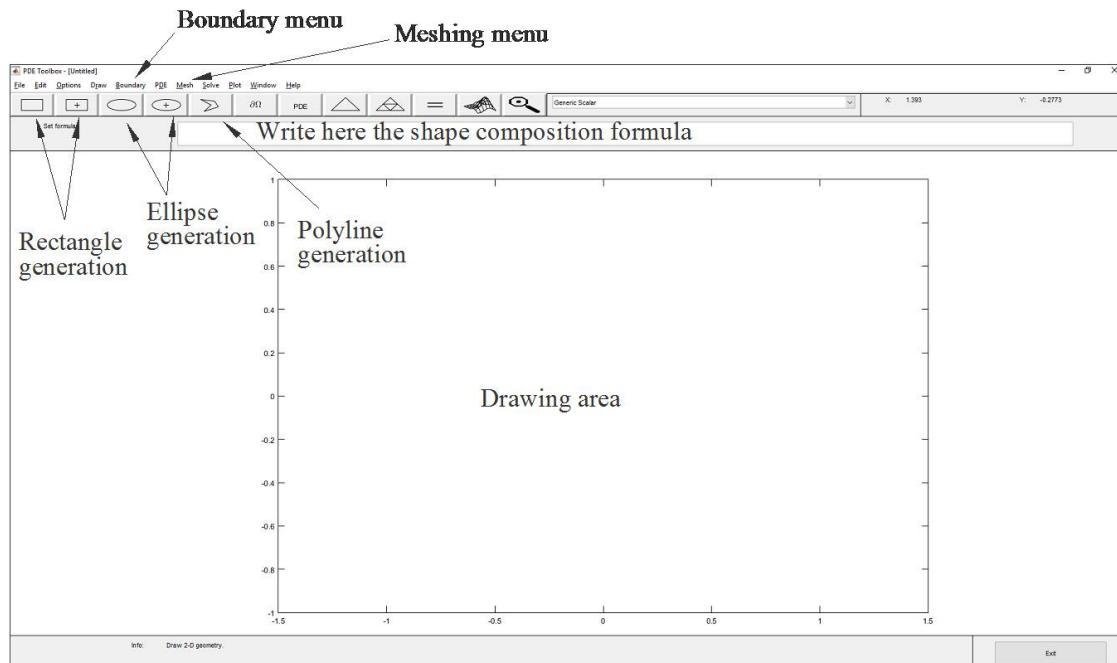


Figure 6. Initial pdetool screen

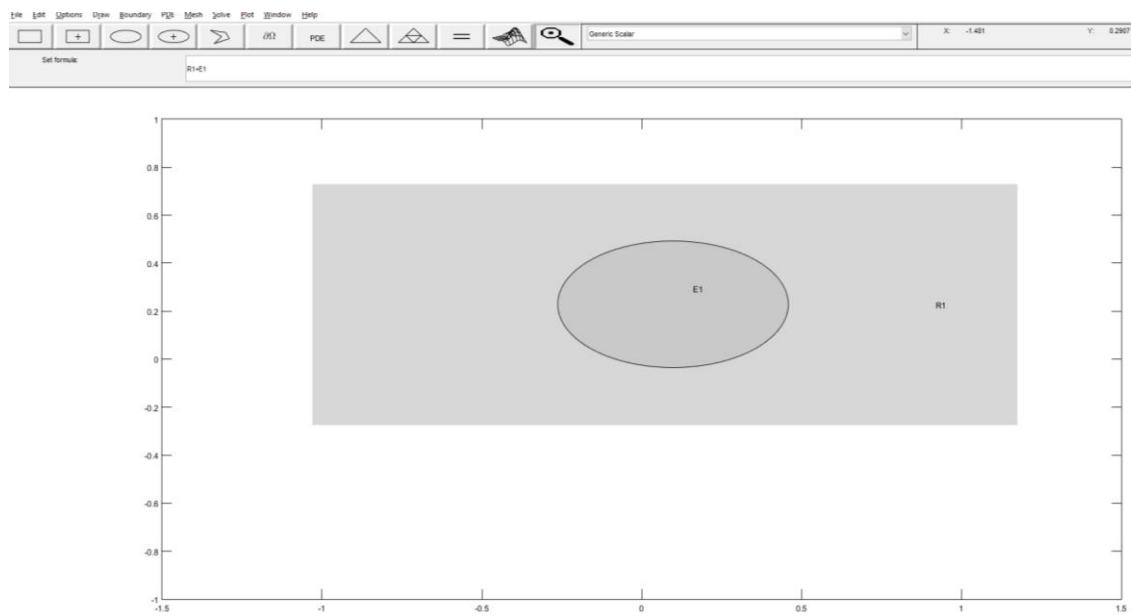


Figure 7. Initial sketch of the geometry

- *define the composition formula in the 'Set formula' zone:* Composition formulae let you add and/or subtract the simple shapes to get the final geometry of the structure. For our case, you will want to create the opening E1 in the rectangle R1, so the correct formula should read R1-E1. After setting up the composition formula, click on *Boundary->Boundary Mode* to apply it to your structure (Figure 9).

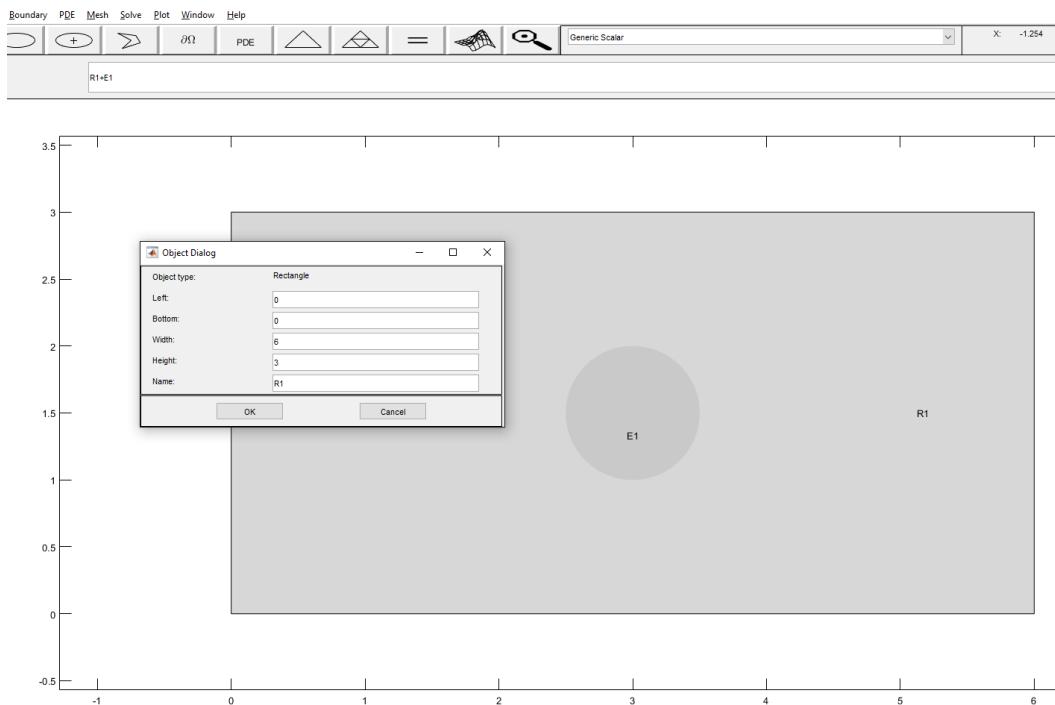


Figure 8. Object definition dialog. Constituent shapes in final form

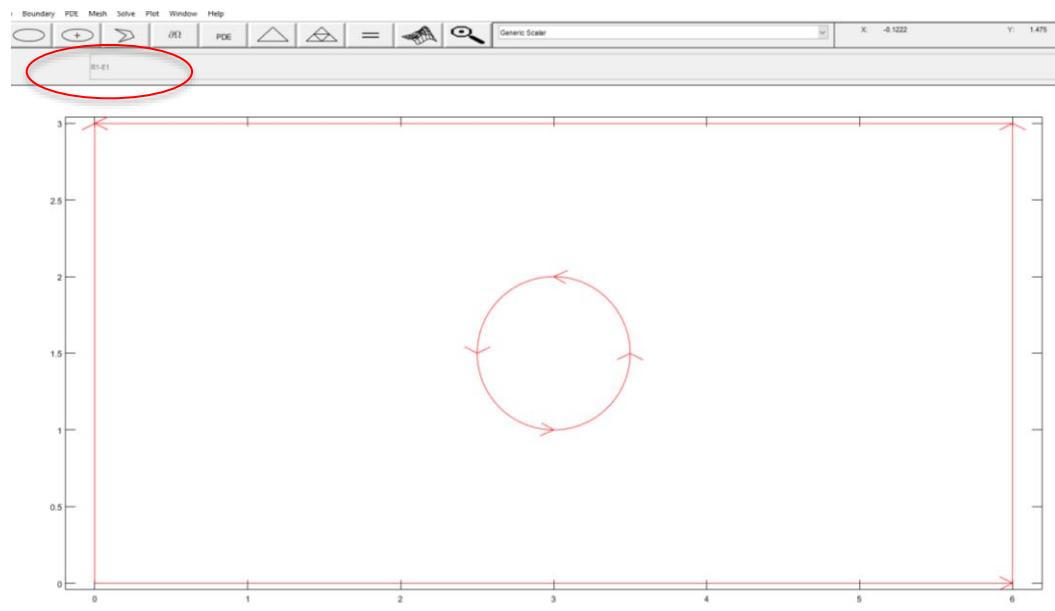


Figure 9. Shape composition formula. Final geometry concluded

*Mesh definition using pdetool.* Once the geometry definition is concluded, it is time to generate the finite element mesh. The procedure involves the following steps:

- *set up the meshing parameters:* Use the *Parameters* option in the *Mesh* menu of the *pdetool* interface. The available meshing options are explained [here](#). The most

critical options, which we shall tune to define the meshes for our hollow body structure, are:

- *maximum edge size*: Defines the largest triangle edge length. Two refinement levels are used for the hollow body, corresponding to the values of 2 and `inf` (from *infinity*, that is, as large as it can possibly be);
- *mesh growth rate*: Defines the rate at which the mesh size increases away from the small parts of the geometry. The value must be between 1 (all geometry is meshed according to the finest region) and 2 (for some reason, `pdetool` does not go beyond 2). Since hybrid-Trefftz elements can be large, we use the value of 1.99 for both refinement levels;
- *generate the mesh*: Use the *Initialize Mesh* option in the *Mesh* menu of the `pdetool` interface to generate the mesh. The meshes obtained for the two refinement levels are presented in Figure 10.

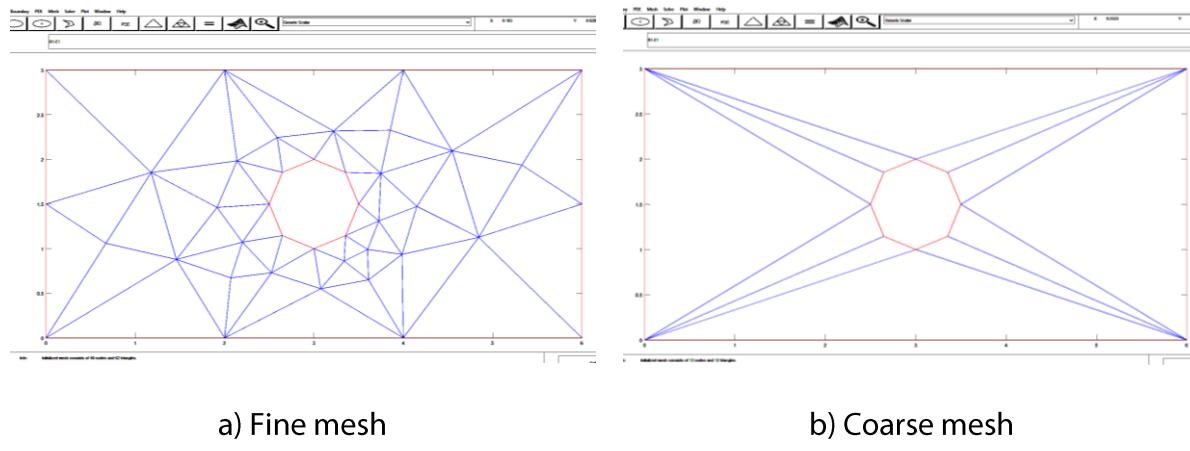


Figure 10. Meshes used for the hollow structure

- *export the mesh*: The mesh information must be exported to the base workspace in order to be read by **FreeHyTE – Heat HTTE**. To do so, go to the *Export Mesh* option in the *Mesh* menu and just click *Ok* in the *Export Mesh* dialog. Do **not** change the mesh variable names from the default `p`, `e` and `t` since doing so will render the mesh information illegible to **FreeHyTE – Heat HTTE**;
- *save the mesh as an \*.m file* (*File->Save As*) if you wish to be able to modify it later using `pdetool`.

From Figure 10, it is clear that the main factor conditioning the mesh generation is the presence of the circular hole. Since the native Matlab mesh generator uses only triangles to discretize the structure, the circular hole is modelled as a polygon (a regular octagon in our case).

In the case of the fine mesh, the modelling of the circular hole requires relatively small finite elements in its vicinity, but larger finite elements are used in the next layer, as indicated by the *Mesh growth rate* parameter. For the coarse mesh, setting the *Maximum edge size* parameter to `inf` causes extremely large and/or distorted finite elements to emerge. This is obviously a poor quality mesh and is not recommended, but is useful in the context of this manual to illustrate the behaviour of the hybrid-Trefftz finite elements under more extreme conditions.



You can call `pdetool` from the Matlab command window before executing **FreeHyTE – Heat HTTE**, or you can execute **FreeHyTE – Heat HTTE** and let it launch `pdetool` for you at the right time during the execution. **The latter option is recommended.**



A new description of the mesh (the [FEMesh](#) object) was introduced in Matlab version 2015a. For backward compatibility, **FreeHyTE – Heat HTTE** uses the legacy [Pet](#) mesh format. If you need to convert FEMesh to Pet, please use the [meshToPet](#) command. However, **you should not need any conversions if you use `pdetool`** as described above.

## 4.5. BASIS REFINEMENT

### 4.5.1. Strategies for basis refinement

As discussed in Section 3.1.2, the possibility of controlling the basis ( $p$ -)refinement of each finite element and essential (i.e. Dirichlet and interior) boundary is one of the most important traits of the hybrid-Trefftz finite elements implemented in **FreeHyTE – Heat HTTE**, but also one of the trickiest obstacles to the use of these elements by inexperienced users. Three reasons may cause an inexperienced user to hold back from embracing this additional flexibility (as compared to the conventional finite elements):

- first, the independent boundary flux approximation and the localized  $p$ -refinement are not common concepts in conventional finite elements, so having to deal with them may be puzzling;
- second, while the correct calibration of the bases refinements enables one to obtain excellent results with very coarse meshes and very few degrees of freedom, the incorrect calibration of the  $p$ -refinements risks to swiftly compromise the results altogether, so some experience is needed to take advantage of the former while avoiding the pitfalls of the latter; and,
- third, the orders of  $p$ -refinements must be chosen such as to guarantee that the finite element solving system is **kinematically indeterminate**, which is a trivial requirement in conventional elements, but not in hybrid-Trefftz elements.

To respond to such concerns, **FreeHyTE – Heat HTTE features two alternatives for defining the refinement of the bases:**

- the **manual definition** of the orders of  $p$ -refinement; and,
- the **automatic basis refinement** mode.



In the **manual  $p$ -refinement mode, the orders of refinement of all finite elements and of all essential boundaries are uniform**. The difficulties associated with securing a kinematically indeterminate system are eliminated by choosing these orders according to the rules presented in the next section.



In the **automatic p-refinement mode, the orders of refinement of the finite elements and essential boundaries are chosen by FreeHyTE – Heat HTTE**. The program also makes sure that the system is kinematically indeterminate, but the execution time is usually much larger than in the manual p-refinement mode.

The two alternatives are explained in more detail in the next sections and some ‘thumb rules’ for valid choices are also given.

#### 4.5.2. Manual basis refinement

The *order of basis refinement* is conceptually similar to the *order of the polynomial basis* used in conventional finite elements. Approximation bases of the conventional finite elements for plane problems are built by combining monomials from the Pascal’s triangle. The order of the basis is commonly defined as the degree of the last *complete* row in the Pascal’s triangle. Consequently, three and four nodes elements are linear, six nodes triangular and eight nodes rectangular elements are quadratic and so on. Hybrid-Trefftz finite elements implemented in **FreeHyTE – Heat HTTE** also use polynomial bases (harmonic polynomials in a complex description, to be exact), so the concept is similar and can be intuitively applied in the same way.

The (uniform) orders of p-refinement in the finite elements (denoted here by  $n_D$ ) and on the essential boundaries (denoted here by  $n_\Gamma$ ) must be chosen to ensure that, for each finite element, **the total number of temperature degrees of freedom** in the domain of the elements **is larger than the total number of heat flux degrees of freedom** on the essential boundaries of the mesh. This restriction is imposed in order to secure the kinematic indeterminacy of the solving system.



While, *in the limit*, observation of the kinematic indeterminacy criterion is not trivial, for its *loose* observation it is enough to ensure that,

$$\begin{cases} n_D \geq 2n_\Gamma + 2, & \text{for the rectangular element meshes, and} \\ n_D \geq 1.5n_\Gamma + 2, & \text{for the triangular element meshes} \end{cases}$$

Besides the observation of the above inequalities, it is recommended that  $1 \leq n_{\Gamma} \leq 7$  in order to avoid the ill-conditioning of the solving system. The orders of refinement respecting these criteria, and thus recommended for a numerically stable analysis with **FreeHyTE – Heat HTTE**, are listed in Table 1, along with the recommended number of Gauss-Legendre quadrature points for the integration of the solving system's coefficients.

It is noted that the values listed in Table 1 should serve as a guidance, not as a guarantee for correct or stable results. They should, however, provide a good starting point for less experienced users.

$n_{\Gamma}$	$n_D$ (triangular)	$n_D$ (rectangular)	Nº of Gauss points
1	4	4	10
2	5	6	10
3	7	8	10
4	8	10	15
5	10	12	15
6	11	14	20
7	13	16	20

Table 1. Recommended orders of refinement  
for domains and essential boundaries



### If you get it wrong, that's alright!

**FreeHyTE – Heat HTTE** automatically corrects the orders of the domain bases such as to ensure that the system is kinematically indeterminate. However, we still recommend using the orders listed in Table 1.

*p-refinement for the solid structure.* For the solid structure presented in Section 4.3.1, two levels of p-refinement are used. They must be chosen in accordance to the levels of h-refinement described in Section 4.4.1, being lower for the finer mesh (Figure 5a) and higher for the coarser (Figure 5b). Accordingly,  $n_{\Gamma} = 5$  and  $n_D = 12$  is the choice for the finer mesh and  $n_{\Gamma} = 10$  and  $n_D = 25$  is the choice for the single-element mesh.

*p-refinement for the hollow beam structure.* Two levels of p-refinement are also used for the hollow beam structure presented in Section 4.3.2. The finer h-refinement (Figure 10a)

corresponds to orders of boundary and domain bases of  $n_{\Gamma} = 5$  and  $n_D = 10$ , while for the coarser mesh (Figure 10b),  $n_{\Gamma} = 10$  and  $n_D = 16$  was the choice.

In all cases, 20 Gauss-Legendre quadrature points were used for the computation of the solving system's coefficients.

#### 4.5.3. Automatic basis refinement – the basics

The automatic basis refinement (or ‘automatic p-adaptive refinement’, in a more scientific slang) is a way to circumvent the difficulties related to the balancing of domain and essential boundary refinements, by passing them over to the computer instead. As compared to the manual basis refinement presented in the previous section, the automatic algorithm presents the following advantages and drawbacks.

*Advantages of the automatic basis refinement:*

- user does not have to worry about calibrating domain and essential boundary refinements, meaning that from a user's perspective, the hybrid-Trefftz finite elements are just as easy to use as the conventional elements;
- **FreeHyTE – Heat HTTE** takes full advantage of the localized p-refinement capability of the hybrid-Trefftz elements and tunes the refinements of each finite element and essential boundary independently. The restriction of uniform domain and boundary refinements present in the manual refinement mode is thus dropped;
- advanced stability control algorithms should help avoid the divergence of the refinement process, meaning that **FreeHyTE – Heat HTTE** is able to detect when numerical issues (typically related to ill-conditioning of the solving system) kick in and will do its best to solve them. Should it not manage to find a solution, **FreeHyTE – Heat HTTE** is designed to return the last stable results.

*Drawbacks of the automatic basis refinement:*

- the quest for the best domain and boundary refinements is an iterative process. Consequently, it is typically *much slower* than running a single analysis using some large refinements taken from Table 1;
- the automatic p-refinement typically fails when the original problem involves  $C^0$  temperature discontinuities (or ‘jumps’). It should be noted that such situations are physically ill-posed problems, so you probably shouldn't try solving them in the first place, but if you do it anyway, you'll get better results by using some large,

uniform, domain refinement, coupled with some low, uniform, boundary refinement, than by using the automatic refinement algorithm.



**Manual basis refinement should be the first line option.** The automatic refinement option should be reserved for those cases where the refinements in Table 1 fail to produce decent results and you just don't want to refine the mesh. Never use automatic p-refinement to solve problems involving discontinuous temperature fields!

See Section 5.7.3 for an illustrative example of a situation where the automatic refinement is worth being used.

#### 4.5.4. Automatic basis refinement – selection criteria

Two refinement criteria are currently implemented into **FreeHyTE – Heat HTTE** to identify the essential boundary (or boundaries) where a basis refinement is deemed necessary. One criterion is local and is related to a (boundary specific) error indicator. The other criterion is global and has to do with the thermal energy variations from one increment to the next.

*Error density reduction on the boundary.* This is a local selection criterion – you opt to refine the essential boundary where doing so **maximizes the improvement of the temperature compatibility** between the two elements adjacent to that boundary (if the boundary is interior), or between the applied temperature and its finite element prediction (if the boundary is exterior).

*Increment of the thermal energy.* This is a global selection criterion – the essential boundary whose refinement causes **the largest thermal energy increment between two successive iterations** is chosen for refinement.

At each refinement iteration, all essential boundaries are ranked according to the selection criterion chosen by the user and the highest-ranked is always selected for refinement. Besides the selection criterion, the user is expected to define the **selection tolerance** (`SelectionTol`<sup>2</sup>) to control **multiple boundary refinement**. Multiple boundary refinement is useful to avoid situations where more boundaries have the same score for

---

<sup>2</sup> Parameters written in `ThisFont` must be defined by the user. The full list of user-supplied parameters for the automatic basis refinement is given in Section 4.5.6.

the selection criterion and only one of them is selected for refinement (a typical case occurs in symmetric setups). By setting `SelectionTol` smaller than 1.0, all boundaries whose refinement criteria are larger than

$$RC_{\text{critical}} = RC_{\text{max}} \cdot \text{SelectionTol}$$

are also listed for refinement. In the above expression,  $RC_{\text{max}}$  is the maximum value of the refinement criterion and  $RC_{\text{critical}}$  the threshold value of the same criterion below which a boundary is not selected for refinement. The default value for `SelectionTol` is 0.99.

#### 4.5.5. Automatic basis refinement – stopping criteria

Three types of stopping criteria are implemented to control the execution of the basis refinement algorithm: the convergence criteria, the maximum order criterion and the system instability criteria. A brief description of each criterion follows.

*Convergence criteria.* This is the most desirable stopping criterion, essentially meaning that the solution convergence has been reached. The user must choose between the **thermal energy convergence criterion** and the **maximum boundary error criterion**. Besides convergence under one of these criteria, the execution may stop if **all available refinements fail to improve the solution**.

- the **thermal energy convergence** is achieved when the average value of the relative energy variation taken over the last `AvgNVal` iterations is below a certain threshold, `TargetErrorNorm`. Both the threshold and the number of iterations used to compute the average are defined by the user;
- in the **maximum boundary error criterion**, the convergence is achieved when the average error density reduction (as defined in Section 4.5.4) taken over the last `AvgNVal` iterations is less than `TargetErrorNorm` times its value in the **first** iteration. Essentially, this means that the algorithm stops when the temperature continuity correction in the current iteration is considered negligible in respect to its value in the first iteration;
- in rare cases, all available refinements may fail to reduce the error density, as defined in Section 4.5.4. This happens when the error density reduction falls short of some user-defined numerical zero threshold (`thresh`) on all essential boundaries, and essentially means that any refinement the algorithm may produce fails to improve the solution in the current iteration, so **FreeHyTE – Heat HTTE** is unable to determine which essential boundary should be refined next. **This is typically the case when the exact solution has been achieved by the**

**algorithm**, so further refinement is futile. However, its occurrence in other (very rare) situations can neither be overruled, nor predicted. User is advised to simply rerun the analysis using different initial refinements if this situation is reached without convergence.

*Maximum order criterion.* Increasing the order of p-refinement is the best way to improve the quality of the solution when using hybrid-Trefftz finite elements. However, as a thumb rule, **the larger the order of the approximation functions, the tougher their numerical handling becomes**. In order to avoid the numerical problems caused by exceedingly large orders of approximation functions, user should define their maximum allowable order (`MaxOrder`). If `MaxOrder` is reached, the execution ends and the most refined solution is plotted. This point is typically reached if the convergence threshold (`TargetErrorNorm`) is set too small, but may also mean that the problem is physically ill posed, as is the case, for instance, of  $C^0$  discontinuous temperature fields.

*System instability criteria.* As shown in Section 3.2.2, one of the drawbacks of the hybrid-Trefftz finite elements is their proneness to yield numerically unstable solving systems. The most well-known symptom of system instability is the ill-conditioning, that is, the near linear-dependence between sets of equations in the solving system. Controlling the condition number is rather straightforward, but unfortunately insufficient in many cases, so new approaches to the evaluation of the system's stability had to be designed. **FreeHyTE – Heat HTTE** performs a thorough assessment of the regularity of the singular values of the system and **identifies those singular values that break the regular distribution pattern** (the outliers). This situation can lead to a forced exit under the following circumstances:

- it occurs in the first analysis of an automatic basis refinement process. Before the iterative process starts, a preliminary analysis of the structure is made using the starting (and uniform) refinement orders supplied by the user. If singular value outliers are detected in this analysis, the iterative process does not even start and **FreeHyTE – Heat HTTE** exits with an error message;
- it occurs in more than `MaxOutlierIter` successive iterations, despite the stability improvement measures that **FreeHyTE – Heat HTTE** takes to improve the solving system. Parameter `MaxOutlierIter` must be supplied by the user.

#### 4.5.6. Automatic basis refinement – user input

The parameters requested from the user to calibrate the automatic basis refinement are listed in Table 2, along with some recommended values.

Code	Role	Recomm. values
SelectionTol	Tolerance for choosing multiple boundaries	0.99
thresh	Threshold for numerical zero	1.E-12
MaxOutlierIter	Maximum consecutive iterations with singular values outliers	10
MinIter	Minimum number of iterations to run (to avoid spurious early convergence)	5
AvgNVal	Number of iterations to average the relative energy variation or boundary error reduction	3
TargetErrorNorm	Convergence threshold (if using the thermal energy convergence criterion)	1.E-4
TargetErrorNorm	Convergence threshold (if using the maximum boundary error criterion)	1.E-2
MaxOrder	Maximum allowable refinement order	20

Table 2. Parameters for calibrating the automatic refinement process and recommended values

## 4.6. BOUNDARY CONDITIONS

### 4.6.1. General definition

Two types of boundary conditions can be defined in **FreeHyTE – Heat HTTE**:

- **Dirichlet boundary conditions**, where the **boundary temperatures** are specified; and,
- **Neumann boundary conditions**, where the **boundary-normal heat fluxes** are specified.

No Robin (e.g. convective) boundary conditions are currently implemented in **FreeHyTE – Heat HTTE**.



All **boundary conditions** are applied to the exterior edges of the structure. They are **defined by polynomials of arbitrary degrees**.

The positive directions of the normal heat flux boundary conditions are in accordance with the side referentials defined in Figure 1.

The definition of a boundary condition is made by **specifying its values in as many equally spaced points along the boundary as needed** to define its polynomial variation. The first point of the sequence must correspond to the beginning of the side (according to its orientation specified in Figure 1) and the last point must correspond to the final point of the side.

Consider, for example, that on edge 4 of the structure presented in Figure 1 a heat flux is applied in the normal direction, as shown in Figure 11. The flux is defined by a polynomial of degree  $N$ . In **FreeHyTE – Heat HTTE**, such flux is input as a list containing its values in  $N+1$  equally spaced points along side 4, including the first and the last, for instance,

$$f(t_4) \equiv (f_1 \quad f_2 \quad f_3 \quad \cdots \quad f_N \quad f_{N+1})$$

The values are given in the order prescribed by the orientation  $t_4$  of the side, and the signs are in accordance with the orientation of the normal axis,  $n_4$  (meaning, for instance, that  $f_1$  to  $f_3$  are positive and  $f_N$  and  $f_{N+1}$  are negative).

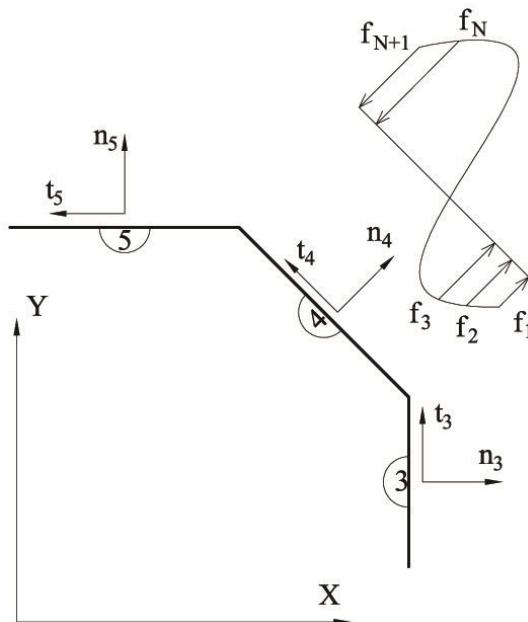


Figure 11. Normal polynomial flux applied to side 4

If the boundary condition is constant, it is, of course, sufficient to specify its value in a single point.

#### 4.6.2. Boundary conditions for the solid and hollow bodies

Following the recipes given in Section 4.6.1, the boundary conditions for the structures defined in Sections 4.3.1 (Figure 2) and 4.3.2 (Figure 3) are listed in Table 3.

Location	Boundary type	Solid body	Hollow body
Left boundary, $X = 0$	Dirichlet	100	100
Left boundary, $X = 6$	Dirichlet	100	100
Bottom boundary, $Y = 0$	Neumann	0	0
Top boundary, $Y = 3$	Neumann	-60 to 0 (linear)	-60 to 0 (linear)
All edges of the opening	Dirichlet	-	100

Table 3. Boundary conditions for the solid and hollow bodies

## 5. GRAPHICAL USER INTERFACE

### 5.1. INTRODUCTION

This chapter describes the Graphical User Interface (GUI) implemented in **FreeHyTE – Heat HTTE**, using the examples presented in Chapter 4 to illustrate its behaviour.

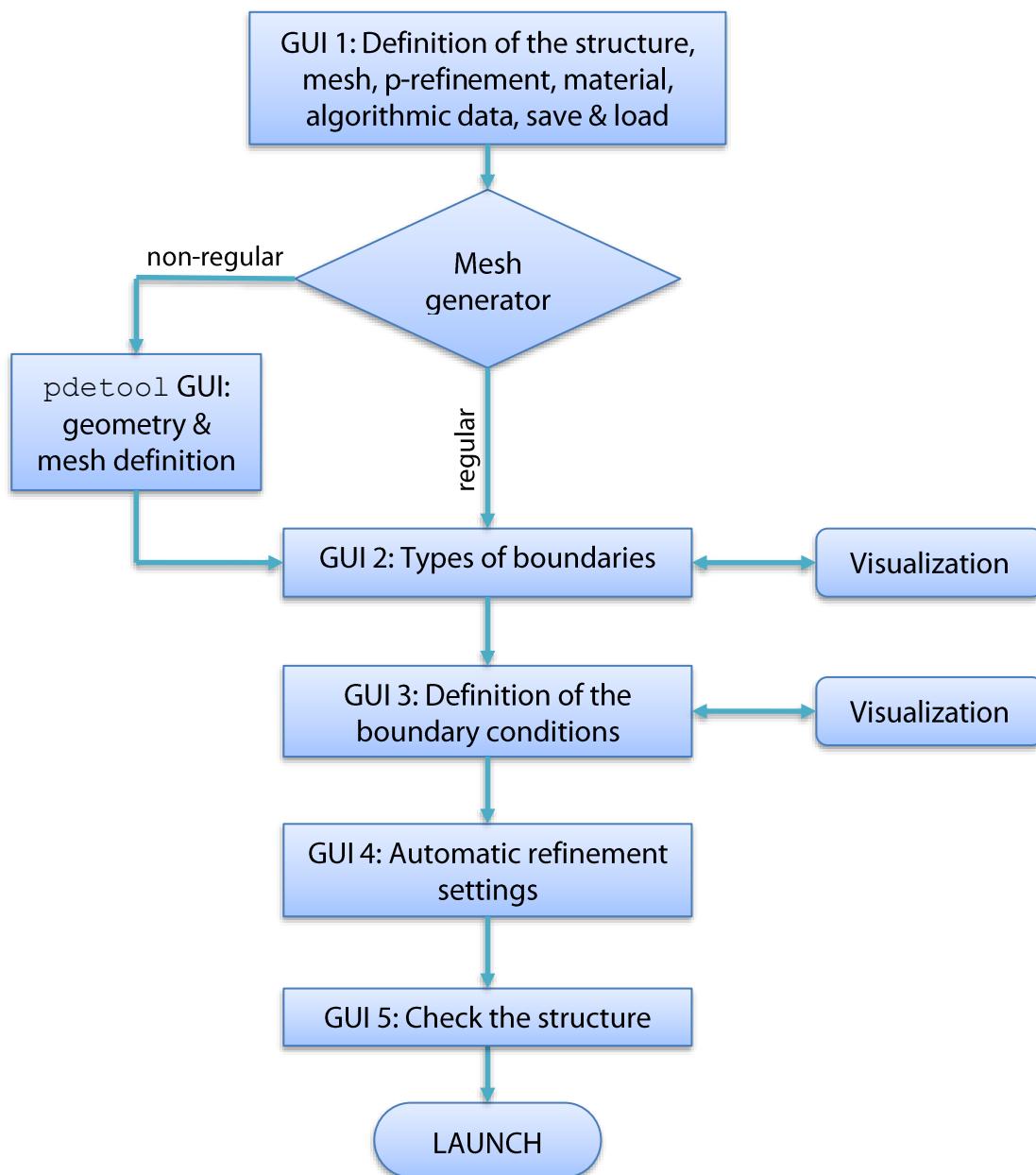


Figure 12. GUI flowchart

The general structure of the GUI is presented in Figure 12, in a flowchart format. It consists of five main GUI, complemented by the `pdetool` GUI for the definition of non-regular bodies and meshes, and an optional visualization interface to assist with the definition of boundary conditions.

Free sequential navigation is supported between interfaces, in both directions. Upon exiting an interface and moving to the next, **FreeHyTE – Heat HTTE** writes the data acquired from the ended interface to a `*.mat` file. As a consequence,



- you must run **FreeHyTE – Heat HTTE** from a folder where you have writing rights; and,
- when **FreeHyTE – Heat HTTE** is ran from the source (\*.m) files, it is possible to start its execution from *any point* of the flowchart in Figure 12, provided you want to reuse all data from the interfaces you skipped.

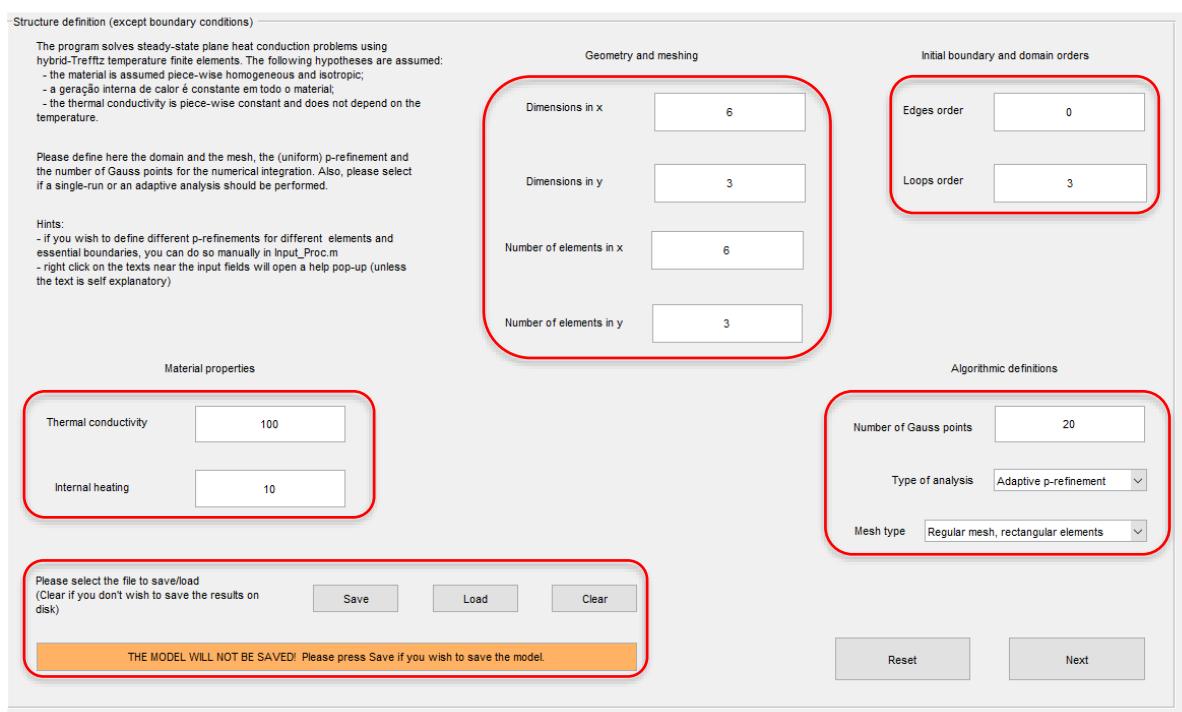
The latter capability is not available when you run **FreeHyTE – Heat HTTE** as a Matlab APP, but since data from the previous run is always loaded at the launching of an interface, reusing it without changes should be possible by just clicking the **Next** button (note that in this chapter the buttons are designated by their label in a small button frame).

The five main GUI and the visualization interface are described in the next sections. The `pdetool` GUI was described in Section 4.4.2 and is not revisited here.

## 5.2. GUI 1: STRUCTURAL AND ALGORITHMIC DEFINITIONS

The first GUI is used to define the structure's geometry, h- and p-refinements, material characteristics, algorithmic options, to save the current model and to load previously saved models.

A typical configuration of GUI 1 as **FreeHyTE – Heat HTTE** is initiated is presented in Figure 13. The main data zones of the interface are identified with red frames. Each of these zones is briefly described below.



Structure definition (except boundary conditions)

The program solves steady-state plane heat conduction problems using hybrid-Trefftz temperature finite elements. The following hypotheses are assumed:

- the material is assumed piece-wise homogeneous and isotropic;
- a geração interna de calor é constante em todo o material;
- the thermal conductivity is piece-wise constant and does not depend on the temperature.

Please define here the domain and the mesh, the (uniform) p-refinement and the number of Gauss points for the numerical integration. Also, please select if a single-run or an adaptive analysis should be performed.

Hints:

- If you wish to define different p-refinements for different elements and essential boundaries, you can do so manually in Input\_Proc.m
- right click on the texts near the input fields will open a help pop-up (unless the text is self explanatory)

Geometry and meshing

Initial boundary and domain orders

Material properties

Algorithmic definitions

THE MODEL WILL NOT BE SAVED! Please press Save if you wish to save the model.

Figure 13. Layout of GUI 1

### 5.2.1. General features of the interface

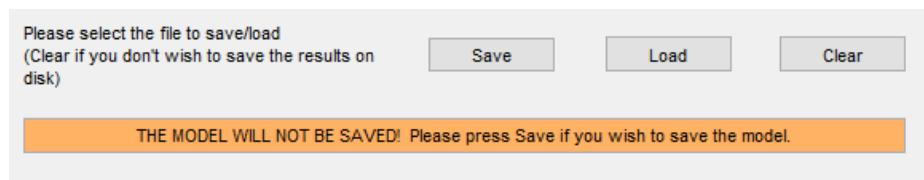
- GUI 1 is **pre-filled with the data from the last run** before it is made visible, if such data is available in the working folder. The **Reset** button deletes pre-filled data;
- hovering the mouse over the text accompanying the edit boxes opens up **context help** explaining in more detail what input is expected from the user;
- edit boxes are protected from incorrect input. Inserting data of incorrect type will open an error window explaining what data is acceptable for that field and prompting the user to correct it. However, you should be minimally careful when feeding in the data, since bullet-proofing the data checks was not on our plans.

### 5.2.2. Saving and loading

**FreeHyTE – Heat HTTE** offers **two levels of save & load**, namely an implicit level and an explicit level.

*Implicit save & load.* Implicit saving and loading is an automatic procedure performed without user's intervention. Each time the **Next** button is clicked in a GUI, the data input by the user is stored in some default \*.mat files which remain in the working folder. As you load a GUI, the data from the corresponding \*.mat file is implicitly loaded.

*Explicit save & load.* Only the data from the last run is implicitly saved. This means that if you define another model, the data from the previous model is overwritten. If you wish to store the data from a model, you need to use the **Save** button in GUI 1. **FreeHyTE – Heat HTTE** prompts you to specify the saving folder and the file name, and changes the message from the text box in the lower left side of GUI 1 from,



to,



using the path and the name specified for your model. If you choose to save the model, the data from each GUI is stored not only in the default files, for the implicit loading procedure, but also in the \*.mat file specified by the user. Use the **Clear** button if you wish to cancel the saving of the model.



**Explicitly saving the model also saves a file with the values of the temperatures and heat fluxes in the Gauss points of each element.** This file is named *MyModelName\_NDxx\_NByy.dat*, where **xx** is the domain basis refinement and **yy** is the boundary basis refinement, if a

single run was executed, or *My modelName\_ltxx.dat*, where **xx** is the number of the last iteration, for automatic basis refinement analyses.

The output file is formatted for direct loading in the post-processing software [Tecplot](#), but can be used in other visualization software as well since what it contains is just a list of points and temperature/flux values. **The results are not saved in an output file if you choose not to save your model, but plots of temperatures and heat fluxes are always shown as the analysis is completed.**

Loading a model overwrites the default \*.mat files of each GUI with the values from the save file.



**Modifying the data from a saved model** in a GUI and running it with the new data **does not overwrite the old save file**. If you wish to store the changes, you must use the **Save** button again.

**The mesh data is saved** with both explicit and implicit processes, **enabling the reuse of the same mesh in subsequent runs. However, the saved mesh data is not sufficient for modifying the mesh in pdetool**. If you wish to store the mesh information such as to be possible to operate on it using pdetool, please do so from the pdetool GUI, as explained in Section 4.4.2.

The **Save** and **Load** buttons are only available in GUI 1.

### 5.2.3. Data input in GUI 1

*Algorithmic definitions.* Choose between the regular and non-regular geometry and mesh generators, and fixed (single run) or automatic (iterative) basis refinements. For details on the automatic basis refinement option, the reader is directed to Sections 4.5.3 to 4.5.6. Additionally, user must specify the number of Gauss-Legendre quadrature points to perform the integrations needed to compute the conductivity matrix. Less Gauss-Legendre points decrease the duration of the analysis, but increase the numerical integration errors. For some guidance on choosing the number of Gauss-Legendre points, please see Section 4.5.2.

*Geometry and meshing.* This area is **only editable if the regular mesh generator is used** to define the structure. Its fields correspond to the geometrical and mesh data  $D_x$ ,  $D_y$ ,  $N_x$  and  $N_y$ , detailed in Section 4.4.1. **If the non-regular mesh generator is chosen in the algorithmic definitions, pdetool is launched automatically** when you click the **Next** button in GUI 1 (see Section 4.4.2 for details on its usage).



If you selected the non-regular mesh generator and wish to **reuse the mesh data from the previous run** (or from a loaded file) rather than defining a new mesh, **just close the pdetool GUI immediately after it opens and proceed to the next GUI.**

*Boundary and domain orders.* For **fixed basis refinement analyses**, insert the (uniform) orders of p-refinement on the essential boundaries and in the domain of the elements. For guidance on how to choose these orders, please consult Section 4.5.2 of this manual. For **automatic basis refinement analyses**, insert the (uniform) **initial** orders of p-refinement on the essential boundaries and in the domain of the elements. Some small orders taken from Table 1 should be alright.

*Material parameters.* Specify the values of the thermal conductivity and internal heat generation in the body.

#### 5.2.4. GUI 1 data for the solid and hollow bodies

The GUI 1 data for the beam problems defined in Sections 4.3.1 (Figure 2) and 4.3.2 (Figure 3) are listed in Figure 14 and Figure 15, respectively. Both sets of data refer to the finer meshes shown in Figure 5a and Figure 10a, respectively, and to the fixed refinement analysis.

Structure definition (except boundary conditions)

The program solves steady-state plane heat conduction problems using hybrid-Trefftz temperature finite elements. The following hypotheses are assumed:  
- the material is assumed piece-wise homogeneous and isotropic;  
- a geração interna de calor é constante em todo o material;  
- the thermal conductivity is piece-wise constant and does not depend on the temperature.

<p>Please define here the domain and the mesh, the (uniform) p-refinement and the number of Gauss points for the numerical integration. Also, please select if a single-run or an adaptive analysis should be performed.</p> <p>Hints:  - if you wish to define different p-refinements for different elements and essential boundaries, you can do so manually in Input_Proc.m  - right click on the texts near the input fields will open a help pop-up (unless the text is self explanatory)</p>	<b>Geometry and meshing</b> Dimensions in x: <input type="text" value="6"/> Dimensions in y: <input type="text" value="3"/> Number of elements in x: <input type="text" value="6"/> Number of elements in y: <input type="text" value="3"/>	<b>Initial boundary and domain orders</b> Edges order: <input type="text" value="5"/> Loops order: <input type="text" value="13"/>
---	---	--

Material properties

Thermal conductivity: <input type="text" value="100"/> Internal heating: <input type="text" value="10"/>	<b>Algorithmic definitions</b> Number of Gauss points: <input type="text" value="20"/> Type of analysis: <input type="button" value="Single run"/>
---	--

Please select the file to save/load  
(Clear if you don't wish to save the results on disk)

THE MODEL WILL NOT BE SAVED! Please press Save if you wish to save the model.

Figure 14. GUI 1 data for the solid structure (finer mesh)

Structure definition (except boundary conditions)

The program solves steady-state plane heat conduction problems using hybrid-Trefftz temperature finite elements. The following hypotheses are assumed:  
- the material is assumed piece-wise homogeneous and isotropic;  
- a geração interna de calor é constante em todo o material;  
- the thermal conductivity is piece-wise constant and does not depend on the temperature.

<p>Please define here the domain and the mesh, the (uniform) p-refinement and the number of Gauss points for the numerical integration. Also, please select if a single-run or an adaptive analysis should be performed.</p> <p>Hints:  - if you wish to define different p-refinements for different elements and essential boundaries, you can do so manually in Input_Proc.m  - right click on the texts near the input fields will open a help pop-up (unless the text is self explanatory)</p>	<b>Geometry and meshing</b> Dimensions in x: <input type="text" value="6"/> Dimensions in y: <input type="text" value="3"/> Number of elements in x: <input type="text" value="1"/> Number of elements in y: <input type="text" value="1"/>	<b>Initial boundary and domain orders</b> Edges order: <input type="text" value="5"/> Loops order: <input type="text" value="9"/>
---	---	---

Material properties

Thermal conductivity: <input type="text" value="100"/> Internal heating: <input type="text" value="10"/>	<b>Algorithmic definitions</b> Number of Gauss points: <input type="text" value="20"/> Type of analysis: <input type="button" value="Single run"/>
---	--

Please select the file to save/load  
(Clear if you don't wish to save the results on disk)

THE MODEL WILL NOT BE SAVED! Please press Save if you wish to save the model.

Figure 15. GUI 1 data for the hollow structure (finer mesh)

### 5.3. GUI 2: DEFINITION OF THE BOUNDARY TYPES

The second GUI is used to define the type (Dirichlet or Neumann) for each exterior side of the structure.

The configuration of GUI 2 for the finer mesh of the solid body problem is presented in Figure 16. The main data zones of the interface are identified with red frames. Each of these zones is briefly described below.

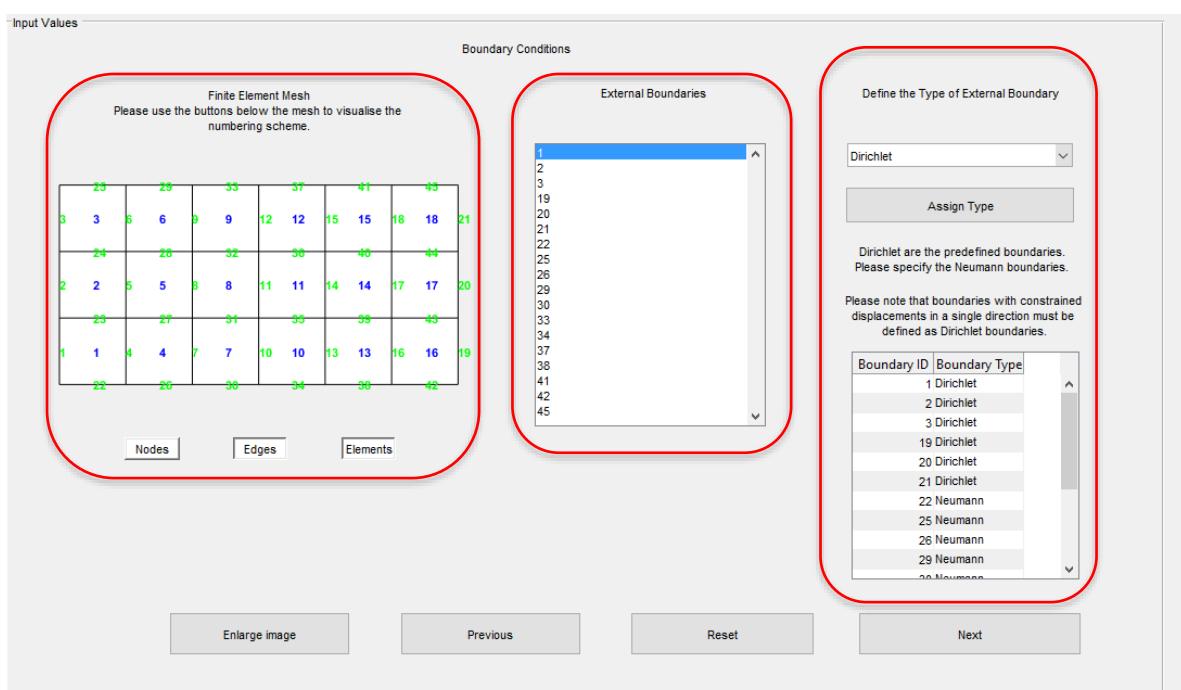


Figure 16. Layout of GUI 2

#### 5.3.1. Structure visualization zone

The structure visualization zone is located on the left side of GUI 2. It consists of an interactive plot of the structure with three buttons controlling what information should be displayed in the plot. In the case presented in Figure 16, this information consists of the edge and element numbers, as the respective buttons are pressed. Un-pressing a button deactivates the information associated to it.

In some cases, the visualization area in GUI 2 may be too small to support a clear read of the structural data. If this is the case, the **Enlarge image** button can be used to open a

separate, visualization-only interface, with zoom, pan and data cursor capabilities, where the structure can be visualized with any degree of detail (Figure 17).

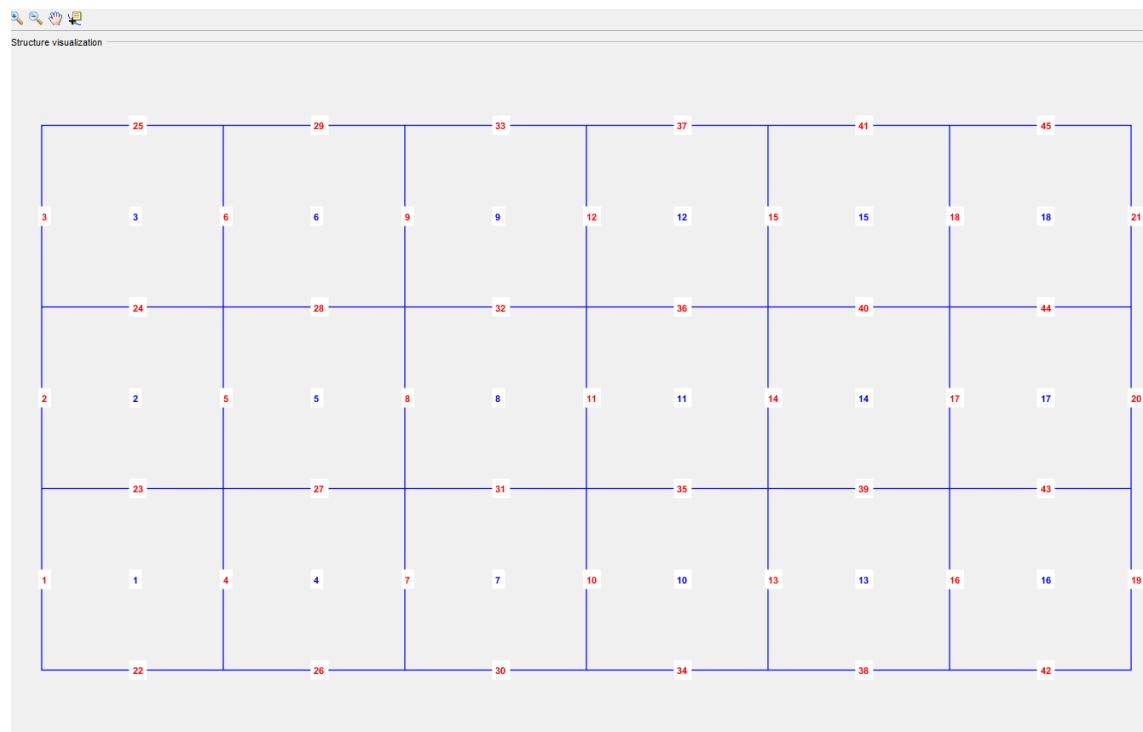


Figure 17. Layout of the visualization interface

### 5.3.2. Definition of the external boundary types

The external boundaries of the structure are listed in the *External boundaries* table, in the central zone of GUI 2. The boundary types are listed in the table situated in the right side of the interface.



At startup, **GUI 2 pre-loads the boundary types that were auto-saved in the previous run or loaded from a save file if no changes were made to the mesh in GUI 1.** If changes were made, all boundary types are reset as Dirichlet.

To define the boundary types, user must select the boundaries in the *External boundaries* table (multiple selection is possible), select the desired boundary type from the pop-up menu on the right side of the interface and click the **Assign type** button. The boundary types in the table on the right should automatically adjust to reflect the changes. Press the **Next** button to move to GUI 3 or the **Previous** button to return to GUI 1.

## 5.4. GUI 3: DEFINITION OF THE BOUNDARY CONDITIONS

The third GUI is used to define the boundary conditions according to the boundary types defined in the previous interface. A more insightful discussion on the definition of the boundary conditions in **FreeHyTE – Heat HTTE** is given in Section 4.6, so this section is restricted to the presentation of the interface used to input those definitions.

The configuration of GUI 3 for the finer mesh of the solid body problem is presented in Figure 18. The main data zones of the interface are identified with red frames. Each of these zones is briefly described below.

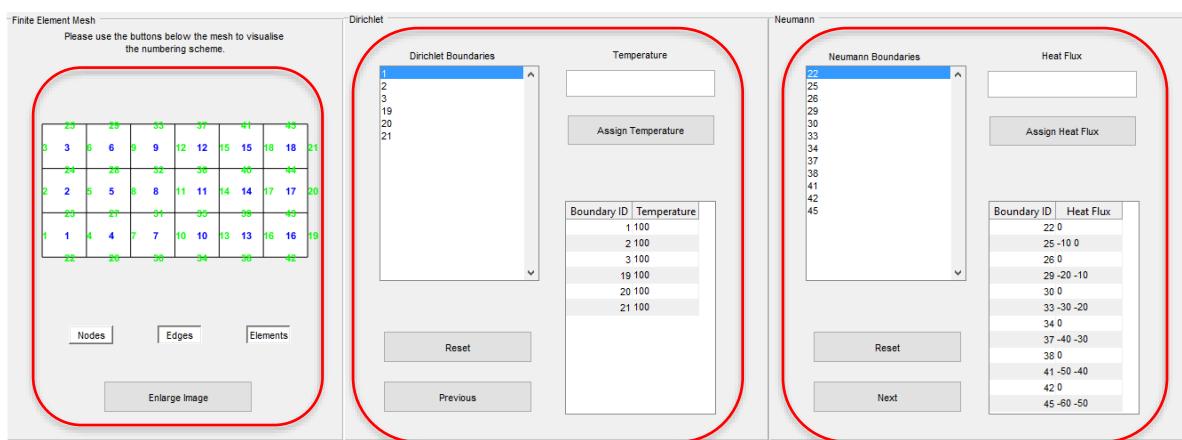


Figure 18. Layout of GUI 3 for the solid body problem

The structure visualization zone is identical to that of GUI 2, as presented in Section 5.3.1.

The centre and right input zones are used to define the Dirichlet and Neumann boundary conditions, respectively.



At startup, **GUI 3 pre-loads the boundary conditions that were auto-saved in the previous run or loaded from a save file if no changes were made to the mesh in GUI 1, nor to the boundary types in GUI 2.** If changes were made, all boundary conditions are predefined as zero.

Regarding the definition of the Dirichlet boundary conditions, the exterior boundaries specified as Dirichlet in GUI 2 are listed in the *Dirichlet boundaries* table. In this table, the

user must select the boundary where he/she wishes to specify the enforced temperature (multiple selection is possible). Finally, the applied temperatures are specified in the *Temperature* area, according to the procedure detailed in Section 4.6.1. **If the field to be applied is not constant, its values along the boundary must be inserted all at once, separated by space** (comma also works). Press the **Assign temperature** button when you are done. The temperature values in the table below should update automatically.

The procedure for the definition of the Neumann boundary conditions is exactly the same as for the Dirichlet boundary conditions. On boundaries 25, 29, 33, 37, 41 and 45, the linearly distributed normal flux (see Figure 2) is specified using its values at the beginning and at the end of each side (see Figure 18).

For completeness, the configuration of GUI 3 for the finer mesh of the hollow beam problem is also presented in Figure 19.

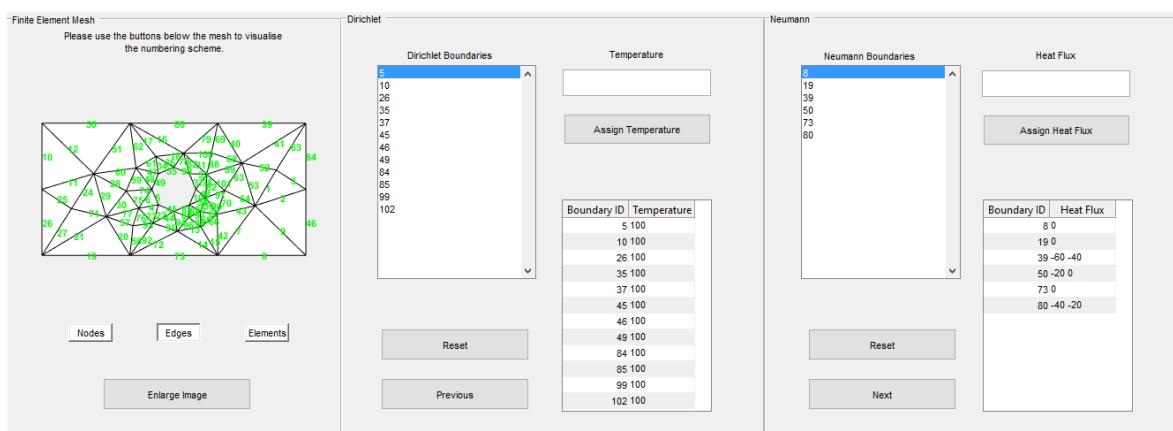


Figure 19. Layout of GUI 3 for the hollow body problem

After completing the definition of the boundary conditions, press the **Next** button to move to the next GUI or the **Previous** button to return to GUI 2.

## 5.5. GUI 4: AUTOMATIC REFINEMENT

GUI 4 only emerges if an adaptive basis refinement analysis was requested in GUI 1. It is used to define the adaptive procedures and parameters that steer the iterative basis refinement process described in Sections 4.5.3 to 4.5.6. Besides choosing the desired selection criterion (see Section 4.5.4) and stopping criterion (see Section 4.5.5), user is asked to input the tuning parameters listed in Table 2 (see Section 4.5.6).

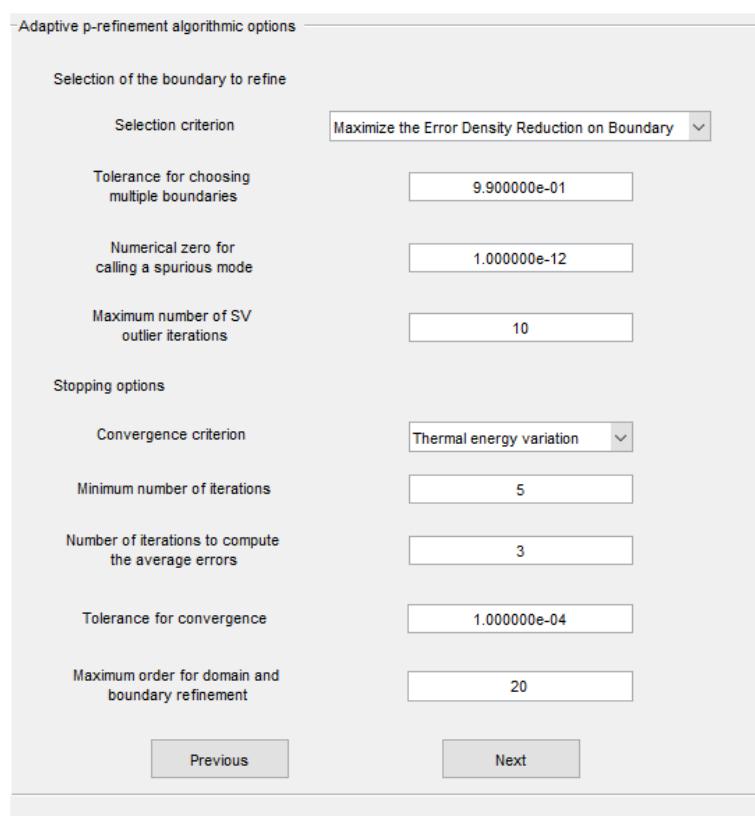


Figure 20. Layout of GUI 4

The layout of GUI 4 is presented in Figure 20.



At startup, **GUI 4 pre-loads the data that was auto-saved in the previous run or loaded from a save file**. If there is no such data in the working folder, it uses the recommended data, as specified in Table 2.

To assist the user, hovering the mouse over the text accompanying the edit boxes opens up **context help** explaining in more detail what input is expected. Edit boxes are protected from incorrect input. Inserting data of incorrect type will open an error window explaining what data is acceptable for that field and prompting the user to correct it. As stated before, however, you should be minimally careful when feeding in the data, since the data checks are not bullet-proofed.



## 5.6. VERIFICATION GUI

Verification GUI is the last stop before launching the execution of **FreeHyTE – Heat HTTE** calculation module. It is meant to allow user to verify the definitions of the structure and boundary conditions, and features a simple interface with a single pop-up menu to choose from the visualizations of the mesh numbering and boundary conditions.

The interface is launched with the mesh numbering visualization on, as presented in Figure 21 for the finer mesh of the hollow beam problem.

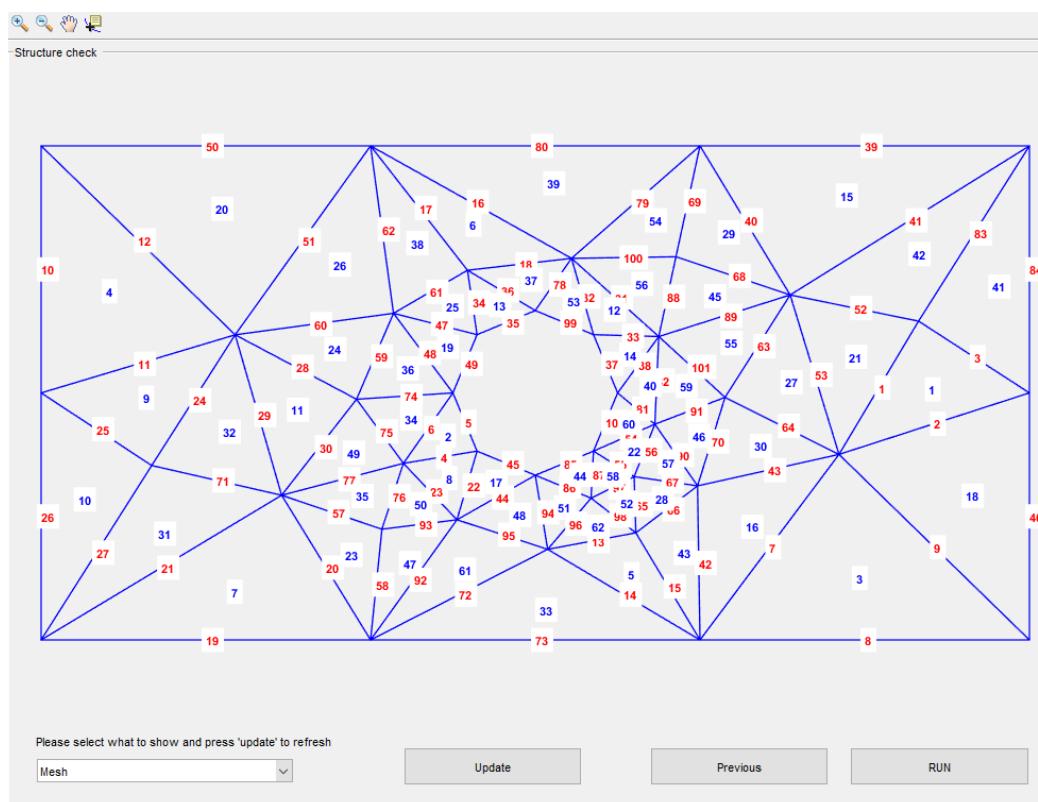


Figure 21. Visualization GUI in the mesh numbering mode

Selecting the *Boundary conditions* option in the pop-up menu and then pressing the **Update** button triggers the boundary conditions visualization mode, where the enforced boundary values are plotted at the beginning and the end of each exterior side. **Dirichlet sides** are plotted in **black**, while **Neumann sides** are plotted in **red**. The visualization interface for the boundary conditions of the mesh presented in Figure 21 is shown in Figure 22. Press the **RUN** button to launch the analysis of the model.

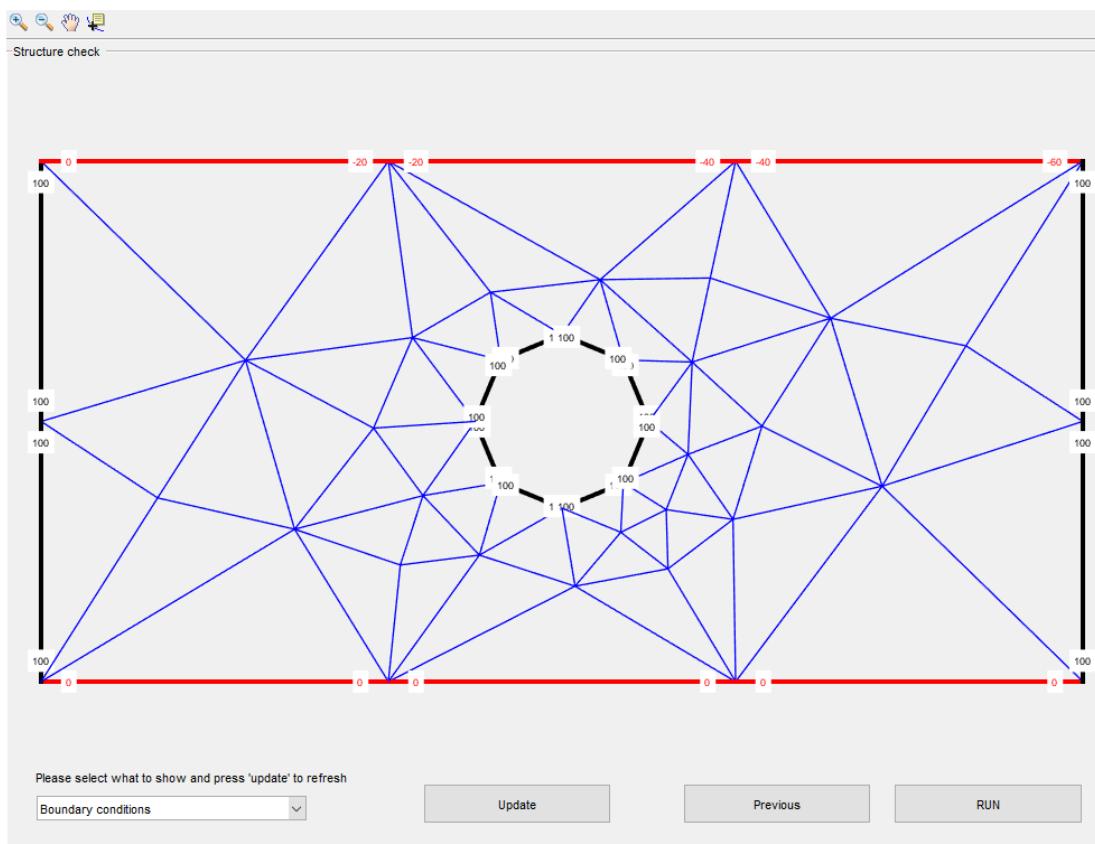


Figure 22. Visualization GUI in the boundary conditions mode

## 5.7. POST-PROCESSING

Post-processing in **FreeHyTE – Heat HTTE** is limited to the writing of the output data file (if the user chooses to save the model, see Section 5.2.2), the plotting of the analysis results and the plotting of the convergence graphs, if an automatic refinement analysis has been performed. The temperature and heat flux plots obtained for the solid and hollow bodies presented in Sections 4.3.1 and 4.3.2 are given in Sections 5.7.1 and 5.7.2, to endorse the comparison between the various h- and p-refinement options. The results for the hollow body problem, coarsely meshed, but using automatic basis refinement are presented in Section 5.7.3.

### 5.7.1. The solid body results

Two refinement schemes were used for the solution of the solid body problem, as detailed in Sections 4.4.1 and 4.5.2. The first scheme involved an 18-element mesh and refinement orders of  $n_\Gamma = 5$  and  $n_D = 12$  on the essential boundaries and in the finite element domains, respectively. The second scheme used one single finite element, but increased the refinement orders to  $n_\Gamma = 10$  and  $n_D = 25$  to compensate for the coarseness of the mesh.

The temperature and heat flux predictions as plotted by the **FreeHyTE – Heat HTTE** post-processor are presented in Figure 23 and Figure 24, respectively.

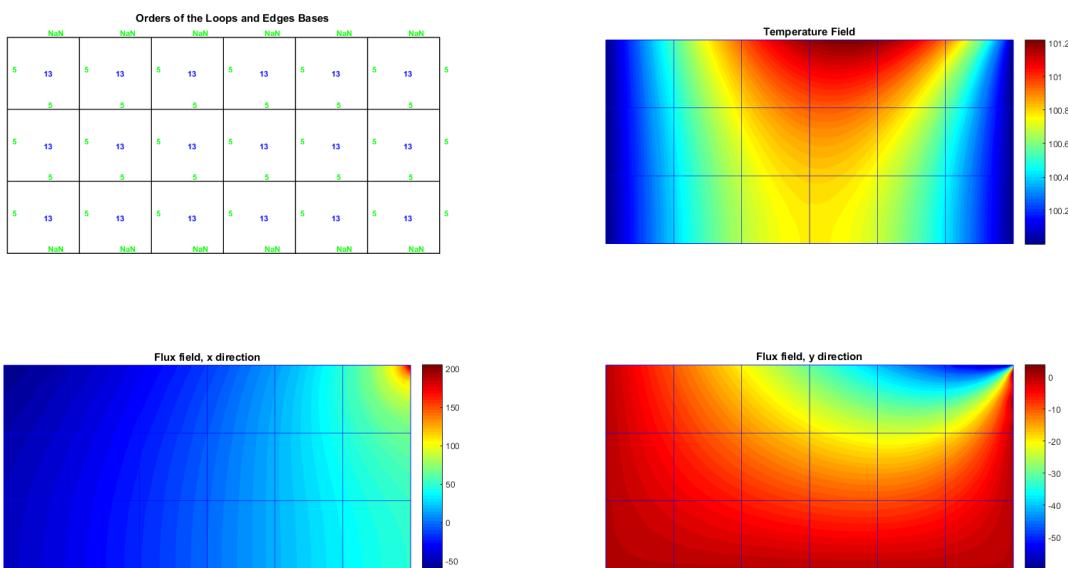


Figure 23. Temperature and heat flux plots for the solid body problem, 18 finite elements

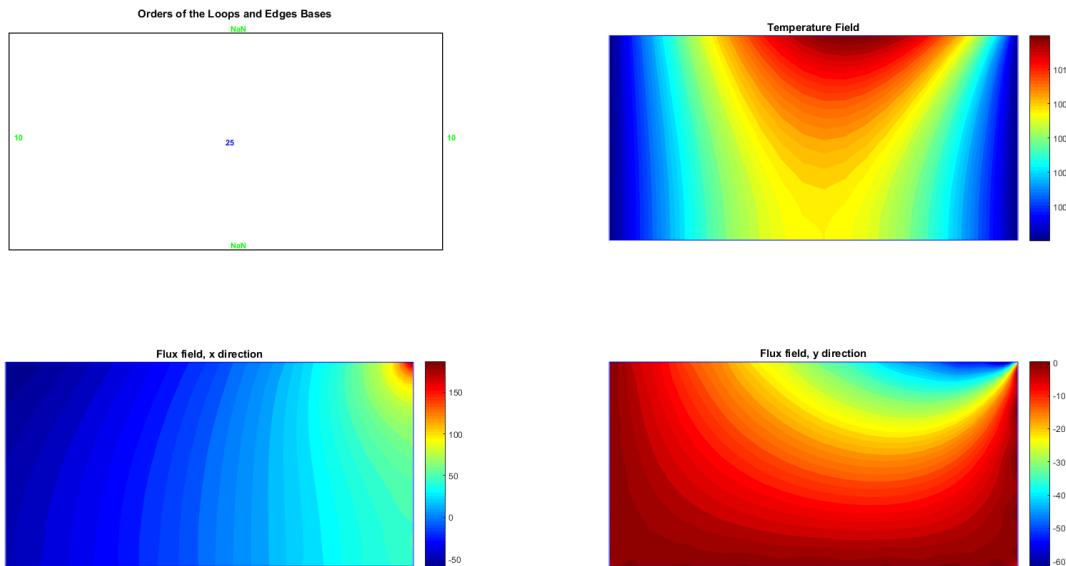


Figure 24. Temperature and heat flux plots for the solid body problem, 1 finite element

It is clear that the results are practically identical in both cases for both temperature and heat flux fields. Reader's attention is called to the smoothness in the stress transitions between neighbouring elements in Figure 23, despite having used no flux averaging in the plots. As explained in more detail in Section 3.1.1, this is one of the most important advantages in using hybrid-Trefftz finite elements.

The total analysis times were 0.59sec for the finer mesh case (with 684 degrees of freedom) and 0.10sec for the coarser mesh case (with 73 degrees of freedom). No ill-conditioning of the solving system was reported in either case.

### 5.7.2. The hollow body results

Two refinement schemes were used for the solution of the hollow body problem, as detailed in Sections 4.4.2 and 4.5.2. The first scheme involved a 62-element mesh and refinement orders of  $n_\Gamma = 5$  and  $n_D = 10$  on the essential boundaries and in the finite element domains, respectively. The second scheme used a (heavily distorted) 12-element mesh, but increased the refinement orders to  $n_\Gamma = 10$  and  $n_D = 16$  to compensate for its coarseness.

The temperature and heat flux predictions as plotted by the **FreeHyTE – Heat HTTE** post-processor are presented in Figure 25 and Figure 26, respectively.

While the temperature field predictions are practically identical for both refinements, despite the very strong mesh distortion and large element sizes present in the coarse model, more significant discontinuities are present in the heat balance between the finite elements neighbouring the internal hole, in the coarse mesh case. However, even here,

the results are usable from a practical standpoint. The vertical flux discontinuity at the upper-right corner is recovered sharply in both cases.

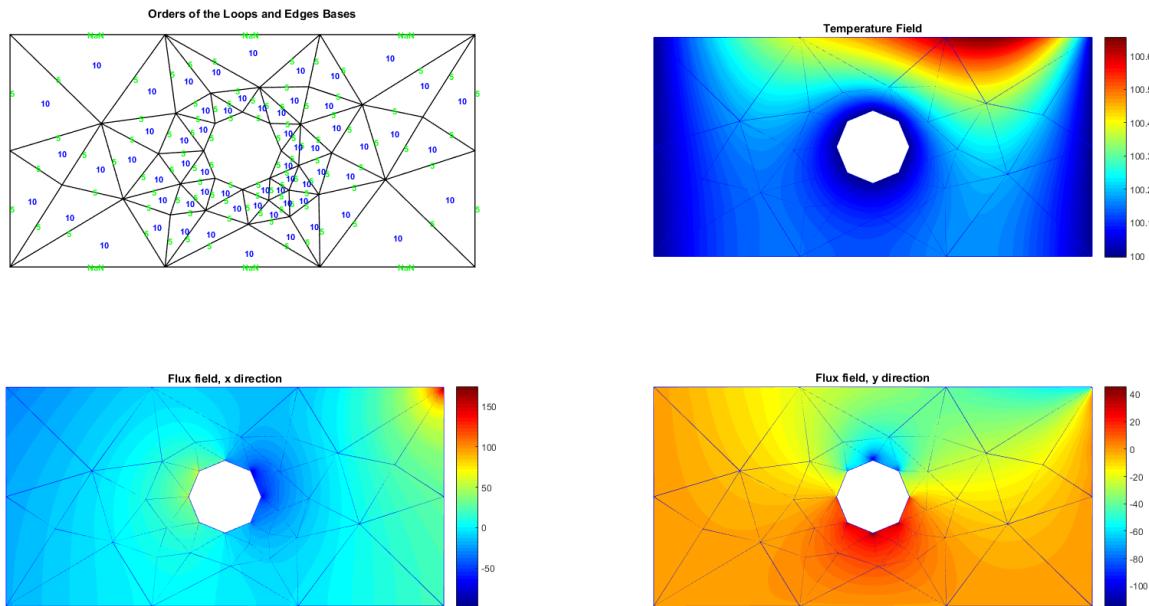


Figure 25. Temperature and heat fluxes for the hollow body problem, 62 finite elements

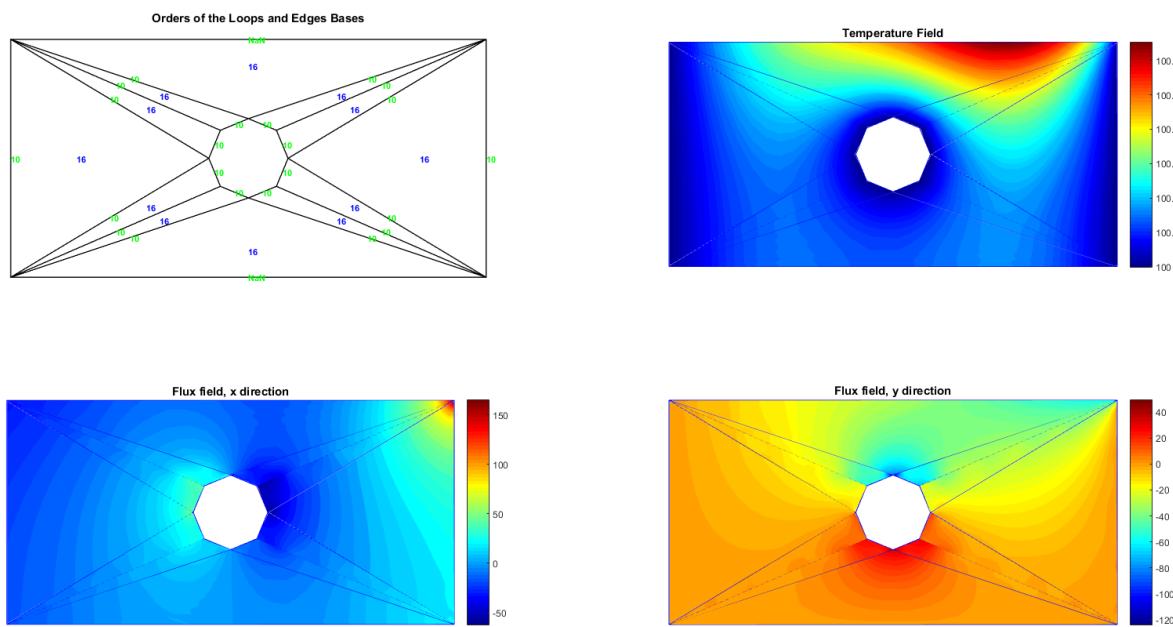


Figure 26. Temperature and heat fluxes for the hollow body problem, 12 finite elements

The total analysis times were 12.17sec for the finer mesh case (with 1878 degrees of freedom) and 0.61sec for the coarser mesh case (with 638 degrees of freedom). Outliers were detected in the singular values of the solving system in the finer mesh case (see Section 4.5.5), prompting the use of the special solver for unstable systems, which, in turn, led to a considerable increase in the execution time.

### 5.7.3. The hollow body results – adaptive basis refinement

The only unsatisfactory result obtained using the ‘manual’ refinement strategy is that of the hollow body problem, using the coarser finite element mesh. Moreover, subsequent experiences (not detailed here) showed that further increasing the uniform domain and essential boundary refinements fails to improve significantly on the solution presented in Figure 26.

In order to improve the quality of these results without refining the finite element mesh, an automatic basis refinement process is launched. The starting orders for the approximation bases in the domains and on the essential boundaries of the elements are defined as  $n_D = 3$  and  $n_\Gamma = 0$  in GUI 1. The selection criterion for the boundaries to refine is the maximization of the error density reduction (Section 4.5.4). The stopping criterion is the thermal energy convergence criterion (Section 4.5.5). All parameters for tuning the automated refinement process are defined according to the values recommended in Table 2, except for the convergence threshold (TargetErrorNorm), which is defined as  $10^{-5}$ .

The refinement orders and the temperature and heat flux solutions resulting from the iterative process are presented in Figure 27.

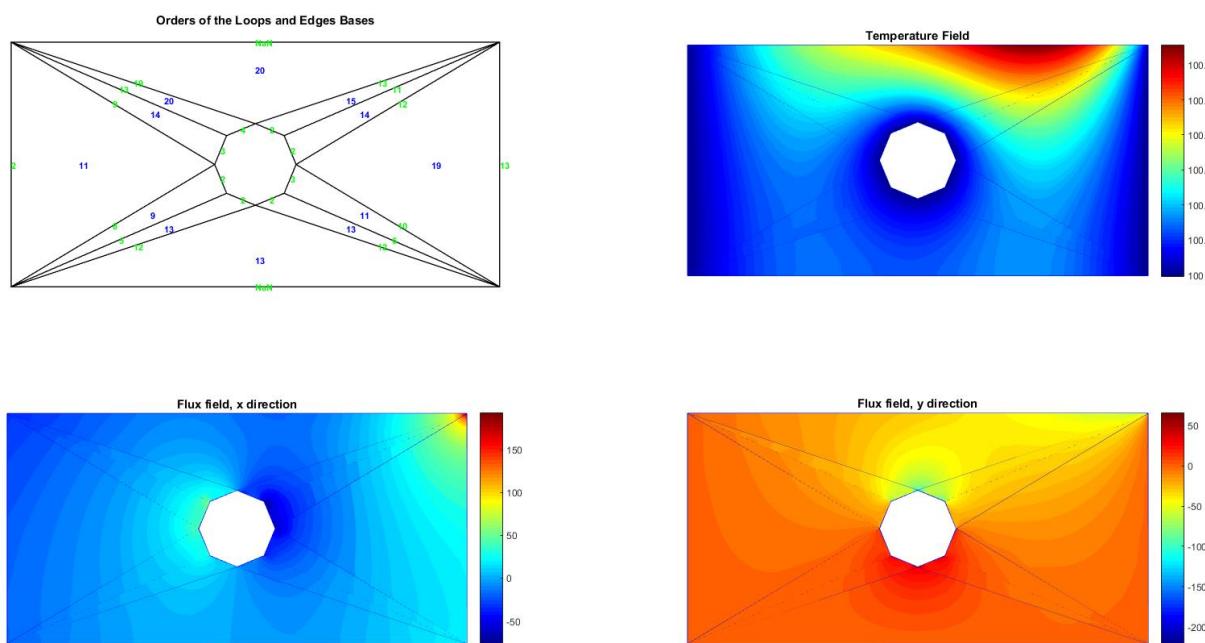


Figure 27. Temperature and heat fluxes for the hollow body problem, 12 finite elements, with adaptive basis refinement

The effect of the automatic p-refinement is obvious in the upper-left plot, where the final orders of domain and boundary refinements are listed on the respective mesh entities, as

the orders are not only no longer uniform, but also quite dissimilar. Indeed, the p-refinement orders of the finite elements vary between 9 and 20, and those of the essential boundaries between 2 and 13. The orders tend to be larger in the upper-right part of the structure, which is to be expected, since in this region the variation of the solution fields is more pronounced.

Unlike the results obtained using uniform p-refinement orders (Figure 26), the flux fields here are considerably smoother, with a better balance between neighbouring elements. However, the analysis took 153 iterations and 185.36 sec, roughly 300 times more than the analysis of the same problem, but with a fixed basis refinement (Section 5.7.2).

Besides the final refinement orders and solution fields presented in Figure 27, **FreeHyTE – Heat HTTE** also plots the convergence graphs, as shown in Figure 28. Each of the four plots are briefly described next.

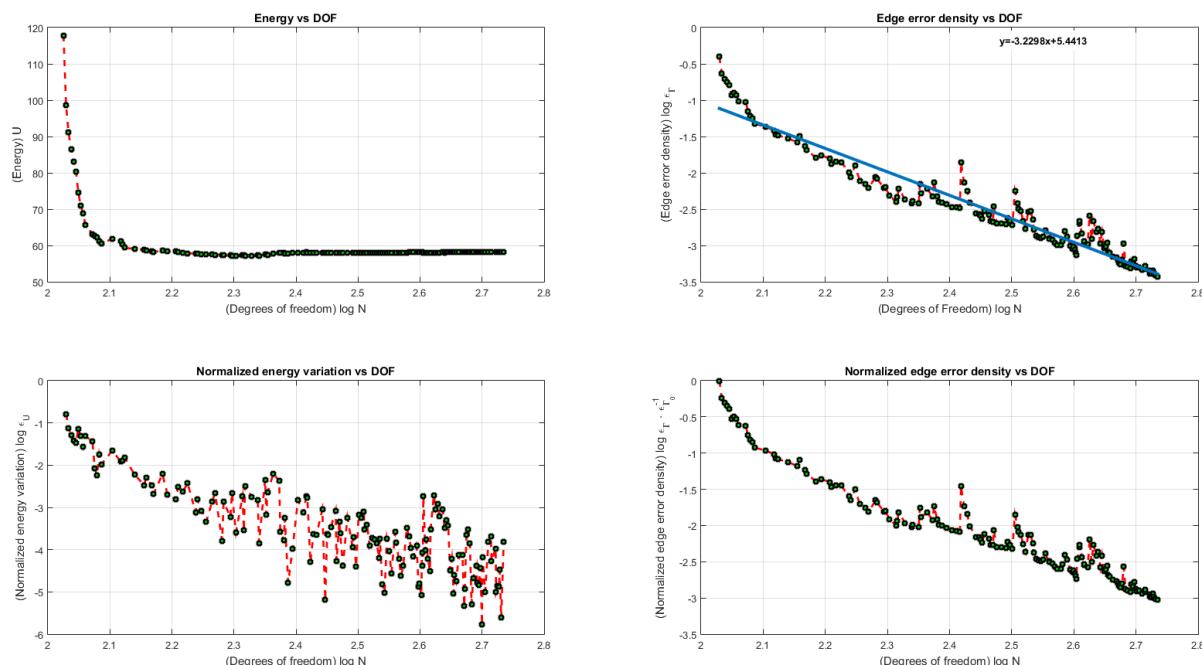


Figure 28. Convergence graphs after the automatic p-refinement process

- *Energy vs DOF*: Plots the variation of the thermal energy with the total number of degrees of freedom of the model. Each dot corresponds to an iteration. The abscissa is logarithmic. While the plotted energy lacks the (constant) particular solution contribution in problems involving internal heat generation, this is a quantity that steers the p-refinement algorithm and its stabilization is an important indicator of the convergence;
- *Normalized energy variation norm vs DOF*: Plots the normalized difference between the thermal energies,  $U_i$  and  $U_{i-1}$ , in two successive iterations (that is, two successive points in the *Energy vs DOF* plot),  $\|U_i - U_{i-1}\| \cdot U_{i-1}^{-1}$ . Both axes are

logarithmic. This quantity is used by the thermal energy convergence criterion, as discussed in Section 4.5.5;

- *Edge error density vs DOF:* Plots the maximum error density on the boundary against the total number of degrees of freedom of the model. The maximum error density on the boundary is used as a selection criterion (see Section 4.5.4) and reflects the maximum temperature imbalance between adjacent elements. Both axes are logarithmic. The blue line in this plot is a linear fit of the error density decay and the  $y(x)$  function is its mathematical expression. The (absolute value of the) coefficient of  $x$  in this expression is thus a measure of the convergence of the refinement process;
- *Normalized edge error density vs DOF:* This contains the same information as the previous plot, but the maximum error density on the boundary is normalized to its value in the first iteration. The plotted quantity is used as a convergence criterion (see Section 4.5.5).



As obvious from the plots in Figure 28, **the convergence of the model is not monotonic.**

After reaching a certain level of precision, the gains obtained with adding functions to the finite element basis are counterbalanced by the numerical errors, justifying the oscillations in the convergence graphs.

## 6. ADVANCED STRUCTURAL DEFINITION

### 6.1. INTRODUCTION

While appropriate for the vast majority of applications, the standard operation of **FreeHyTE – Heat HTTE** does not provide user with a way to choose different p-refinements for different mesh entities. However, while the Graphical User Interface (GUI) of **FreeHyTE – Heat HTTE** only allows uniform basis refinements to be defined, these definitions can be programmatically overwritten to specify different orders of refinement for different finite elements and essential boundaries. A simple procedure for this is described in Section 6.2.

Another limitation of the GUI-based **FreeHyTE – Heat HTTE** is that a single material type can be used in the structure definition. This limitation can be overcome following the same procedure as for the localized p-refinement. This topic is covered in Section 6.3.



Please note, however, that **no contact resistance can be prescribed between layers of different materials**, meaning that flux and temperature continuity are enforced on the interface boundaries. This limitation may be removed in future versions of **FreeHyTE – Heat HTTE**.

## 6.2. LOCALIZED P-REFINEMENT

The concept of localized p-refinement was introduced in Sections 3.1.4 and 4.5. The procedure may improve the finite element solutions in areas where localized effects like temperature or heat flux concentrations/discontinuities are expected to occur, without the need of incrementing the order of the bases in areas where doing so would bring no significant improvement.

One way of taking advantage of the localized p-refinement capability is to use the automatic basis refinement procedure detailed in Sections 4.5.3 to 4.5.6. However, the procedure puts the silicone beast in charge of defining the p-refinement, with no control from the user, and the execution time is considerably increased. Alternatively,



localized p-refinement can be defined programmatically, by slightly modifying the *InputProcReg.m* file if you use the regular mesh generator or *InputProcTri.m* file if you use the non-regular mesh generator.

Both of these files are found in the installation folder of **FreeHyTE – Heat HTTE**. The procedure for the definition of localized p-refinement involves the following steps:

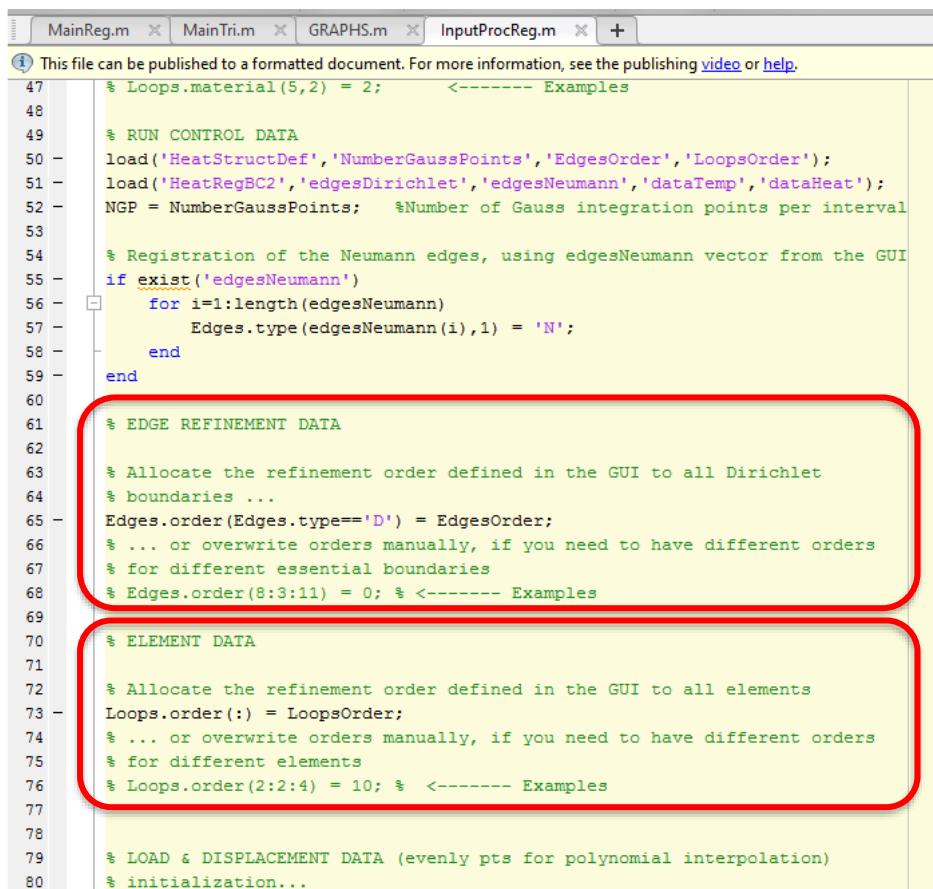
- define your model as usual, inserting the typical orders of boundary and finite element refinements in GUI 1;
- at any point between the creation of the mesh and pressing the **RUN** button in the verification interface, open the *InputProcReg.m* or *InputProcTri.m* file, according to the mesh generator you use;
- in the Matlab code, look for the areas marked with '`% EDGE REFINEMENT DATA`' or '`% ELEMENT DATA`' (see Figure 29);
- set the new orders of approximation for the desired Dirichlet boundaries and finite elements according to the examples indicated in the \*.m file. For instance, to set the order *a* for the boundary *b* and element *c*, the input lines should be something like,

```
Edges.order(b) = a;
```

```
Loops.order(c) = a;
```

- use as many lines as needed to overwrite all orders that you wish to change;

- remember to change the code back after the analysis is completed.



```

MainReg.m X MainTri.m X GRAPHs.m X InputProcReg.m X +
① This file can be published to a formatted document. For more information, see the publishing video or help.
47 % Loops.material(5,2) = 2; <----- Examples
48
49 % RUN CONTROL DATA
50 - load('HeatStructDef','NumberGaussPoints','EdgesOrder','LoopsOrder');
51 - load('HeatRegBC2','edgesDirichlet','edgesNeumann','dataTemp','dataHeat');
52 - NGP = NumberGaussPoints; %Number of Gauss integration points per interval
53
54 % Registration of the Neumann edges, using edgesNeumann vector from the GUI
55 - if exist('edgesNeumann')
56 -     for i=1:length(edgesNeumann)
57 -         Edges.type(edgesNeumann(i),1) = 'N';
58 -     end
59 - end
60
61 % EDGE REFINEMENT DATA
62
63 % Allocate the refinement order defined in the GUI to all Dirichlet
64 % boundaries ...
65 - Edges.order(Edges.type=='D') = EdgesOrder;
66 % ... or overwrite orders manually, if you need to have different orders
67 % for different essential boundaries
68 % Edges.order(8:3:11) = 0; % <----- Examples
69
70 % ELEMENT DATA
71
72 % Allocate the refinement order defined in the GUI to all elements
73 - Loops.order(:) = LoopsOrder;
74 % ... or overwrite orders manually, if you need to have different orders
75 % for different elements
76 % Loops.order(2:2:4) = 10; % <----- Examples
77
78
79 % LOAD & DISPLACEMENT DATA (evenly pts for polynomial interpolation)
80 % initialization...

```

Figure 29. Code areas to modify for localized p-refinement

The localized p-refinement must be defined such as to **satisfy the kinematic indeterminacy condition**, as explained in Section 4.5.2. This means that, for every finite element, **the number of degrees of freedom inside the element must be superior to the sum of the degrees of freedom on its essential (Dirichlet and interior) boundaries**. Denoting by  $n_D$  the order of the domain basis of a finite element and by  $n_\Gamma^i$  the order of the approximation basis on its essential boundary  $i$ ,

the **kinematic indeterminacy criterion is always satisfied if**



$$2n_D \geq \sum_{i=1}^S (n_\Gamma^i + 1)$$

where  $S$  is the total number of essential boundaries of the element.

### 6.3. DEFINITION OF DIFFERENT MATERIALS

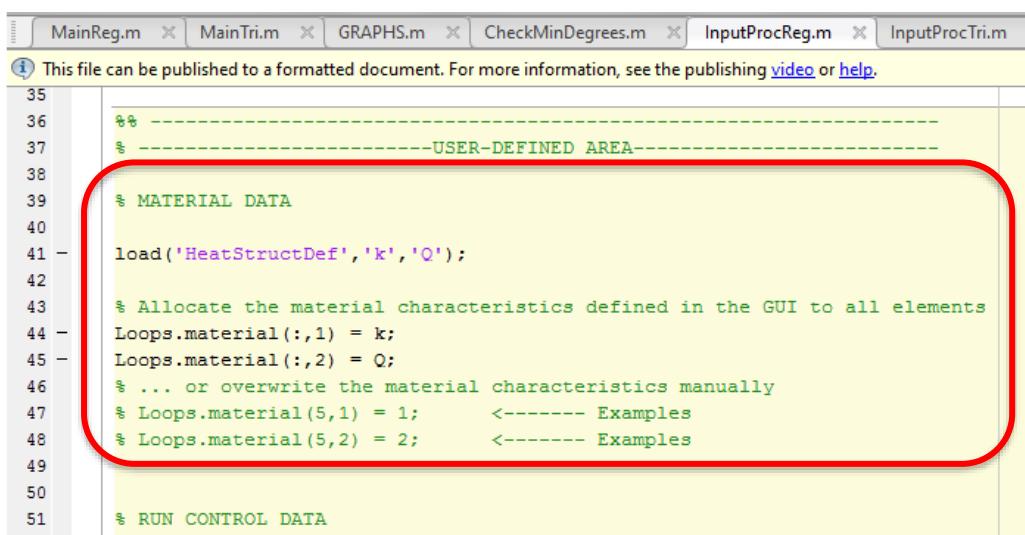
The same procedure described in Section 6.2 can be applied to overwrite the uniform material description given in GUI 1. The code (\*.m) files to change are the same and the necessary steps are listed below:

- define your model as usual, inserting the typical material characteristics in GUI 1;
- at any point between the creation of the mesh and pressing the **RUN** button in the verification interface, open the *InputProcReg.m* or *InputProcTri.m* file, according to the mesh generator you use (regular or non-regular);
- in the Matlab code, look for the area marked with '`% MATERIAL DATA`' (see Figure 30);
- set the new material characteristics for the desired finite elements according to the examples indicated in the \*.m file. For instance, to set the conductivity *a* and internal heat generation *b* for the element *c*, the input lines should be something like,

```
Loops.material(c,1) = a; % '1' designates the conductivity
```

```
Loops.material(c,2) = b; % '2' designates the internal heat
```

- use as many lines as needed to overwrite the material characteristics for all elements that you wish to change;
- remember to change the code back after the analysis is completed.



This screenshot shows a MATLAB editor window with several tabs at the top: MainReg.m, MainTri.m, GRAPHs.m, CheckMinDegrees.m, InputProcReg.m (which is the active tab), and InputProcTri.m. A yellow status bar at the bottom says: "This file can be published to a formatted document. For more information, see the publishing [video](#) or [help](#)". The main code area contains the following:

```

35 %
36 %
37 % -----USER-DEFINED AREA-----
38 %
39 % MATERIAL DATA
40 %
41 - load('HeatStructDef','k','Q');
42 %
43 % Allocate the material characteristics defined in the GUI to all elements
44 - Loops.material(:,1) = k;
45 - Loops.material(:,2) = Q;
46 %
47 % ... or overwrite the material characteristics manually
48 % Loops.material(5,1) = 1;      <----- Examples
49 % Loops.material(5,2) = 2;      <----- Examples
50 %
51 %

```

A red rectangular box highlights the code block from line 37 to line 48, which is labeled as the 'USER-DEFINED AREA'. Below this area, another section is labeled '% RUN CONTROL DATA'.

Figure 30. Code areas to modify for defining different materials