



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

CAMPUS D'ALCOI



# ***STUDY AND UNDERSTANDING OF THE SOFTWARE OPENWAM AND USE FOR ENGINES SIMULATION***

---

Degree Thesis by:

***JOSÉ MARÍA PÉREZ LÓPEZ***

UNDER DIRECTION OF:

***Dr. Ing. SANTIAGO RUIZ ROSALES***

***Dr. Ing HANS-PETER RABL***

**Presentation of Degree Thesis: SEPTEMBER 2017**



# DEDICATION

First of all, I would like to express my gratitude to my family and friends for their support and patience in the bad moments of this Degree Thesis (there were quite a few). Sometimes these persons have been the reason for not using devious methods or not writing a Thesis with less attention to the little details. Of course my sincere thanks go to my tutors Santi Ruiz and Hans-Peter Rabl for their time, hundreds of emails and meticulous explications. They are the other important part of this Thesis and finally Rudi Beer, the Walter Röhrli in my support team and I cannot forget Gonzalo Muñoz, Sarah Krauser and Michael Pöppel for their assistance for this text, which is also a part of my life.

Thank you very much for being there when I needed you.

José



# TABLE OF CONTENTS

1. INTRODUCTION .....	1
1.2 MOTIVATION OF THE PROJECT .....	2
1.2.1 PRINCIPAL MOTIVATION .....	2
1.2.2 SECONDARY MOTIVATION.....	2
1.3 PRESENTATION OF THE PROBLEM .....	3
1.4 OBJETIVES OF THE PROJECT .....	4
2.1 HEAT RELEASE LAWS .....	5
2.2 PHASES OF COMBUSTION. WHAT IS NECESSARY TO KNOW? .....	7
2.3 LAMBDA. WHAT IS THE VARIABLE LAMBDA WHEN ITS TALKS OVER FUEL AND COMBUSTION .....	8
2.4 DISCHARGE COEFFICIENT.....	8
3. WAMer .....	10
4. HOW TO CREATE AN ICE WITH OpenWAM.....	11
4.2 SELECTION OF GENERAL ELEMENTS.....	16
4.3 DECISION OF ENGINE .....	17
4.3.1 DEFINITION OF MODEL OF CYLINDER BLOCK.....	17
4.3.2 INTRODUCTION OF DIFFERENT DATA IN THE ENGINE .....	19
4.4 DEFINITION OF VALVES .....	23
4.4.1 NUMERATION OF ELEMENTS - AUTOMATIC OR MANUAL FORM .....	26
4.4.2 ASSOCIATION OF VALVES IN TYPES .....	27
4.5 PLENUM.....	29
4.5.1 CONFIGURATION OF PLENUM AND NODE.....	31
4.6 PIPES, DUCTS AND ELEMENTS SUSCEPTIBLE OF UNION.....	31
5. MEASURING OF PARAMETERS IN THE MODEL .....	39
6. HOW TO LAUNCH A SIMULATION WITH OpenWAM.....	41
6.1 PROCESS FOR WORKING WITH EXCEL .....	46
CONCLUSION .....	46
BIBLIOGRAPHY.....	47
ESTIMATE .....	48



# TABLE OF FIGURES

Figure 1. Different values of the factors of the form of $m_0$ .....	6
Figure 2. Blank desktop in WAMer.....	10
Figure 3. Process to define the HEAT RELEASE LAWS.....	13
Figure 4. Menu to define HEAT RELEASE LAWS.....	15
Figure 5. The layers are already introduced.....	15
Figure 6. Example of curve of graphic of representation of HEAT RELEASE LAW.....	16
Figure 7. All tabs for choosing the elements to build the engine.....	16
Figure 8. Options of engines.....	17
Figure 9. Menu to introduce the general engine data.....	19
Figure 10. Menu with the option USING ACT MODEL enabled.....	19
Figure 11. Second step to create the cylinder.....	20
Figure 12. Workspace with the four cylinders already created.....	21
Figure 13. The four options to use as a union.....	21
Figure 14. Workspace with four cylinders and sixteen valves.....	22
Figure 15. Configuration of the valves.....	23
Figure 16. Different options and data for the FLOW COEFFICIENTS.....	24
Figure 17. Valve without valve lift configuration.....	25
Figure 18. Elementst unnumbered.....	26
Image 19. Elements numbered.....	26
Figure 20. This process saves a lot of time.....	27
Figure 21. Association of valves.....	28
Figure 22. Plenum and nodes to deposits without connection.....	29
Figure 23. Plenum and node already connected.....	30





<b>Figure 24. Process of decoupling the node.....</b>	<b>30</b>
<b>Figure 25. Different positions of the couple Plenum-Node.....</b>	<b>30</b>
<b>Figure 26. All components of our example in the absence of pipes, ducts and ramifications.....</b>	<b>31</b>
<b>Figure 27. Duct without definition.....</b>	<b>32</b>
<b>Figure 28. Menu to define the duct.....</b>	<b>33</b>
<b>Figure 29. Tab of GEOMETRY and its options.....</b>	<b>34</b>
<b>Figure 30. Menu to build an intercooler.....</b>	<b>36</b>
<b>Figure 31. Configuration of the options for heat transfer in this duct.....</b>	<b>37</b>
<b>Figure 32. Intercooler with different layers.....</b>	<b>38</b>
<b>Figure 33. Menu of duct with its variables.....</b>	<b>39</b>
<b>Figure 34. Menu of cylinder with its variables.....</b>	<b>39</b>
<b>Figure 35. Menu of connection with its variables.....</b>	<b>39</b>
<b>Figure 36. Menu of plenum with its variables.....</b>	<b>39</b>
<b>Figure 37. The option of measuring in a pipe.....</b>	<b>40</b>
<b>Figure 38. Different elements with points of measurement.....</b>	<b>40</b>
<b>Figure 39. Structure of engine with four cylinders and sixteen valves ready for simulation.....</b>	<b>41</b>
<b>Figure 40. Moment to create the *.WAM file.....</b>	<b>42</b>
<b>Figure 41. Diverse types of messages caused by mistakes.....</b>	<b>43</b>
<b>Figure 42. Menu where the simulation is launched.....</b>	<b>43</b>
<b>Figure 43. Menu where the file for the simulation is introduced and which realizes the simulation.....</b>	<b>44</b>
<b>Figure 44. List of file for simulation. It is possible to simulate many files simultaneously.....</b>	<b>44</b>
<b>Figure 45. Menu to run or stop the simulation.....</b>	<b>45</b>
<b>Figure 46. Menu to configurate the file *.DAT.....</b>	<b>46</b>
<b>Figure 47. Example of data once they are configurated for the right lecture.....</b>	<b>47</b>



# 1. INTRODUCTION

The software OpenWam is a free tool created for Catedra de Motores Térmicos (from this point on CMT) at Universidad Politécnica de Valencia (UPV) for the use within this department. Therefore, this institute did not provide any tutorial in order to explain which is the best form (or simply the form) to work with it. Until today the form for knowing how to run or how to use the tools was transmitted orally between the members of the department.

OpenWam is an extraordinary and powerful instrument for the development of four-stroke combustion engines, both gasoline and gas-oil, because it can be used to predict the flow motion through the elements of an internal combustion engine or in other facilities involving compressible flow with a big precision in its data. The advantage of it is that this software is free. It is based on 1-dimensional gas-dynamic code, which is especially appropriate to solve the thermos- and fluid-dynamics of compressible flow in different elements that can be considered as 0 and 1-dimensional.

Modelling is an important tool for the optimization of internal combustion engines (ICE). The use of calculation models together with experimental tests produces unquestionable successes due to the fact that both techniques complement each other.

In 1D wave action models the engine is represented by means of ducts, where wave propagation is considered in only one dimension, and cavities where mass accumulation is considered and the gas properties are assumed to be uniform across the entire element. Finally, non-dimensional models are used to solve any connections between 1D and 0D elements.

## **1.2 MOTIVATION OF THE PROJECT**

### **1.2.1 PRINCIPAL MOTIVATION**

When I visited Professor Rabl for the first time in order to talk about my future dissertation, I already had an idea of what I wanted to do and we both thought that this work was interesting. But when I visited the second time, we talked about my project again, other options were discussed and finally my actual dissertation was agreed upon. I saw that it was a perfect opportunity for learning. As said before, the simulation has turned into one of the cornerstones of current engineering, but the form to access it is a little restricted because the personnel in this area are less by number in comparison to other areas like for example production. So, if the object of this project has been commented to a greater extent before, my motivation comes mainly from the fact that this thesis can provide a lot for me, and here I am not talking about the academic requisite, but about the possibility of it to be a big chance for my self-development as an engineer in a world so complicated to access, where it is very difficult to learn but also possible to improve professionally. That was the principal reason that I decided to be a part in this technical area.

### **1.2.2 SECONDARY MOTIVATION**

The other point, although it was not the most important one, but is of special interest for me, is to do work which has the goal to be useful for other persons. And, is there a better option than helping students? In this way, the future engineers can learn faster and better in this field and I can make a contribution (of course an almost insignificant one) to improve the training in this area, of which I hope that it will also be the area of my job in the near future.

## 1.3 PRESENTATION OF THE PROBLEM

As I explained before, the modelling is an important tool for the creation of new reciprocating engines. For this reason it is very important that students and future engineers know how to work with those software tools, even more so if possible, when they study a subject about ICE. Therefore Professor Rabl has a big interest in explaining these tools in his "VERBRENNUNGSMOTOREN" class, but he would have to deal with the purchase of a high number of software licenses, which are very expensive, in order to provide the best training for every student. One possible solution to this problem is OpenWam, however it presents a big disadvantage as there is not any tutorial available. As it was said before, this software was created for internal use in the department and its authors did not write any tutorial because when a colleague needs information about the software, they provide this information or program the tools for the right function.

## 1.4 OBJETIVES OF THE PROJECT

For all these reasons, it is convenient to discover how to use and operate OpenWam, knowing that the advantage it can offer is for both the professor and students, because it is a precise tool and, in addition to this, it is free. Furthermore, it is an open source software. That means that the users can modify or improve the functions of the tool.

For obvious reasons, the authors cannot explain to me every detail of OpenWam and I have to discover how it works by changing different parameters, running the software and searching for a relationship between the modified or varied data and the variation in the result, trying time and time again. By this process it is possible to know the exact relationship between one parameter and the function inside the program. But this *modus operandi* presents big problems. The first one is the great deal of time required to analyze the data. The second one is that some parameters are used in a lot of internal equations, which makes it more complicated to find the real relationship between input and output.

In addition to this problem the precision of the results has to be confirmed because, although the tools have been used in the preliminary phase, it is important that its results are as close as possible to the outcome of the real engine. For this reason, in the second phase of the project the real data of the engine can be used, if it is available, to run a different simulation with another similar software whose accuracy has already been proved.

This second phase of the project has the problem that the inputs are not similar. A lot of inputs are similar or they are relatively easy to calculate (for example one program needs the radius and the other the diameter) but in the other case one expects data that are very different. Therefore, the doubt exists whether the comparison is possible.

We have to be conscious of the big problem that this project may contain, considering that the "world of simulation" is not very well-known and there is not a lot of information to learn about this interesting and complex matter. Therefore the reader is asked a little empathy for possible mistakes or imprecision that this text can include.

## 2. FUNDAMENTS THEORICAL THAT THE USER MUST TO KNOW BEFORE OF DESIGN THE ENGINE

### 2.1 HEAT RELEASE LAWS

The Heat Realease Laws (HRL) 'predicts' as will to be the energy performance (or heat in this case) in the combustion process. The HRL is the most important characteristics of a ICE, because it will to indicate wich are the intrinsic characteristics (torque, emisions, efficients, etc...) of this engine.

In order to calculate that, WARMer uses the popular Law of Wiebe. When the Law or Function of Wiebe (VIBE also) is applied, it refers to a formula, which can represent the values of Heat and Pressure. With these values the diagram FQL-Angle (or dFQL for extension) inside the combustion chamber can also be calculated.

The function is:

$$x_b = \beta * (1 - \exp [-C * (\frac{\alpha - \alpha_0}{\Delta\alpha_0})^{m_0 + 1}])$$

Where:

- $x_b$  represents the energy performance which is the heat liberated in the combustion.
- $\beta$  represents the proportion of combustion already made. This value can grow from 0 (no combustion in this phase) to 1 (all fuel is burned in this phase). The maximum number of phases will be four: **Pilot**, **Premixed**, **Diffusion** and **Late Diffusion**. This point will be explained in more detail afterwards.
- $C$  represents the parameter of modulation that controls the percentage of injected fuel, which is burned during the combustion phase.
- $\alpha$  represents the crankshaft angle.

- $\alpha_0$  represents the crankshaft angle when the combustion phase starts.
- $\Delta\alpha_0$  represents the duration of the combustion in degrees.
- $m_0$  represents the factor form of the dFQL curve in this phase. This value can grow from 0.1 (the 'mountain peak' is completely on the left side) to 10 (the peak is on the right side but not totally). Figure 1 shows different forms of curves and their respective values, but they are not tabulated and it is possible to use any value between 0.1 and 10.

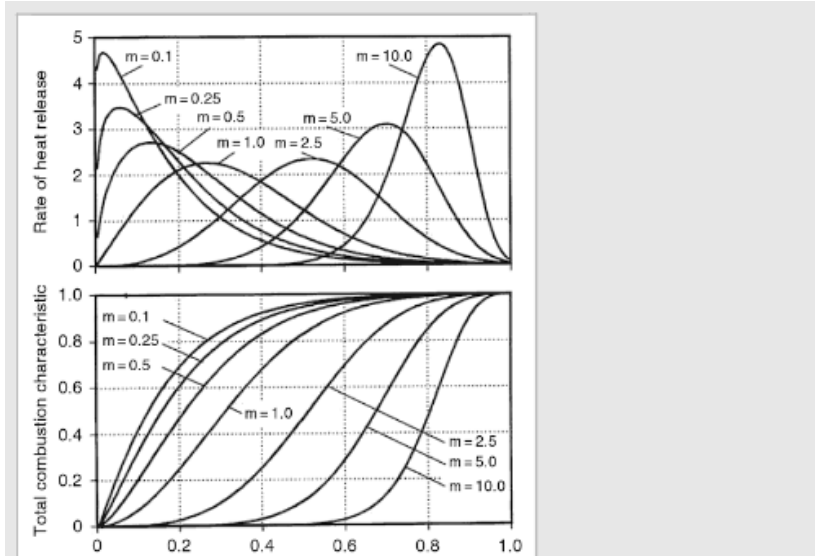


Figure 1. Different values of the factors of the form of  $m_0$



## 2.2 PHASES OF COMBUSTION. WHAT IS NECESSARY TO KNOW?

- The first and second phases are **PREMIXED** and **PILOT**. These phases help to achieve the best conditions (high Temperature and Pressure) in the combustion chamber to make the fuel combustion more effective in the **Diffusion**. The first injection is designated as **Pilot** and the second one as **Premixed**. In the system with one injection per cycle, **Premixed** is the phase between the point when the injection starts and the combustion begins. It is important not to confuse that with the **Premixed** referred to in this work, since this “time gap of combustion” belongs to the **Diffusion** phase. In WAMer the **Premixed** is an independent injection that can be completely defined. One of the big problems with the Diesel engine is its high level of acoustic emissions, provoked by the accumulation of unburned fuel since the start of the injection and the start of the combustion – the temperature is not high enough. Between 1 and 2 g of fuel are burned during the **Premixed/Pilot** so as to increase the temperature and carry out the principal combustion more easily. Here,  $\beta$  can vary between 0.03 and 0.15.

- The third and principal phase is the **DIFFUSION**. It is the “big combustion” of fuel in the chamber. It provides the energy to move the system (crankshaft rotation) and obtain the power that the user expects. Thus, the proportion of fuel burned in this phase is the largest one, which means that  $\beta$  here is always bigger than the rest of  $\beta$ 's and generally around 0.4 - 0.5.

-The fourth phase is the **LATE DIFFUSION**. This part has the function of helping to control gas emissions in the combustion. Thanks to this heat it burns better the soot particles produced in the principal combustion.

If an actual Diesel engine is modelled, the four phases can be used, but an Otto ICE is possible to be modelled with the principal phase.

## 2.3 LAMBDA. WHAT IS THE VARIABLE LAMBDA WHEN ITS TALKS OVER FUEL AND COMBUSTION

Here it is recommendable to make a pause and talk a bit about lambda  $\rightarrow \lambda$ . Lambda is the momentary ratio between air and fuel division for the stoichiometry relation between air and fuel.

$$\lambda = \frac{\text{Air Fuel Relation}}{\text{Air Fuel Relation}_{\text{stoichiometry}}}$$

Lambda is important for peak combustion efficiency in the engine, for peak efficiency and improved reliability, Fuel Economy and Lower Emissions. One should always pay attention to remember that the relation between air and fuel (the ratio LAMBDA) is not constant during all ranges of revolutions. This value goes up from idle to the point of maximal torque and down again to the maximal speed of engine.

## 2.4 DISCHARGE COEFFICIENT

The **discharge coefficient** (also known as **coefficient of discharge**) is the ratio of the real discharge to the theoretical discharge, that is, the real mass of the air or fluid that is discharged in the end of the pipe to mass of the same fluid in an ideal pipe and with the identical conditions of pressure and temperature. It is possible to say that it is a ratio of efficiency of discharge of this fluid.

Mathematically the coefficient of discharge may be related to the mass flow rate of a fluid through a straight tube of constant cross-sectional area through the next equation:

$$Cd = \frac{\dot{m}}{\rho V} = \frac{\dot{m}}{\rho A u} = \frac{\dot{m}}{\rho A \sqrt{\frac{2\Delta P}{\rho}}}$$

$$C_d = Q_{exp} / Q_{theo}$$

Where:

$C_d$  = Discharge Coefficient through the constriction (unit-less).

$\dot{m}$  = Mass flow rate of fluid through constriction (unit mass of fluid per unit time).

$\rho$  = Density of fluid (unit mass per unit volume).

$\dot{V}$  = Volumetric flow rate of fluid through constriction (unit volume of fluid per unit time).

$A$  = Cross-sectional area of flow constriction (unit length squared).

$u$  = Velocity of fluid through constriction (unit length per unit time).

$\Delta P$  = Pressure drop across constriction (unit force per unit area).

This parameter is useful for determining the irrecoverable losses associated with a certain piece of element (constriction for a valve for example) in a fluid system, or the "resistance" that an element of equipment imposes upon the flow.

This flow resistance, often expressed as a unit-less parameter,  $K$ , is related to the discharge coefficient through the equation:

$$K = \frac{1}{C_d^2}$$

This may be obtained by replacing  $\Delta P$  in the first equation above with the resistance,  $K$ , multiplied by the dynamic pressure of the fluid,  $q$ .

### 3. WAMer

When WAMer is run, the Figure 2 is shown on the screen. That is the workspace of the interface. We have the normal Menus so renowned from other programs such as FILE (New, Open, Close, Save, etc....), EDIT (Undo, Copy, Paste, Erase and very importantly, Numerate), VIEW, WINDOW and HELP.

A Zoom can be done when there are many elements on the desk. The engine diagram should be simple for an easy understanding. Therefore, the distribution of elements should be logical and it is recommendable to gather the elements according to their function or section (exhaust system, turbocharger, etc.). Afterwards, the numbering can be done element by element or in an automatic way. The second option is less intuitive, but it is recommendable because it is very fast and not totally illogical. The only problem may be that another different numeration is preferred.

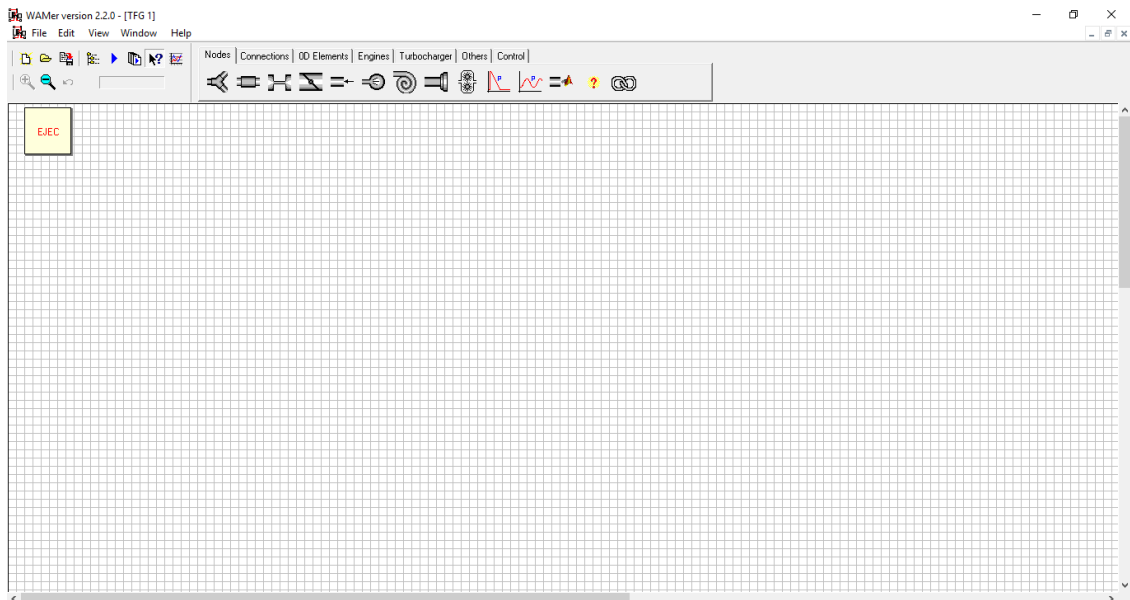


Figure 2. Blank desktop in WAMer

## 4. HOW TO CREATE AN ICE WITH OpenWAM

From this point of view, we have to distinguish between OpenWam, which is the software of simulation and the WAMer, which is the interface that means that here we “build” our engine, with all elements and its characteristics. We will make the ICE, element for element and indicate the conditions and the respective union with the other components in every element. We can see the enormous work, as an actual engine is a compound of hundreds of different pieces, and every piece has a lot of variable points. Luckily, we have not defined all valves of the operating gear system (only this system has around one hundred pieces), only the angles of the open and the closed valves and the profile of the cam corresponding. However, I don't exaggerate when I say that we need to introduce thousands of data to create this first part of the simulation.

Having done that, Warmer “translates” all the elements, its characteristics, measurements, disposition and boundary conditions into a language (C++) that OpenWam can understand to process this information in order to achieve a result.

The software is an open source software, which means that it is possible to modify the algorithm or the method to govern the simulation, but it is not the objective of this project to understand the area of OpenWam because of its high complexity, which would exceed what a grade dissertation can include.

That is the reason why this dissertation sets the focus on how to control and how to use the WAMer, considering that it is enough (and essential) to do a simulation and to valuate possible modifications in a real model. As stated before, there are hundreds of parameters and some of these parameters, for its physical origin or because the scientists do not understand totally how they work, are very complicated to explain. Normally this tutorial will have a short explanation for the user to know because it is best to use one specific value or configuration, or will provide a predetermined value for the student as a reference in a concrete point.

The web of OpenWAM, which belongs to CMT's web, has a forum for asking questions or presenting doubts about the tools. It is necessary to register previously but it is an interesting form to learn how the software works because it is possible to talk directly with the author of OpenWAM and in addition to that, one can read other doubts/solutions of the other users.

The web direction of OpenWAM is: <http://openwam.webs.upv.es/docs/>

The web direction of the forum is: [http://openwam.webs.upv.es/docs/?page\\_id=99](http://openwam.webs.upv.es/docs/?page_id=99)

And it is possible to access to the Manual of OpenWAM here:  
[http://openwam.webs.upv.es/manual/index.php?title=Main\\_Page](http://openwam.webs.upv.es/manual/index.php?title=Main_Page)

In this bachelor thesis, a four stroke Diesel engine with four cylinders and four valves (two for intake and two more for exhaust) with intercooler and with the options of EGR or not, is built. This example was chosen because it represents the popular structure of engines in most of the current cars with a Diesel engine. There are other options with three, six or eight cylinders or with only two valves per cylinder but I repeat that this composition is a minority.

In the rest of the text it is important to pay attention to always write a comma (,) and not a full stop (.) when the intention is to write a number with decimals. WAMer does not recognize the full stop for that.

## 4.1 TO DEFINE HEAT RELEASE LAWS

When the ICE icon appears on the desktop, a click with the right button of the mouse opens a menu with different options available to define a correct simulation. The first step is HEAT RELEASE LAWS (Figure 3), which allows to know the energy performance in the combustion process.

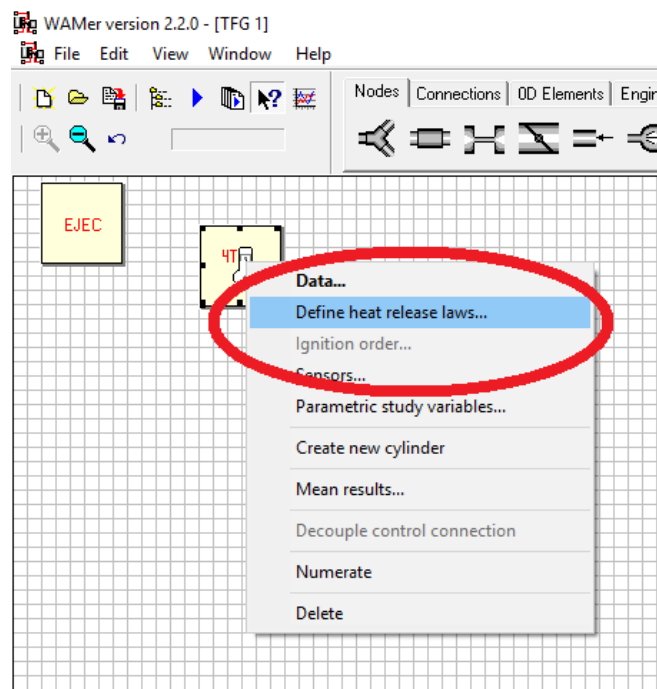


Figure 3. Process to define the HEAT RELEASE LAWS

Normal values to define the HRL in WAMer

- $\beta$  This value can grow from 0 (no combustion in this phase) to 1 (all fuel is burned in this phase). The maximum number of phases will be four: **Pilot**, **Premixed**, **Diffusion** and **Late Diffusion**. This point will be explained in more detail afterwards.
- $C$  Its value is normally 6.908 and that means the combustion has an efficiency of over 99%. Less combustion efficiency implies a lower value of  $C$ .
- $\alpha$  all free.

-  $\alpha_0$  This value is normally between:

In the pilot phase: 25 and 40

In the premixed phase: -1 and 9

In the diffusion phase: -4 and 5

In the late diffusion: -4 and 5

-  $\Delta\alpha_0$  This value is normally between:

In the pilot phase: 20 and 50

In the premixed phase: 6 and 7

In the diffusion phase: 20 and 65

In the late diffusion: 60 and 130

-  $m_0$  This value can grow from 0.1 (the 'mountain peak' is completely on the left side) to 10 (the peak is on the right side but not totally). The Figure 1 shows different forms of curves and their respective values, but they are not tabulated and it is possible to use any value between 0.1 and 10.

Now, every phase will be defined.

WAMer does not permit to click the random form option. The first and obligatory phase is undoubtedly the **Diffusion**. When this option is available, the interface allows to choose **Premixed**. Afterwards, **Late Diffusion** and, at the end, **Pilot**. But that does not mean that all phases have to be done. The **Late Diffusion** option can be clicked because two pilot injections (**Pilot + Premixed**) can be done, but after that, it is not defining its values in its respective line or a different option is defined in the line for **Late Diffusion** like  $\beta=0$ . This example is good at any phase except **Diffusion**, because for obvious reasons it is always present in the combustion.

When the submenu DEFINE HEAT RELEASE LAW is clicked (Figure 3), WAMer shows this menu (Figure 4). First, the desired phases of the cycle (red circle Figure 4) are chosen; remember the obligation of choosing the phases in the correct order and in a second step, a sufficient number of laws is picked in order to achieve a good simulation. The number of laws is chosen in the option NUMBER OF LAWS (blue ellipse Figure 4).



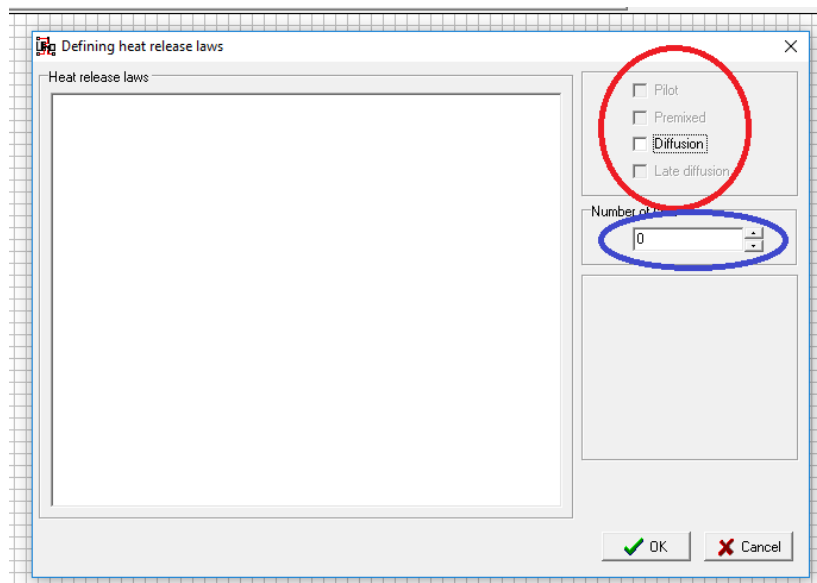


Figure 4. Menu to define HEAT RELEASE LAWS

When we have defined all parameters, something like the Figure 5 is obtained. In every law the correct respective mass of air, fuel and the rpm has to be introduced.

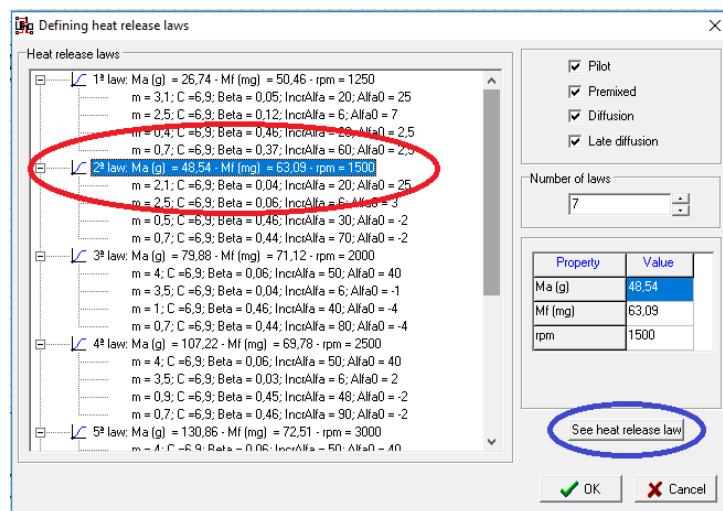


Figure 5. The layers are already introduced

Once the number of different phases is known and how to define them – using the Function of Wiebe –, the next step is learning how to implement this knowledge in WAMer.

Then click on the law that we want to know (red ellipse of Figure 5) and eject SEE HEAT RELEASE LAW (blue ellipse of Figure 5) and we discover something like the Figure 6:

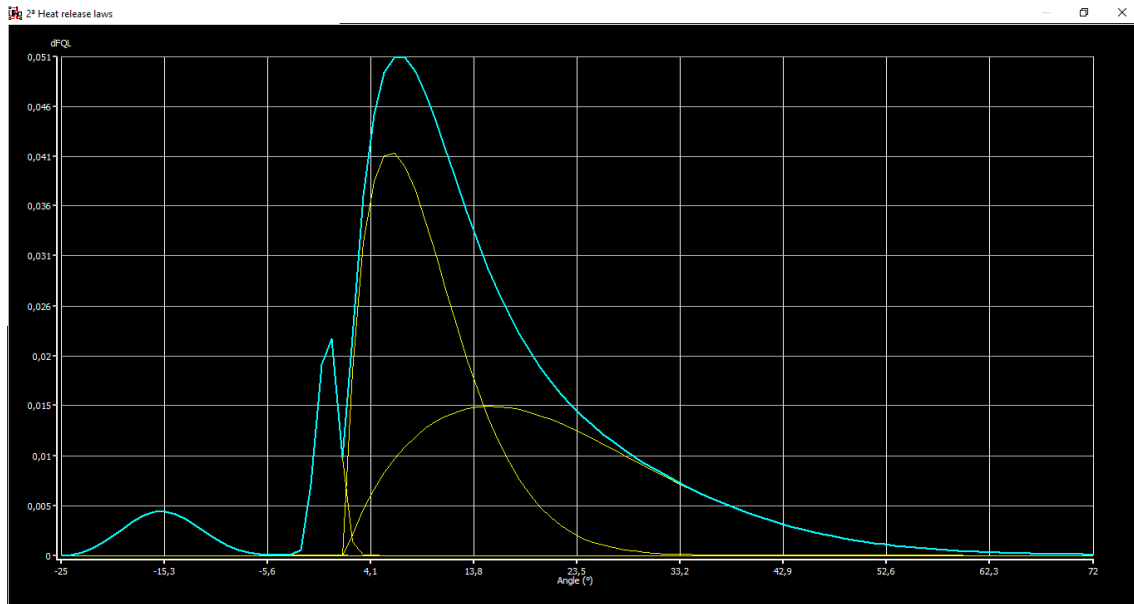


Figure 6. Example of curve of graphic of representation of HEAT RELEASE LAW

The yellow lines show every phase that forms HRL and the blue line represents the sum of all phases for this particular grade of crankshaft.

If we want to see only the dFQL of one specific phase, the procedure is the same but this time we click only on the line of the respective phase that we want to show and eject SEE HEAT RELEASE LAW.

## 4.2 SELECTION OF GENERAL ELEMENTS

We can find the menu to choose the different elements in the next image, inside of the red oval. All we need for building our engine is here.

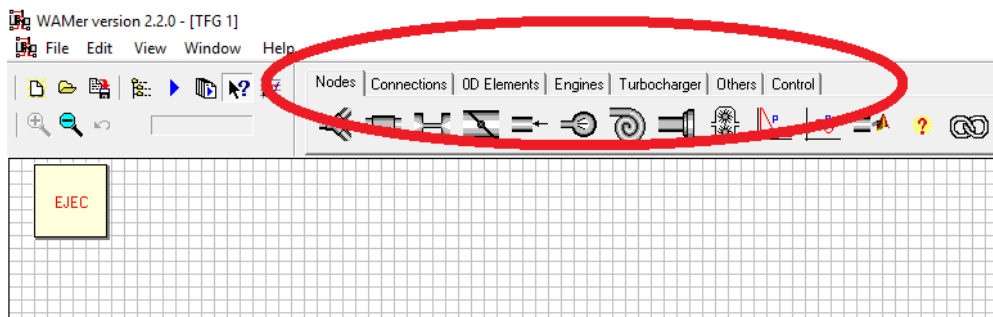


Figure 7. All tabs for choosing the elements to build the engine

## 4.3 DECISION OF ENGINE

The first step is to define what engine to use. Therefore click in the submenu of ENGINES and choose the respective option (in our example '4 Stroke' Figure 8). Click on this icon inside the red circle and drag the mouse to the desk. This process of looking for something in the submenu is equal for all families, and after one has performed the “click and drag” it will be common for the conception of the engine or other elements, except for the combining of the different pieces of tubes; but this operation will be explained afterwards.

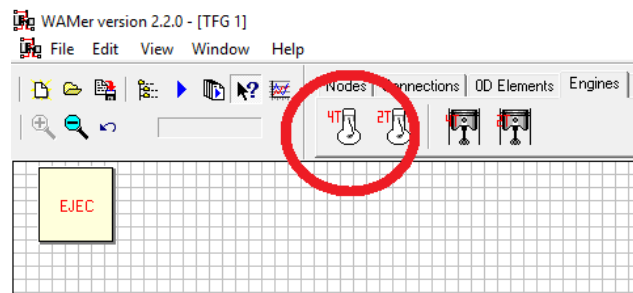


Figure 8. Options of engines

### 4.3.1 DEFINITION OF MODEL OF CYLINDER BLOCK

In order to define the model to use and the different parameters of calculation, click with the right button of the mouse on the EJECT icon and after that on DATA.

The data to define are:

- CALCULATION METHODOLOGY: COMMON is used for a unique method and INDEPENDENT in order to have more precision in the measurement of the pipe.
- NUMBER OF CYCLES FOR SIMULATION: In this point it is important to get a convergence in the calculation, for this reason a minimal number is used to obtain it.

- SPECIFIC HEAT RATIO: The ratio of specific heat is the heat that is the necessary supply for one mass unit of a certain material to raise its temperature by one unit. For this variable OpenWAM offers three options:

SPECIFIC HEAT RATIO CONSTANT → As its name indicates, this value is constant all the time.

SPECIFIC HEAT RATIO =  $f(\gamma)$  → This value is a function of the adiabatic coefficient  $\gamma$ .

SPECIFIC HEAT RATIO =  $f(\gamma T)$  → This value is a function of the adiabatic coefficient  $\gamma$  and the temperature of gas.

- AMBIENT PRESSURE (bar): These values are chosen for the user.
- AMBIENT TEMPERATURE (°): These values are chosen for the user.
- SPECIES CALCULATION: Species SIMPLIFIED (CO<sub>2</sub>, C<sub>x</sub>H<sub>x</sub>, O<sub>2</sub>, H<sub>2</sub>O, N<sub>2</sub>), or the other option EXTENDED with more different components like NO, CO, etc.
- ALLOW EXTERNAL CALCULATIONS: This option applies only for the connection with MATLAB/SIMULINK
- CONSIDER FUEL SPECIES
- FUEL SUBSTANCE

The second tab, NUMERICAL METHOD, allows to choose between some calculation models. The first model is Lax Wendroff, which solves the differential partial hyperbolic equations on the basis of finite differences. The second model is CE-SE, a method of numeric solution development with the cooperation of the NASA. And the third model is the TVD method (TOTAL VARIATION DIMINISHING). This method gives a solution to differential partial hyperbolic equations without producing oscillations.

Lax Wendroff allows to get results faster but with less precision. The TVD is the most precise of all methods.

### 4.3.2 INTRODUCTION OF DIFFERENT DATA IN THE ENGINE

The next step is introducing the different and multitudinous data that define the engine. Therefore click with the right button on the icon “4T engine” and open a menu with some options. Then click on DATA (red ellipse Figure 9) , as shown in the image on the right.

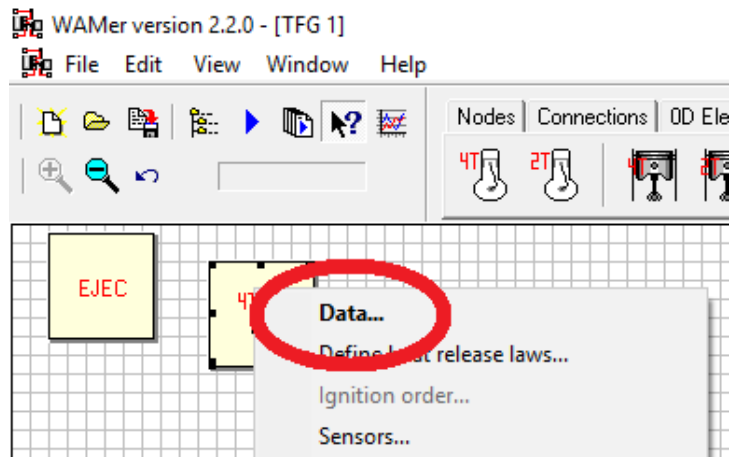


Figure 9. Menu to introduce the general engine data

In this menu we find the next options (red ellipse Figure 10): WORKING CONDITIONS, HEAT TRANSFER, GEOMETRY, MECHANICAL LOSSES, COMPOSITION and CONTROL. If we choose inside of WORKING CONDITIONS the option of

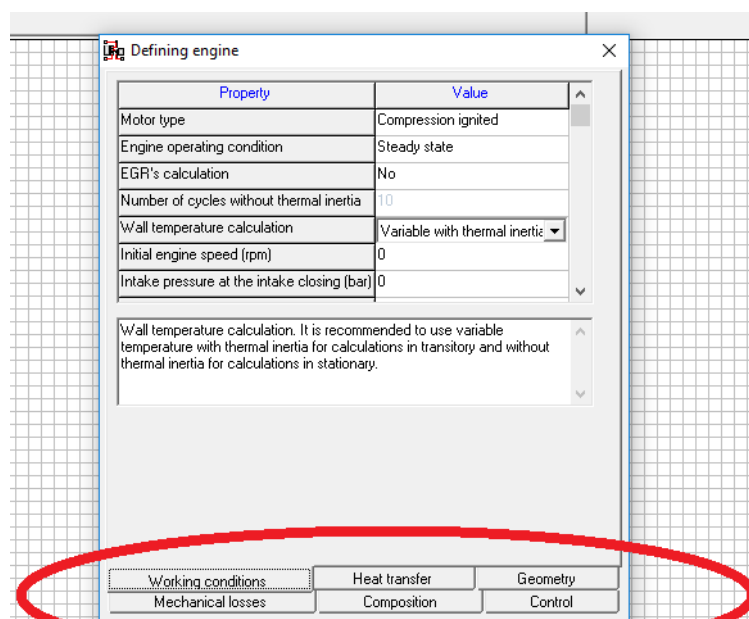


Figure 10. Menu with the option USING ACT MODEL enabled

COMBUSTION MODEL → USING ACT MODEL, we have in addition to this the options of INJECTION SYSTEM and INJECTION PULSES.

This point is now explained. In COMBUSTION MODEL, we can choose between two options:

- IMPOSING HEAT RELEASE LAW or;
- USING ACT MODEL

As it was stated before, the HEAT RELEASE LAW defines what the “form” of production of heat inside the combustion chamber is like and for this reason, we do not define the injection system, because it is not necessary. All variables and elements that define the injection system are replaced simply for this law.

Now we are going to create the cylinder. Click with the right button on the engine block icon and choose CREATE NEW CYLINDER (red ellipse in the Figure 11). We do that as many times as we want to create new cylinders, in our example four times.

The cylinders appear just below the icon of “4T Block” and automatically the next cylinders are appearing below the last cylinder that we created. Keep in mind the right distribution of elements in the work place because that is only the start of the construction and we have to sum up a lot of more elements.

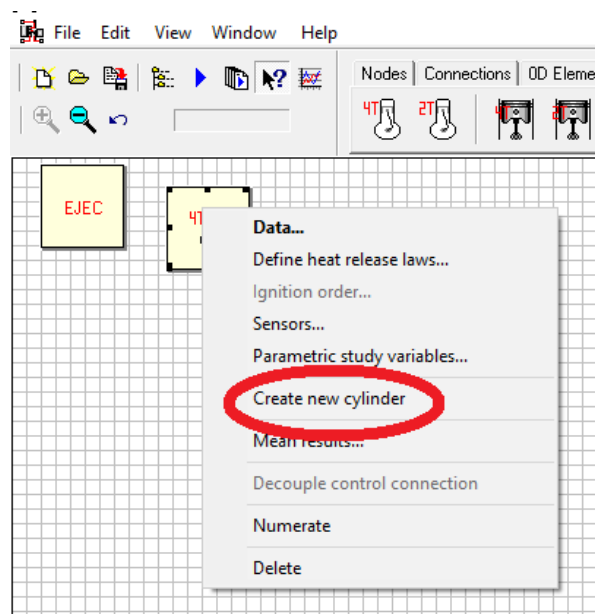


Figure 11. Second step to create the cylinder

At the same time as we have the four cylinders (Figure 12), we create the valves, both the intake and the exhaust. Therefore with the right button we click on the icon of cylinder and choose CREATE UNION.

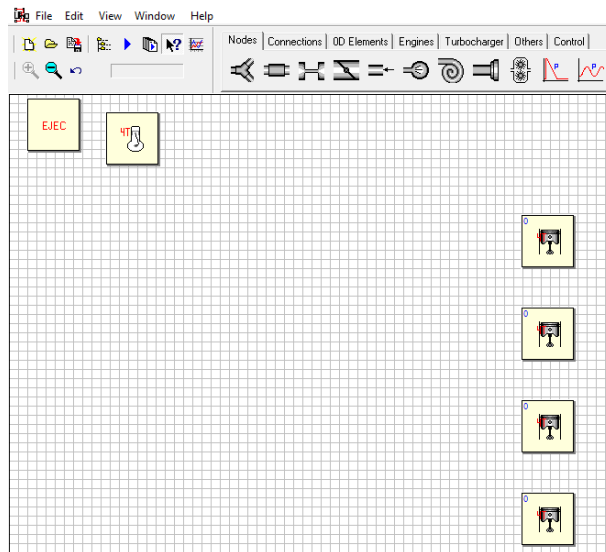


Figure 12. Workspace with the four cylinders already created

When this submenu appears (Figure 13), one can choose what type of element one wants (VALVE, NODE TO DEPOSIT or NODE OF EXTERNAL CALCULATION), the function (INTAKE or EXHAUST), the position (LEFT or RIGHT) and the number. In our case, two intake valves in the left position and two exhaust valves in the right position. We repeat these steps for the other three cylinders.

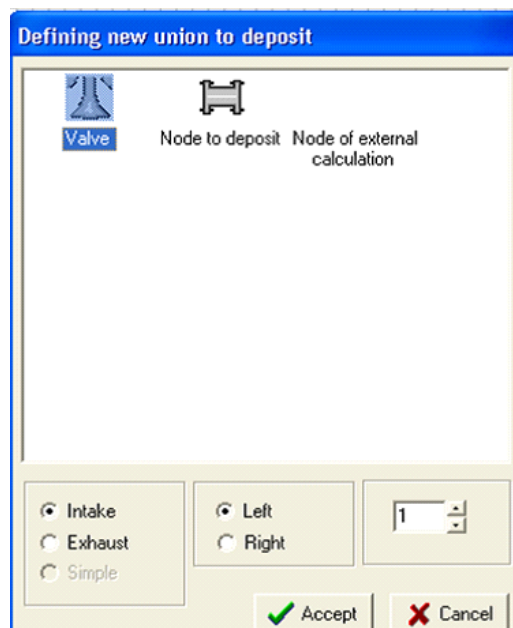


Figure 13. The four options to use as a union

Finally, we will have a work space like that:

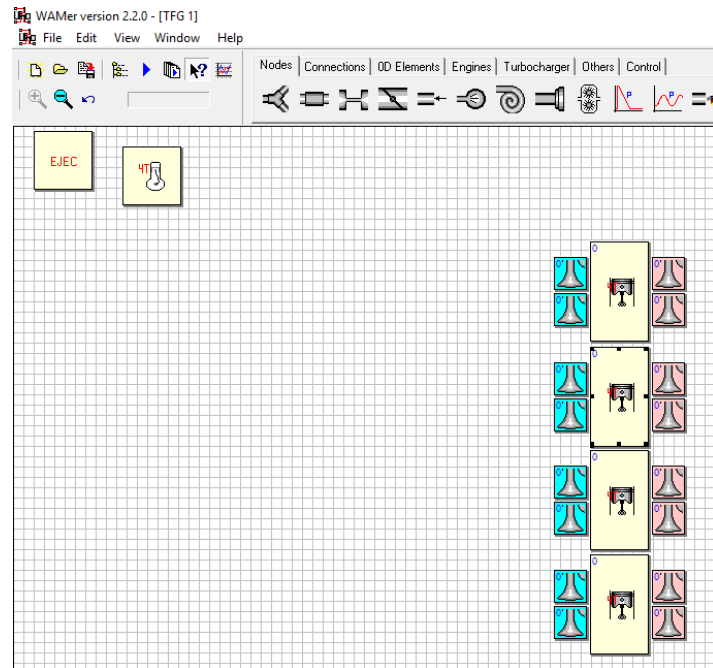


Figure 14. Workspace with four cylinders and sixteen valves



## 4.4 DEFINITION OF VALVES

Now the valves are defined. With a double click on the valves icon one obtains a menu like that of Figure 15, where in the first tab DATA you decide the:

- VALVE DIAMETER (mm): measure of the valve head in millimeter
- VALVE OPENING (°): The angle of the crankshaft when this valve is open
- SPEED CONTROL
- SPEED
- SPEED RATIO

Property	Value
Valve diameter (mm)	0
Valve opening (°)	0
Speed control	Self control
Speed	0
Speed ratio	1

Note: valve opening (°)

Figure 15. Configuration of the valves

On the second tab, FLOW COEFFICIENTS (Figure 16), choose all data relative to the coefficients of load/download to the valves as the effective section of valves.

- LIFT INTERVAL (mm): The distance between points that is chosen in order to determinate the different coefficients.
- REFERENCE CALCULATION DIAMETER (mm).
- NUMBER: Indicates the position (in mm) of the valve head. For example, 1 is position zero; 2 is 0 plus 1 time the LIFT INTERVAL (LI); 3 is 0 plus 2 times the LI; and etc.
- OUTLET FLOW COEFFICIENT: The coefficient that the gas has when it flows from the combustion chamber to the pipe.

- **INLET FLOW COEFFICIENT:** The coefficient that the gas has when it flows from the pipe to the combustion chamber.
- **SWIRL COEFFICIENT:** The coefficient of gas (only on INLET) in the case that it produces a swirl effect when it flows around of valve.

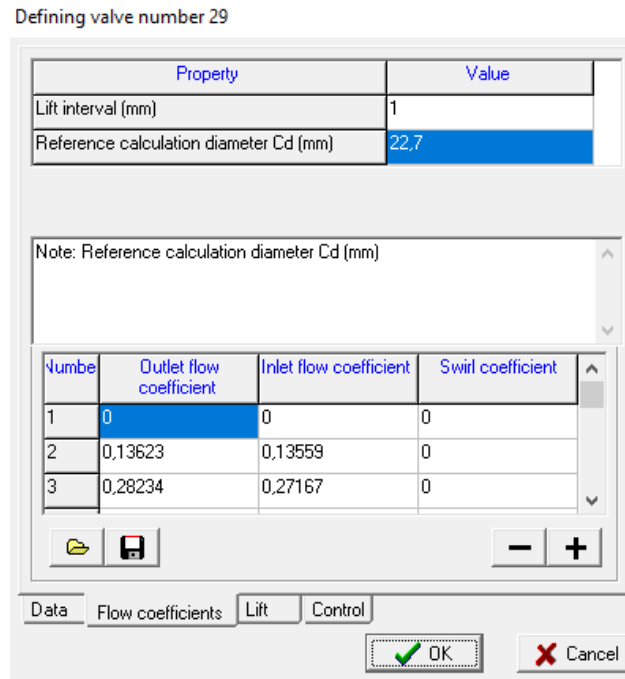


Figure 16. Different options and data for the FLOW COEFFICIENTS

The third tab is LIFT and here are introduced all data relative to the profile of the cam. On:

- **ANGLE STEP(°):** You indicate how many degrees there are between one point and the next point in the table of VALVE LIFT.

Attention must be paid to how to introduce this data because there are two systems to do that.

The **first option** is to click on the symbol + (red circle in Figure 15) and introduce one data of VALVE LIFT. You do this every time that you need to introduce a data of lift, but if you think for a minute that for a reliable representation of the profile of the cam, around 250/300 data are necessary, you will understand immediately that this form of work is not efficient.

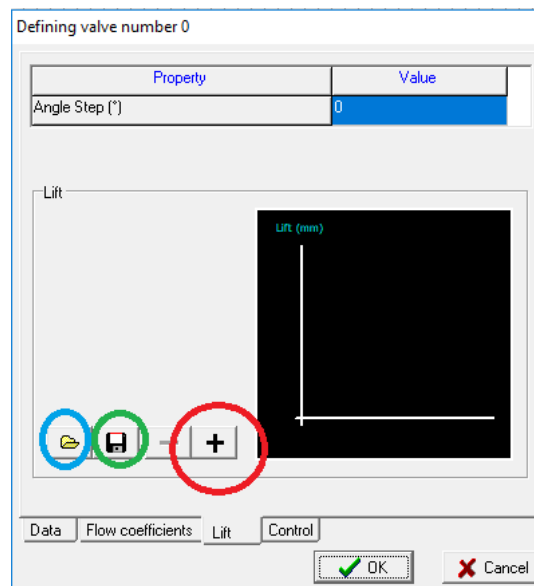


Figure 17. Valve without valve lift configuration

The **second option** is absolutely easier and faster. You generate the profile (with data, not graphically) with a program like Matlab for example or manually, and copy this data in a file of Note Block (Windows

Accessory) in the form of a column and afterwards, on top of it we write the number of data that we introduce. Here it is important to mark that **the separation between unit and decimal is only valid by using a full stop (.) and not a comma (,)** and WAMer only admits six decimals for data but it is not obligatory to use all decimals. Once we have the file finished, we come to the LIFT tab and click on the **folder icon** (blue circle of Figure 17). Now WINDOWS EXPLORER is opened and one can look for the file and introduce it. You can also export the date. Therefore click on the **Diskette icon** (green circle Figure 17) and we generate a LEV file, that we can work on with Note Block. The work is the same for the intake valve and exhaust valve. We do not have to do this task for every valve. You can ASSOCIATE elements but before this, it is recommendable to explain another point.

#### 4.4.1 NUMERATION OF ELEMENTS - AUTOMATIC OR MANUAL FORM

We can use a lot of elements to design an ICE and the biggest problem is that possibly some parts are used many times. When this happens, how can we name one element to distinguish it from other identical elements?

The answer is the NUMERATION. Not all components can be numbered, but almost all.

Every element susceptible for numeration can be numbered manually if we click with the right button and choose NUMERATE MANUAL. I recommend this form, when for a specific reason we want one element to have a specific number and no other but this system can be very laborious, therefore it is almost always better to use automatic NUMERATE. In the upper left corner we find the normal menus of other programs and if we click on Edit, we find the option of NUMERATE. When we use this function all elements get a number that makes it easier to work with.

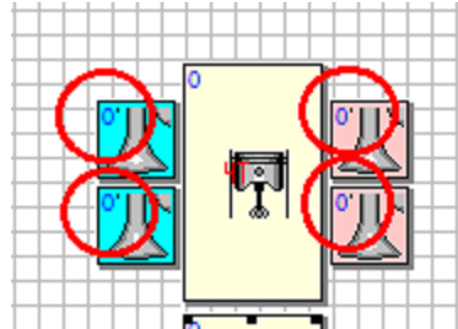


Figure 18. Elementst unnumbered

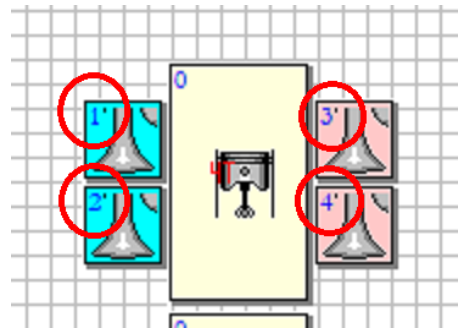


Image 19. Elements numbered

## 4.4.2 ASSOCIATION OF VALVES IN TYPES

Now, every valve has a number that makes it different from other valves. The association of valves in types has one big advantage: we save time because we do not have to write the same parameters several times, but attention, if we change some parameters of some elements from this “family”, this element is out of the group automatically. If we want to modify some data, first we do this modification and after that we associate every element to the new modified element one by one again.

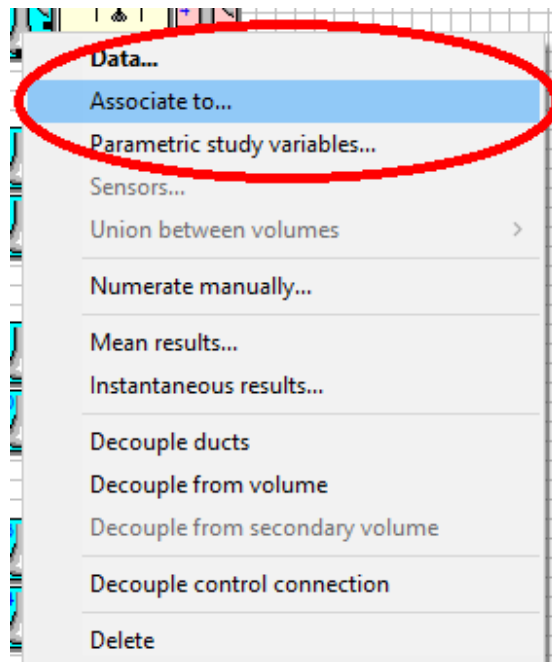


Figure 20. This process saves a lot of time

When we numerate the elements automatically, WAMer produces a type for every valve and, in our example, we connect all Intake valves to valve 1 and the Exhaust valve to valve 3 (both previously defined. See section **DEFINE VALVE**); therefore click with the right button on the icon of the intake valve that we want to connect to valve 1 and click straightaway on **ASSOCIATE TO...** (Figure 20). Next, WAMer shows a menu like the one of Figure 18 and we look for the principal valve that will be associated. We do that for Intake group as well as Exhaust. We repeat this process for every element to be associated. In Figure 18 we can see that the valves 2, 5, 6, 9, 10, 13 and 14 are already integrated in the same group (red square).

If we are not sure during the process of association or if one of the elements has the parameters we want, we can check its configuration by marking the element (it changes to blue colour) and next click on DATA (blue circle Figure 21).

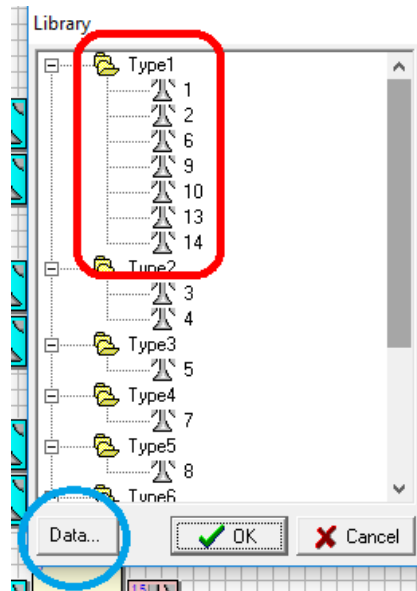


Figure 21. Association of valves

## 4.5 PLENUM

A plenum is any chamber or space where the air is accumulated. This accumulation can be produced on purpose because this chamber has a special function for balancing the pressure in a special point or it can be produced for the geometry of the part of engine – for example a two end part of an intercooler or the union of two or more pipes in a manifold. NODE TO DEPOSITS is used because there is normally a loss of pressure between the plenum and the duct, and the node to deposits helps to introduce this loss. If there is not a loss of pressure, write only a zero in this variable because it is obligatory to use this element. Therefore pay attention to use a Plenum at every point where an accumulation of air without a pipe is possible.

One time that we have the cylinders and the valves already, we introduce the PLENUM and its connections in relation to NODE TO DEPOSITS, a minimum of two for a Plenum (one for Intake and another one for Exhaust). Therefore we choose the principal menu (Image 7) and select the option OD ELEMENTS, and within this menu, choose the option CONSTANTS VOLUME CHAMBER (red arrow in figure 22) and, in the submenu connections, NODE TO DEPOSITS (blue arrows in figure 22).

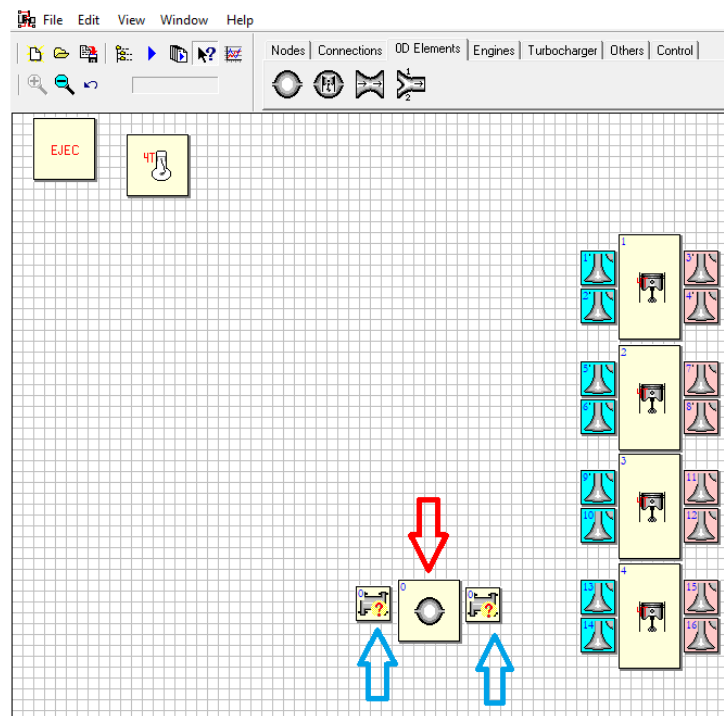


Figure 22. Plenum and nodes to deposits without connection

In order to put the elements in the work space, we proceed with the process of click and pull. We use this method for the implementation of the rest of the elements of the engine.

Now, we should COUPLE or DECOUPLE the Node to the Deposit with a process that is very easy, as we show in the next images.

COUPLE: Click on the icon of Node and pull it on the icon of Deposit. WAMer shows a positive sign (+) when it recognises the element. Let go and the two elements are connected (Figure 23) and that creates a unique body. If we move one element, the other is moved, too.

DECOUPLE: Here the process is different. We click with the right button on the Node and a menu appears like the one in Figure 24. Choose the option DECOUPLE FROM VOLUME and the Node is again free to be united with another element. If we want to just delete the element, we only have to mark it with the mouse and to use the button of DELETE on our keyboard.

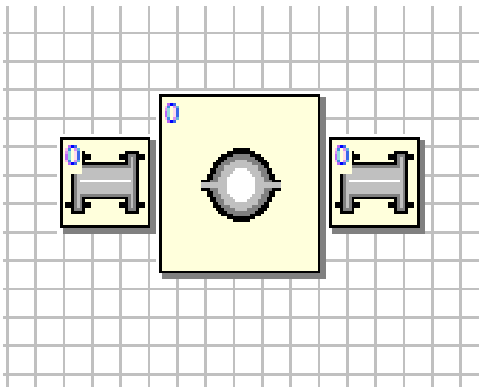


Figure 23. Plenum and node already connected

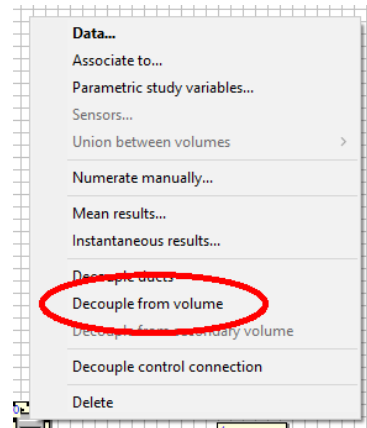


Figure 24. Process of decoupling the node

The ensemble of Plenum-Node has numerous options of positions. We choose one way or another one dependent on which is the optimal position for our drawing (Figure 25).

We repeat this process for every plenum that we need in our diagram.

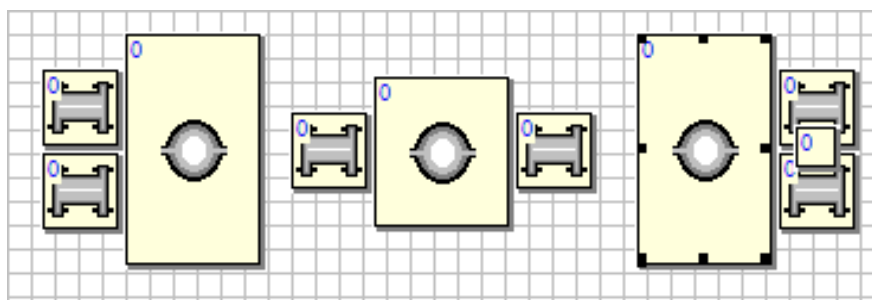


Figure 25. Different positions of the couple Plenum-Node



### 4.5.1 CONFIGURATION OF PLENUM AND NODE

The next Figure 26 is our example with all plenums and the turbocharger (the process for placing any elements is similar to the one described previously).

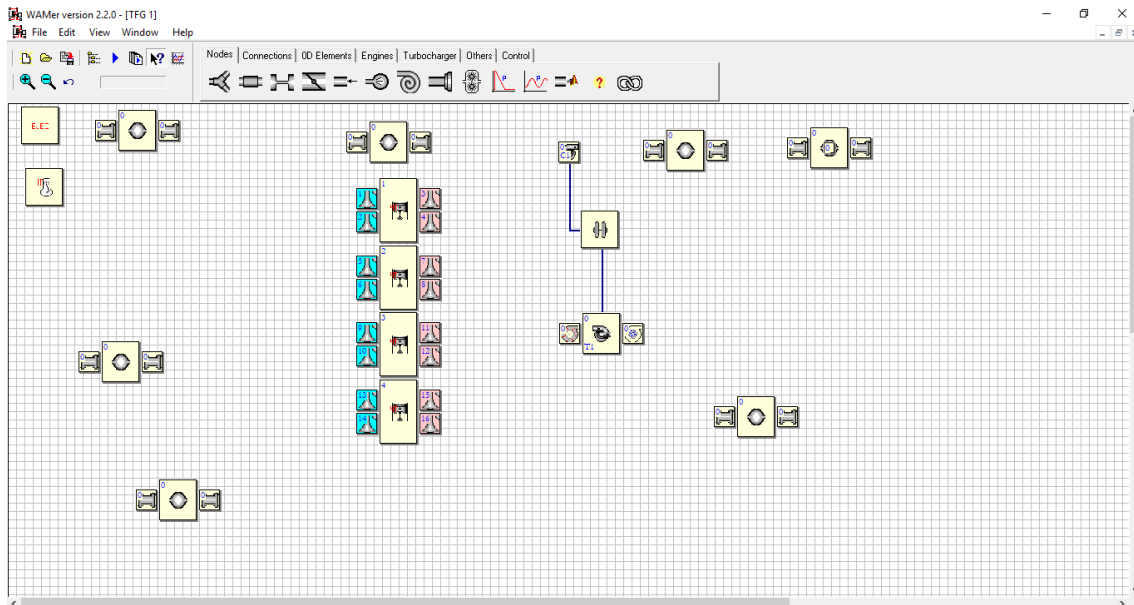


Figure 26. All components of our example in the absence of pipes, ducts and ramifications

And now it is time to connect the pipes and ducts with the plenums and other components.

### 4.6 PIPES, DUCTS AND ELEMENTS SUSCEPTIBLE OF UNION

It has to be emphasised that with ducts it is not the same as with the elements that we can join. An element is simply something that has a function (valves, turbocharger or compressor) or represents a physical phenomenon inside of the engine (laminar pressure loss or Extreme to or contact with the atmosphere). The icon of this element indicates a point where just this function or phenomenon is produced. Every element, when it is produced in real life, has a “chamber” through which the air flows. Certainly, the duct means this “chamber” where the air moves from point to point (if there is a big increase of volume, it is named Plenum and that will be explained afterwards). For example, inside the cylinder head, between the manifold and the valves, there is a pipe

where the air flows. Indeed, this pipe is a part of the cylinder head, but it is considered to be a duct, that joins the valve with the other element. Every time that a pipe changes some characteristics, it is considered a different duct. Therefore inside of manifolds you have to use a lot of ducts, because in a few centimeters it is possible that you find some unions or change of sections. As will be explained in the future, the conic section presents problems in OpenWAM. Summary: All “chambers” through which the air flows are defined as DUCT. It is possible to connect all elements that are available in the menu of WAMer, then why has WAMer the menu of CONNECTION if we can connect different parts only with a duct? The answer is that the option of CONNECTION provides a lot of alternatives, that adapt better to our conditions or requirements (for example the end part of the duct with direct connection to the environment or part where laminar or turbulent conditions are simulated, a compressor or the end part of the duct without exhaust) and the ducts do not. At first glance we could think that the union of two parts is as easy as creating the union of the elements with a line but immediately you will see that the point of conduction is a little more complicated.

The elements more used for the union between the different parts is the UNION OF TWO PIPES if we connect only two bodies and the RAMIFICATION if we carry out the union between three or more pieces.

In order to accomplish the duct, click with the left button of the mouse on the first element and pull the cursor (without dropping the button) to the second element and now drop. The theoretical direction of flow is determined by the direction in which the mouse is pulled. That means, if the cursor is pulled from element X to Y, the direction flow will be  $X \rightarrow Y$ . The direction is indicated by an arrow close to the square of the duct (red circle Figure 27).

The duct is ready but we should define it now.

By a click on the square of the duct (red circle Figure 27) a menu as Figure 28 is shown.

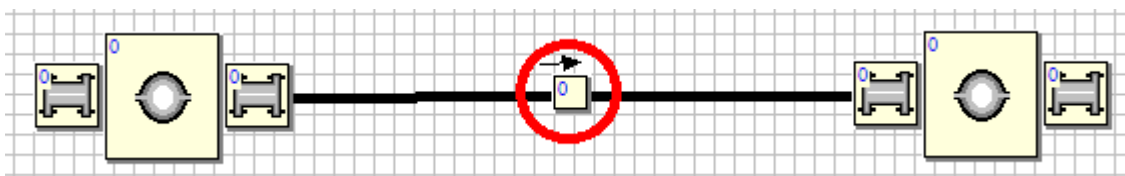


Figure 27. Duct without definition

Here we find four more submenus that help us to better define the tip of the duct. In the first submenu, DATA, we enter the general data as:

- SPATIAL MESH (mm): It defines the calculation used in this duct. If the measure between two or more points is small, the precision is more exact but the amount of time of the processing of the data is bigger, too. Normally values of cm are used for the secondary ducts and values of mm for the critic ducts, for example ports.
- DUCT TYPE: Here are four options; INTAKE PIPES, INTAKE PORTS (cone valve seat), EXHAUST PIPES and EXHAUST PORTS. For all intake pipes you use the INTAKE PIPE. For the exhaust manifold → EXHAUST PORT and for the rest of exhaust ducts, EXHAUST PIPES.
- WALL TEMPERATURE CALCULATION: Here you should consider if you want that this duct has the function of INTERCOOLER or not. For makes an INTERCOOLER see the respective section later, in the other case, choose CONSTANT.
- DUCT ROUGHNESS (mm): The normal value is 0,15 mm.
- INITIAL WALL TEMPERATURE (°C): Temperature of the duct wall. In the case of Intake Manifold use the same temperature T0 of the bloq EJEC. For a duct placed behind the compressor, the admissible temperature is 50° C over the T0. For the exhaust system, a values are used between 500 to 600° C for Diesel engines and 700 to 800° C for Gasoline engines.
- INITIAL GAS TEMPERATURE (°C): Temperature of air or gas respectively inside of the conduction. You have to do the same rationing as in the INITIAL WALL TEMPERATURE
- INITIAL GAS PRESSURE (bar): The pressure of air or gas respectively inside of the conduction. You have to do the same rationing as in the INITIAL WALL TEMPERATURE
- INITIAL GAS VELOCITY (m/s): Speed of air or gas respectively in the conduction. A value of 0,01 m/s gives a stabler computacion.
- HEAT TRANSFER CORRECTION COEFFICIENT
- FRICTION CORRECTION COEFFICIENT
- NUMBER OF INTERCOOLER DUCTS: See Intercooler Section.

Defining duct number 99

Property	Value
Spatial mesh (mm)	5
Duct type	Intake pipes
Wall temperature calculation	Constant
Duct roughness (mm)	0,1
Initial wall temperature (°C)	115
Initial gas temperature (°C)	115
Initial gas pressure (bar)	2
Initial gas velocity (m/s)	0,001
Heat transfer correction coefficient	1
Friction correction coefficient	1
Nº of intercooler ducts	1

Note: Duct roughness (mm)

Data | Geometry | Numerical method | Composition

Figure 28. Menu to define the duct

In the tab of GEOMETRY, the geometry of the duct can be defined (Figure 29). There are four options in this point:

- INLET DIAMETER (mm): It is the diameter of conduction in its beginning.
- LENGTH (mm): Length of this section.
- OUTLET DIAMETER (mm): It is the diameter of conduction in its last point.
- +/-: Introduction of different sections: Later it will be explained why this option is not recommended. By pushing the icon + (red circle Figure 29) it is possible to add more sections and to create a connection between all of them, with identical inlet diameters. In the same manner, it is possible to remove this section by pushing on the icon --.

It is of vital importance to check the direction of flow (Figure 27) when some sections are used in an identical duct because the conicity reverses its value and the results can be totally different.

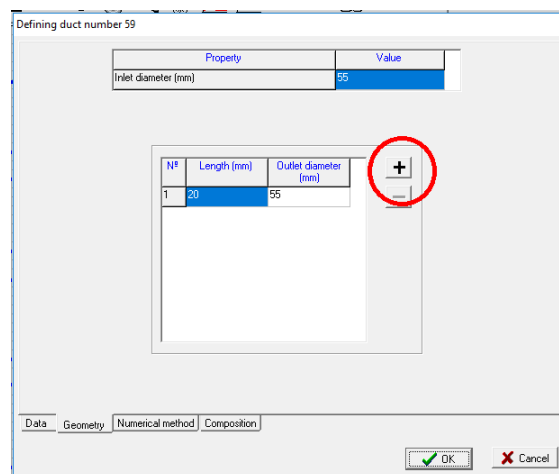


Figure 29. Tab of GEOMETRY and its options

At this point it should be explained that the duct has a conical form, this may cause the software to produce a problem of instability. Therefore it is better to do a division of the duct into a number of elements and introduce a different diameter in every component. By definition, a conical section has an infinite diameter. As is logical we cannot introduce an infinite section of ducts for the development of a simulation. It is enough if we know that more sections produce more precision in our simulation. In this case, we insert some UNION OF TWO PIPES in order to do the division.

The third tab is NUMERICAL METHOD. Here there are the same options as for the definition of the EJEC block. For the intake system the Lax Wendroff method is recommended, for the exhaust, the TVD is better.

The fourth tab is COMPOSITION of gas. Here there are the same options as for the definition of the EJEC block.

## 4.7 HOW TO BUILD AN INTERCOOLER

OpenWAM does not offer the direct option of intercooler, therefore it has to be modeled in a process as follows:

The first step is to connect two elements with a duct (see the part PIPES, DUCTS AND ELEMENT SUSCEPTIBLE OF UNION). This pipe is transformed into a heat exchanger. The next step is to change the option WALL TEMPERATURE CALCULATION → VARIABLE WITHOUT THERMAL INERTIA and N° OF INTECOOLER DUCTS (red ellipse Figure 30) and to change the different values of the duct.

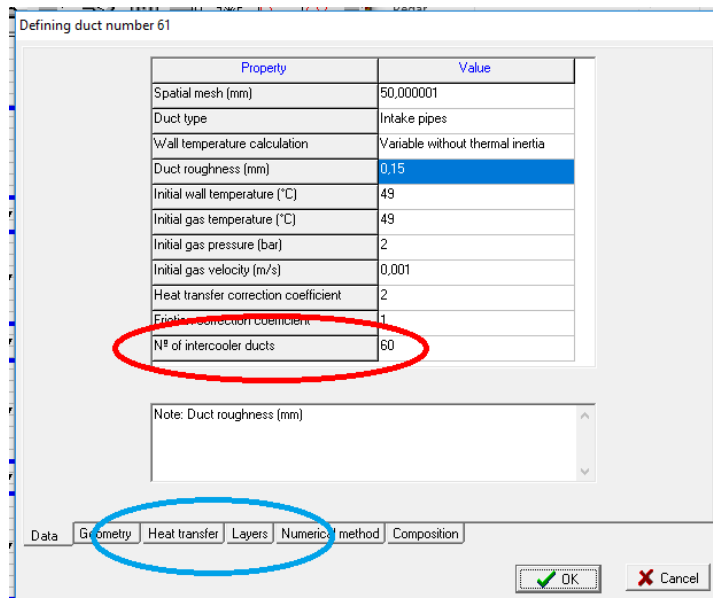


Figure 30. Menu to build an intercooler

Automatically, when the option VARIABLE WITHOUT THERMAL INERTIA is marked, it shows two more tabs, HEAT TRANSFER and LAYERS (blue ellipse Figure 30). It is necessary to modify the data of these options to find the correct value that defines our intercooler.

In the tab of HEAT TRANSFER the Figure 31 is shown to introduce the type of refrigerant. If the option AIR is marked, WAMer uses the environment temperature as the

temperature of coolant, but if one chooses the option of WATER, it is necessary to determine also the temperature of this fluid. In EXTERNAL CONVECTION CORRECTION COEFFICIENT and EXTERNAL EMISSIVITY the values that define our element have to be introduced.

Defining duct number 61

Property	Value
Coolant type	Air
Cooling water temperature (°C)	0
External convection correction coefficient	10
External emissivity	0,5

Note: Coolant type

Data Geometry Heat transfer Layers Numerical method Composition

OK Cancel

Figure 31. Configuration of the options for heat transfer in this duct

In the option of LAYER values are defined such as WALL THICKNESS, MATERIAL DENSITY, MATERIAL SPECIFIC HEAT, MATERIAL CONDUCTIVITY, INTERNAL EMISSIVITY or EXTERNAL EMISSIVITY, but it is not possible to use parameters in a single layer. Why is this so? Because the options are used in order to define the duct of intercooler and the other one to define the coolant fluid. The difference is in the tab MAIN LAYER (for the duct) or FLUID LAYER (for the coolant medium). If the option of FLUID LAYER is marked, the option of MAIN LAYER changes automatically to NO and it is not possible to introduce the value of MATERIAL DENSITY and MATERIAL SPECIFIC HEAT. However, the option of INTERNAL EMISSIVITY and EXTERNAL EMISSIVITY is open. It allows to add as many layers as wanted, the form that is possible to build an intercooler with different layers. As in the other menus of WAMer, a click on the symbol '+' or '-' to add or delete a layer, as one can see in Figure 32.

Figure 32. Intercooler with different layers

Once the data are defined, one has to modify the values of 'Wall temperature', 'Air temperature' and 'Length' of the duct to find out the correct amount of heat that the intercooler needs to dispel. The rest of the tab, GEOMETRY and COMPOSITION, are known already from similar configurations in other elements.



## 5. MEASURING OF PARAMETERS IN THE MODEL

### MODEL

Once the model is built, this is the moment to select what to measure and where. The answer to the first point has two more options, because OpenWAM gives the options of instantaneous or average measure. This said, normally it is possible to know the data of pressure, temperature, speed, mass flow rate, species mass fraction among others in every element, but also every element has its special variable that it measures. (Figure 33, 34, 35 and 36). The other variables specific of an element (for example Wochni's coefficient in the cylinder, or discharge coefficient in a connection) are shown in their respective menu.

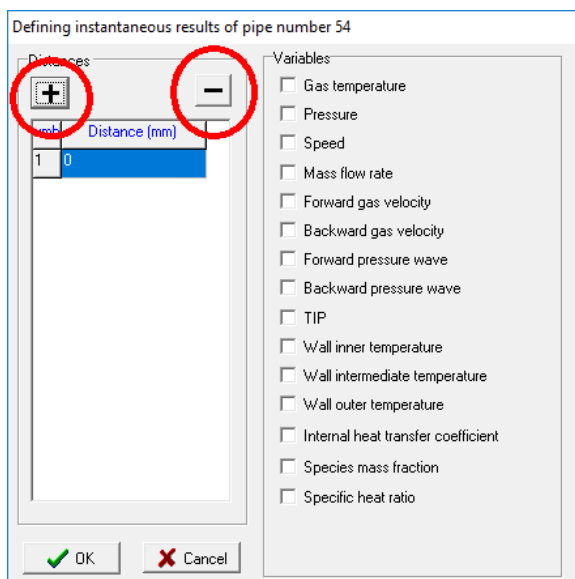


Figure 33. Menu of duct with its variables

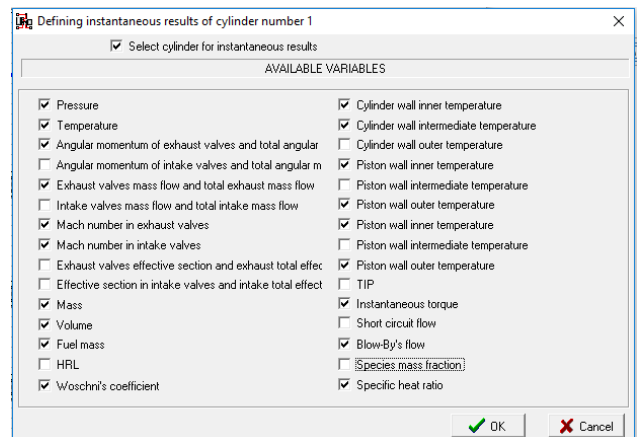


Figure 34. Menu of cylinder with its variables

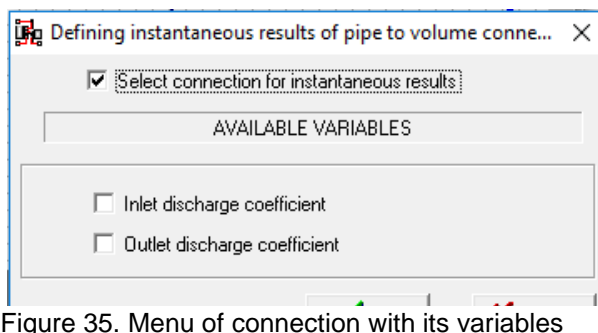


Figure 35. Menu of connection with its variables

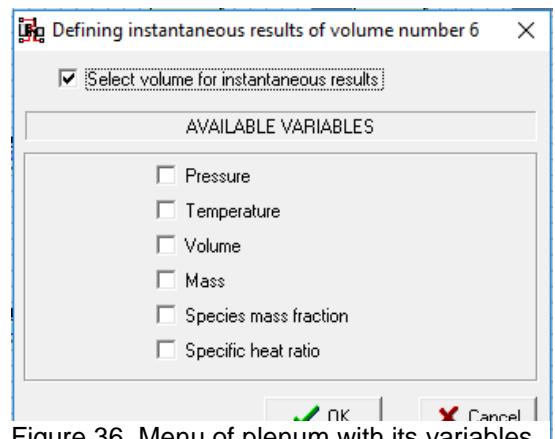


Figure 36. Menu of plenum with its variables

In order to know this measurement, click with the right button of the mouse on the element of which you want to know the variable and choose MEAN/INSTANTANEOUS RESULTS (red ellipse of Figure 37).

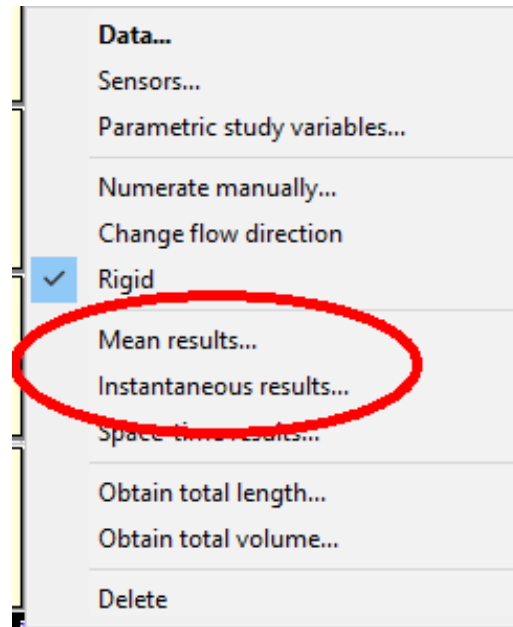


Figure 37. The option of measuring in a pipe

The similar process is used for the study of average measures and therefore the steps of explication are not repeated here.

We know that an element has a point of measurement when there is a symbol in the form of the graphic as it is possible to see in Figure 38 (eight red circles).

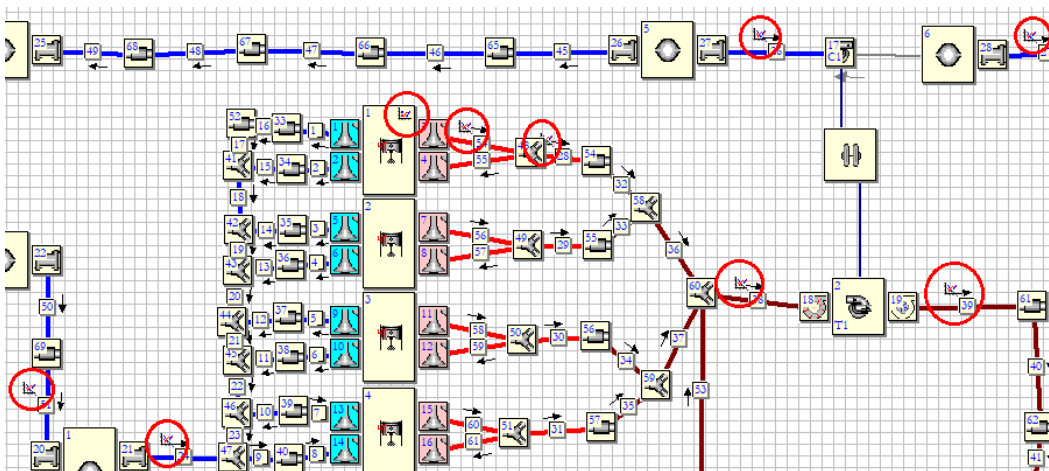


Figure 38. Different elements with points of measurement

## 6. HOW TO LAUNCH A SIMULATION WITH OpenWAM

Now that the engine is correctly configured with WAMer, it is time to launch the simulation and to check the result.

In this directory [http://personales.upv.es/hcliment/openwam/running\\_simulation.flv](http://personales.upv.es/hcliment/openwam/running_simulation.flv) it is possible to download a video with the instruction for the execution of a simulation; but just in case the link is not available or presents other different problems, it indicates a detailed process to launch the simulation under the operative system of Windows. The second process is possible through MS-DOS, but for reasons of space, it will not be explained in this thesis.

Once that all data, variables, etc...are introduced in the engine, it has a structure as the Figure 39 shows. The next step is to create a file \*.WAM, that is the file compatible with OpenWAM. For that, click on the icon of “play” (red circle of Figure 39).

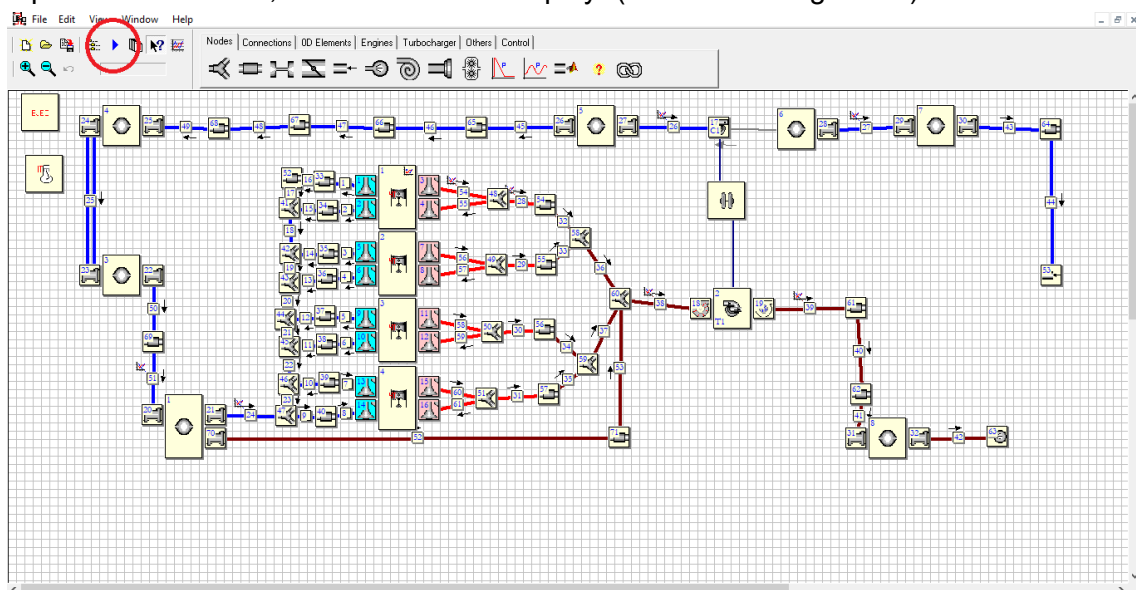


Figure 39. Structure of engine with four cylinders and sixteen valves ready for simulation

If there is any change since the last time that it saved the engine model, WAMer will save for the first time a file in the format of \*.PCS – this type of file is compatible only with WAMer - and finally, another type \*.WAM (Figure 40). As always, WAMer asks where the user wants to save the document and with which name. That is a routine.

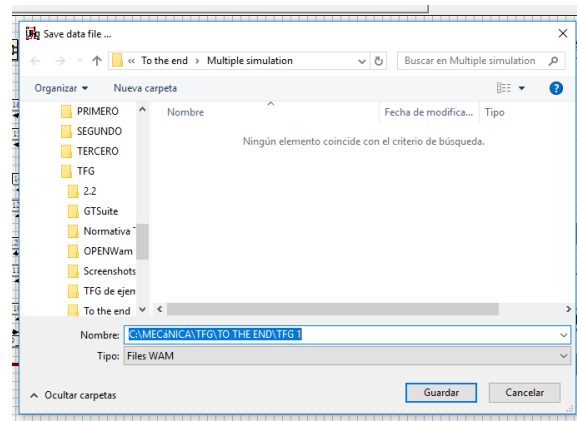


Figure 40. Moment to create the \*.WAM file

Before creating this document, the interface checks the sketch, all data, configuration and variables, and then claims that this model can be simulated. Therefore, the software can give a message if the engine has a mistake or is not correctly configured. This message can be WARNING (red ellipse of Figure 41) if the problem can vary or influence the result of the simulation or ERROR (blue ellipse of Figure 41) if the mistake is important and impedes the correct simulation of the model.

**IMPORTANT:** If this model has an ERROR message, it cannot be saved as a \*.WAM file. The unique solution is to check this mistake and solve it. This is common with the other software of programming. On the other hand and as previously recited, the WARNING messages are not vital but attention must be paid to them because these mistakes may not totally disable our model but can have a significant effect on the final result. In the example of Figure 41, one must go to duct 45 and in the tab of composition of gas check that the sum of all gas elements is 1.

Once all mistakes are corrected (at least the ERROR), do the same step as before and then finally create the \*.WAM file once more.

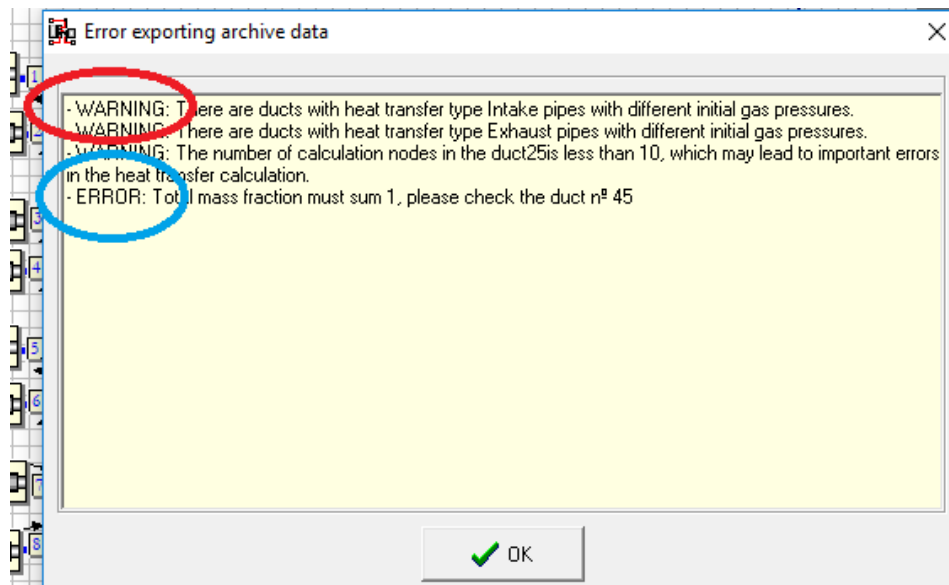


Figure 41. Diverse types of messages caused by mistakes

When the problem of a message has been resolved, WAMer shows a display as the Figure 42. Once one has gotten here, click on the icon **+** (red circle of Figure 42) and introduce the corresponding files (Figure 42).

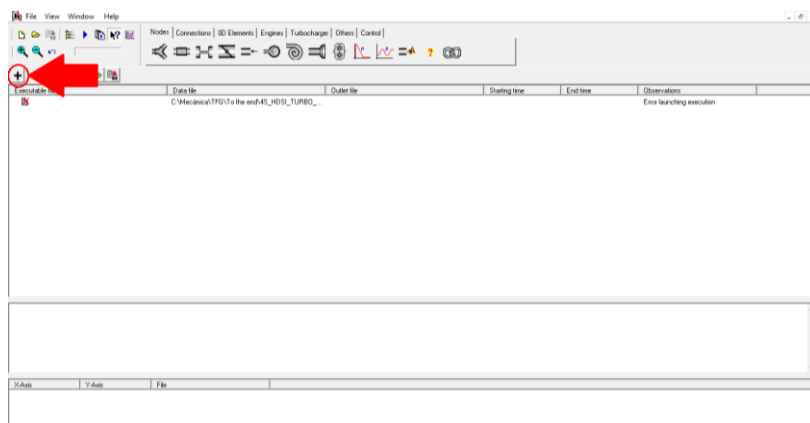


Figure 42. Menu where the simulation is launched

The next menu (Figure 43) is then shown. In EXECUTABLE FILE introduce (this is important and can lead to confusion) the file OpenWAM2.2-win-x64 (or OpenWAM2.2-win-x32 if the operating system runs with 32 bits). In DATA FILE introduce the model in the format \*.WAM that you want to simulate. This determines in DATA FILE the file, which will be simulated, in EXECUTABLE the file that realises the simulation. Afterwards, as a logical consequence, push OK.

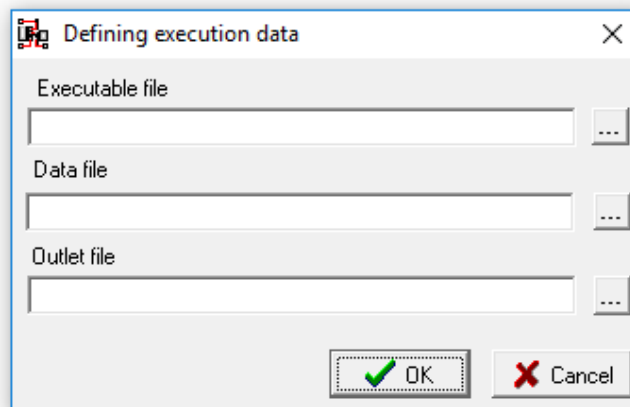


Figura 43. Menu where the file for the simulation is introduced and which realizes the simulation

At that point, WAMer shows (Figure 44) the list of files introduced for the simulation. Click on the file and look for the option START CALCULATION (Figure 45). If it is necessary to stop the realization while the file is running, the process is the same but in the last step, click on STOP CALCULATION.

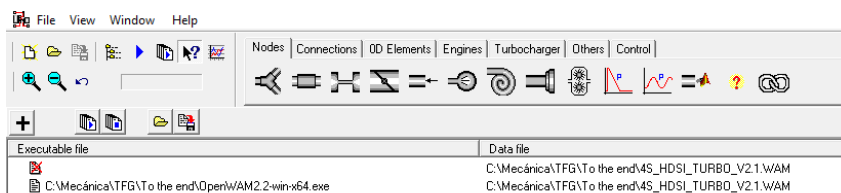


Figure 44. List of file for simulation. It is possible to simulate many files simultaneously

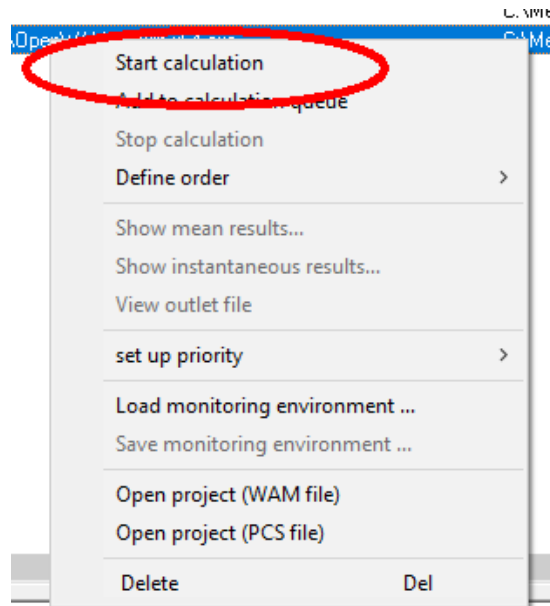


Figure 45. Menu to run or stop the simulation

Once the simulation is running, OpenWAM creates two different files. One for the average (\*.AVG) measures and the other one for the instantaneous (\*.INS) measures. Both files are of the type DAT, but it is possible to open these documents with the programme Excel. These files are saved in the same folder where the data file can be found (Figure 40).

## 6.1 PROCESS FOR WORKING WITH EXCEL

The following will give an explanation on how to adapt a DAT file for a good usage of Excel. This explanation is valid for the version of Office 2013. For other versions it is recommended to look for more information but the principal steps will be almost the same.

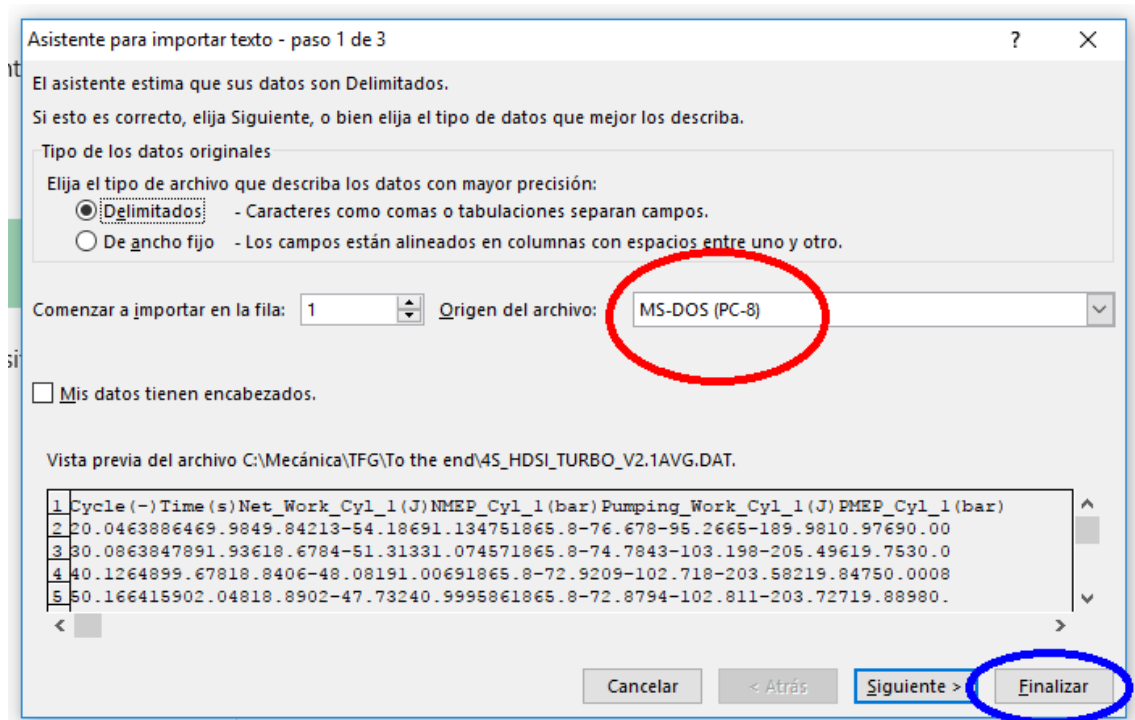


Figure 46. Menu to configurate the file \*.DAT



When the menu of Figure 47 is shown (here you can see it in Spanish but the positions of the options or the names are the same), it is basic to configurate the option in MS-DOS (PC-8) in the origin of the file (red ellipse, Figure 46) and push on FINISH (blue ellipse, Figure 46). After this process, a normal grid of Excel is obtained to analyse or work with these data, like in Figure 47.

	A	B	C	D	E	F	G	H	I	J	K
1	Cycle(-)	Time(s)	Net_Work_Cyl_1(I)	NMEP_Cyl_1(bar)	Pumping_Work_Cyl_1(I)	PMEP_Cyl_1(bar)	Combustion_Heat_Cyl_1(I)	Heat_Through_Liner_Cyl_1(I)	Heat_Through_Cylinder_Head_Cyl_1(I)	Heat_Through_Piston_Cyl_1(I)	IMEP_Cyl_1(bar)
2	2	0.0463886	469.984	994.213	-541.869	113.475	1865.8	-76.678	-952.665	-189.98	109.769
3	3	0.0683847	891.936	186.784	-513.133	107.457	1865.8	-747.843	-103.198	-205.496	19.753
4	4	0.1264	899.678	188.406	-480.819	10.069	1865.8	-729.209	-102.718	-203.582	198.475
5	5	0.166415	902.048	188.902	-477.324	0.999586	1865.8	-728.794	-102.811	-203.727	198.898
6	6	0.206454	903.417	189.188	-476.793	0.998472	1865.8	-72.671	-103.312	-204.709	199.173
7	7	0.246391	899.666	188.403	-481.226	100.776	1865.8	-720.288	-100.35	-198.731	198.481
8	8	0.286335	903.13	189.128	-485.423	101.655	1865.8	-717.268	-100.259	-198.613	199.294
9	9	0.326403	905.478	18.962	-488.666	102.334	1865.8	-713.096	-102.174	-202.436	199.854
10	10	0.366348	902.302	188.955	-490.046	102.623	1865.8	-706.861	-990.497	-196.138	199.217
11	11	0.40641	904.419	189.398	-491.018	102.826	1865.8	-707.335	-101.091	-200.281	199.681
12	12	0.446346	900.817	188.644	-491.766	102.983	1865.8	-703.071	-981.493	-194.359	198.942
13	13	0.486399	903.246	189.153	-491.169	102.858	1865.8	-704.786	-100.417	-198.954	199.438
14	14	0.526448	904.381	18.939	-494.125	103.477	1865.8	-701.901	-100.219	-198.503	199.738
15	15	0.566333	885.011	185.334	-626.093	131.113	1865.8	-716.526	-981.566	-194.367	198.445
16	16	0.606347	893.286	187.067	-549.916	115.116	1865.8	-710.204	-996.287	-197.287	198.583
17	17	0.646333	895.586	187.758	-531.768	111.136	1865.8	-704.018	-990.138	-196.033	198.894
18	18	0.686426	900.114	188.497	-532.223	111.455	1865.8	-703.391	-101.003	-200.026	199.642
19	19	0.726382	896.276	187.693	-536.488	112.348	1865.8	-700.878	-981.329	-194.248	198.928
20	20	0.766444	900.535	188.585	-541.179	113.331	1865.8	-703.441	-100.459	-198.953	199.918
21	21	0.80637	894.749	187.373	-545.981	114.336	1865.8	-700.578	-973.702	-192.719	198.807
22	22	0.8464	897.996	188.053	-550.114	115.202	1865.8	-703.276	-997.244	-197.482	199.573
23	23	0.886415	897.403	187.929	-555.142	116.255	1865.8	-700.256	-99.238	-196.446	199.555
24	24	0.926414	896.783	187.799	-559.397	117.146	1865.8	-70.035	-989.221	-195.817	199.514
25	25	0.966395	895.739	187.581	-564.126	118.137	1865.8	-700.225	-985.469	-195.065	199.394
26	26	1.00636	895.086	187.444	-569.064	119.117	1865.8	-70	-981.887	-194.351	199.361
27	27	1.04643	897.973	188.048	-573.521	120.104	1865.8	-699.727	-100.167	-198.296	200.059
28	28	1.08636	893.57	187.126	-577.572	120.952	1865.8	-697.069	-972.081	-192.329	199.222

Figure 47. Example of data once they are configurated for the right lecture

## CONCLUSION

This Thesis has been a fantastic opportunity to get to know the world of simulation. A contact which was longer than productive but enough to enable a first valuation of the power of this tool.

First of all I would like to say that OpenWAM is really precise in its results, but we can see that it is not the best software for the calculation of emissions, although in the part of fluid mechanics it is very powerful. In fact it is so powerful that it is hard to lay out what is the advantage of buying another simulation software of some other commercial company if it costs a lot of money.

The first answer is the information for a correct use of this tool, but with this “thesis-tutorial” I hope that actually it will not be a problem, but it is true that the information that other types of software provide is more complete than the present text. Another important point can be the friend's usability. The graphics, icons and other points in relation with the interface are best solved in the GT-Power for example. And the last point is the complexity to find all data that the tools need for the correct execution of the simulation. But, if we leave out this point, OpenWAM is a perfect simulation software for its use in a private company, and of course, in the student world. The negative points do exist but with a complete education (which is possible to find with time, with a good relation with the CMT or with both), I am sure that this tool can be a real alternative for the other types of private and commercial simulation software.

## BIBLIOGRAPHY

- x Motores de combustion interna alternativos, Prof. F. Payri, Prof J. M. Desantes
- x [https://en.wikipedia.org/wiki/Heat\\_capacity\\_ratio](https://en.wikipedia.org/wiki/Heat_capacity_ratio)
- x [https://en.wikipedia.org/wiki/Discharge\\_coefficient](https://en.wikipedia.org/wiki/Discharge_coefficient)
- x <http://www.fem.unicamp.br/~phoenics/EM974/PROJETOS/PROJETOS%201%20S EM-10/TURMA%20B/G8/Artigo%20Venturi.pdf>
- x <https://www.soloingenieria.net/foros/viewtopic.php?t=1977>
- x Degree Thesis by Hofherr Thomas (Motorprozesssimulation eines direkteinspritzenden 4-Takt Ottomotors)
- x Degree Thesis by Braun Tobias (Erstellung eines Motorsimulationsmodells anhand eines direkteinspritzenden und aufgeladenen 4-Takt-Ottomotors in GT-POWER)
- X Degree Thesis by José Antón Ruiz (Análisis de un motor de combustion interna alternativo con un sistema de distribución innovador mediante modelado 1D)

## ESTIMATE

As almost all project, it is advisable to calculate a valuation of the money spend in it. In this case, we can suppose that we work for a software company and this company needs to know how much resources the should invest to have a appropriate tutorial in English to say this tools. With this perspective was wrote this part of degree thesis.

We estimate the cost per hour of engineer in 40 €/h [4] and the work hour of a professor, in 60 €/h. This price is esteemed, because we had not to find the precise preis, but if we think in the experience and bigger knowledge of the professor with relation with the normal engineer, can to calculate the preis per hour of this professional a 50 % bigger that the engineer.

	TRANSLATE FRENCH TUTORIA	TO KNOW OpenWAM	TUTORIAL WITH PROFESSOR	TO REDACCION OF DEGREE THESIS
HOURS	10	125	24,5	150
PREIS PER HOUR (€)	40	40	100*	40
TOTAL PREIS FOR SECTION (€)	400	5000	2450	6000

Table 1. Real estimate

**TOTAL.....13850 €**

\* Here we make the summ of the preis for normal engineer (student) and the professor, therefore is 100 €. The summ of 60 € of the professor and the respective 40 € of engineer.

The amount of **13.850 €** may seem a very expensive for a simple tutorial, which not delve into the part of programation of the tools OpenWAM and it is possible reduce this monen without a doubt with other sistem of work, but if it is considered the conditions and means use to know the software and to development of tutorial and the great theoretical information provide fot both professor, we estimate that the appropriate could not to be fewer at 13 %. This estimation is explained in the next table and paragraph.

	TO TRANSLATE FRENCH TUTORIAL	TO KNOW OpenWAM	TUTORIAL WITH PROFESSOR	TO REDACCION OF DEGREE THESIS
HOURS	5	125	21	120
PREIS PER HOUR (€)	40	40	100*	40
TOTAL PREIS FOR SECTION (€)	200	5000	2100	4800

Table 2. Possible estimate

**TOTAL.....12100 €**

This reduccion of hours is because:

- x The translation and layout of the French tutorial is not obligatory and we could use only 5 hours for that.
- x If we started direct with the elaboration of the tutorial, we could save 3.5 hours in the phase of reconsideration of the thesis.
- x And if we have had as first objetive, the redaction of a tutorial, the part of redaction have been writte with other structure, wich have saved a 20 % of time in this phase.

\* Here we make the summ of the preis for normal engineer (student) and the professor, therefore is 100 €. The summ of 60 € of the professor and the respective 40 € of engineer.