Final Year Project

Bachelor's degree in Industrial Technology Engineering

Analysis of nuclear fusion reactor's discharge simulations using METIS

REPORT

Author: Xavier Guilera Alonso
Director: Guillem Cortes Rossell
Tutor: Albert Riego Perez
Call: January 2022

ETSEIB

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

UPC

Abstract

The Nuclear Engineering Department of UPC has the need of a nuclear fusion plasma simulator for its research activities and educational tasks. For many years, PRETOR has been the program used for this purpose, but it has become obsolete. Nowadays there are more modern simulation codes for fusion reactor's plasma shots, like CRONOS and METIS. Due to the complexity of CRONOS, METIS was more suitable.

The Nuclear Engineering Section of UPC has been authorised by ITER Organisation to use METIS for educational and research purposes after the signing of an agreement named "Agreement on Cooperation on the ITER Modelling and Analysis Suite (IMAS) and Related Repositories by Universitat Politècnica de Catalunya".

This project has gotten inside METIS program, to analyse the obtained simulations on fusion plasma, trying to understand some pieces of code as well. METIS is a code based on MATLAB, combined with some modules programmed with Fortran, C and C++, developed by J.F. Artaud from the Institute for Magnetic Fusion Research, in France.

Despite being METIS a very powerful tool on the simulation of fusion reactors with a great quantity of applications, as it is a restricted program only available for researchers from the same field, it does not offer enough help information for those who begin to work with it.

This document includes a brief introduction on fusion physics, the principles on magnetic confinement reactors and the presentation of some different integrated modelling codes for fusion plasma.

In order to complement the METIS help guide, we have used the knowledge acquired during our work to prepare a draft manual that we hope to be useful whenever it is intended to work on a teaching practices program. During this process we have tried to understand on which models the program is based, getting to analyse the source code, study and compare simulations.

Furthermore, a great amount of work done in this project has been focused on the development of a setup of MATLAB functions designed to make easier the study of METIS outputs. This has permitted us to outline an example on how it would be possible to incorporate new functionalities inside METIS source code.

Resum

El Departament d’Enginyeria Nuclear de la UPC té la necessitat d’un simulador de plasmes de fusió nuclear per dur a terme activitats de recerca i educació. Durant anys, PRETOR ha estat el programa utilitzat amb aquest propòsit, però ha quedat obsolet. Actualment hi ha simuladors més moderns per les descàrregues de reactors de fusió com CRONOS i METIS. Degut a la complexitat de CRONOS, METIS era més adequat.

La Secció d’Enginyeria Nuclear de la UPC està autoritzada per l’Organització ITER a poder utilitzar METIS amb propòsits educatius i de recerca després de signar un acord anomenat “Agreement on Cooperation on the ITER Modelling and Analysis Suite (IMAS) and Related Repositories by Universitat Politècnica de Catalunya".

Aquest projecte s’ha endinsat dins el programa METIS, per tal d’analitzar simulacions obtingudes sobre el plasma de fusió, procurant entendre part del codi del programa en el procés. METIS és un codi basat en MATLAB, combinat amb alguns mòduls en Fortran, C i C++, elaborat per J.F. Artaud de l’Institut per la Recerca en Fusió Magnètica, a França.

Tot i ser METIS una eina molt potent en la simulació de reactors de fusió amb una gran quantitat d’aplicacions, al tractar-se d’un programa restringit només a investigadors del mateix camp de recerca, no ofereix una informació suficient per a tot aquell que comença a treballar amb ell.

Aquest document incorpora una breu introducció a la física de fusió, els principis de funcionament dels reactors de confinament magnètic i la presentació de diferents codis de modelat integrat per plasmes de fusió.

Per intentar complementar la guia d’ajuda de METIS, hem utilitzat els coneixements adquirits durant aquest projecte per elaborar un esbós de manual d’usuari que podria ser útil a l’hora de confeccionar un manual de pràctiques acadèmiques. Durant aquest procés hem intentat entendre sobre quins models es basa el programa, arribant a analitzar el codi font, estudiar i comparar simulacions.

A més a més, un dels gruixos de feina més importants d’aquest projecte, ha estat tot el desenvolupament d’un seguit de funcions de MATLAB destinades a facilitar l’estudi de les sortides de METIS. Això ens ha permès també plantejar un exemple de com es podrien incorporar noves funcionalitats dins el codi font de METIS.

Resumen

El Departamento de Ingeniería Nuclear de la UPC tiene la necesidad de un simulador de plasmas de fusión nuclear para llevar a cabo actividades de investigación y educación. Durante años, PRETOR ha sido el programa utilizado con este propósito, pero ha quedado obsoleto. Actualmente hay simuladores más modernos para las descargas de reactores de fusión, como CRONOS y METIS. Debido a la complejidad de CRONOS, METIS era el más adecuado.

La Sección de Ingeniería Nuclear de la UPC está autorizada por la Organización ITER a poder utilizar METIS con propósitos educativos y de investigación después de firmar un acuerdo llamado "Agreement on Cooperation on the ITER Modelling and Analysis Suite (IMAS) and Related Repositories by Universitat Politècnica de Catalunya".

Durante este proyecto nos hemos adentrado en el programa METIS, para analizar las simulaciones obtenidas sobre el plasma de fusión, procurando entender parte del código del programa en el proceso. METIS es un programa basado en MATLAB, combinado con algunos módulos en Fortran, C y C++, elaborado por J.F. Artaud del Instituto de Investigación en Fusión Magnética, en Francia.

A pesar de ser METIS una herramienta muy potente en la simulación de reactores de fusión con una gran cantidad de aplicaciones, al tratarse de un programa restringido únicamente a investigadores del mismo campo, no ofrece información suficiente para todo aquel que empieza a trabajar con él.

Este documento incorpora una breve introducción a la física de fusión, los principios de funcionamiento de los reactores de confinamiento magnético y la presentación de diferentes códigos de modelado integrado para plasmas de fusión.

Para intentar complementar la guía de ayuda de METIS, hemos usado los conocimientos adquiridos durante este proyecto para elaborar un esbozo de manual de usuario que podría ser útil a la hora de confeccionar un manual de prácticas académicas. Durante este proceso hemos intentado entender sobre qué modelos se basa el programa, llegando a analizar el código fuente, estudiar y comparar simulaciones.

Además, uno de los principales trabajos en este proyecto ha sido el desarrollo de un conjunto de funciones de MATLAB destinadas a facilitar el estudio de las salidas de METIS. Esto nos ha permitido también plantear un ejemplo de cómo se podrían incorporar nuevas funcionalidades dentro del código fuente de METIS.

Table of contents

1. Glossary

The objective of this glossary is not presenting an exhaustive list on fusion, ITER and METIS terms, but to give a brief explanation for the most relevant acronyms from this project. This was also a first task to do during the documentation phase of this project, so we hope this alphabetically sorted list of acronyms helps the lector to understand this report.

Acronyms and Abbreviations List

ASDEX	Axially Symmetric Divertor Experiment
B_{tout}	Total magnetic field in the middle plane at the low magnetic field side (T)
DEMO	Demonstration power plant
ECRH	Electron Cyclotron Resonance Heating
H-mode	Plasma High confinement mode
ICRH	Ion Cyclotron Resonance Heating
IMAS	ITER Modelling and Analysis Suite
ITER	International Thermonuclear Experimental Reactor
JET	Joint European Torus
$L2h_{slope}$	METIS H-mode parameter for soft transitions
LCFS	Last Closed Flux Surface
LCMS	Last Closed Magnetic Surface
LH	Lower Hybrid oscillation
LHCD	Lower Hybrid Current Drive
L-mode	Plasma Low confinement mode
METIS	Minute Embedded Tokamak Integrated Simulator
n_{bar}	Line-averaged electron density (m^{-3})

NBI	Neutral Beam Injection
P_{add} / P_h	Additional power or heating power (W)
P_{alpha}	Alpha fusion power (W)
P_{Br}	Bremsstrahlung power loss (W)
P_{Cyclo}	Cyclotron power loss (W)
P_{ECRH}	Electron cyclotron power (W)
P_{ICRH}	Ion cyclotron power (W)
P_{in}	Input power (W)
P_{ioniz}	Power losses due to cold neutral ionization (W)
P_{LH}	Lower Hybrid power (W)
P_{lhthr} / P_{Pl2h}	METIS power compared to threshold power for H-mode transitions (W)
P_{loss}	Plasma loss power, as defined in ITER Physics Basis (W)
$P_{lossl2h}$ / $P_{scaling}$	Switch-on power for transition from L to H-mode
P_{NBI}	Neutral Beam Injection power (W)
P_{ohm}	Ohmic power (W)
P_{rad}	Impurity radiation losses, without Bremsstrahlung (W)
S_{ext}	External plasma surface (m^2)
TFTR	Tokamak Fusion Test Reactor

2. Introduction

"The ultimate test of your knowledge is to convey it to another."

- Richard P. Feynman -

It is a really hard task to organize our own thoughts but it is an even more difficult work to do it clearly enough for somebody else to understand them. This project tries to understand thoughts. More precisely, to learn how theory behind fusion physics is applied to simulate fusion plasma and what surmises scientists behind METIS [1] made to create this code. This has been done studying METIS ways of operation and its logic.

2.1. Overview

The Nuclear Engineering Section of UPC has the need of a nuclear fusion plasma simulator for its research activities and educational tasks. For many years, PRETOR has been the code used for this purpose, but it has become obsolete. Nowadays there are several simulation codes for fusion reactor's plasma shots, like CRONOS [8] and METIS. Due to the complexity of CRONOS, METIS was preferable.

METIS (Minute Embedded Tokamak Integrated Simulator) is a fast integrated tokamak simulation tool for the CRONOS suite. The motivation for its creation was to simulate a full plasma discharge in a time of the order of one minute, while for a 1.5D simulation of tokamak plasma discharges, as made for example with the CRONOS suite of codes, would require a lot more of computing time (typically, a factor 10^4 longer). Then, METIS provides a first and faster but yet meaningful alternative to most classical integrated transport suites.

Of course this is a very useful code, but in order to use it properly, be able to take profitable conclusions from the simulations results, and predict plasma behaviours for different scenarios, we should figure out how METIS does its calculations and manages different events throughout a plasma discharge.

2.2. Objectives

This paper aims to be an addition to METIS user's guide oriented to provide a more specific explanation of what METIS code does when we surf through its parameters and options. With this work, we want to understand and learn how METIS works and what possibilities offers when analysing simulations from different fusion reactors scenarios. Then, our main objective is to provide a technical guide of METIS oriented to the academic and research activities of the Nuclear Engineering Section (UPC).

The specific objectives to accomplish this goal are to understand METIS functioning and run simulations through METIS in the process. Another specific objective is to study METIS outputs directly from MATLAB data and display our acquired knowledge of the program developing our own set of MATLAB functions.

2.3. Reach

Only once we have learnt this insight into METIS, we will be ready to carry out teaching practices and research activities. In addition, this project contemplates the option of using our MATLAB setup to add new features through METIS source code.

2.4. Literature survey

When starting to work with METIS, we found laborious to make it work with the official user guide [1] that we received together with the software and the paper on METIS topic [2]. It should be kept in mind that the reference documents we would be able to consult were these ones, due to the technic character of IMAS and METIS software.

That said, this project has been oriented to incorporate this addition to METIS documentation for anybody who starts working with it.

2.5. Organisation of this document

This document begins with a brief theoretical explanation for nuclear fusion and plasma characteristics in section 3 and the presentation of two different simulation codes for fusion plasmas such as CRONOS or PRETOR in section 4. Later on, in section 5, the physical model for METIS and its way to reach different events are explained, and a brief METIS guide is shown in section 6, together with an educational example. This guide continues in section 7, with the output possibilities of METIS, analysing the first simulations to check the theory on section 5 applying our MATLAB setup, explained in section 8. After this, in section 9, a more thorough explanation on METIS data management is treated.

Thanks to our MATLAB code, we can show in section 10 an example on how to implement new scaling laws, for threshold power or power losses involved in H-mode transitions, taken from more recent papers. To finish up this project the work plan, the environmental, social and economic impact about this project and a commentary on the point of view for gender equality in this field, are mentioned from sections 11 to 14.

The Annex, complement to this study, includes a simulations log and our self-developed MATLAB code.

3. Nuclear fusion

3.1. Fusion reactions

Nuclear fusion is a reaction in which two or more atomic nuclei are combined to form one or more different atomic nuclei and subatomic particles. Nuclear fusion uses light elements, such as hydrogen and helium, which are in general more fusible; while the heavier elements, such as uranium, thorium and plutonium, are more fissionable. Reaction 3.1 shows an exothermic fusion reaction between a neutron and a hydrogen nucleus.

$$n + {}^{2}H \rightarrow {}^{3}He + e^{-} + E \;\rightarrow E = 6.291 \text{ MeV} \quad (3.1)$$

The idea behind inducing fusion is replacing neutrons with other light element, like Deuterium, Tritium or Helium-3.

i. $D + D \rightarrow \begin{cases} {}^{3}He + n + 3.27 \text{ MeV} \\ T + p + 4.03 \text{ MeV} \end{cases}$ (3.2)

ii. $\text{D} + {}^{3}\text{He} \rightarrow \alpha + \text{p} + 18.3 \text{ MeV}$ (3.3)

iii. $\text{D} + \text{T} \rightarrow \alpha + \text{n} + 17.6 \text{ MeV}$ (3.4)

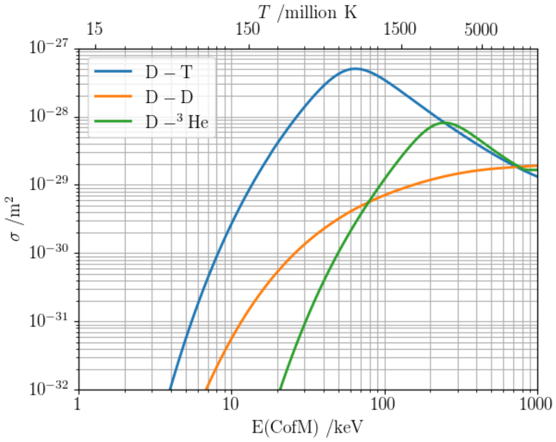


Fig.1: Cross section towards projectile energy required for fusion reaction pairs

The D-T reaction (3.4) requires the least energy to start the reaction (Fig. 1), so it is the easiest one to achieve. Nevertheless, there is no natural T on Earth. This lack of T is an inconvenient, but the generation of neutrons makes it possible to breed Tritium by adding Lithium.

From now on, generally we are going to be talking about D-T reactions when talking about fusion plasma. The energy released by the D-T fusion reaction is carried out by the alpha particles (3.5 MeV) and neutrons (14.1 MeV) in the form of kinetic energy, so the alphas are absorbed by the plasma transforming to heating energy and neutrons are transmitted by the plasma making their thermal energy output available for electric energy generation.

3.2. About plasma

To obtain fusion reactions, it is necessary to heat D-T to high temperatures combined with appropriate pressure conditions:

T [K]	P [atm]	E	
291.75	1	0.025 eV	No fusions. Elastic collisions.
4988	40	0.43 eV	No fusions. Elastic collisions.
116000	1500	10 eV	1 fusion each 500 years.
116·106	$1.5 \cdot 10^6$	10 keV	Fusion produced. $100 \cdot 10^6$ kJ/s per litter of Deuterium.

Tab.1: Plasma temperature and pressure conditions and rate of fusion reactions

At these temperatures, D-T gas appears as plasma, an ionized gas whose behaviour is dominated by collective effects (the way the plasma as a whole reacts is dependent on the behaviour of each and every particle in the plasma)) and possessing a very high electrical conductivity.

3.3. Reaction rate and reactivity

The power generated by fusion reactions, depends of course on the fusion reactions rate, meaning the number of particles colliding per unit of time and volume.

To define this rate, R_{12}, we assume collisions between hard-spheres, considering both nuclei, the target nucleus and the incident particle densities ($n_1 n_2$), before fusion, in a first approximation, resulting:

$$R_{12} = n_1 n_2 \sigma v \quad (3.5)$$

being σ the reaction cross-section and v the particle velocity. If each fusion reaction generates an energy E_f, then the fusion power density is:

$$\rho_f = E_f n_1 n_2 \sigma v \quad (3.6)$$

The R_{12} expression rate is correct, but as mentioned, is a first simplification (one quiet sphere and a moving sphere). We must consider relative velocities, a distribution function (Maxwell distribution) and mean variables for the rate parameters. With these, we obtain:

$$R_{12} = n_1 n_2 \langle \sigma v \rangle \quad (3.7)$$

where $\langle \sigma v \rangle$ is the parameter of the reaction and depends on its species and the plasma temperature (mean kinetic energy of particles). This allows determining the rate of reactions in a simple way. While n_1 and n_2 depend on position and time, $\langle \sigma v \rangle$ can be determined through a maxwellian distribution function:

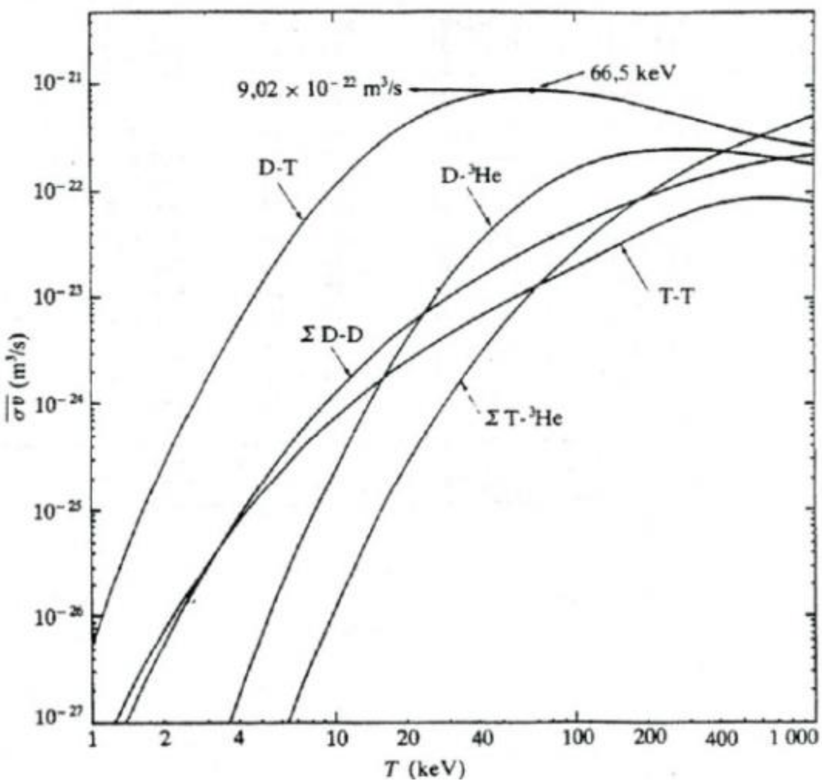


Fig.2: $\langle \sigma v \rangle$ towards plasma temperature

$$\overline{\sigma v}(T) \simeq 8{,}32 \times 10^{-27}\, T^5 \quad \text{for} \quad 1{,}4 \text{ keV} \leq T \leq 2{,}6 \text{ keV}$$
$$\overline{\sigma v}(T) \simeq 2{,}15 \times 10^{-26}\, T^4 \quad \text{for} \quad 2{,}6 \text{ keV} \leq T \leq 5{,}3 \text{ keV}$$
$$\overline{\sigma v}(T) \simeq 1{,}15 \times 10^{-25}\, T^3 \quad \text{for} \quad 5{,}3 \text{ keV} \leq T \leq 10{,}3 \text{ keV}$$
$$\overline{\sigma v}(T) \simeq 1{,}18 \times 10^{-24}\, T^2 \quad \text{for} \quad 10{,}3 \text{ keV} \leq T \leq 18{,}5 \text{ keV}$$
$$\overline{\sigma v}(T) \simeq 2{,}18 \times 10^{-23}\, T \quad \text{for} \quad 18{,}5 \text{ keV} \leq T \leq 39{,}9 \text{ keV}$$
$$\overline{\sigma v}(T) \simeq 8{,}69 \times 10^{-22} \quad \text{for} \quad 39{,}9 \text{ keV} \leq T \leq 100 \text{ keV}$$

Fig.3: $\langle \sigma v \rangle$ approximation for D-T plasmas

3.4. Fusion power

As with equation 3.6, we can compute different power densities such as the one given by alphas to plasma, determined for a D-T reaction as:

$$\rho_\alpha = f_\alpha E_\alpha R_{DT} \quad (3.8)$$

where f_α is the fraction of alpha particles confined in a plasma (≈1).

The density power transported by neutrons can be expressed as well as:

$$\rho_n = E_n R_{DT} \quad (3.9)$$

The energies of these reactions are E_α = 3.5 MeV and E_n = 14.1 MeV as seen before.

3.5. Power balance

We are going to study a very simplified model of a fusion reactor, so we'll treat the plasma as if it were uniform and quasi-neutral, characterizing it by electron, fuel ion, and alpha particle densities and temperature. Therefore, the hypotheses considered about the fusion plasma behaviour are:

- Equal concentrations of positive and negative charges per unit volume, n:

$$2n_D = 2n_T = n_e = n \quad (3.10)$$

- All the fuel components are at the same temperature, T:

$$T_D = T_T = T_e = T \quad (3.11)$$

For 3.10 and 3.11, the plasma species considered are deuterium, tritium and electrons, for both density and temperature.

- If we consider the fuel as gaseous plasma fully ionized near the thermodynamic equilibrium on the Maxwell distribution, we must assume:

$$U = \frac{3}{2}p \quad (3.12)$$

where U is the internal energy density and p is the pressure of the plasma particles.

In a small fixed volume, we can describe the conservation of energy in fluids as:

$$\frac{3}{2}\frac{\partial p}{\partial t} + \frac{3}{2}\nabla \cdot p\vec{v} + p\nabla \cdot \vec{v} + \nabla\vec{q} = S \quad (3.13)$$

1 2 3 4 5

Being,

1. Internal energy variation in time.
2. Density of power flux leaving the volume by convection.
3. Density of power losses due to the expansion of the fluid.
4. Density of power losses by diffusion. Mainly heat loss by conduction $\vec{q} = -\kappa \nabla T$.
5. Sources and sinks of power density contributing to the power balance.

$$S = \rho_f - \rho_{Br} - \rho_C + \rho_h \quad (3.14)$$

These power densities (fusion, Bremsstrahlung and Cyclotron radiation losses, and external heating) are going to be some of the most important quantities studied during the METIS simulations, so in the next points we'll give a brief explanation about them.

We can simplify even more the physics by considering the steady state 0-D phase of the plasma and give its power balance defining an also 0-D energy confinement time τ_e, a relaxation time for e- due to heat conduction, valid for a generic geometry. During this time, there are fusion reactions in the reactor. It depends on p and T, but we will consider it as independent and that it is known. The balance, after several simplifications, would be:

$$\rho_\alpha + \rho_h = \rho_{Br} + \frac{3}{2}\frac{p}{\tau_e} \qquad (3.15)$$

3.6. Power loss

Our previous power balance (3.14) takes in consideration two of the main power loss causes for tokamak plasmas. Despite existing different reasons behind power loss, the Bremsstrahlung and Cyclotron radiations are the most important ones.

The Bremsstrahlung radiation is produced when a charged particle is deviated or decelerated by another charged particle, generating electromagnetic radiation. These losses are significant, so they should be considered in the plasma power balance. The bremsstrahlung power increases because of the higher value of the ionic charge of the impurities. Its emission is on the ultraviolet range.

Cyclotron radiation is emitted by charged particles accelerated by a magnetic field. It is known as well as Synchrotron radiation, due to the wavelength it emits. Because of its mass, only electrons are considered over ions. In comparison to Bremsstrahlung, its emissions are infrared, so it is less energetic.

Cyclotron losses grow faster with temperature, and can take large power such as 1 MW/m3 under reactor conditions. At low T, though, Bremsstrahlung radiation is greater than Cyclotron. However, this power is not lost from the plasma, due to its optical thickness to radiation at the fundamental frequency. Overall, the main power is lost in the harmonics, where the Cyclotron losses are in the order of 10-2 MW/m3 and then, negligible.

3.7. Magnetic confinement reactors

Since the beginning of fusion research in 1950s, two different perspectives have been in use to generate the magnetic surface necessary to confine all the plasma particles: the tokamak and the stellarator configurations.

The development of tokamak technology has focused almost all the efforts from the start, and although the stellarator concept has never been abandoned, the first large-size stellarator was put into operation in Japan only in 1998, when different tokamak reactors like TFTR or JET had already produced MW of fusion power.

For this reason, as nuclear fusion research keeps concentrating on tokamak development and the upcoming reactor, ITER, will be the main representative of tokamak fusion reactors, we will study the physics and simulations concerning this approach.

The International Thermonuclear Experimental Reactor (ITER) located near Aix-en-Provence, in southern France, will be dedicated to the investigation and demonstration of burning plasmas. In these plasmas, the energy of the helium nuclei produced by the fusion reactions is enough to maintain the temperature of the plasma, reducing or eliminating the need for external heating, offering the highest efficiency until now.

ITER will also test the availability and integration of technologies essential for a fusion reactor (such as superconducting magnets, remote maintenance, and systems to exhaust power from the plasma) and the validity of tritium breeding module concepts that would lead in a future reactor to tritium self-sufficiency.

Generally speaking, a modern tokamak consists of a toroidal vacuum vessel (with a D-shaped cross-section) around which coils are wound. These coils generate a toroidal magnetic field (green field lines).

The current variation in the central ohmic transformer coils induces an electric field along these lines, driving a toroidal flow for ions and electrons in opposite directions (Fig. 4). This constitutes a current, the plasma current (big red arrows), which generates a poloidal magnetic field (yellow field lines). The superposition of the toroidal and poloidal field lines results in magnetic field lines winding around the torus, shown in black, confining the charged plasma particles.

Viewed as an electrical system, a tokamak is a transformer with ohmic transformer coils as the primary winding and the single-loop conducting plasma torus as the secondary winding.

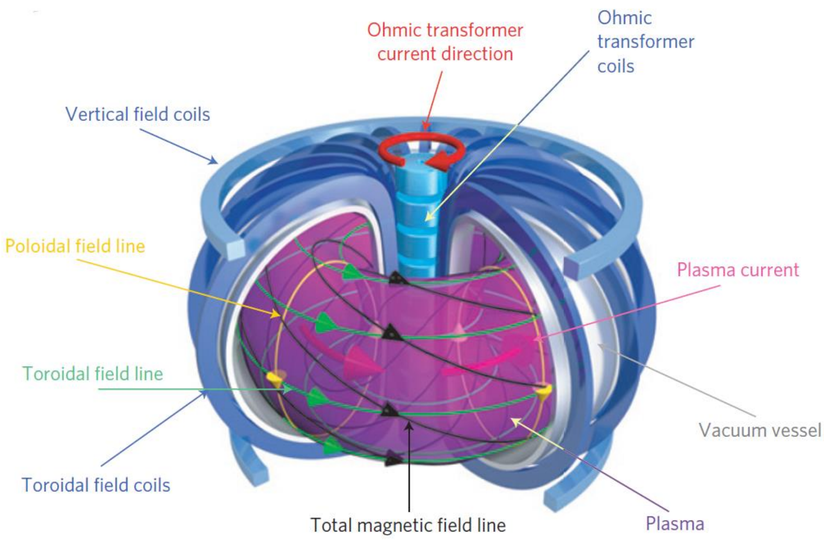


Fig. 4: Tokamak scheme of operation

The pure tokamak scheme of operation alone cannot provide sufficiently high temperatures to generate large amounts of fusion power only through its ohmic-heating power produced by the plasma current. This state of plasma is known as L-mode, referring to a low magnetic confinement regime. To further increase the temperature of the plasma, additional heating methods must be used, explained further on in section 3.8.

In 1982, the discovery of the H-mode (High confinement) in ASDEX experiment, trying to achieve enough plasma stability and increase its temperature up to the necessary for fusion, improved the plasma energy confinement. These resulted in the conditions with best prospects for tokamak operation. Then, we can define H-mode as the stabilization of unstable modes located on the vicinity of the LCFS (Last Closed Flux Surface), reducing the electron heat and particle diffusion.

In order to get to the H-regime, the transport power losses must exceed a certain threshold power. Although half of the H-mode behaviour has been solved, there are lots of unknown considerations yet to produce a predictive theory for the scaling of the power threshold that triggers the evolution from L-mode to H-mode. For now, the threshold power is determined by an extensive database of H-mode power thresholds, and there are several empirical scaling options for the L-H transition.

3.8. Additional heating power

There are several ways of externally heating the plasma, and we will present them from the ITER reactor design [5] and its physics basis. This tokamak will use up to four sources of external heating working together to provide the energy required to achieve the temperature necessary for fusion. These are two neutral beam injectors and two sources of high-frequency electromagnetic waves. The main heating processes are:

- **Neutral Beam Injection**

Neutral beam injectors are used to shoot uncharged high-energy particles into the plasma where, by way of collision, they transfer their energy to the plasma particles. Before injection, deuterium atoms must be accelerated outside of the tokamak to a kinetic energy of 1 MeV. Only atoms with a positive or a negative charge can be accelerated by electric field; for this, electrons must be removed from neutral atoms to create a positively-charged ion. The process must then be reversed before injection into the fusion plasma; otherwise the electrically-charged ion would be deflected by the magnetic field of the plasma cage. In neutral beam injection systems, the ions pass through a cell containing gas where they recover their missing electron and can be injected as fast neutrals into the plasma.

Two neutral beam injectors—each one delivering a deuterium beam of 16.5 MW with particle energies of 1 MeV—are currently foreseen for ITER. A third neutral beam will be used for diagnostic purposes.

- **Electron Cyclotron Heating**

Electron Cyclotron Resonance Heating (ECRH) heats the electrons in the plasma with a high-intensity beam of electromagnetic radiation at a frequency of 170 GHz, the resonant frequency of electrons. The electrons in turn transfer the absorbed energy to the ions by collision. Power will be provided by powerful, high-frequency gyrotrons as power sources. The ITER design includes the development of a 1 MW gyrotron operating at 170 GHz with pulse duration of more than 500 s.

- **Ion Cyclotron Heating**

In ion cyclotron resonance heating (ICRH), energy is transferred to the ions in the plasma by a high-intensity beam of electromagnetic radiation with a frequency of 40 to 55 MHz.

A generator, transmission lines and an antenna are necessary for ion cyclotron heating. A generator produces high-power radio frequency waves that are carried along a transmission line to an antenna located in the vacuum vessel, sending the waves into the plasma.

4. Simulation codes for fusion plasmas

ITER includes a novelty in tokamak history, since it is planned to simulate every plasma experiment planned to run on ITER by an Integrated Modelling tool to check that the pulse is achievable. To model the plasma dynamics of entire experiments the current paradigm is to use 1.5D Integrated Modelling codes. Simulation codes like CRONOS or PRETOR, introduced in this section, solve transport equations in the plasma core for quantities like energy, poloidal flux, particles or toroidal momentum in the radial direction through their average over determinate flux surfaces.

While for simulations of short experiments of a few seconds it is reasonable to use the best available modules, to simulate ITER experiments, which could last up to a few thousands of seconds, we face a computational challenge. Due to the different time scales on which different plasma qualities evolve, like plasma turbulence (~10^{-6} s), transported quantities (~1 s) or diffusion of the poloidal flux (~1000 s); and the high degree of stiffness these transport models usually present, create a numerical challenge for transport solvers. In fact, with these conditions, the simulation of a full discharge on ITER with 1.5D transport code using sophisticated modules typically takes a few days.

4.1. CRONOS

The CRONOS [8] suite of codes is a modular environment dedicated to 1.5D integrated simulation for tokamak discharges. It integrates, in a modular structure, a 1D transport solver with general 2D magnetic equilibria, several heat, particle and impurities transport models, as well as heat, particle and momentum sources. The main body of the program and the graphic interface have both been developed through MATLAB environment. However, many of the modules have been written in Fortran and a few in C or C++.

4.2. PRETOR

PRETOR [9] code is a 1.5D based code which can simulate the temporal and radial evolution of the main macroscopic physical magnitudes of a thermonuclear plasma magnetically confined inside a tokamak or a stellarator of a given geometry and under some defined conditions in a completely predictive way.

PRETOR is mentioned in this project because it has been a useful tool during many years for teaching practices and academic purposes at UPC.

5. METIS

METIS is a numerical code aiming at fast full tokamak plasma analyses and predictions. As it is oriented to perform investigations on this field, is a reserved program for those who would work with it, and despite being an open-source code, it is not a public tool and can only be acquired with the specific permit from the creators.

The Nuclear Engineering Section of UPC is authorized by ITER Organisation to use METIS for educational and research purposes after the signing of an agreement named "Agreement on Cooperation on the ITER Modelling and Analysis Suite (IMAS) and Related Repositories by Universitat Politècnica de Catalunya.

5.1. Physical model

In order to shorten the computational time it takes to simulate a plasma shot with a 1.5D modelling code, it is required to simplify the physics model. METIS takes four principles for its design [2]. These, of course, will make the result of the simulation tend to deviate from the real experiment, losing reliability on its prediction. Anyway, these principles have been carefully chosen to minimize its effect on this loss of reliability, these being:

- Keep the 1.5D paradigm on what can be reliably modelled with accuracy, typically plasma equilibrium and resistive current diffusion.
- Use a quasi-0D approach for what is usually modelled with less reliability, overall turbulent transport.
- Keep the non-linear interactions between the transported quantities, plasma equilibrium and source terms.
- Keep in the model a realistic modelling of the dynamic character of the sources and plasma response.

A further explanation on how these principles are implemented in METIS can be found in [2].

It is appropriate to say that, in a first approach, it seems METIS interests are focused on the steady-state phase of the discharge, making it possible to simplify even more some of the differential equations for certain quantities as well as the power balance, as mentioned in section 3.5.

5.2. Events of interest and computation

To avoid getting lost inside the great amount of options, parameters, and scenarios we could simulate through METIS, we tried to find a way inside the program through some events we could define as interesting from the academic point of view. Achievements like breakeven, ignition or the transitions between L and H-mode have been a window into METIS definitions and mechanics, and these are just a small sample of the interesting physics we could study through this software. However, in order to understand how METIS computes H-mode transitions, for example, we should understand what this event means, how it affects plasma, and later, what assumptions METIS takes to compute it.

5.2.1. Breakeven and ignition

Theoretically, the breakeven is defined as the trespassing of the produced fusion power over all the input power the plasma receives. In this moment the reactor produces the same or more energy than the externally supplied. Still, at these conditions, if the external heating is turned off, the fusion reactions will stop.

METIS does not provide the instant where this event happens, because it is only interesting from the academic point of view and it has no direct repercussions on the plasma behaviour. Despite of this, through our MATLAB setup presented in section 8, we can define and find the moment of the analysed simulation where this occurs. Then, in terms of power densities, breakeven should be defined as the surpassing of fusion power over all the heating powers (resonance heating, neutrals injection, lower hybrid and ohmic):

$$\rho_\alpha \geq \rho_h = \rho_{ecrh} + \rho_{icrh} + \rho_{nbi} + \rho_{lh} + \rho_{ohm} \qquad (5.1)$$

The reactivity of fusion plasmas is quantified by the power amplification or gain factor (the fusion Q-factor). The first important landmark for the value of Q is breakeven, when the heating power is equal to the fusion power produced:

$$Q = \frac{P_\alpha}{P_h} = 1 \; (Breakeven) \qquad (5.2)$$

The second important landmark would be the plasma ignition. We understand this event as the instant when the fusion reaction is self-sustained through its fusion power produced and the input additional heating power can be turned off. This way the gain factor would tend to an infinity value and the plasma would be ignited. ITER goal is $Q \geq 10$

On paper, in order to determine the number of fast alpha particles produced by the D-T reaction is large enough to sustain the plasma temperature and the reaction becomes self-sustainable, we can use the Lawson criterion (eq. 5.3). This relates the electron density, energy confinement time and temperature of the plasma, establishing a minimum value for the triple product of these quantities to consider the plasma ignition (Fig. 5)

$$n_e \cdot \tau_e \cdot T \geq 3 \cdot 10^{21} \qquad (5.3)$$

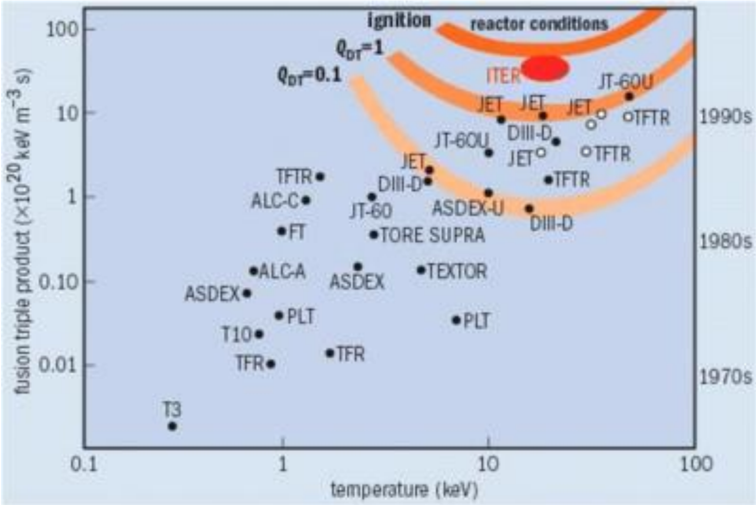


Fig. 5: Lawson criterion for different magnetic confinement experiments

5.2.2. Safety factor and plasma stability

Physics limitations become important when trying to increase the plasma current I_p, in order to reach the necessary temperatures for fusion through the ohmic-heating power:

$$P_{OH} = R_p \cdot I_p^2 \qquad (5.4)$$

where R_p is the electrical resistance of the plasma torus. The maximum plasma current allowed is limited by instabilities that destroy the plasma confinement whenever the safety factor q, which characterizes the LCFS confining the plasma, gets close or below 2 [3]. This is why it was necessary to externally heat the plasma to obtain the required temperatures for fusion, but this resulted in an increase of the degradation of the confinement. As mentioned in section 3.7, the situation of fusion plasma in these conditions is known as the L-mode.

In terms of METIS, the safety factor q_{95} is described as the safety factor at 95% of the enclosed toroidal flux, as defined in the ITER physics basis expression.

5.2.3. H-mode transitions

The discovery of the H-mode implied a paradigm change on the possibilities of heating plasma preserving its stability. This is why we have chosen this event as a way inside METIS, trying to comprehend how the program defines the achievement of the H-mode. According to METIS user guide [1], the conditions for the transition are set when the transport power loss exceeds a certain threshold power. When the power named P_{lhthr}

(Power compared to the power threshold for the beginning of the transition) surpasses $P_{threshold}$. The threshold power is computed as:

$$P_{threshold} = P_{scaling} + Offset \qquad (5.5)$$

where the offset, "l2hmul" option in METIS output data, and $P_{scaling}$, also named $P_{lossl2h}$ inside METIS, depend on the scaling law used for the simulation, empirically extracted from the previously mentioned database. The standard scaling law used for METIS ITER-like simulations corresponds to a 2.4 MW offset and to the next scaling law [4]:

$$P_{lossl2h} = P_{scaling} = 0.0488 \cdot {n_{bar}}^{0.717} \cdot {B_{tout}}^{0.7} \cdot {S_{ext}}^{0.941} \qquad (5.6)$$

The scaling laws depend usually on n_{bar}, the line-averaged electron density, B_{tout}, the total magnetic field in the middle plane at the low magnetic field side, and the external plasma surface, S_{ext}.

As for P_{lhthr}, it also depends on the paper we choose to perform the simulation. The standard option for METIS is named 'P_LCFS' and is defined as the thermal power lost through the last closed flux surface:

$$P_{lhthr} = \min\left(P_{in} - P_{rad} - P_{Br} - P_{Cyclo} - P_{ioniz}\, , P_{in} - \frac{dW}{dt}\right) \text{ with } P_{in} = P_{\alpha} + P_{h} \qquad (5.7 - 5.8)$$

P_{rad} corresponds to the impurity radiation losses, P_{ioniz} are the power losses due to cold neutral ionization and W is the total plasma energy.

The determination of the L to H-mode transition sounds like a trivial matter, but in reality it's not that easy. This transition is in fact softer, and to simulate this effect METIS offers the option of the $l2h_{slope}$ variable, which triggers the transition when the power crossing the separatrix and the threshold is above or equal to $P_{threshold} * l2h_{slope}$. The standard value of this parameter is 0.5. As it affects to the confinement regime of the plasma, it also changes the way the confinement time is computed.

In the case of the transition between modes H to L, the theoretical criterion followed by METIS can vary between comparing $P_{in} < P_{threshold}$ or $P_{lhthr} < P_{threshold}$. For this purpose, it defines the hysteresis phenomenon where the power compared to the threshold power is:

$$P_{hysteresis} = hysteresis \cdot P_{in} + (1 - hysteresis) \cdot P_{lhthr} \qquad (5.9)$$

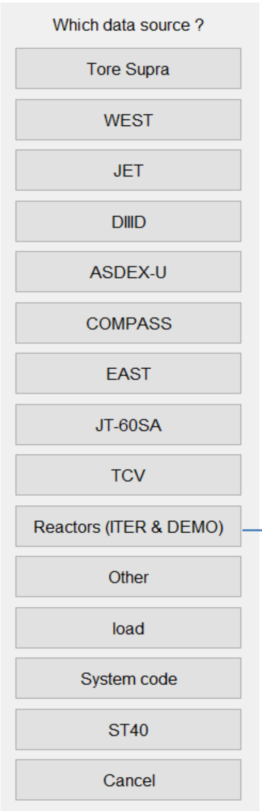
with the $hysteresis$ parameter null by default. So the standard model would be in fact $P_{lhthr} < P_{threshold}$ to determine the loss of the H-mode.

Being able to understand how METIS implements these different scaling laws made us propose a way of carrying out different scaling laws based on more recent papers like [7]. This possible use of our work is executed as an example in section 10.

The study of this important event through different simulations to fully comprehend the way METIS has of computing took us to check if the output results coincided with the methods mentioned above. This resulted in a big effort during this project, because it finally took us to analyse the source code, of a high complexity, and the apparent inconsistencies we found between METIS functioning and the user guide [1], for example. These doubts analysing the simulations results, explained in section 7.2, delayed us in our purpose of studying different simulation scenarios but brought us to a whole new mind-set, using these simulations now to understand METIS functioning and its code.

6. METIS input data

In this section, we intend to provide an easy guide for the main features of METIS we have used during the development of this project, setting the base for a proper practices manual. It begins from the first window after the execution of the program, going through the input data used for a specific simulation, as an example, showing later some of the possible output plots in section 7.1. We also mention some functions and data used by the source code during the execution of these features to describe some of the inner functioning process of METIS.



When we run METIS, it pops up the data source choosing window (Fig. 6). METIS uses several databases for each type of reactor, based on existing experiments. Additionally, for future machines such as ITER, DEMO or JT-60SA, METIS includes a dedicated scenario generator allowing an easy preparation of scenario template. We're focusing then, on these ITER reactor simulations.

When we select this option we are executing the *reactormetissimulation.m* module, which initiates the creation of the option file with standard ITER parameters. This also shows the first options editor interface (Fig. 7) where we can change the value of the initiation parameters.

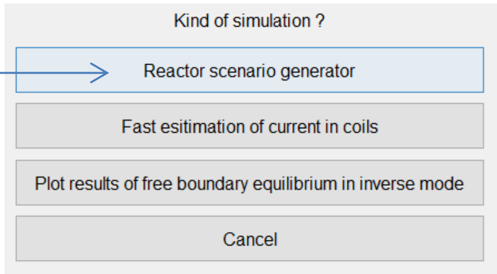
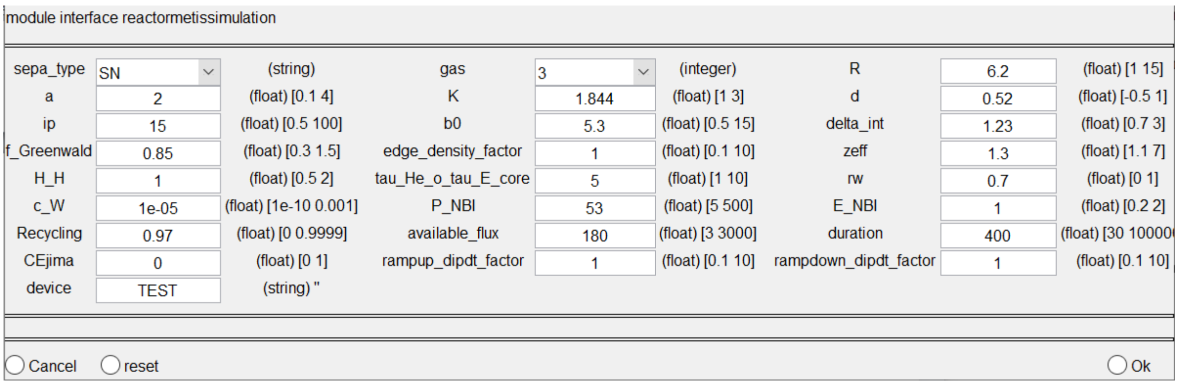


Fig. 6: METIS first interface (left) and the choosing of reactor scenario generator for ITER window (right)



module interface reactormetissimulation

sepa_type	SN	(string)	gas	3	(integer)	R	6.2	(float) [1 15]
a	2	(float) [0.1 4]	K	1.844	(float) [1 3]	d	0.52	(float) [-0.5 1]
ip	15	(float) [0.5 100]	b0	5.3	(float) [0.5 15]	delta_int	1.23	(float) [0.7 3]
f_Greenwald	0.85	(float) [0.3 1.5]	edge_density_factor	1	(float) [0.1 10]	zeff	1.3	(float) [1.1 7]
H_H	1	(float) [0.5 2]	tau_He_o_tau_E_core	5	(float) [1 10]	rw	0.7	(float) [0 1]
c_W	1e-05	(float) [1e-10 0.001]	P_NBI	53	(float) [5 500]	E_NBI	1	(float) [0.2 2]
Recycling	0.97	(float) [0 0.9999]	available_flux	180	(float) [3 3000]	duration	400	(float) [30 10000
CEjima	0	(float) [0 1]	rampup_dipdt_factor	1	(float) [0.1 10]	rampdown_dipdt_factor	1	(float) [0.1 10]
device	TEST	(string) ''						

Cancel reset Ok

Fig. 7: First interface for the ITER reactor generator

Some of these window's parameters we've worked with are defined by METIS as:

- **ip**: plasma current (MA)
- **P_NBI**: maximum power of NBI during flat-top (MW)
- **b0**: vacuum magnetic field (T)
- **duration**: shot duration including ramp-up and flat-top but without ramp-down (s)
- **rampup_dipdt_factor**: factor applied to ramp-up rate current increase
- **rampdown_dipdt_factor**: factor applied to ramp-down rate current decrease
- **Plasma geometry parameters**: **a** (minor radius (m)), **R** (major radius (m))

These starting parameters are necessary to compute the ***z0dinput*** file that contains the input data used to run the simulation. This data is calculated by the *sycomore2metis.m* module, called by the *reactormetissimulation.m* module. The *sycomore2metis* function is the starting point for METIS coupling to SYCOMORE system code syntax. This modular code includes physics and technology models coupled to an optimizer in order to explore a large design parameter space, and currently it is used to provide an approximation model for demonstration fusion power plants (DEMO).

Once we accept this parameter window (Fig. 7), METIS shows the main edition window (Fig.8) where we can edit almost all options and parameters for the simulation as well as the waveform references computed in first place by the sycomore2metis module.

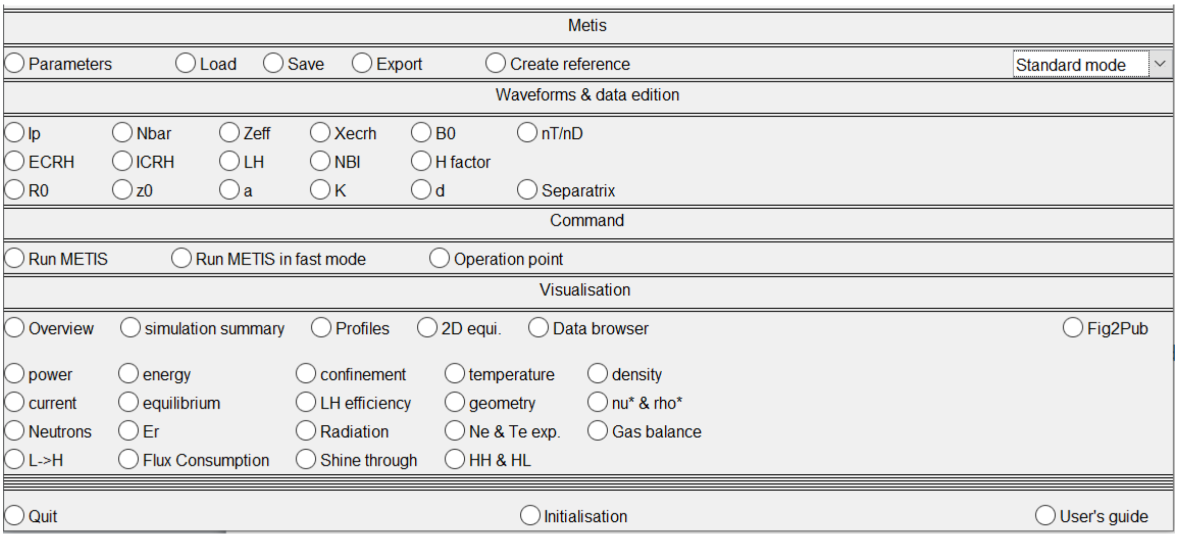


Fig. 8: METIS main interface window in standard mode

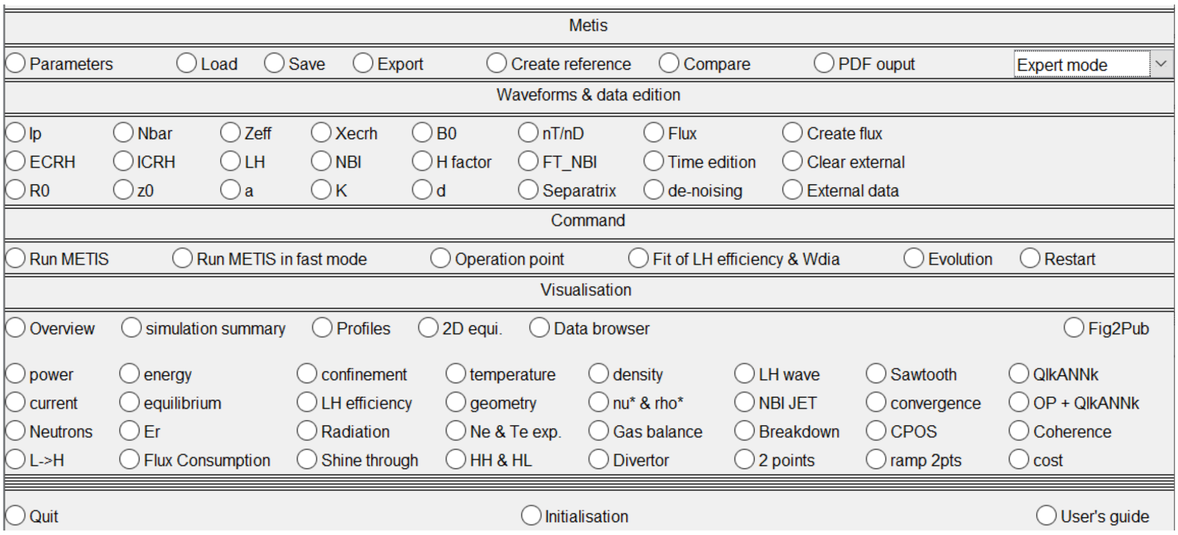


Fig. 9: METIS main interface window in expert mode

As we can see, METIS has two main modes, the standard (Fig. 8) and the expert (Fig. 9) mode. During this project we've ended up using some of the features from the expert mode, so we will be working from now on with it.

These input parameters, inside ***z0dinput*** structure, are classified in the next sub-structures:

- **option**: scalar parameters
- **info**: scalar parameters descriptions
- **zsinfo**: 0D data descriptions
- **profinfo**: profile descriptions
- **exp0d**: 0D data coming from measurements or from a CRONOS data set
- **cons**: references for the simulation
- **geo**: geometrical parameters of the plasma and the vacuum magnetic field
- **machine**: name of the tokamak
- **shot**: in case of real shot simulation, contains the shot number

In the first place, we are going to focus on the study of the ***z0dinput.cons*** sub-structure, since some of the waveforms it contains (named after the corresponding variable in the matfile save) are:

- time intervals and duration of the simulation, **temps** (only editable in expert mode)
- boundary condition for the current diffusion equations (plasma current (**ip**) or poloidal flux at LCFS (**flux**))
- injected power for heat sources (**pecrh**, **picrh**, **plh** (considered as pecrh2 in METIS output), **pnbi**)
- effective charge, **zeff**
- line-averaged density, **nbar**
- plasma geometry, (**a**, **r0**, **z0**, **K**, **d**)
- isotopic plasma composition, nT/nD (**iso**)
- confinement enhancement factor (**hmore**)

If we click the NBI option, two windows (Fig. 10-11) open showing the input reference for the two ITER neutral injectors, considered one as “real” and the other one as “imaginary” due to the injection direction in relation of the plasma.

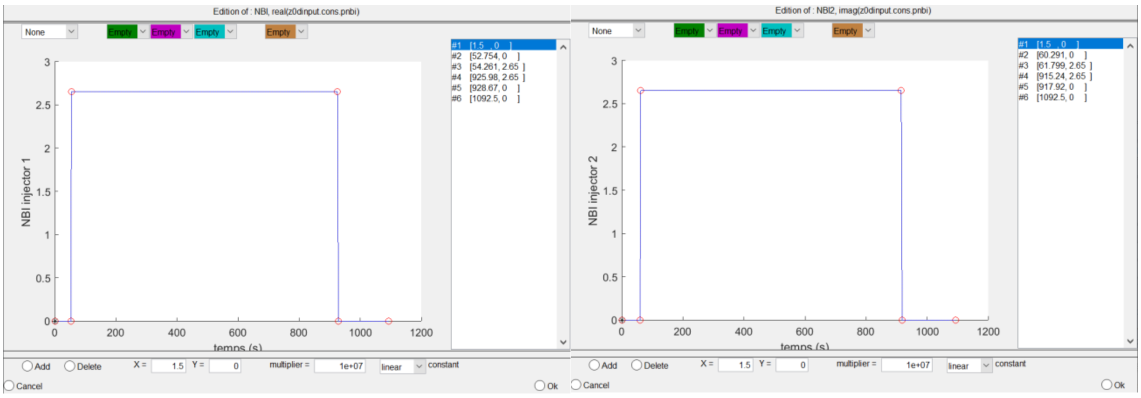


Fig. 10-11: Standard input NBI Power waveform, for first and second injectors, respectively

We can check how the combination of the two injectors provides 53 MW of power to the plasma, as specified in the “reactormetissimulation” window (Fig. 7), with the first parameters.

For educational purposes, we will see how we should edit some parameters and waveforms to simulate scenario 053, where we want to increase the neutral beam injection power during a long linear ramp-up and to change the standard way METIS has to determine the H-mode transition.

The first step, would be editing the neutrals power injection, clicking the NBI option and changing the value for some of the reference points. We are looking for a ramp-up starting from 0 MW at the same time as the original waveforms do, so the only change we must apply is to delete reference point number 3, through the option "Delete". Of course, if we'd like to add or edit any reference point, it would be enough to click on "Add" and, selecting it through the right list, change its X and Y values. This window also offers the possibility to approximate the reference points through linear, spline or pchip interpolation. After our changes, NBI waveforms should look like:

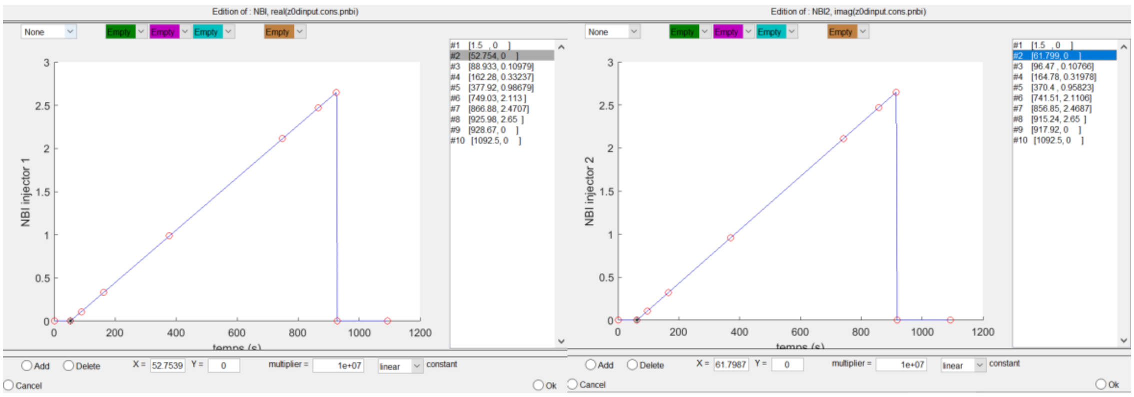
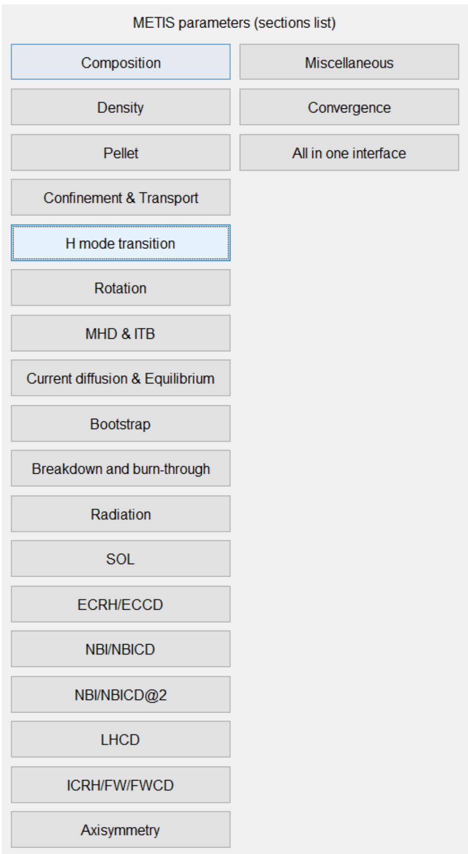


Fig. 12-13: Input NBI Power waveform edited for simulation 053, for first and second injectors, respectively

After this, we want to change the way of computation for the power (P_{lhthr}) compared to threshold power when calculating the H-mode. This option will only be available when working with the expert mode, and we find it clicking on "parameters" -> "H mode transition":



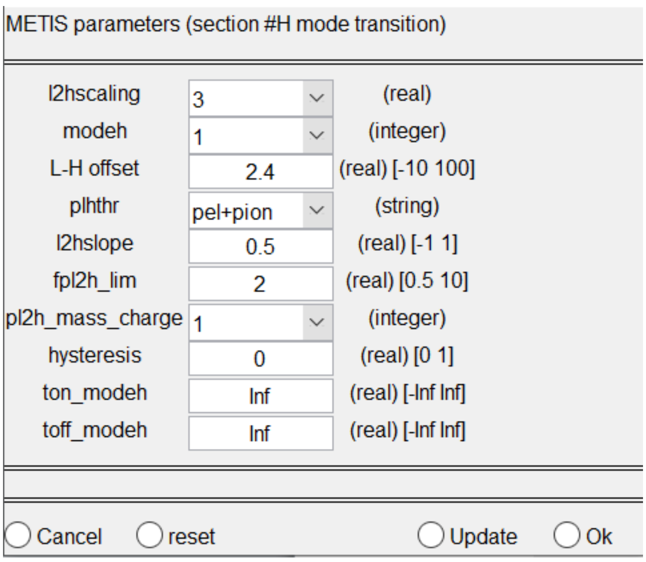


Fig. 14-15: Parameter selection list window (left) and the H-mode transition parameters editor (up)

Once inside this option window (Fig. 15), only one of the many METIS has to edit simulation parameters, the option we have to change is the “plhthr” to “pel + pion”, a scaling law defined as:

$$P_{lhthr} = P_{loss} = P_{el} + P_{ion} - f_{rad} \cdot P_{rad} \qquad (6.1)$$

where P_{el} and P_{ion} are the total thermal power deposition for electrons and ions, P_{rad} is the impurity radiation losses, without Bremsstrahlung and f_{rad} is a multiplicative constant applied to the line radiated power.

In this last window, we can also change the scaling law for the computing of $P_{lossl2h}$ and the offset value to calculate the threshold power, for example. There’s also the option to vary the “l2hslope” and “hysteresis” values.

Finally, to run the simulation we could execute four simulation models, from the “Command” options: “Run METIS”, “Run METIS in fast mode”, “Fit of LH efficiency & Wdia” and “Evolution”. The difference between these computation modes is treated in section 9. For this example, we will execute the Evolution run mode. This option will start a computing process that can last for over 10 minutes and, when finished, it will pop up the simulation overview output. For our simulation, it should look like this:

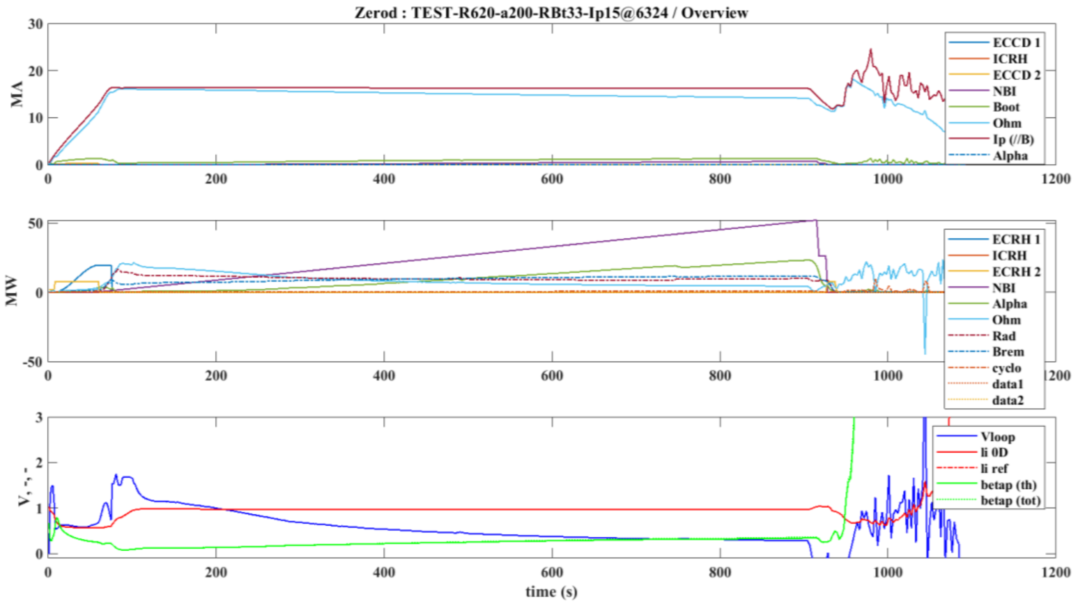


Fig. 16: METIS-053 overview

We can see how this simulation does not reach the H-mode, nor breakeven. So in terms of analysis, it does not show an interesting output. In spite of this, in the next section, we continue this guide talking about the output options, for more interesting simulations.

After running our simulation, we would be able to save all the data through the main interface, by clicking on "save" creating a MATLAB structure called ***post***, including a z0dinput copy, and the ***zerod*** (time-dependent output data) and ***profil0d*** (profile output data) substructures. We could also export this output through the "export" option for a following analysis through CRONOS.

7. Analysis of METIS simulations and results verification

All our simulations have been focused on ITER-like scenarios, so the first simulation we ran through METIS was of course the standard model presented by the program, without changing any input options, parameters or reference values, explained thoroughly in section 6. We started simulating using the RUN FAST mode of METIS, but as explained in section 9.6, we encountered huge differences and inconsistencies between the three modes we used (RUN FAST, RUN and EVOLUTION), so we went for this last one for the simulations we wanted to study due to its higher fidelity.

This first simulation was named "000". This numerical classification was intended for utility purposes concerning our MATLAB setup and a whole explanation on the way it works is treated in section 8.1. Moreover, a simulation log can be found in the Annex A.1 section to know the characteristics over each simulation. For this starting scenario each number corresponds, respectively, to: ITER standard (first "0"), reference (second "0") and standard simulation ("0"). As said before, this means no parameters or options were changed from the original ones METIS loads. In the next part, we show the options METIS has to show the output results from a simulation, and following, some particular analysis are explained.

7.1. METIS output visualisation

In section 6, we developed a quick guide oriented to run simulations of our interest, editing the input reference waveforms and parameters for a specific simulation. Now we are focusing on how to see and plot several outputs from METIS. The next outputs correspond to the previously mentioned simulation '000'.

When METIS ends a simulation run, shows the output overview, but we should be able to check many more quantities and plots from the program. A very useful tool to find specific variables is the "Data browser". By clicking on this option a window will open (Fig.17) giving us the opportunity to look for a determinate parameter or quantity through the "METIS 0D" or "METIS profiles" menus.

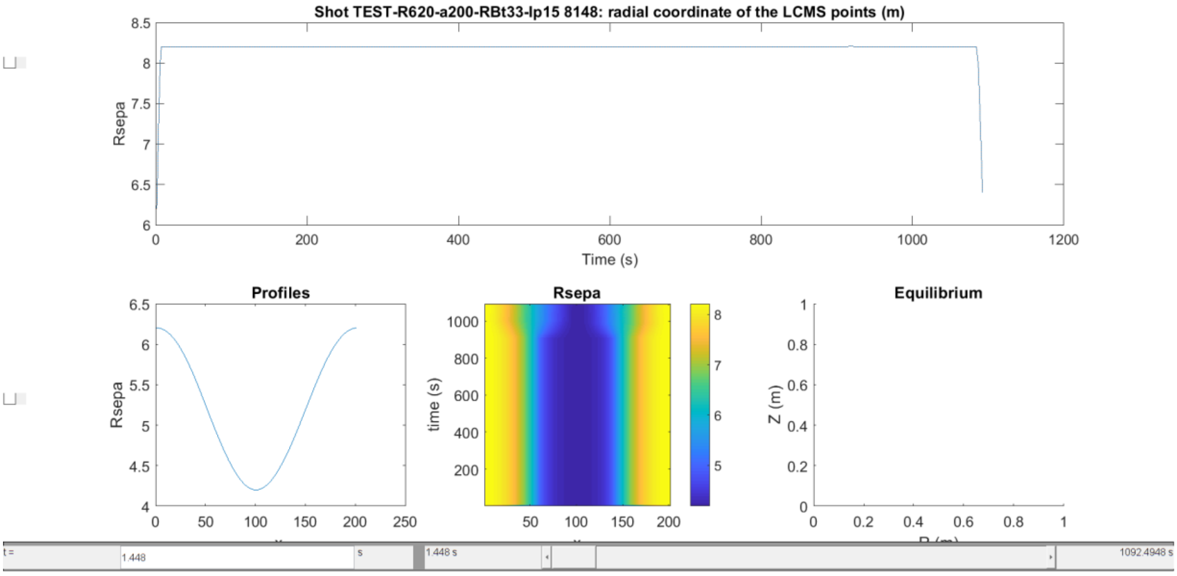


Fig. 17: METIS Data browser showing Rsepa evolution for simulation 000

We can see in Fig. 17, for example, the plot window from the Data browser showing the radial coordinate of the LCMS points (Rsepa), found through the "METIS profiles" menu. The Data browser has been a very useful tool to identify certain quantities and parameters we needed when trying to understand what each variable meant and represented from the simulation and, overall, when working with the output post structure through MATLAB, where the variables are not sorted.

Apart from the overview output, which we can replot clicking on the "Overview" option from the main METIS interface, we could use the "Simulation summary" option, which shows the plots of some waveform references from the input data and asks us to choose through our cursor a time slice for the profiles visualization (let's choose some instant from the flat-top, around 600 s) and three other time slices for the evolution of the safety factor, (for example, one during the ramp-up, one during the flat-top, and another just beginning the ramp-down.

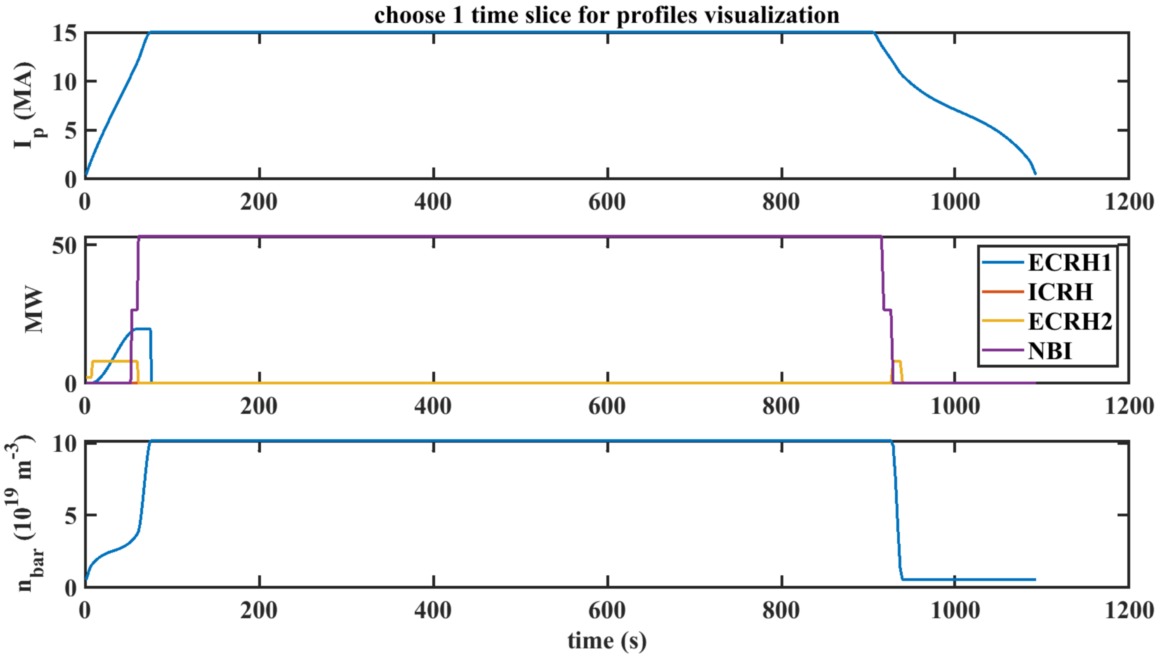


Fig. 18: METIS-000 simulation summary ready for time slicing

Several figures will appear, including the safety factor evolution vs time as well as for the three chosen time slices, some quantities over profiles for the first time slice and some of the main quantities like plasma current, fusion or auxiliary powers through the whole simulation.

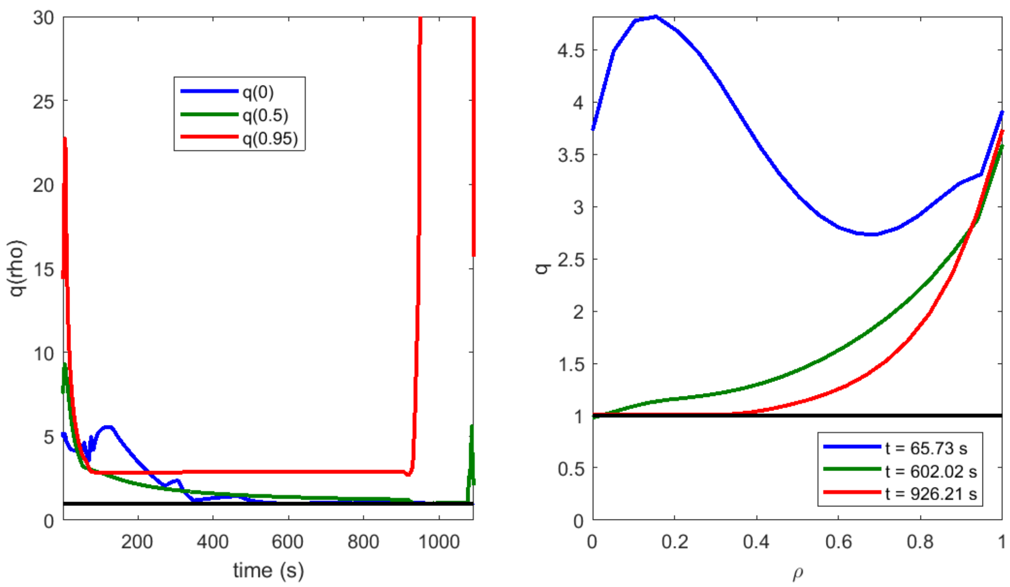


Fig. 19-20: METIS-000 safety factor through time (left) and three time-slices safety factor through profiles (right)

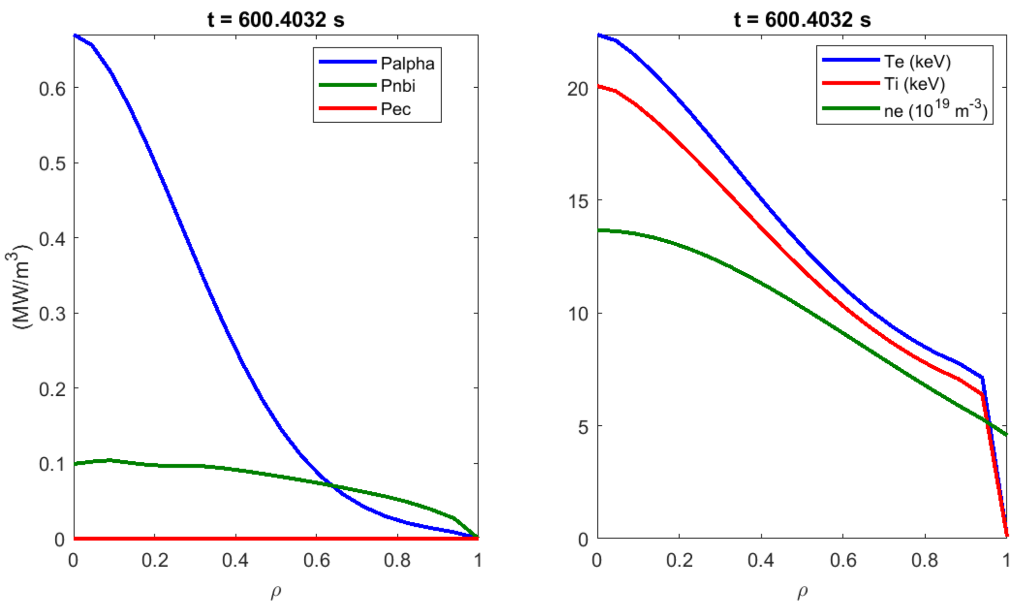


Fig. 21-22: METIS-000 fusion, neutral and EC powers (left), electron and ion temperatures, and electron density (right) through profiles at t = 600.4032 s

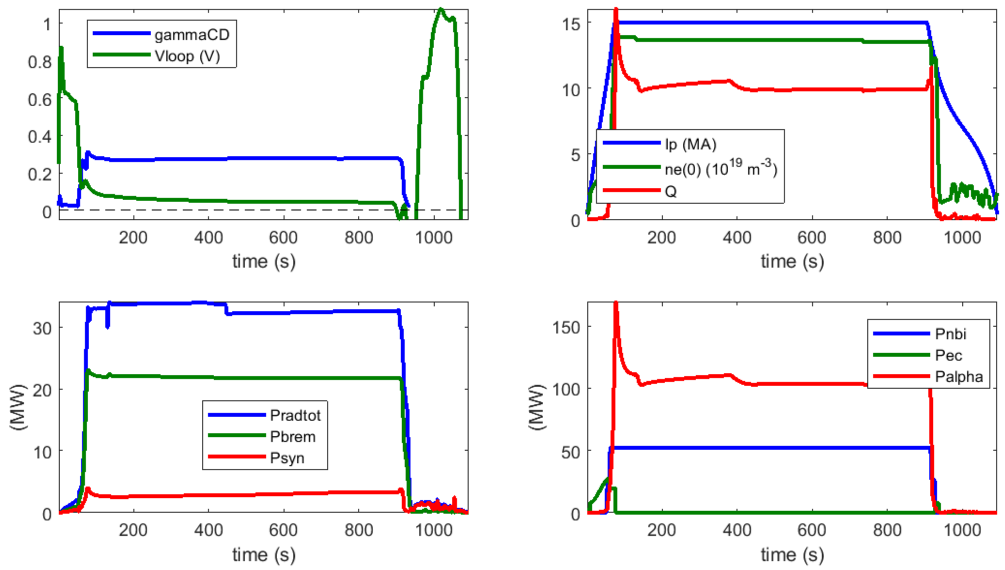


Fig. 23-24: METIS-000 different quantities through time

The last worthy plots to mention in this project are the ones related with the H-mode transition, due to the important paper it has had in our study of METIS. We can find them clicking the "confinement" visualisation option, resulting in the next multi-figure:

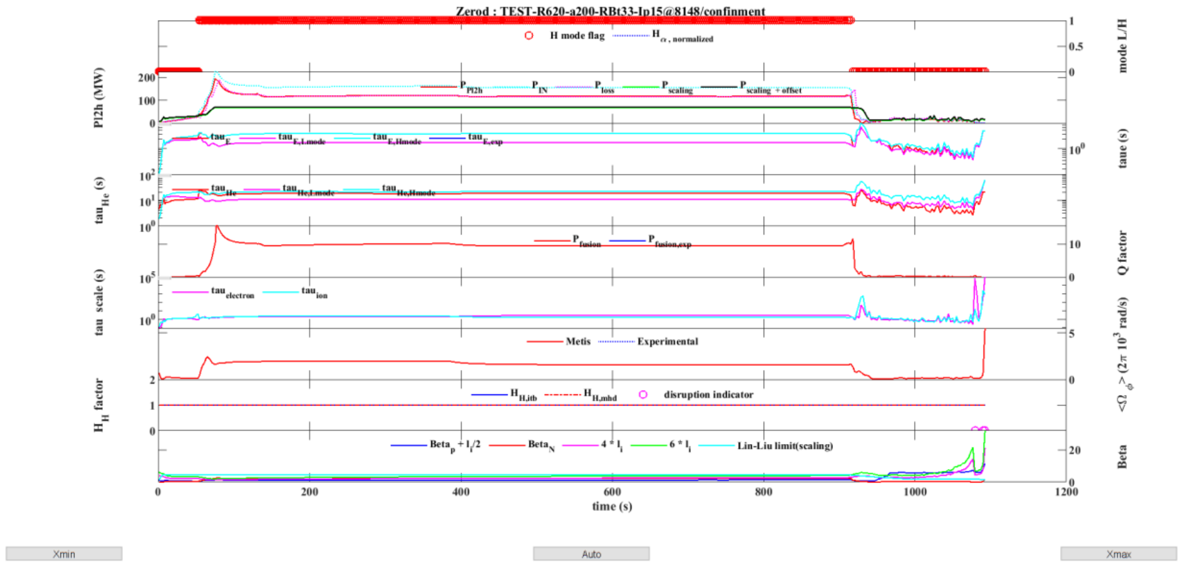


Fig. 25: METIS-000 confinement plots

As we can see, this format is a bit difficult to understand as there is a lot of information too squeezed for only one window. To separate this figure into single plots, we can use the "Fig2pub" option from the main window, selecting the figure we want to subdivide and we will have some better quality outputs. Some of the interesting ones are the H-mode flags, the threshold power comparison or the energy confinement times:

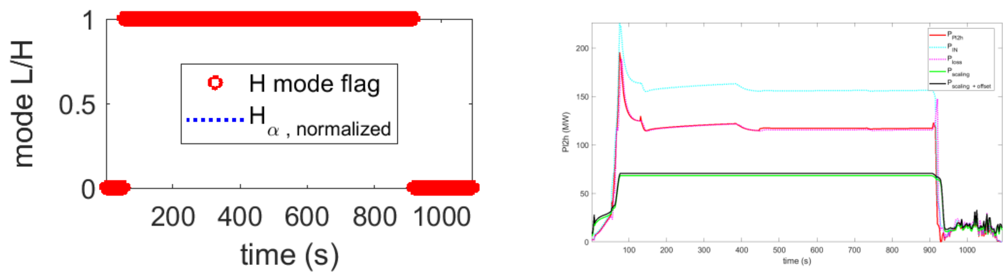


Fig. 26-27: METIS-000 H-mode flag (left) and threshold power comparison (right)

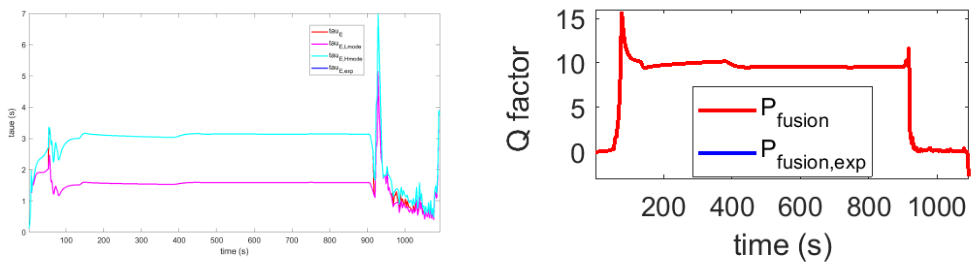


Fig. 28-29: METIS-000 energy confinement times (left) and gain factor (right)

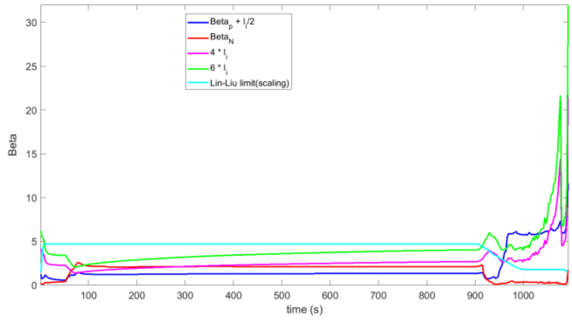


Fig. 30: METIS-000 plasma pressure

These are only some of the options we have used during the project, but we hope this guide helps anybody who needs to get inside METIS for the first time. Done this, we now can begin some simulation analysis with the goal to check the assumptions made in section 5.2 for the different events of interest.

It is important to have in mind that we wanted to be sure we understood how METIS worked before jumping into wrong conclusions during the simulations analysis and this is why we started down this path.

7.2. Simulation 000: Standard ITER simulation

The input values for simulation 000, are the default values given by METIS for an ITER simulation. We can see some of them in Fig. 7, like a plasma current (Ip) of 15 MA, the ramp-up factor (rampup_dipdt = 1), or the duration of the ramp-up and the flat-top (400 s). The waveform values for input powers are P_{NBI} = 53 MW, P_{ECRH} = 20 MW during ramp-up, a null P_{ICRH} and P_{LH} = 8 MW for the ramp-up and the beginning of the ramp-down.

This simulation’s overview (Fig.31) shows a really clear flat-top phase after a ramp-up during which both breakeven and H-mode transition happen. We checked these events using the METIS output and our own MATLAB function for the simulation overview (*sim_overview*), explained in section 8:

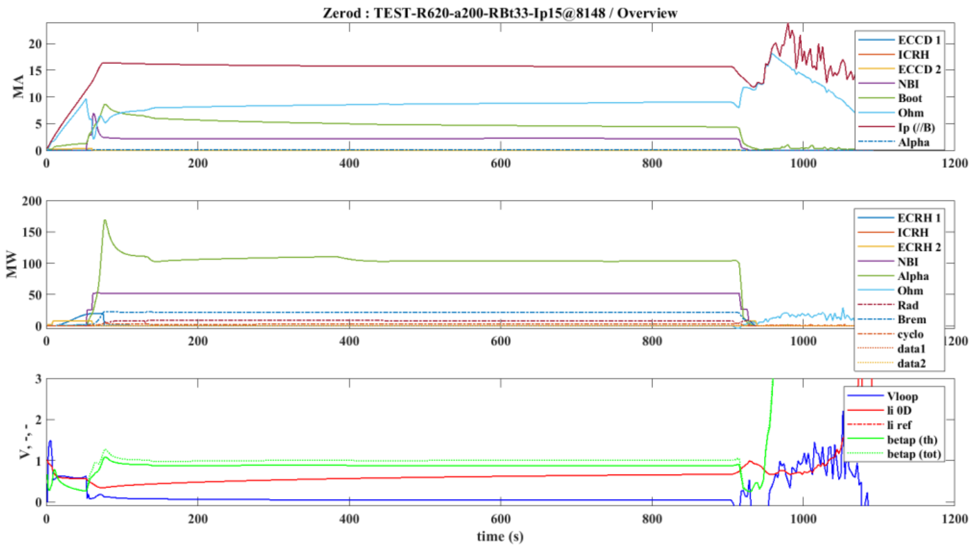


Fig.31: METIS-000 overview

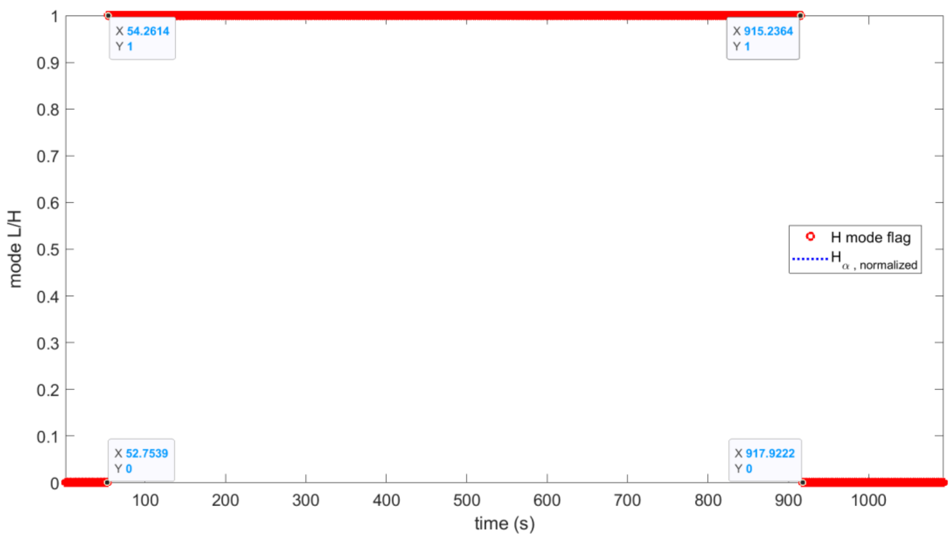


Fig. 32: METIS-000 H-mode flag

In Fig. 32 we can see the H-mode transitions according to METIS, so we should see in the threshold power plot (Fig. 33-35) two intervals of time corresponding with the ones from the H-mode flag where P_{lhthr} (named P_{Pl2h} in METIS Fig. 33) is over or under $P_{threshold} = P_{scaling} + Offset$,for each transition respectively. We should check then for the red line (P_{lhthr}) crossing the black line ($P_{threshold}$).

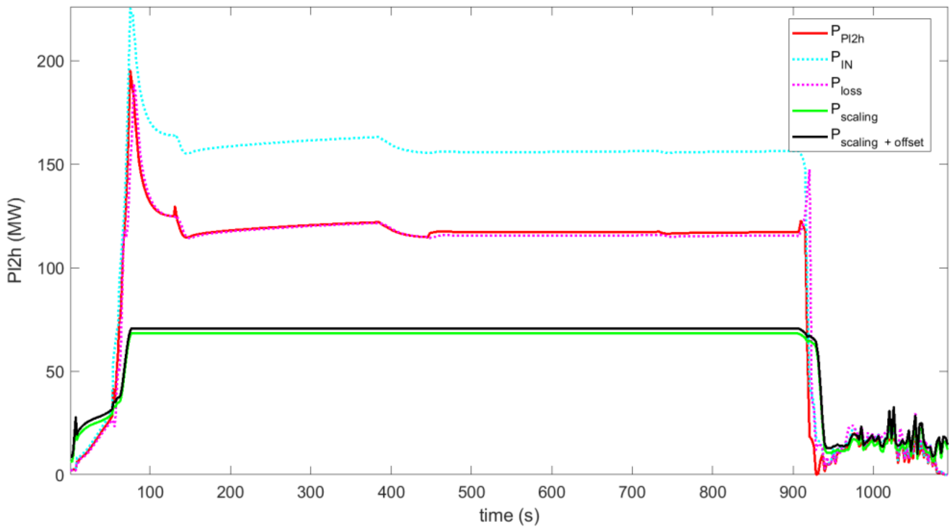


Fig. 33: METIS-000 powers comparison to threshold power

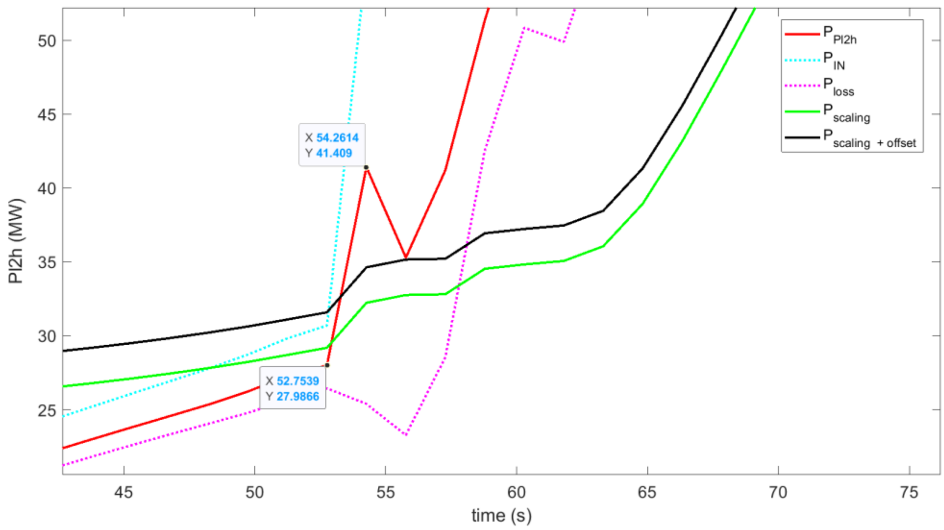


Fig. 34: METIS-000 H-mode transition

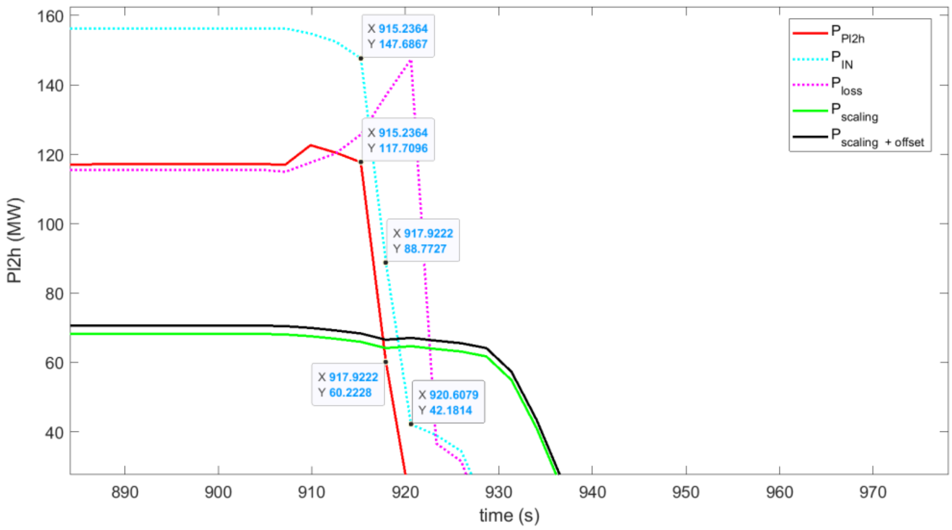


Fig. 35: METIS-000 H-mode loss transition

As we can see in Fig. 34-35, the times match with the H-mode plot (Fig. 32) for both the achievement of the H-regime and its loss. As for hysteresis, the time obtained for the transition, works as expected and works only with P_{lhthr} as mentioned in section 5.2.3. We should now compare these results with our MATLAB setup outputs, where we used the calculations explained in section 5.2. To get our own results for each simulation, and to compare and understand them better, we have programmed new functions with MATLAB that are described in section 8. The main one though, is *sim_overview*, through which we can obtain the next figures.

```
>> sim_overview('000',1)
Key times:
Starting time of the simulation: 1.5 s @ Ip = 0.4 MA
X-point formation: 7.61234 s @ Ip = 1.9 MA
Full power time: 60.8987 s @ Ip = 12 MA
Start of flat-top: 76.1234 s @ Ip = 15 MA
End of full power assited phase: 304.493 s @ Ip = 15 MA
End of flat-top: 904.493 s @ Ip = 15 MA
H to L back transition: 938.734 s @ Ip = 10.6066 MA
back transition to limiter: 1084.88 s @ Ip = 1.9 MA
End of simulation/plasma  : 1092.49 s @ Ip = 0.4 MA
L->H: 52.7539 [s]
H->L: 915.2364 [s]
Breakeven achieved
Breakeven: 70.8435 [s] ; 72.5053 [Padd, MW] ;83.1757 [Palpha, MW] ;
Qavg = 9.6712

ans =

  4×2 string array

    "Max Palpha = 169.1855"    "MW"
    "Max te0 = 31.6981"        "KeV"
    "Max ne0 = 13.8416"        "1e+19 1/m^3"
    "Max taue = 5.148"         "NaN"
```

Fig. 36: MATLAB-000 simulation summary

In Fig. 36 we can see the MATLAB command window output obtained by the call to the function *sim_overview* for the simulation 000. We are now focusing on the time results for the H-mode transitions, which are the same as the ones seen in METIS outputs, and the average value of the gain factor during the flat-top phase for a simulation with a Breakeven achieved, which is very close to ITER's objective of Q=10.

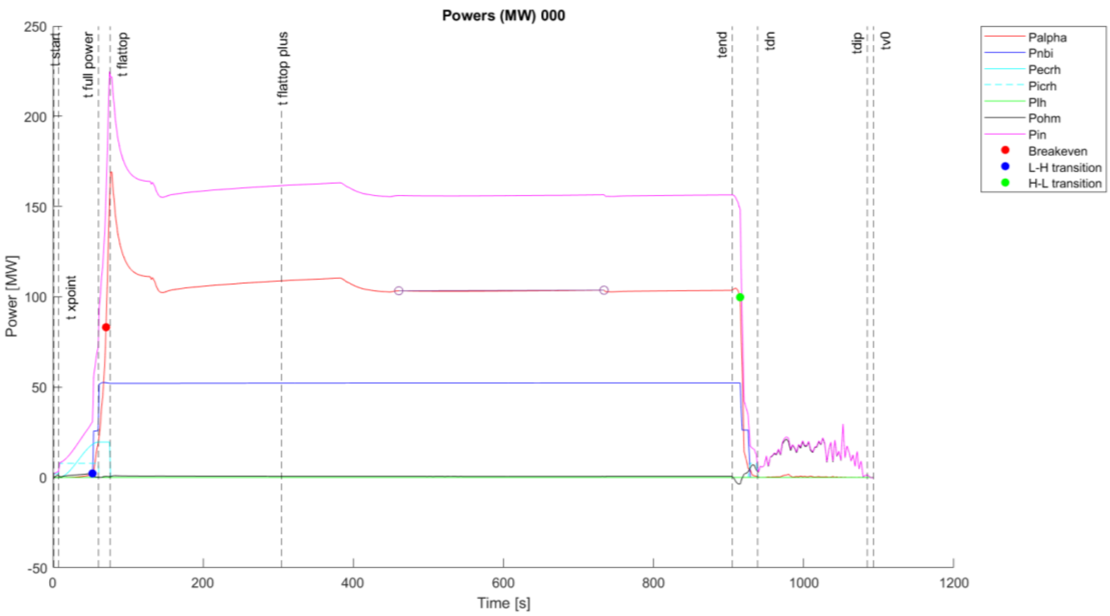


Fig. 37: MATLAB-000 powers and events

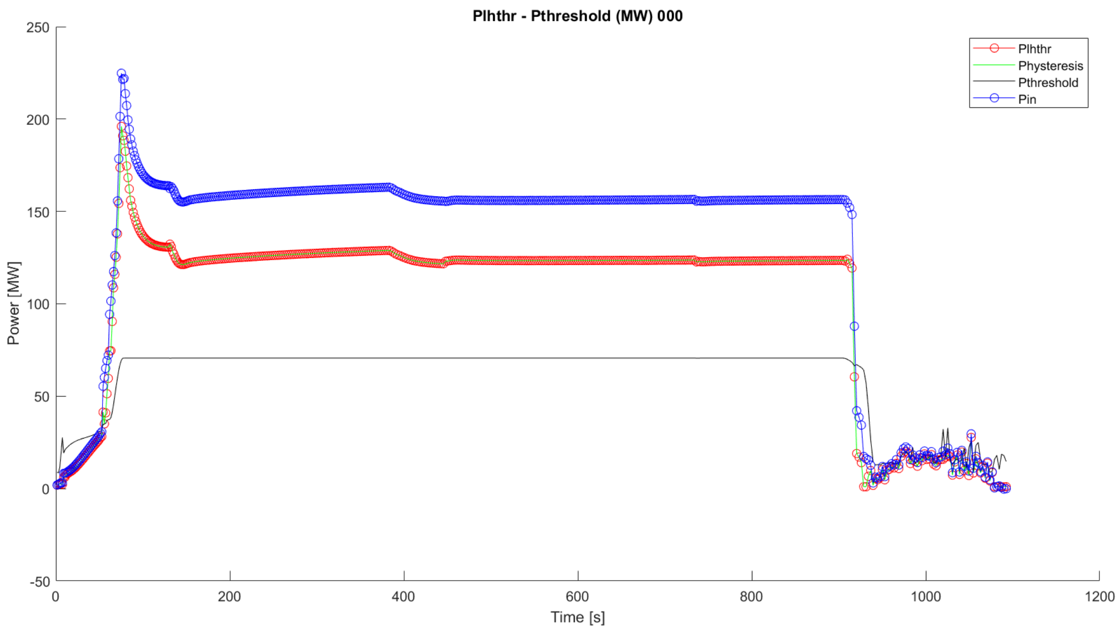


Fig. 38: MATLAB-000 powers comparison to threshold power

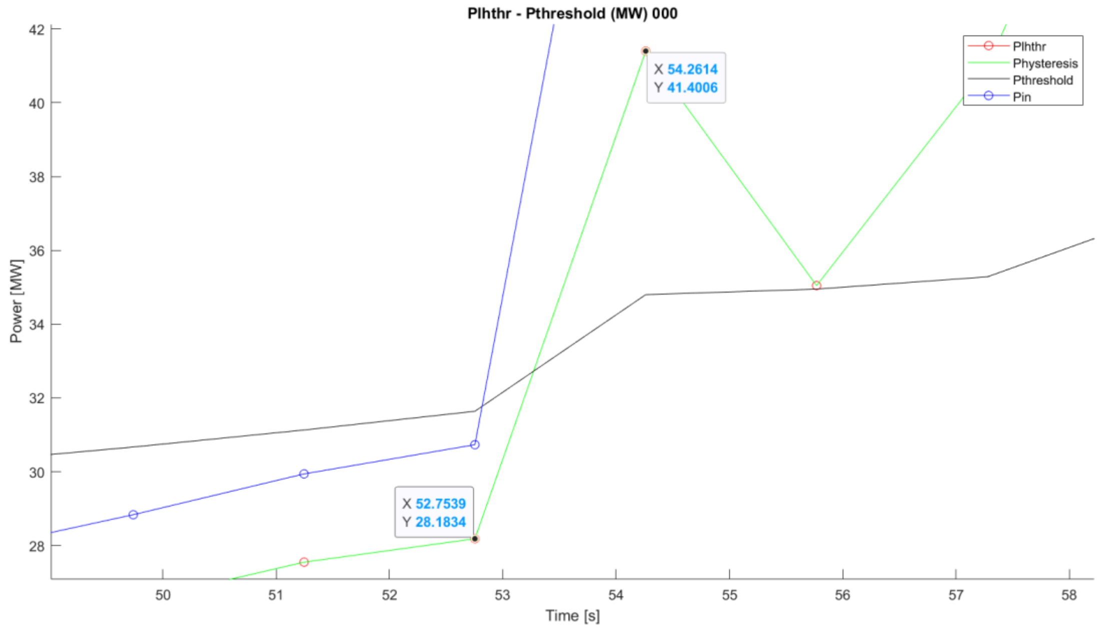


Fig. 39: MATLAB-000 H-mode transition

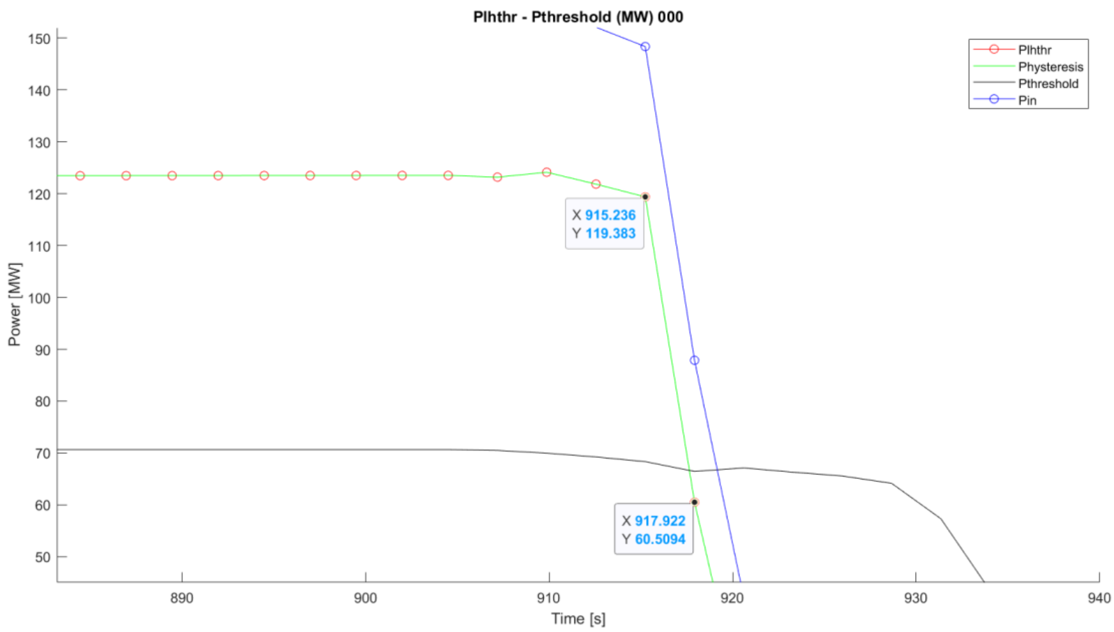


Fig. 40: MATLAB-000 H-mode loss transition

We can check how the plots comparing our own computed threshold power and the loss of power through the last flux surface (P_{lhthr}) (Fig. 38-40) are really similar to the METIS plots (Fig. 33-35) and work as well to give an accurate answer for H-mode transitions.

7.3. Simulation 032: P_{NBI} = 26.5 MW

In simulation 032, the neutrals power injected is half the original from simulation 000. The other parameters and waveforms are the same. With this simulation we are trying to check what happens if the H-mode is achieved but the plasma has not enough input power to sustain it.

From now on, we'll repeat the previous comparison process in order to have a variety wide enough to be sure this works for most of the simulations, focusing on the H-mode.

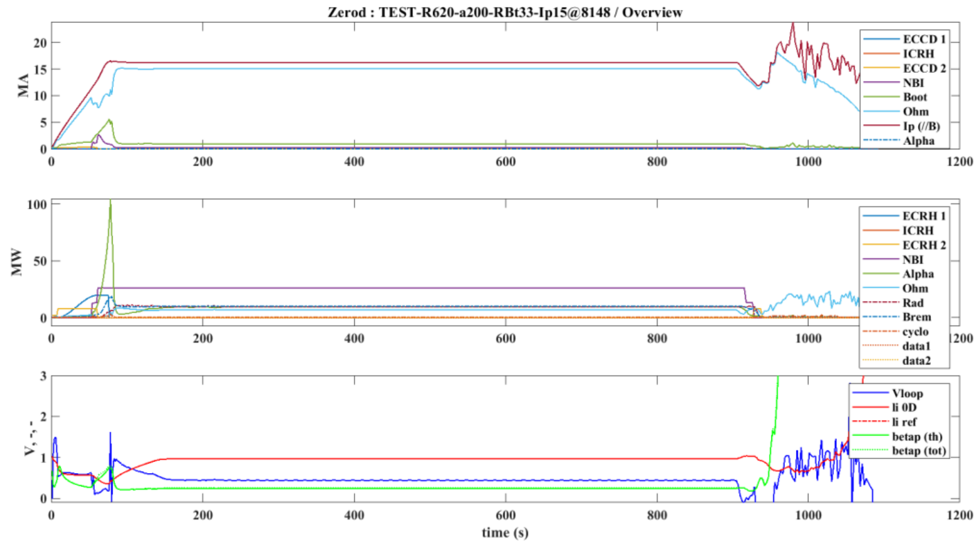


Fig. 41: METIS-032 overview

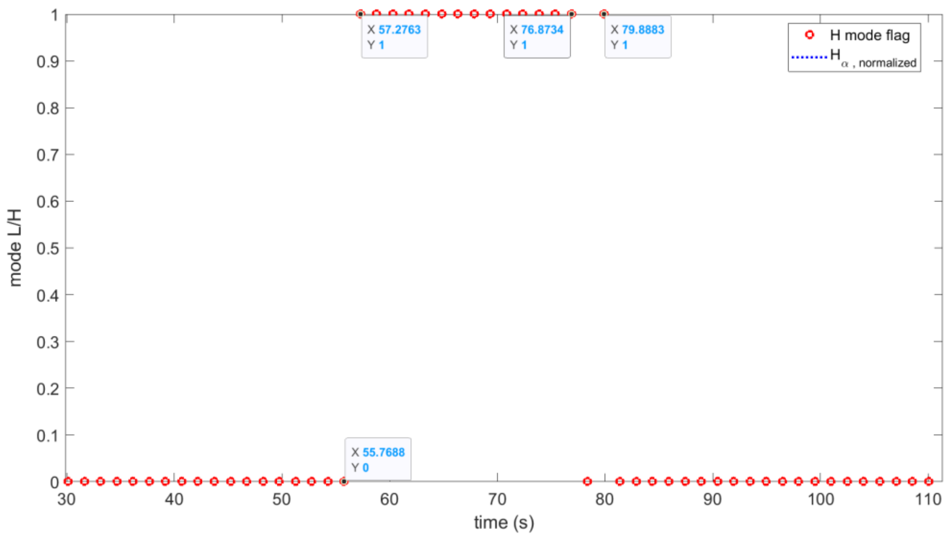


Fig. 42: METIS-032 H-mode flag

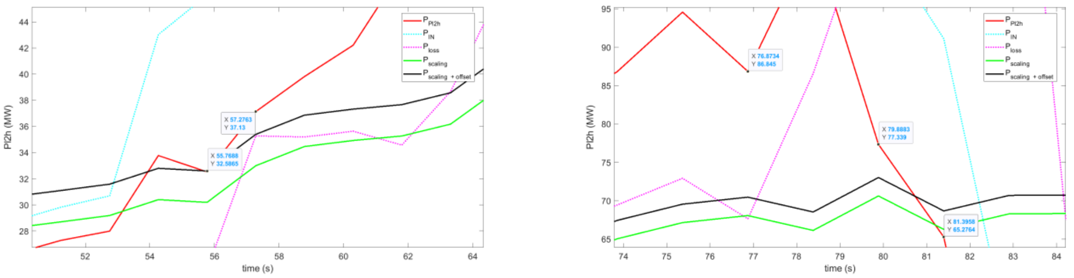


Fig. 43-44: METIS-032 H-mode transitions: achievement (left) and loss (right)

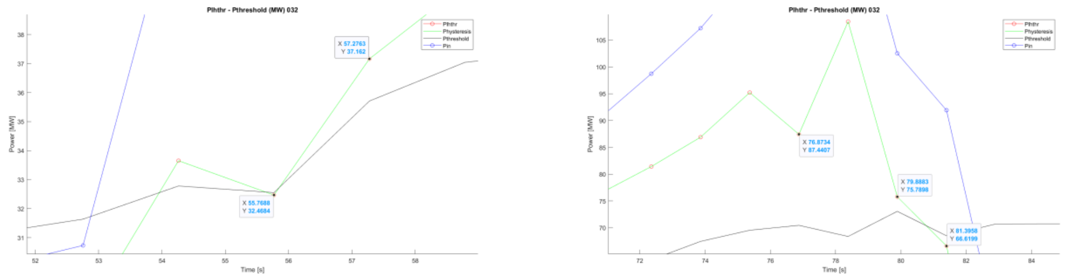


Fig. 45-46: MATLAB-032 H-mode transitions: achievement (left) and loss (right)

In this case, there's a first trespassing of P_{lhthr} (red line) over the threshold power (black line) in Fig. 43, before the H-mode flag shows the transition to the H confinement mode (Fig. 42). This happens because of the $l2h_{slope}$ parameter, which holds back the transition until the value of P_{lhthr} is well above the threshold power.

We can also see how METIS identifies a quick loss of the H-mode at t = 76.87 s, when P_{lhthr} doesn't get close enough to the threshold power. This wasn't an expected result, because it didn't follow the hypothesis we stated in section 5.2.3. We decided to run some other simulations, like 033 analysed in the next section, to check if this kept happening.

7.4. Simulation 033: P_{NBI} = 39.75 MW

Simulation 033 is the next variant after 032 for the increasing P_{NBI} series of simulations; in this case, the neutrals power injected is at 75% of the standard simulation's P_{NBI}. The other parameters and waveforms are the same. In this simulation we see (Fig. 47) how the plasma enters the H-mode during the ramp-up and it sustains it for over 100 s before losing it.

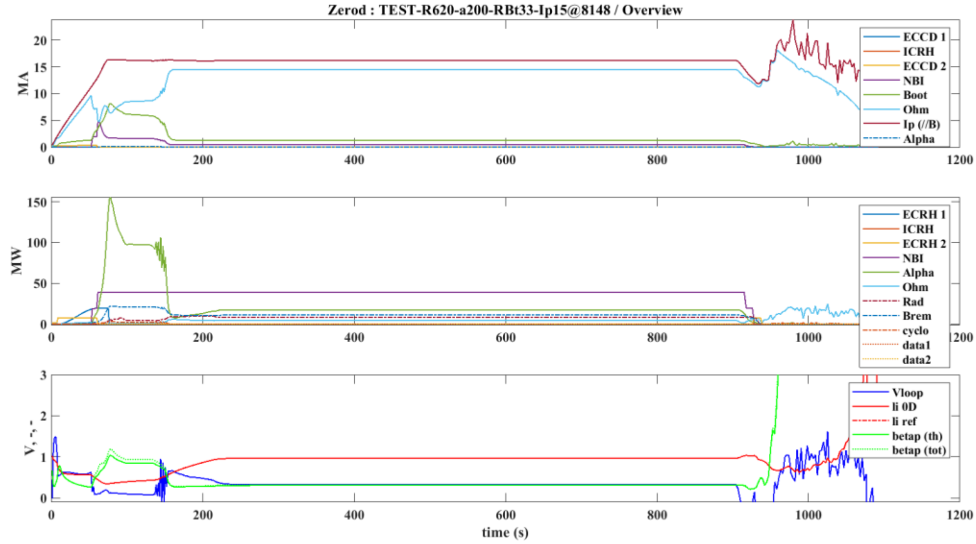


Fig. 47: METIS-033 overview

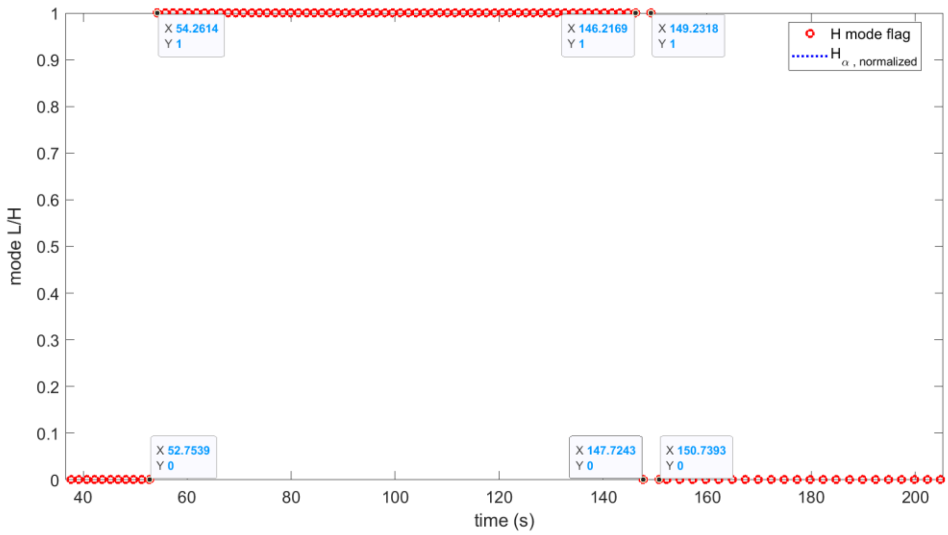


Fig. 48: METIS-033 H-mode flag

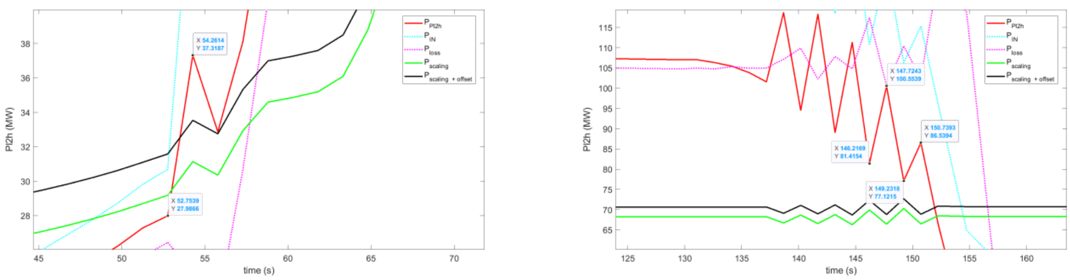


Fig. 49-50: METIS-033 H-mode transitions: achievement (left) and loss (right)

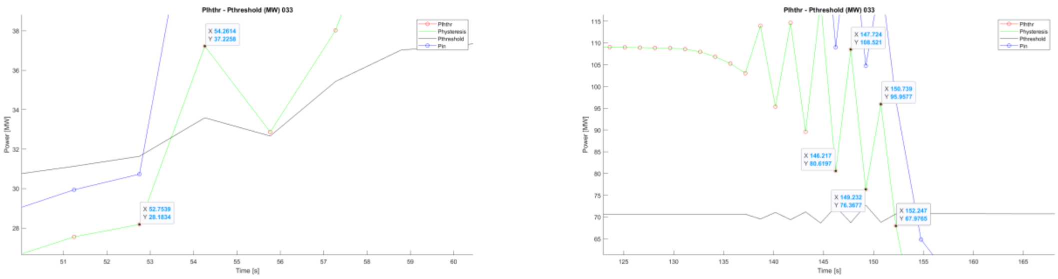


Fig. 51-52: MATLAB-033 H-mode transitions: achievement (left) and loss (right)

For this last scenario, we can see a situation in Fig. 52 similar to the previous simulation. During the loss transition of the H-mode, there's a first flag (Fig.48) at t = 147.72 s, in a moment where clearly P_{lhthr} hasn't reached the threshold power and in t = 150.74 s it definitely losses the H-mode confinement without even having crossed the threshold.

In this moment we decided to check through the source code how METIS actually computed the H-mode. We found out that the lines of code in charge did what we assumed through our hypothesis and the indications in the user guide [1]. This was a difficult moment during the project because we needed to decide whether keep going forward and delve into the source code, meaning changing our plans on studying more simulations on different scenarios, or go ahead and simulate new scenarios. We were not comfortable with this last option, because it meant that we would probably have jumped into wrong conclusions. It was clear that something was missing in our idea of how METIS worked; we weren't sure how it computed any of its output data. Therefore, we decided to follow each step METIS did in the process of running a simulation.

This more technical explanation on how the input data was generated and each computation mode for METIS worked, is split between the METIS input guide in section 6, and the METIS data computation in section 9.

The goal by taking this decision was, once cleared this problem, to analyse our simulations with more confidence. Finally, this analysis of METIS code took us all the time from our project and it became an important priority in it, so studying more reactor scenarios will have to wait for another project following all the knowledge acquired during our work. We must say though, that in the process of studying METIS outputs during this section, the results obtained through our MATLAB setup were very close to the ones from METIS and sufficient to give good predictions for the events we treated.

We used this new MATLAB setup to compare the original results with our approximations following the knowledge we kept acquiring while studying the source code. Some examples of our results have been shown in this section (Fig. 36-40). In the next section, we explain thoroughly this set of new functions.

8. MATLAB setup for METIS output analysis

One of the main objectives we had in mind when starting this project, was to, not only be able to comprehend METIS output and study it through MATLAB, but also create a set of tools that would help us to analyse these results. Then, to compare our assumptions about how METIS works with the output data obtained for every simulation, we needed a known and useful MATLAB workspace to compute our own results and check if we could confirm our entire hypothesis.

The next set of functions was created and updated during the project with the objective of returning an easy visual representation of some of the parameters and quantities we were interested from METIS, like input and output powers or the instants when breakeven, ignition or H-mode are achieved. Above all, the option of working outside METIS gives us the opportunity to compare two or more different simulations, either plotting the relation between different quantities like temperature and density for different input powers, or comparing how the change of a parameter can vary the behaviour of the simulations, like the ramp-up factor.

For the creation of this setup we have gotten inside METIS source code to understand how certain calculations were made, and by doing this, we have learnt a way to introduce new features inside METIS. This possibility is presented in section 10, introducing an example of a new scaling law for the threshold power used for the determination of H-mode.

8.1. New MATLAB functions

sim_overview
<u>Function description</u>: This is our main module, through which we analyse the results from a determinate METIS simulation. The most important trait of this function, (of our setup as well), is that we compute certain quantities the same way METIS does as an example of how we could work with the source code of METIS in the future. This way, we could study certain events for academic purposes, like the ones mentioned in section 5.2, and add new ways of computing METIS outputs. As a main function, it calls several different modules, computing the different "achievements" of the simulation, like ***plasma_ignition*** or ***L_H_mode***. If the **extraplot** option is activated, the function plots some more information like the threshold and scaling powers comparison, the key times estimation (explained in section 9.1) or the steady state for the output quantities (in case of ignition).

Input parameters: **sim** (simulation of study, in string format and with our three number code naming system), [**extraplot** (logical 1 or 0, whether we want extra plots from secondary modules), **ignore_null_padd** (logical 1 or 0, whether we'd like to ignore the condition of Q = infinity for the achievement of ignition), **ip** (plasma current (MA)), **rampup_dipdt_factor** (factor applied to ramp-up rate current increase), **rampdown_dipdt_factor** (factor applied to ramp-down rate current decrease), **duration** (shot duration including ramp-up and flat-top, without ramp-down)].

Output: text and plots.

Example:

```
>> sim_overview('000',1)
Key times:
Starting time of the simulation: 1.5 s @ Ip = 0.4 MA
X-point formation: 7.61234 s @ Ip = 1.9 MA
Full power time: 60.8987 s @ Ip = 12 MA
Start of flat-top: 76.1234 s @ Ip = 15 MA
End of full power assited phase: 304.493 s @ Ip = 15 MA
End of flat-top: 904.493 s @ Ip = 15 MA
H to L back transition: 938.734 s @ Ip = 10.6066 MA
back transition to limiter: 1084.88 s @ Ip = 1.9 MA
End of simulation/plasma  : 1092.49 s @ Ip = 0.4 MA
L->H: 52.7539 [s]
H->L: 915.2364 [s]
Breakeven achieved
Breakeven: 70.8435 [s] ; 72.5053 [Padd, MW] ;83.1757 [Palpha, MW] ;
Qavg = 9.6712

ans =

  4×2 string array

    "Max Palpha = 169.1855"    "MW"
    "Max te0 = 31.6981"        "KeV"
    "Max ne0 = 13.8416"        "1e+19 1/m^3"
    "Max taue = 5.148"         "NaN"
```

Fig. 53: MATLAB-000 simulation summary

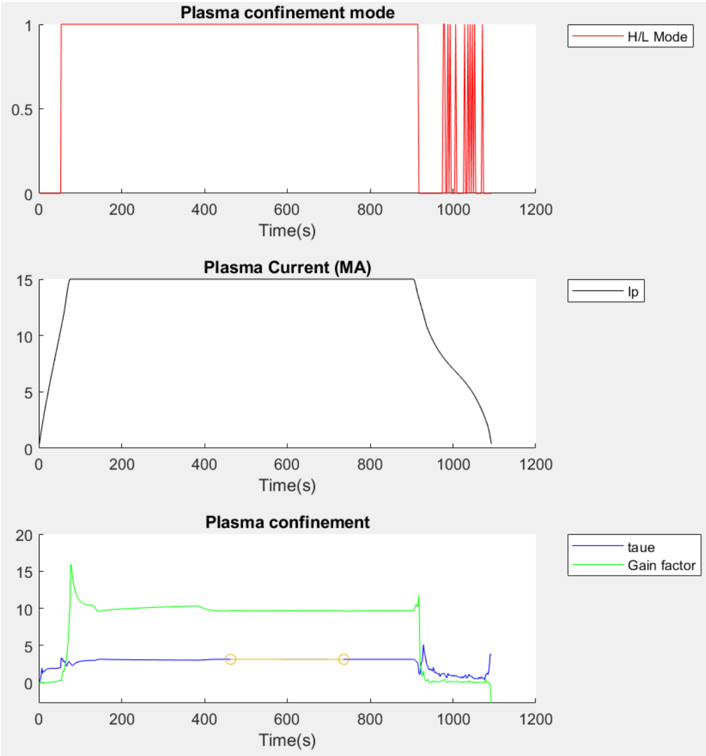


Fig. 54: MATLAB-000 simulation overview

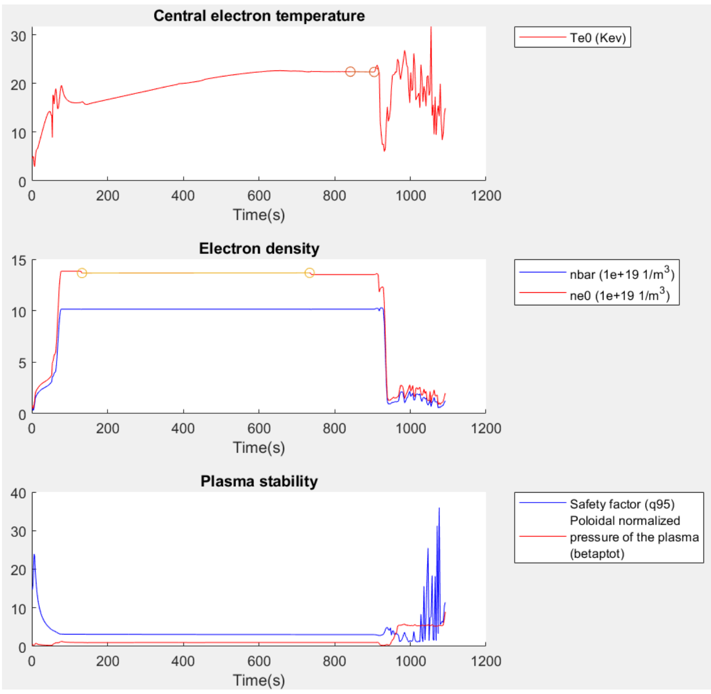


Fig. 55: MATLAB-000 simulation overview

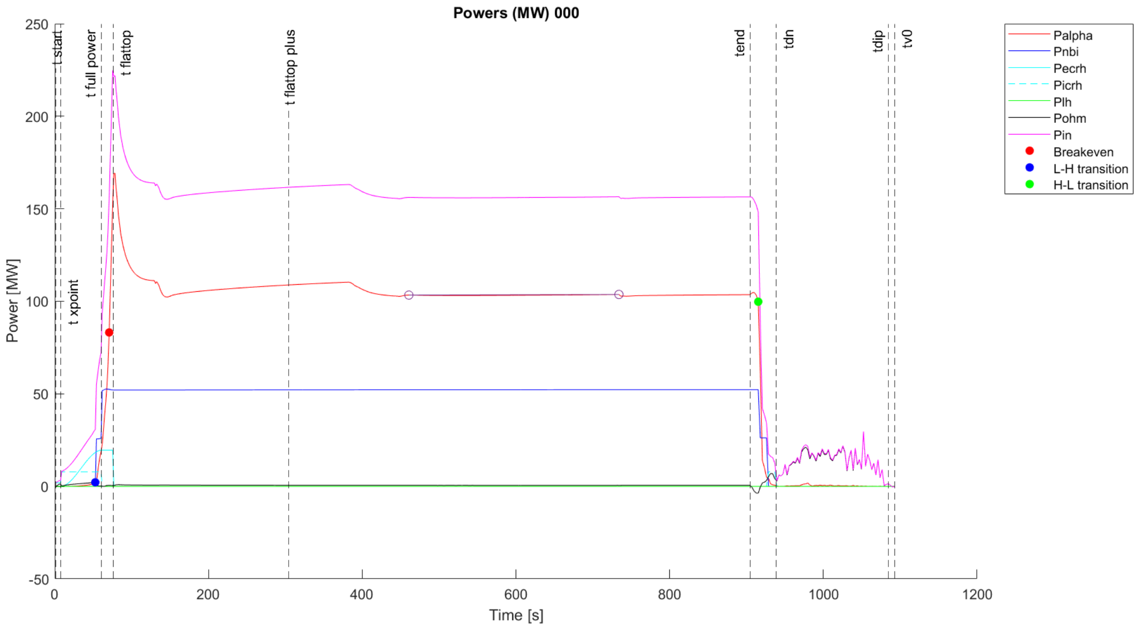


Fig. 56: MATLAB-000 powers and events

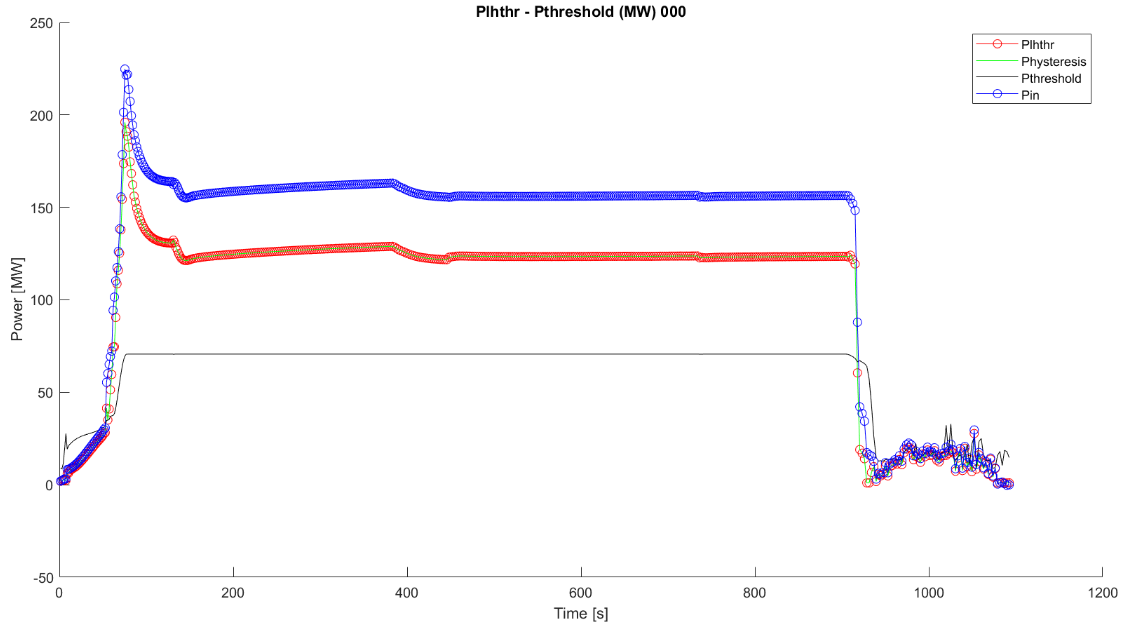


Fig. 57: MATLAB-000 powers comparison to threshold power

powers
Function description: Returns a list of powers (**P**), obtained directly from METIS output structure, all in MW; if required, returns the gain factor (**Q**) as it is computed by METIS.
Input parameters: **sim** (simulation of study, in string format and with our three number code naming system).
Output: **P** (array, Palpha, Pnbi, Pecrh, Picrh, Plh, Pohm, Prad, Pcyclo, Pbrem, Pioniz, Pin, Padd), **Q** (float, gain factor).

plasma_ignition
Function description: This function returns whether a discharge reaches or not the plasma ignition or the breakeven, considering the power balance and the Lawson criterion. If any of these two events happen, returns the index of the instant of time when it occurs. To avoid getting these events during the ramp-down phase, where we have seen some inconsistencies, we have limited the possible range of time where they may occur to a range previous to the maximum Pin or Padd and before the end of the estimated H-mode loss transition (tdn).
Input parameters: **sim** (simulation of study, in string format and with our three number code naming system), [**ignore_null_padd** (logical 1 or 0, whether we'd like to ignore the condition of Q = infinity for the achievement of ignition)].
Output: **achievement** (string, "Not breakeven"/"Breakeven"/"Ignition"), **ins** (float, index of time).

L_H_mode
Function description: This module determines the plasma confinement regime, whether it's L or H, evaluating and comparing the loss power to the threshold value through a determinate scaling law (option.plhthr). Returns an array with the index of time instants of the simulation where a transition from L to H (**L_H**) or H to L (**H_L**) occurs. If required, it can return the complete array of H-mode logical flags (**modeh**).
Input parameters: **sim** (simulation of study, in string format and with our three number code

naming system), [**extraplot** (logical 1 or 0, whether we want extra plots from secondary modules)].

Output: **L_H** (array, index of time), **H_L** (array, index of time), **modeh** (array, H-mode flags).

Plossl2h

Function description: This function computes the scaling power used for the calculation of the threshold power in the determination of the H-mode, depending on which scaling law is used (option.l2hscaling).

Input parameters: **sim** (simulation of study, in string format and with our three number code naming system), [**extraplot** (logical 1 or 0, whether we want extra plots from secondary modules)].

Output: **plossl2h** (array, scaling power values),

Example:

```
>> plossl2h = Plossl2h('000',1)

plossl2h =

   1.0e+07 *

    0.6368
    0.6380
    0.6380
    0.6152
    0.9062
    1.6769
    2.5241
    1.6822
    1.8607
    1.9472
    2.0202
    2.0852
```

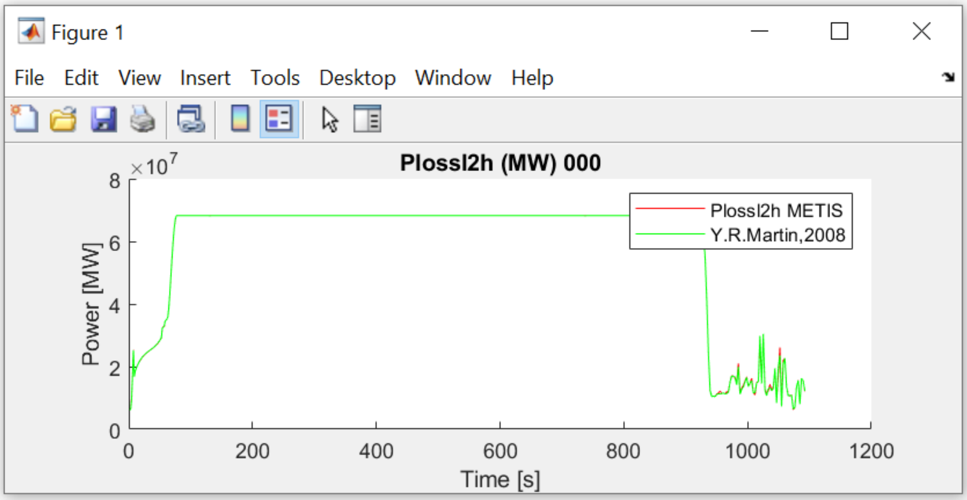


Fig. 58: MATLAB-000 Plossl2h or Pscaling

steady_state

Function description: This function's purpose is to calculate and return the time interval of the discharge for which a certain output variable reaches and sustains a steady state given a determinate percentage tolerance. If it does, returns the average value of the variable during this steady state.

Input parameters: **sim** (simulation of study, in string format and with our three number code naming system), **var** (array containing values for the variable of study), **tol** (var tolerance (%) accepted between consecutive time intervals to determine the steady state).

Output: **int** (array, time interval), **int_var** (array, value interval), **avg** (float, average value).

Example:

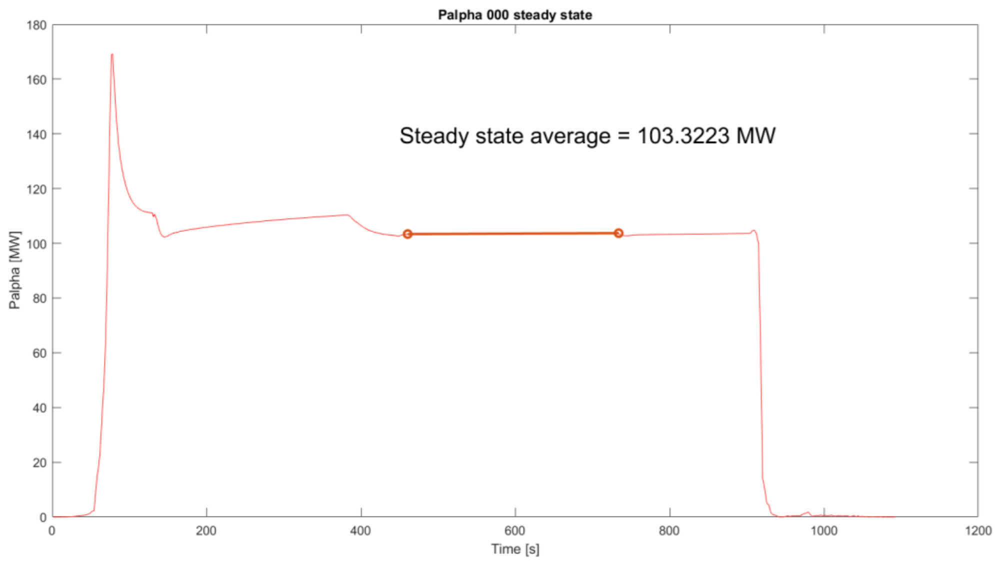


Fig. 59: MATLAB-000 Palpha steady state average value.

keytimes

Function description: Through this function we compute estimation for the times METIS defines as times of interests where, depending on the plasma current as an input parameter, certain events should happen. This is done the same way as in METIS.

Input parameters: **sim** (simulation of study, in string format and with our three number code naming system), [**extraplot** (logical 1 or 0, whether we want extra plots from secondary modules), **ip** (plasma current (MA)), **rampup_dipdt_factor** (factor applied to ramp-up rate current increase), **rampdown_dipdt_factor** (factor applied to ramp-down rate current decrease), **duration** (shot duration including ramp-up and flat-top, without ramp-down)].
Output: **keytimes** (array of times).

N_sims_overview
Function description: This function represents the simulation overview for N different functions the same way it does *sim_overview* for a single function. These simulations must be the same type, having the same two first numbers following our classification criteria.
Input parameters: **sim_type** (simulations of study folder, in string format and with our two number code naming system for folders), **N** (number of simulations), [**start** (last code number of the first simulation to be performed), **mode** (mode of computation 'EVOLUTION', 'RUNFAST', 'RUN'), **list** (if simulations to compute are not consecutive, specify its third indices in this list), **extraplot** (logical 1 or 0, whether we want extra plots from secondary modules), **ignore_null_padd** (logical 1 or 0, whether we'd like to ignore the condition of Q = infinity for the achievement of ignition)].
Output: text and plots.

compare_sims_2D
Function description: This function compares two different variables from N simulations, during the **steady-state** or a determinate **time**. These simulations must be the same type, having the same two first numbers following our classification criteria.
Input parameters: **vx** and **vy** (path to variables inside METIS post structure in string format), **namevx** and **namevy** (string name for each variable), **sim_type** (simulations of study folder, in string format and with our two number code naming system for folders), **N** (number of simulations), [**start** (last code number of the first simulation to be performed), **time** (time of interest for the comparation)].
Output: plot

Example:

```
compare_sims_2D('real(post.zerod.pnbi)/10^6+imag(post.zerod.pnbi)/10^6',
'post.zerod.pfus/10^6','Pnbi [MW]','Palpha [MW]','03',5,0,700)
```

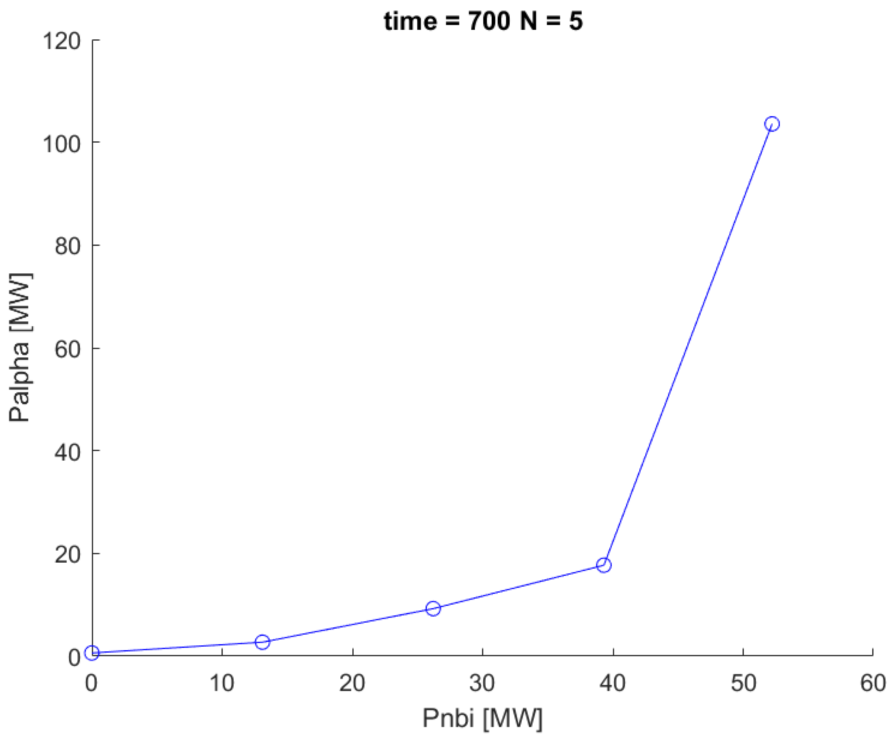


Fig. 60: MATLAB-03 Pnbi vs Palpha at t = 700 s

compare_sims_3D

Function description: This function compares three different variables from N simulations, during the steady-state or a determinate time. These simulations must be the same type, having the same two first numbers following our classification criteria.

Input parameters: **vx**, **vy** and **vz** (path to variables inside METIS post structure in string format), **namevx**, **namevy** and **namevz** (string name for each variable), **sim_type** (simulations of study folder, in string format and with our two number code naming system for folders), **N** (number of simulations), [**start** (last code number of the first simulation to be performed), **time** (time of interest for the comparation)].

Output: plot

Example:

```
compare_sims_3D('real(post.zerod.pnbi)/10^6+imag(post.zerod.pnbi)/10^6',
'post.zerod.pfus/10^6','post.zerod.te0/10^3','Pnbi [MW]','Palpha [MW]',
'Temp [keV]','03',5,0,700)
```

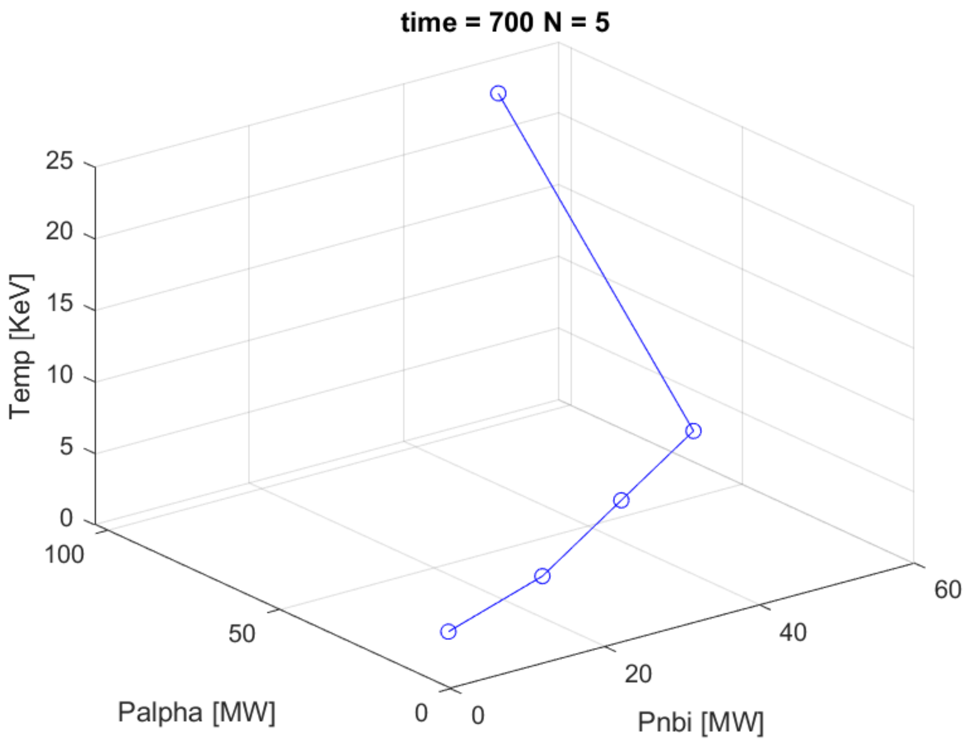


Fig. 61: MATLAB-03 Pnbi vs Palpha vs Temperature at t = 700 s

config_load
Function description: This module is strongly related to section 10, "Implementation of a new model through our MATLAB setup", because is the one in charge of editing all the options inside an output post structure from METIS (usually the ITER standard reference '000'), and generate a new one to be loaded from METIS main interface to conduct a new simulation. It is a first step to control METIS directly from MATLAB.
Input parameters: **sim_init** (name of the post structure from the simulation to edit), **sim_save** (name of the new simulation post file).
Output: **post** (structure, new simulation (sim_save) with new input options)

8.2. Work Methodology

During the entire project, several paths have been taken and we've ended up deviating from our main goal, focusing in studying METIS source code. About this source code analysis, my work has been guided and restricted by my tutor, to avoid losing myself into the huge amount of code lines and helping me find the answer to specific questions. We tried documenting all these decisions along the project and we've tried to organise and explain them through this report.

Furthermore and most important, we conceived a way of arranging the simulations we kept performing through a three number code for each different scenario. To identify each simulation using its name code:

1. We identify the first number as the initialisation of the scenario, which corresponds to the first window showing editable parameters in METIS, mentioned in section 6 (Fig. 7) as the interface module "reactormetissimulation". All scenarios with these same parameters, share the same first number of the code. '0' is understood as the predetermined case for ITER in METIS.

2. The second number refers to changes in the secondary window (Fig. 8-9) shown by METIS, like waveforms edition such as powers or density and option parameters.

3. Finally, the third number of the code is also a variation of waveform references and input data but, of course, sharing a common main change with the rest of the simulations with the same second number in its identification.

This classification is not only useful to register simulations into the simulation log (Annex A.1), but also to navigate through the OS archive system when required from the MATLAB setup.

9. METIS data computation

There is very few information on how METIS manages its data, despite explaining in the METIS user guide [1] which are the most important functions and what do they do, there's nothing on how the first waveforms are computed or about the estimation times METIS returns and uses for the creation of the first variables. These estimation times depend on the plasma current defined in the beginning of the reactor scenario generation (Fig. 7), and parameters like the duration, ramp-up or ramp-down factors are determining for the general output of the simulation. Aspects like these are the ones we have explained in this section.

9.1. "Key times" estimation, waveform and reference computation

In section 6, during a quick explanation on how to manage the input data to perform a simulation, we mentioned the creation of this data after the call to the *sycomore2metis.m* module. This module computes all the waveforms for the reference quantities with the options and parameters loaded from the "reactormetissimulation" window. Of course, if desired, this data can be loaded from the "load" option on the main METIS window

Before studying what happens with these input waveforms during the simulation, we should know how they are calculated in the first place. Inside the *sycomore2metis* function, the computing order for some of these variables is:

1. Geometry and separatrix points. (ln.416)

2. First power values for specific times of interest (ECRH and LH powers are linearly interpolated during ramp-up and ramp-down).

3. Computes **dipdt**, **dipdt**$_{\mathbf{down}}$ and **dipdt**$_{\mathbf{adj}}$ for ramp-up and ramp-down. (ln.531)

4. Dynamical parameters for different times (**ip** and **nbar**). (ln.553)

5. Key times: (ln.581)

   - Start of the simulation

$$t_{start} = 1.5\ s \qquad (11.1)$$

   - Ramp-up duration

$$dt_{ramp\_up} = ip/dipdt \qquad (11.2)$$

   - X-point formation

$$t_{xpoint} = dt_{ramp\_up}/10 \qquad (11.3)$$

- Time for full power

$$t_{full\_power} = dt_{ramp\_up} * ip_{full\_power}/ip \qquad (11.4)$$

- Start of flat-top

$$t_{flattop} = dt_{ramp\_up} \qquad (11.5)$$

- Time for full density and end of full power assisted phase (estimation)

$$t_{flattop\_plus} = t_{flattop} + 3 * dt_{ramp\_up} \qquad (11.6)$$

- Simulation duration (end of flat-top)

$$t_{end} = t_{flattop\_plus} + \frac{3}{2} * duration \qquad (11.7)$$

- H to L back transition

$$t_{dn} = t_{end} + \frac{ip_{flattop} - ip_{xpoint}}{dipdt_{adj} + dipdt_{down}} * 2 \qquad (11.8)$$

- Back transition to limiter

$$t_{dip} = t_{dn} + (ip_{dn} - ip_{xpoint})/dipdt_{down} \qquad (11.9)$$

- End of simulation and plasma

$$t_{v0} = t_{dip} + (ip_{xpoint} - ip_{ini})/dipdt_{adj} \qquad (11.10)$$

6. Pre-parametrised METIS input data, ***z0dinput***, computed by *zerod_scalaire* function. (ln.661)

7. Waveforms control nodes generation. (ln.760)

8. Power scaling (with **rap_power** (ln.433), volume ratio to scale power). (ln.996)

9. Time interpolation of z0dinput cons and geo parameters, using ***pchip*** and ***zinterpnc*** functions. (ln.1016)

10. If only input structure ***z0dinput*** is required, call end. (ln.1214)

Once this module completes its run, we are ready to edit these default input waveforms, if required, from the main interface of METIS, which shows up after concluding this process as seen in section 6 (Fig 8-9).

It is interesting to notice that this pre-simulation computes different key times related to achievements like the L to H-mode transition or the arriving to a flat-top phase. This is of course for all possible different values for the **"reactormetissimulation"** window (Fig. 7). What would it happen for a simulation that doesn't get there and cannot satisfy these cases? We've tried it for example with the simulation **100**, where the maximum PNBI limit during the flat-top is set to 5 MW instead of 53 MW like in the default simulation **000**. Nothing significant occurs. As expected, the key times are computed the same way and once we run the simulation we achieve neither a breakeven nor the flat-top, nor H-mode. Meaning, this times are only estimations of what is expected to happen according to the ODE and boundary conditions given for the simulation.

```
==> Data ready !
Volume separatrice @0.95 = 783.315 m^3
Rmin = 4.2, Rmax = 8.2, Zmin = -3.34477, Zmax = 4.001,
Rzmin = 5.04556, Rzmax = 5.33396
Ra = 6.2, Za = 0.628, Rx = 5.04556, Zx = -3.34477
a = 2, K = 1.83644, d = 0.505121
b0 = 6.80226
dIp/dt_{rampup} = 197.049 (kA/s)
dIp/dt_{rampdown, end of plasma} = 59.5736 (kA/s)
Key times:
Starting time of the simulation: 1.5 s @ Ip = 0.4 MA
X-point formation: 7.61234 s @ Ip = 1.9 MA
Full power time: 60.8987 s @ Ip = 12 MA
Start of flat-top: 76.1234 s @ Ip = 15 MA
End of full power assited phase: 304.493 s @ Ip = 15 MA
End of flat-top: 904.493 s @ Ip = 15 MA
H to L back transition: 938.734 s @ Ip = 10.6066 MA
back transition to limiter: 1084.88 s @ Ip = 1.9 MA
End of simulation/plasma  : 1092.49 s @ Ip = 0.4 MA
```

Fig. 62: METIS-100 terminal pre-simulation output

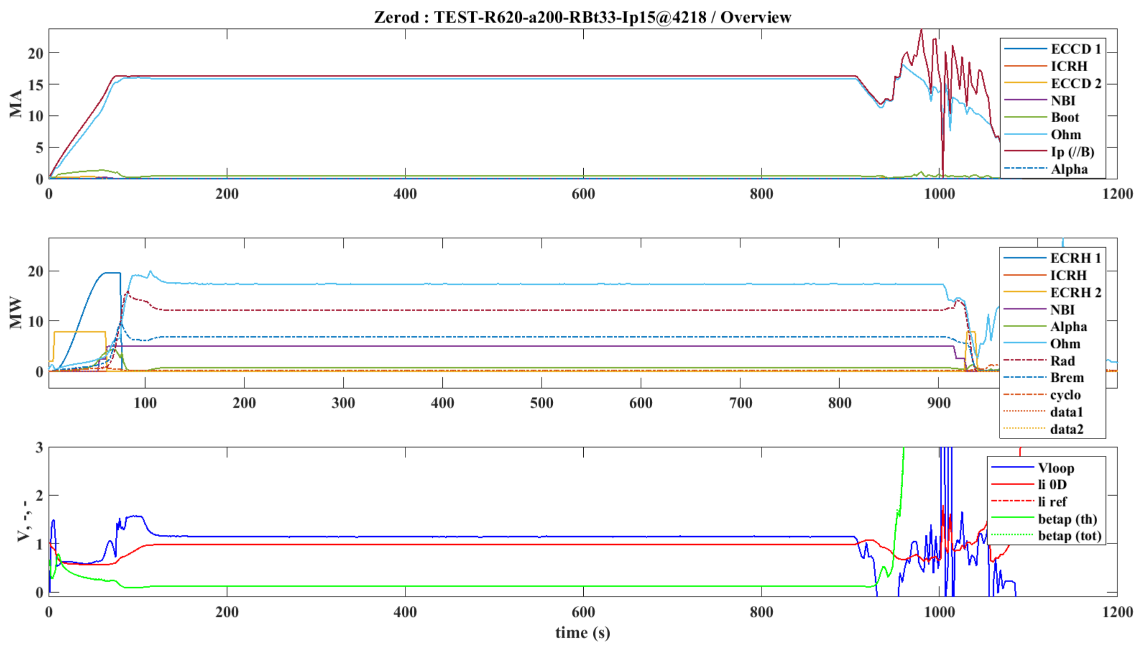


Fig. 63: METIS-100 overview

9.2. Input data towards output. Instructions and constraints

In order to start an efficient strategy to decide which scenarios would be useful to simulate, we decided to check if we could tell apart some instructions from the input data. We can identify as an instruction an input value that, no matter what happens during the simulation, it won't change its value and will directly affect every other aspect and outcome of the run. Constraints, on the other hand, are those values that could restrict the evolution of certain quantities, due to physical or engineering limits.

In the output data post structure we identified the variables that represented the same values as the input data from z0dinput.cons and we compared each input-output pair for the RUNFAST and EVOLUTION modes. The result for simulation 000 (Annex A.3) showed how for the RUNFAST mode we can notice a difference between the input and the output for almost all the variables except for time and P_{LH} while for the EVOLUTION mode, the only inputs that change are the central electron density estimation, neutrals power and effective charge. This generated doubts on how METIS manages its data depending on the mode of computation, so we tried to understand it in the next sections.

9.3. FAST RUN and RUN

To control all the preparations for the simulation, including the option file creation process and the input data, we shall understand how METIS computes all the different outputs.

In METIS, a "waveform relaxation" like algorithm [6] is used for the main convergence loop and all the PDEs are solved by artificially separating time and space for all the computation modes. Only the PDE for current diffusion is completely solved. Getting inside this algorithm would mean extending our objectives into a new whole project; keeping this in mind, we continued our source code analysis.

Fig. 64: Space-time domain decomposed for a waveform relaxation method

We decided to understand, overall, when and where (using which function files) METIS created the ***zerod*** output file. With this objective, we took METIS's terminal window displays and prints as a hint. The starting shot for the METIS calculations is whenever we hit either

“Run METIS” or “Run METIS in fast mode” commands. We are going to start analysing the steps for this last one.

```
Metis sample in fast mode : 471 -> 67 -> 471
Metis: using separatrix given by points (R,Z) :B@:HeBiHeE.HeE.He.He.He.He.He...W............. in 28.014 s (cpu = 58.3125 s)
Warning : bad convergence at some time step : check it !
possible shot disruption @ 1087.12 s (65): radiative power exceeds input power; you must check the result
```

Fig. 65: METIS-000 RUNFAST terminal output

When we execute the “Run METIS in fast mode” option for simulation 000, this output is given through METIS terminal window. Analysing the source code, we’ve been able to summarise this process in the next steps:

1. When we searched for the first sentence appearance inside the different files of code, we found the ***zerodfast*** function, where the first print “Metis sample in fast mode: %d -> %d -> %d\n” is called (ln.357). This file carries out a fast calculation with extraction of relevant points.

2. Further on, the ***zerod*** function is called inside a conditional (ln.409) using the **option.tol0d** variable. **tol0d** is an indicator of the precision demanded for the simulation, in this case, if not defined, it will be given **0.01** as value and zerod will be executed.

3. For this first call of the ***zerod.m*** module (nargin < 8, ln.76), function ***zero1t*** is called (ln.178) returning a first value to the ***zerod*** (named *zs* along the code) output file we are looking for and later will be studied using our own MATLAB setup. The convergence loop is also initiated (ln.242). It would be interesting to study the variables defined to study the convergence of the simulation, but this is the matter treated on section 9.5.

4. The next messages displayed on the terminal correspond to different calculation events, like the definition of the geometry of plasma, (“Metis: using separatrix given by points (R,Z) :”, *zgeo0.m*, ln.137), the start of the convergence computation loop (“@:”, *zerod.m*, ln.381), or each repetition of the loop (“.”, *zerod.m*, ln.866).

5. We can see in ln.472 how ***zero1t*** function is called again for each repetition of the convergence loop. ***zero1t.m*** module is presented in METIS documentation as the central function of the program, computing new guesses of data for all time slices.

6. Finally, the end of the calculations is defined in ln.884 (“in %g s (cpu = %g s)\n”) and the warnings on bad convergence or possible disruptions are displayed (ln.916 and ln.923, respectively).

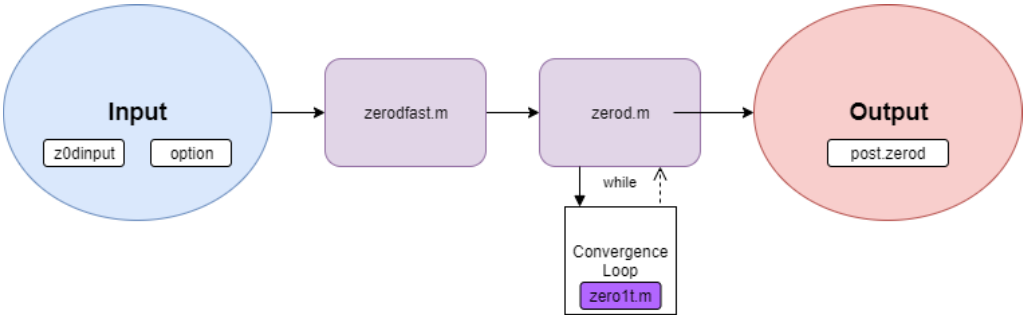


Fig. 66: RUNFAST mode diagram

For the more precise option "**Run METIS**", the calculations follow a different path, with a different outcome:

```
initial run of Metis sample in fast mode : 471 -> 67 -> 471
Metis: using separatrix given by points (R,Z) :B@:HeBiHeE.HeE.He.He.He.He.He...W............. in 25.162 s (cpu = 55.75 s)
Warning : bad convergence at some time step : check it !
possible shot disruption @ 1087.12 s (65): radiative power exceeds input power; you must check the result
start full run of METIS now ->
@:Metis: using separatrix given by points (R,Z) :........W.W..W..W.W.W.W.W.W.W.........W..W.W.W.W.W..........................~~~ in 333.608 s (cpu = 563.703 s)
possible shot disruption @ 1087.12 s (469): radiative power exceeds input power; you must check the result
```

Fig. 67: METIS-000 RUN terminal output

1. For this mode, the "initial run of" Metis is done first by the ***zerod*** function (ln.107) with a **tol0d** of **0.001**.

2. The function ***zerodfast*** is also called in this mode (ln.109), as a first approximation. This execution will do the same as starting with the fast mode, but the difference here is the tolerance. This results in ***zerod*** function being called from the same conditional (ln.407) but with more arguments (nargin == 8, *zerod*, ln.76) skipping the call to *zerodfast.m* inside *zerod.m* module and continuing as mentioned in steps 4 to 6 of fast mode.

3. After doing this first analysis, it proceeds to complete the first call of the zerod function (from ln.110) starting the convergence loop (ln.159), printing "start full run of METIS now ->\n" (ln.167), and computing again steps 4 to 6, taking the simulation to an end.

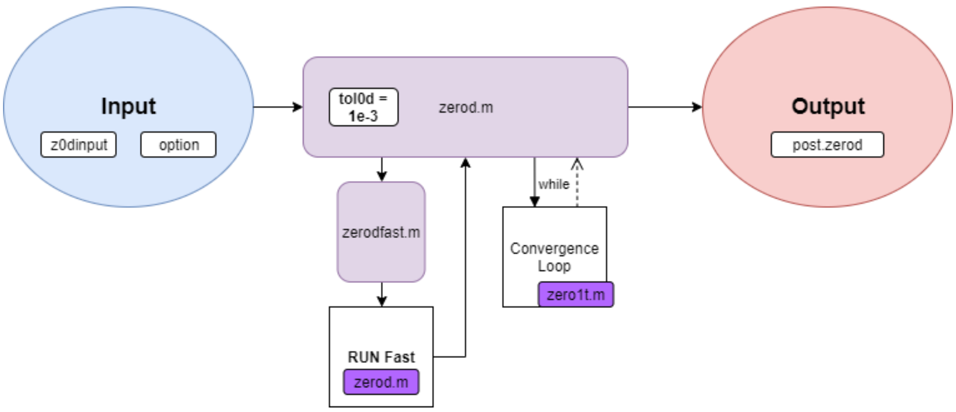
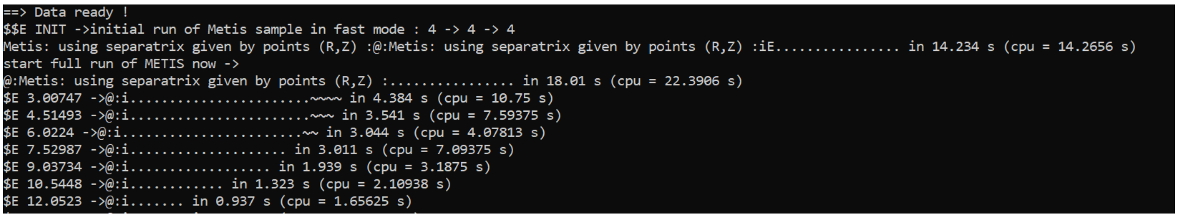


Fig. 68: RUN mode diagram

9.4. Evolution mode

In addition to RUN and RUN Fast modes, METIS offers two more computation modes, the fit of LH computation mode (not studied during this project), and the EVOLUTION mode, that allows to include METIS in a **time loop**. An analysis like the previous ones has been made for this mode, and the result is the diagram in figure 69, showing the way it operates.



```
==> Data ready !
$$E INIT ->initial run of Metis sample in fast mode : 4 -> 4 -> 4
Metis: using separatrix given by points (R,Z) :@:Metis: using separatrix given by points (R,Z) :iE............... in 14.234 s (cpu = 14.2656 s)
start full run of METIS now ->
@:Metis: using separatrix given by points (R,Z) :................ in 18.01 s (cpu = 22.3906 s)
$E 3.00747 ->@:i......................~~~~ in 4.384 s (cpu = 10.75 s)
$E 4.51493 ->@:i......................~~~ in 3.541 s (cpu = 7.59375 s)
$E 6.0224 ->@:i......................~~ in 3.044 s (cpu = 4.07813 s)
$E 7.52987 ->@:i................... in 3.011 s (cpu = 7.09375 s)
$E 9.03734 ->@:i................. in 1.939 s (cpu = 3.1875 s)
$E 10.5448 ->@:i........... in 1.323 s (cpu = 2.10938 s)
$E 12.0523 ->@:i....... in 0.937 s (cpu = 1.65625 s)
```

Fig. 69: METIS-000 EVOLUTION first terminal output section

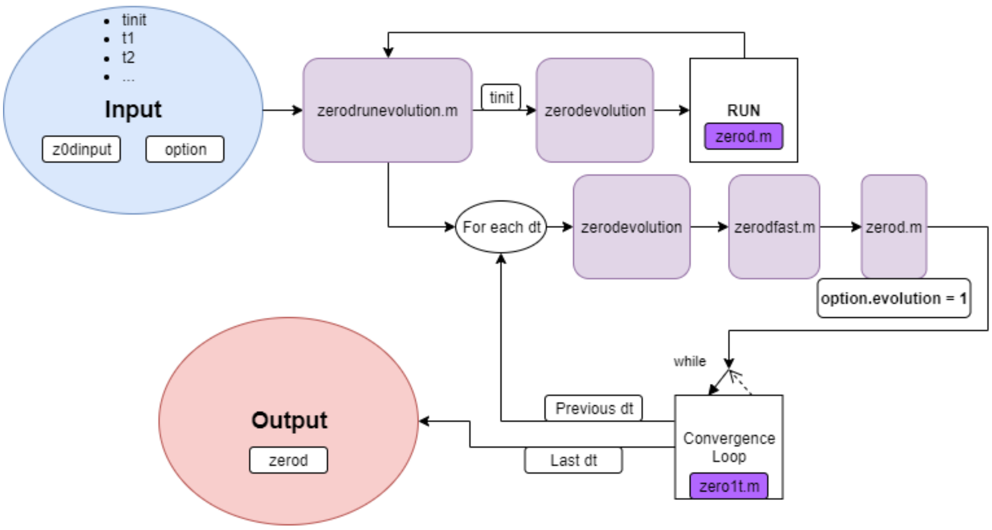


Fig. 70: EVOLUTION mode diagram

9.5. Convergence of the simulation

Obviously, for these three different modes of computation, the precision and algorithm used for the simulation affects the behaviour of the convergence loop and the time it takes to compute it. Then, it would be interesting to know on which variables relies the convergence loop to end. Looking inside the ***zerod*** function, we can see that the convergence of a simulation is determined by the next variables:

- **dw**: relative error on w (total plasma energy), at the end of convergence.
- **dpfus**: relative error on pfus, at the end of convergence.

- **dini**: relative error on dini (total non-inductive current), at the end of convergence.
- **diboot**: relative error on diboot (bootstrap current), at the end of convergence.

While these variables are greater than the tolerance **tol0d**, the simulation won't converge. Furthermore, we have other options and variables that affect this convergence like:

- **vloop**: loop voltage at the edge of the plasma. This key allows choosing the edge condition of the poloidal flux diffusion equation.
- **lhmode**: this key allows to choose the scaling law used to compute the lower hybrid current drive efficiency.
- **scaling**: this key allows selecting the two scaling laws used for the simulation. Depending on the choice, a scaling law gives the thermal energy content of the plasma in L-mode and the second one in H-mode or, alternatively, the first one gives the thermal energy content of the core plasma and the second one the thermal energy content of the pedestal. A special value allows fitting experimental measurements.

9.6. Simulation results comparison between different modes

To finally determine which mode is the most precise, we have compared the differences between their results for certain simulations. They do not seem very different for scenario 000, where the fusion power behaves quite similar (Fig. 71-73).

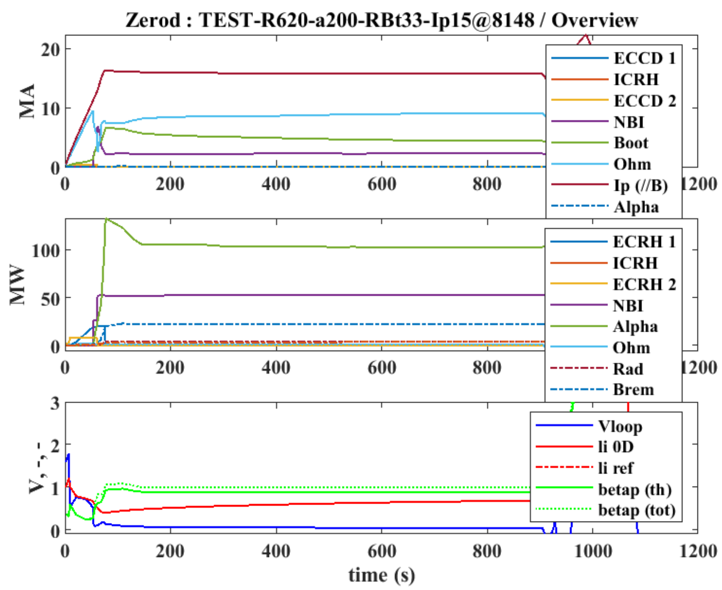


Fig. 71: METIS-000 overview for RUNFAST mode

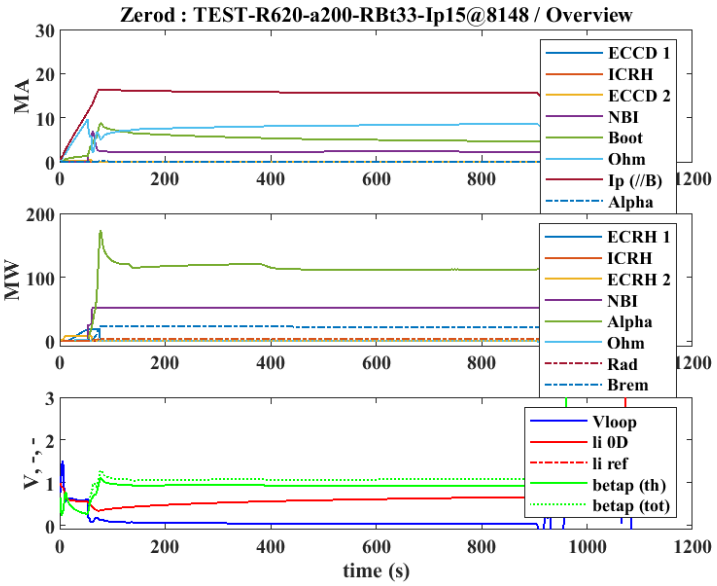


Fig. 72: METIS-000 overview for RUN mode

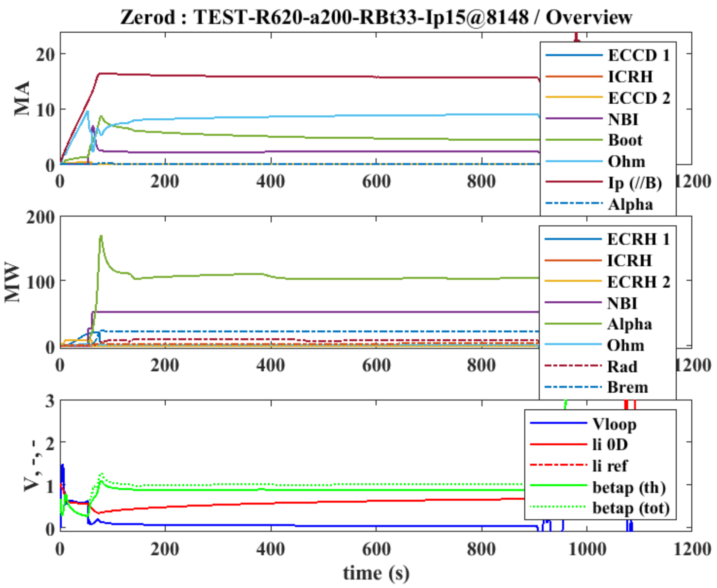


Fig. 73: METIS-000 overview for EVOLUTION mode

But when comparing the results for scenario 051, the output is absolutely different for the evolution mode, as it does not achieve the H-mode, for example, while it does through RUN and RUNFAST modes (Fig. 74-76).

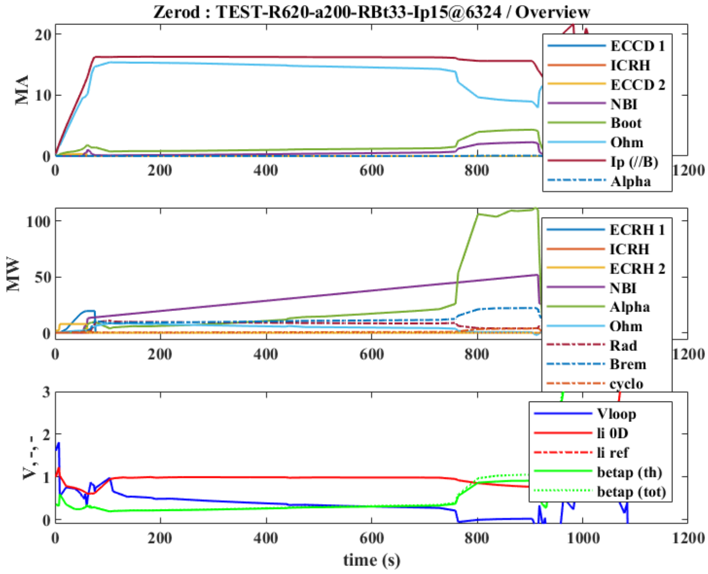


Fig. 74: METIS-051 overview for RUNFAST mode

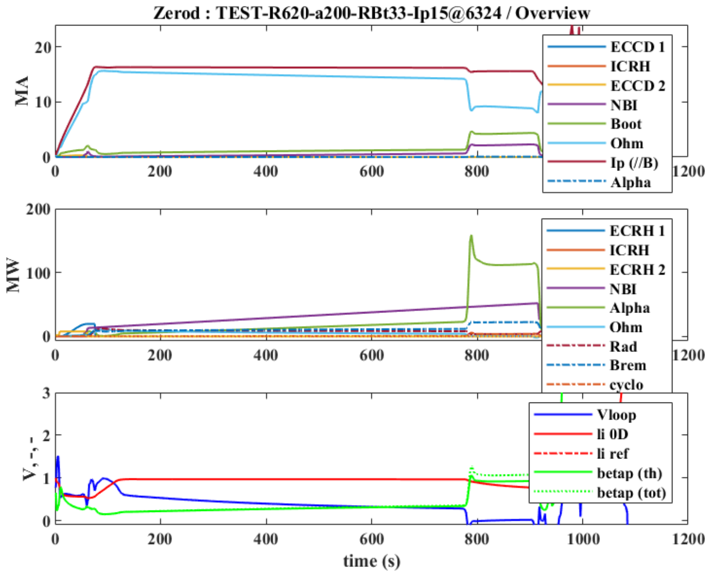


Fig. 75: METIS-051 overview for RUN mode

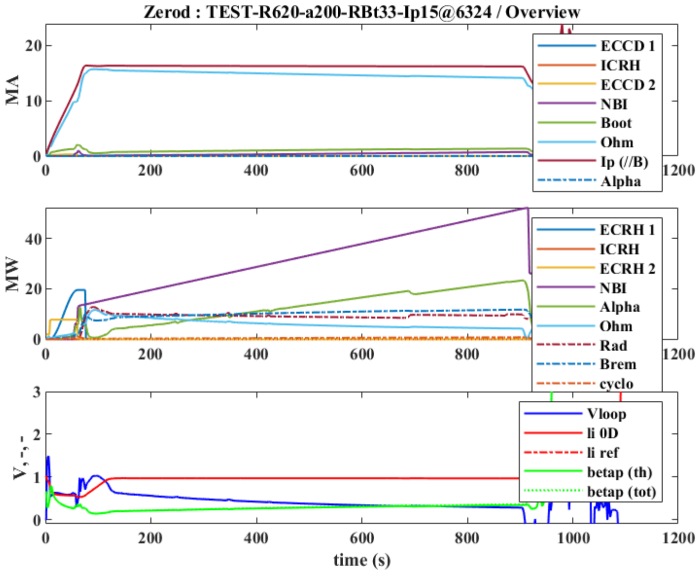


Fig. 76: METIS-051 overview EVOLUTION mode

After seeing this great difference between the three modes, we understood the most accurate and coherent mode is the EVOLUTION one, so we decided to use this mode for all the analysis simulations.

10. Implementation of a new model through our MATLAB setup

One of the most exciting and interesting lines of work we could continue developing to be a METIS complement, following the example of our MATLAB setup, is the possibility to add new ways of computing certain quantities, like power losses or confinement times, new options and parameters to change the behaviour of the main software, or add further scaling laws to change the physics point of view to another from more recent studies.

As for this last one, we've developed an example of what would be a possible computation of the H-mode transition through a different scaling law for the power compared to the power threshold, extracted from Birkenmeier, 2022 [7]. In this study, the variable P_{lhthr} mentioned as P_{SEP}, is the power reaching the separatrix by transport in the plasma, with the radiated power of the bulk plasma:

$$P_{lhthr} = P_{SEP} = P_{loss} - P_{rad} \qquad (10.1)$$

We've introduced a new case (P_{SEP}) for the option switch **option.plhthr** inside our MATLAB setup code, simulating an output for a METIS simulation under this condition:

```
switch option.plhthr
    case '2*pion'
    Plhthr = max(1,max(min(2 .* zs.pion,Pin - dwdt),zs.ploss./3));
    case 'P_LCFS'
    Plhthr = max(1,min(Pin - Prad - Pbrem - Pcyclo - Pioniz,Pin - dwdt));
    case 'PSEP'
    Plhthr = max(1,Ploss./10^6-Prad);
    otherwise
    Plhthr = max(1,max(min(Ploss,Pin - dwdt),Ploss./3));

end
```

When comparing the H-mode transition between simulations '000' and '003', in which we've introduced this change, we can spot differences looking to the power threshold plots from each simulation (Fig. 77 and 78):

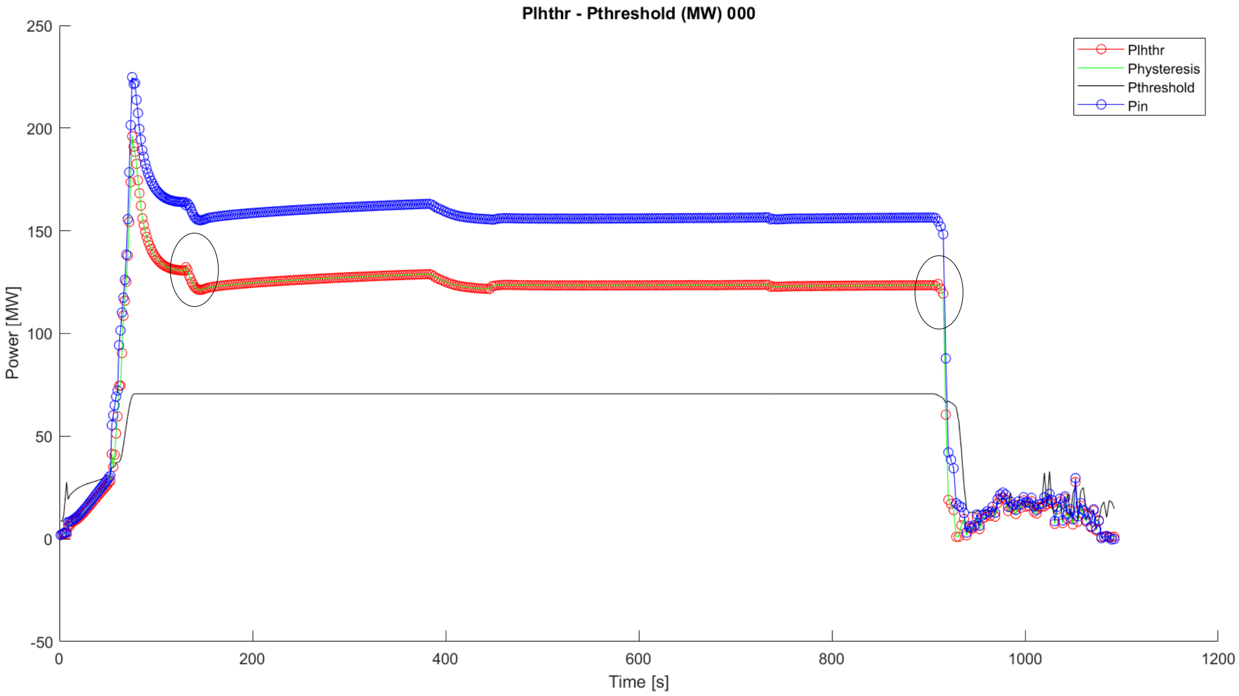


Fig. 77: MATLAB-000 powers comparison to threshold power

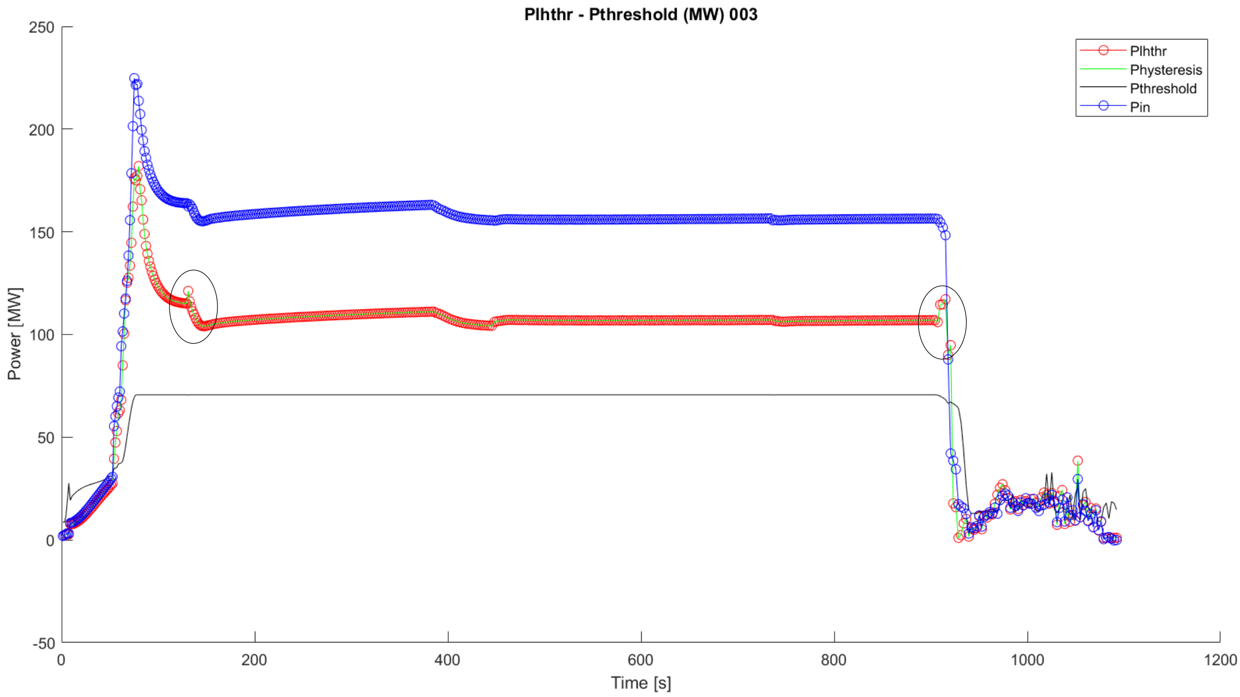


Fig. 78: MATLAB-003 powers comparison to threshold power

11. Work plan

The work plan for this final year project has been summarised in a Gantt diagram (Tab. 2) sorting the work done in five different tasks or phases: starting with the documentation process, the first simulations, programming our MATLAB setup, doing reverse engineering with the METIS source code, and writing the report on all this process.

			Aug-22	Sep-22				Oct-22				Nov-22				Dec-22				Jan-23		
TASK	START	END	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Documentation																						
About METIS	23-8-22	14-9-22																				
About nuclear fusion	10-9-22	30-9-22																				
About ITER	14-9-22	2-10-22																				
Simulations																						
Depending on Powers	8-9-22	19-10-22																				
For H-mode study	3-11-22	18-11-22																				
For ramp-up study	15-11-22	17-11-22																				
MATLAB Coding																						
Basic functions	4-10-22	21-12-22																				
H-mode related	11-11-22	21-12-22																				
Ramp-up related	17-11-22	21-12-22																				
Load configuration	16-12-22	11-1-23																				
Font code analysis																						
H-mode	8-11-22	29-11-22																				
Ramp-up	29-11-22	12-12-22																				
Computation modes	12-12-22	18-12-22																				
Report																						
Writing	8-11-22	17-1-23																				

Tab. 2: Work plan

12. Environmental impact

12.1. This project's impact

The main cause on energy consumption has been the use of the computer during the development of the project. This consumed energy must be considered when computing the total emissions of this project in table 3.

For CO_2 emissions, it has been used the electric MIX from Spain for 2022 and for radioactive waste, data from 2018. The electrical consumption has been computed for a 45 W laptop and a 33 W screen, considering approximately 560 h of work.

Computer equipment	*MIX Spain*	*Electrical energy consumption (kWh)*	*Total emissions*
CO_2	259 g/kWh	43.68	11.31 kg
Low and medium radioactivity	0,002277 cm^3/kWh	43.68	0.0995 cm^3
High radioactivity	0,277 mg/kWh	43.68	12.1 mg

Tab. 3: Environmental impact breakdown

12.2. Nuclear fusion's impact

At present, fusion devices produce more than ten MW of fusion power. ITER will be capable of producing 500 MW; therefore, fusion power is definitely the main technology to develop during the next years as an alternative to greenhouse related energy production [5]. Fusion is among the most environmentally friendly sources of energy as there are no CO2 or other harmful atmospheric emissions from the fusion process. Moreover, its two sources of fuel, hydrogen and lithium, are widely available in many parts of the Earth.

It is also a very important alternative of fission nuclear power. Nuclear fission power plants have the disadvantage of generating unstable nuclei; some of these are radioactive for millions of years. Fusion on the other hand does not create any long-lived radioactive nuclear waste. A fusion reactor produces helium, which is an inert gas. It also produces and consumes tritium within the plant in a closed circuit. Tritium is radioactive (a beta emitter) but its half life is short. It is only used in low amounts so, unlike long-lived radioactive nuclei, it cannot produce any serious danger. The activation of the reactor's structural material by intense neutron fluxes is another issue. This strongly depends on what solution for blanket and other structures has been adopted, and its reduction is an important challenge for future fusion experiments.

In a global context of rising oil and gas prices, decreased accessibility to low-cost fossil fuel sources, and an estimated three-fold increase in world energy demand by the end of this century, the energy question finds itself propelled to centre stage. How will it be possible to supply this new energy without adding to greenhouse gases?

Investing in renewables such as solar, wind and geothermal is important. Just like in fusion R&D... with significant investment, new advancements in technology come, and with advancements in technology comes a decrease in price. All calculations point to an increase in the importance of renewables in the decades to come.

The ideal future energy mix would hold a mixture of generation methods instead of a large reliance on one source. Fusion offers advantages that make it worth pursuing: widely abundant, inexpensive and virtually unlimited fuels, and the ability to operate in a baseload capacity, which is not easy for generation methods based on intermittent sources, such as wind or sun. The fusion community doesn't see itself in competition with renewable forms of energy. Rather, in a world ever more dependent on energy, it is important to follow all of the promising options for our common future [5].

13. Economic study

13.1. This project's budget

The budget for this project can be subdivided in the next different types of costs:

- **Computer costs:** we have considered 10 years for the lifetime of the computer equipment, including a 1,539 € laptop and a 60.79 € monitor.

- **Personnel costs:** the fee for a junior engineer is around 25€/h, and this project has taken 560 h of work approximately.

- **Energetic costs:** as mentioned in section 12.1, the total energy consumption has been of 43.68 kWh. The average kWh price in Spain from August-22 to December-22 has been of 0,182 € [12].

Finally, the taxes are the last addition, adding the VAT 21% and BIC 6%. The total cost of this project ascends to 19,635.77 €. This cost has been broken down in Table 4.

Personnel costs	**Concept**	**Total hours**	**Hour price (€)**		**Total**	
	Junior engineer	560	25			
	Total				14,000	(1)
Energetic costs	**Concept**	**Power (kW)**	**Functioning hours**	**€/kWh**	**Total**	
	Laptop	0.045	560	0.182	4.59	
	Screen	0.033	560	0.182	3.36	
	Total				7.95	(2)
TOTAL	**Concept**				**Total**	
	Total cost	(1)+(2)			14,007.95	(3)
	Unforeseen expense	10% of (3)			1,400.79	(4)
	Total BT	(3)+(4)			15,408.75	(5)
	VAT	21% of (5)			3,235.84	(6)
	BIC	6% of (5)			924.52	(7)
	Total AT	(5)+(6)+(7)			19,569.11	(8)

Amortization	Concept	Purchase price (without VAT)	Lifetime (months)	Use time (months)	Recover value	
	Laptop	1539	120	5	64.125	
	Screen	60.79	120	5	2.53	
	Total				66.66	(9)
Total Project Cost	**(8)+(9)**				**19,635.77**	

Tab. 4: Total project's budget

BT: Before taxes
AT: After taxes

13.2. ITER budget

ITER is financed by seven Members: China, the European Union, India, Japan, Korea, Russia and the United States. In all, 35 countries are sharing the cost of the ITER Project.

Prior to the 2016 budget updating exercise, the European Union had estimated its global contribution to the costs of ITER construction at EUR 6.6 billion, with other Domestic Agency contributions depending on the cost of industrial fabrication in those Member states, which can be higher or lower, and their percentage contribution to the construction of ITER. Based on the European evaluation, the cost of ITER construction for the seven Members had been evaluated in the past at approximately EUR 13 billion (if all the manufacturing was done in Europe) [5].

At the ITER Council meeting in November 2016, the ITER Organization proposed a complete updated project schedule through First Plasma (2025) and on to Deuterium-Tritium Operation (2035). The overall project cost in line with the revised schedule added EUR 4 billion to the original estimate, a cost that was approved by the ITER Members through their domestic budget processes.

Concerning economic benefits, ITER is creating jobs, and not only locally.

First, consider the R&D and fabrication activities that are going on for ITER around the world. In 2020, the ITER Domestic Agencies estimated the number of contracts awarded related to the development and procurement of ITER systems, components and infrastructure at over 3,200—the direct beneficiaries of these contracts are the laboratories, universities and industries in ITER Member countries. (Contracts are also awarded directly by the ITER Organization.) These contracts—many of which demand skilled contributions in engineering—are significantly more labour-intensive than conventional industrial manufacturing. An estimated EUR 4 billion are engaged in ITER manufacturing around the world.

It is estimated that over three-fourths of the total European construction contribution to ITER will be directed to industry, a proportion that is similar in other Members.

Over 1,000 people worked on the preparation of the ITER site, the construction of the Provence-Alpes-Côte d'Azur International School, and the ITER Itinerary. A further 2,500 people were involved in ITER construction for the period mid-2010 to 2014, and an average of 1,800 people for the period of 2014 to 2020. Today, approximately 3,500 people work for the ITER Project in Saint Paul-lez-Durance (ITER staff, contractors, temporary agents, European Domestic Agency staff and subcontractors, site workers); these employees contribute, with their families, to the economic life of the region.

During the peak of construction and assembly works (2019-2024), 1,000-1,500 workers will be employed on the ITER site.

Contracts totalling EUR 8.87 billion have been attributed since 2007 by the ITER Organization, the European Domestic Agency for ITER (responsible for the in-kind contribution of Europe to ITER, including all buildings), and Agence Iter France. Within this total, companies in France have been awarded EUR 5,378 billion worth of contracts, of which 78% (worth EUR 4,206 billion) were attributed to companies based in the PACA region (statistics for the period ending 30 June 2022).

14. Gender equality

Advancing the science, engineering, and technical knowledge necessary for commercializing fusion energy will require a diverse and inclusive workforce.

As with other technical fields, women in fusion have historically comprised just a small fraction of the research ecosystem.

Too often, women working in science, technology, engineering, and mathematics (STEM) face unfair barriers in the workplace, from institutional biases to hostile working environments that discount their contributions. The actual percentages of women working in STEM fields are 3%, 10%, 24% and 10%, respectively [11].

As this field continues to grow, the need for inclusive policies and peer support will increase in importance.

Conclusions

We have accomplished our main objective to provide a technical guide of METIS oriented to the academic and research activities of the Nuclear Engineering Section (UPC), presented in section 6. During this project we have understood METIS functioning and we have ran simulations as it was intended. Some of these simulations are analysed in section 7, using the MATLAB setup created for this purpose, described in section 8.

To achieve these objectives we have needed to apply reverse engineering to understand some METIS outputs that didn't match with our expected results (section 7). This has taken us to analyse some parts of the source code in section 9.

Our work is also the base for a more focused study on METIS simulations from which we have deviated in favour of knowing how METIS managed its data. Studying the code has been a very dedicated reverse engineering work; it has proven very effective when trying to understand the different computation modes METIS has to offer or how the input data defined different simulation characteristics, like the ramp-up or the H-mode.

In spite of not having a certain conclusion on events like the H-mode transitions, we have been capable of creating our own MATLAB setup for the analyses of METIS's simulations outputs. With these new functions now it will be possible to work from a known environment and sets the possibility to, not only program directly from METIS code, introducing new features, but also run METIS from a command line, adding versatility to the code. As an example, we have implemented a new model for the H-mode computation based on the paper Birkenmeier, 2022 [7].

This project sets the base for new academic and research projects, like teaching practices for the nuclear fusion subject at ETSEIB, UPC.

While trying to learn the functioning of METIS, I have been able to use my MATLAB programming knowledge and to expand my learning on nuclear, plasma and tokamak physics.

In the end, this final year project has been of great interest considering how nuclear fusion is one of the most important research fields in the actual physics, environmental and energetic paradigm.

"Don't judge each day by the harvest you reap but by the seeds that you plant."

- Robert Louis Stevenson -

Acknowledgements

First, I would like to thank the opportunity offered by my director Dr. Guillem Cortes and my tutor Albert Riego for me to learn from their work, knowledge and experience in my introduction into the research world. I hope for this to be the first of many works to do with them.

Finally, I won't be able to repay the support I have received from my family and friends, who have always been patient during these years, hoping for me to end up doing what I love.

ETSEIB

Bibliography

Bibliographic references

[1] J.F. Artaud, *METIS User's Guide*: PHY/NTT2008.001, 2008

[2] J. Artaud, et al. *METIS: a fast integrated tokamak modelling tool for scenario design.* Nuclear Fusion Journal of Plasma Physics and Thermonuclear Fusion, 2018

[3] J. Ongena, R. Koch, R.Wolf and H. Zohm, *Magnetic-confinement fusion*, 2016

[4] Y R Martin et al. *Power requirement for accessing the H-mode in ITER*, J. Phys.: Conf. Ser. 123 012033, 2008

[5] Website: https://www.iter.org, last search 17/1/2023

[6] J. White, F. Odeh, A. S. Vincentelli, A. Ruehli, *WAVEFORM RELAXATION: THEORY AND PRACTICE*, 1985

[7] G. Birkenmeier, et al., *The power threshold of H-mode access in mixed hydrogen–tritium and pure tritium plasmas at JET with ITER-like wall*, 2022

[8] J.F. Artaud et al., *Nucl. Fusion 50 043001*, 2010

[9] Boucher D. "Études et modelisation du transport de l'energie de particules dans un plasma du fusion thermonucléaire côntrolée. Aplication au transport anormal et aux conditions du fonctionnement du tokamak", Ph.D. Thesis, Ecole Polytechnique de Paris, 1992

[10] Website: https://www.iaea.org/topics/energy/fusion/faqs, last search 17/1/2023

[11] Website: https://www.womeninfusion.org/women-in-fusion/, last search 17/1/2023

[12] Website: https://www.ree.es/es/datos/mercados/componentes-precio-energia-cierre-desglose, last search 20/1/2023

Complementary bibliography

- ITER Physics Basis Editors, ITER Physics Expert Group Chairs and CoChairs, and ITER Joint Central Team and Physics Integration Unit, *ITER Physics Basis Chapter 1, Overview and Summary*

- IOPP, *Nuclear Fusion Num.2: THE H-MODE OF ASDEX*, 2003

- C. Reux, et al., *DEMO design using the SYCOMORE system code: conservative assumptions and pathways towards the reactor*, 2016

- CIEMAT, *INF-EISSV-SL6.3-00*1, *GLOSARIO DE TÉRMINOS DE FUSIÓN Y DE ITER*, 2003

- Website: https://canviclimatic.gencat.cat/es/actua/factors_demissio_associats_a_lenergia/, last search 19/1/2023

- Website: https://www.notebookcheck.org/Analisis-completo-del-HP-EliteBook-840-G4-7200U-Full-HD.208750.0.html, last search 19/1/2023

- Website: https://www.amazon.es/HP-Compaq-LA1951g-Monitor-45-45%C2%B0/dp/B002TQO7CU, last search 19/1/2023