

Developing a New Tool to Implement Computer-Supported Active Learning Strategies in the Engineering Classroom

JUAN M. TIZÓN^{*a}, PABLO SIERRA^{†a}, LUIS SÁNCHEZ DE LEÓN^{‡a},
 EMILIO NAVARRO^{§a}, JAVIER VILÁ^{¶b}, AND JOSÉ F. MORAL^{||b}

^aDEPARTAMENTO DE MECÁNICA DE FLUIDOS Y PROPULSIÓN AEROESPAZIAL, ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA AERONÁUTICA Y DEL ESPACIO (ETSIAE), UNIVERSIDAD POLITÉCNICA DE MADRID (UPM), PZA. DEL CARDENAL CISNEROS 3, 28040 MADRID, SPAIN

^bEMPRESARIOS AGRUPADOS INTERNACIONAL, S.A., DIVISIÓN DE SIMULACIÓN Y SISTEMAS DE TRANSPORTE, C/ MAGALLANES 3, 28015 MADRID, SPAIN

Abstract

Successful implementation of active learning strategies in the engineering classroom — and in particular in certain subjects which are highly technological in nature such as, for instance, rocket engines and space propulsion — means overcoming certain challenges that arise from the fact that these are extremely complex systems to analyze. In this paper, we address the specific means to overcome one of such challenges: the lack of readily available software tools that are suitable for implementing this sort of teaching strategies within the engineering training. In particular, we develop a new tool for the modeling and simulation of liquid-propellant rocket engines specially tailored for the classroom, taking a systematic approach to the development of such tool based on the needs of modern teaching practices. After a thorough review of the available literature on the topic, the few most critical features that our tool should have in order to serve its purported goal are identified. Subsequently, a pilot experience to assess the impact of the usage of said tool on the learners' performance was carried out, showcasing excellent results, both in terms of the students' perceived quality of their training as well as in terms of their grade of retention and understanding of the matter. The conclusions of this study, especially the guidelines for the development of software tools aimed at the classroom, nevertheless, should be applicable to any other highly technological discipline, extending the scope of this paper beyond merely the subject of rocket science in engineering.

Keywords: Aerospace engineering; Educational software; Active learning; Simulation; Systems thinking.

^{*}E-mail: jm.tizon@upm.es—ORCID iD: <https://orcid.org/0000-0002-8687-6657>

[†]E-mail: pablo.sierra.heras@alumnos.upm.es—ORCID iD: <https://orcid.org/0000-0001-7876-2724>

[‡]Address all correspondence to this author: Luis Sánchez de León Peque, Departamento de Mecánica de Fluidos y Propulsión Aeroespacial, Escuela Técnica Superior de Ingeniería Aeronáutica y del Espacio, Universidad Politécnica de Madrid, Pza. del Cardenal Cisneros 3, 28040—Madrid, Spain. E-mail: luis.sanchezdeleon@upm.es—ORCID iD: <https://orcid.org/0000-0001-6815-3263>

[§]E-mail: emilio.navarro@upm.es

[¶]E-mail: jvv@empre.es

^{||}E-mail: frj@empre.es

Abbreviations

- DAE** differential algebraic equation.
ECLSS environmental control and life support system.
EL EcosimPro Language.
ESA European Space Agency.
ESPSS European Space Propulsion System Simulation.
JiTT just-in-time teaching.
LPRES Liquid-Propellant Rocket Engine Simulation.
LPRM liquid-propellant rocket motor.
ODE ordinary differential equation.
PBL problem-based learning.
SSME Space Shuttle main engine.
UPM Universidad Politécnica de Madrid.

1 Introduction

In the era of artificial intelligence and technological revolution that we are living today, higher education must evolve and adapt, preparing our graduate students to meet the demands of the working environment of tomorrow. As envisioned by Aoun [1], the workers of tomorrow will compete in a much more flexible and demanding market, where jobs that are suitable to be automated will be done by robots, while humans will have to develop their creativity in order to remain competitive. In order for the higher education of today to be able to train the workers of tomorrow, the whole vision of the higher education itself must move towards new paradigms: training the students for a continuous, life-long, self-guided process of learning, to continually adapt and meet the ever changing needs of the market.

As such, and in particular within the engineering field, it will not suffice for the engineer of tomorrow to be an expert on a certain field or a specific part of a device, but rather a much more holistic view of the whole system will be needed. The fundamental understanding of the interactions between several disciplines, the merging between the technical knowledge of a particular device's functioning, data-analysis capabilities and technological —i.e. computer software development— expertise is what big companies in the engineering sector are after, even today [1, p. 36].

According to Aoun [1, p. xix], the workers of tomorrow will need to develop new competencies, among which *systems thinking* will be fundamental; and acquire new abilities, such as *data literacy* and *technological literacy*. For this purpose, Aoun defends the concepts of *experiential learning* and *cooperative education*, in which both the Industry as well as the University share the responsibility for the learners' training. However, the precise tools and means to achieve this sort of education, in a field such as engineering, is not irrevocably established yet. Among the many changes in teaching practices and the new paradigms in learning strategies that we have seen arising in the last decades, two aspects are common to the vast majority of them: *a)* the use of technology and computers as a *cognitive tool* [as defined by 2]; and *b)* the drive towards a more 'active' role from the students in the classroom.

The challenges for implementing this sort of concepts in teaching practices within the engineering field, and in particular within the sector of rocket engines and space propulsion that we are dealing with, respond to some of its technical particularities: for starters, it is a field in which the complex interactions between the many components that form a single system (a rocket engine, in our case) cannot be understood without a profound knowledge of the 'basic science' that underlies beneath, so that active learning is not an attainable paradigm without at least some time devoted to the traditional approach of the student absorbing knowledge from the lecturer in a more 'passive' role; furthermore, the computational tools used in this sector for the simulation of rocket engines either

fall within the category of in-house code developed by the manufacturers, and are consequently inaccessible for the students, or are just generic-purpose tools with such a complex modeling that require experts on that particular tool to be trained just to use them —this is the case, for instance, of the European Space Propulsion System Simulation (ESPSS) libraries developed for EcosimPro®, that will be referred to in Section 3—, which in turn would require a vast amount of time to be spent in such a training, hardly affordable within the time frame of a semester-long course.

In consequence, active learning and, in particular, inquiry-based methods have been used much less extensively in engineering than in many other sciences [3]. In view of the apparent lack of readily available software tools actually useful for the purposes of implementing active learning and problem-based learning (PBL) paradigms in the engineering classroom —more specifically, in teaching the subject of rocket engines—, the objectives of this project are: *a*) to identify the major shortfalls of currently used software tools in the field for this purported goal; *b*) to systematically devise the features that a new tool should have in order to fulfill its expected usage; and *c*) to assess the impact of the usage of said developed tool on the learners' performance. For this purpose, after the tool was developed, a pilot experience was carried out in the context of the Space Systems MSc program at the Universidad Politécnica de Madrid (UPM).

2 Motivation

The courses within any engineering MSc program are highly technological in nature. Basic knowledge of ‘science’ has already been imparted in previous years and should be applied to solving problems in the real world. The trouble with such problems is often associated with two aspects: the degree of realism for describing the different elements, and the complexity —understood as the presence of many interrelated components— of the systems to analyze.

From the old paradigm of ‘passive’ learning, in which students merely ‘absorb’ knowledge from a lecture, a textbook, etc., teaching of such courses nowadays tries to place the students in a more active role. In the particular subject of *Rocket Engines and Space Propulsion* that we are dealing with, nevertheless, there are several issues that somewhat limit the extent to which an active learning paradigm can be adopted. On the one hand, practical experience, hands-on, with an actual rocket engine in a laboratory is unaffordable, both in terms of cost and in terms of safety; on the other, classroom ‘activities’, such as problem-solving carried out with “pen and paper”, tend to be oversimplified (so that the exercises can actually be tackled “by hand”), which triggers in the learners the perception of ‘disconnection’ from reality: what they study feels meaningless, too simplistic and completely unrelated to the amazing complexity of the actual systems.

Computers, as *cognitive tools*, therefore, stand as an interesting alternative. It must be noted, though, the key difference between ‘learning with’ computers and ‘learning from’ computers [2, 4]. In ‘learning from’, the student assumes a passive role, and uses the computer merely as a source of information (as they could get the information from listening to a lecture or reading a textbook); while in ‘learning with’ computers, the emphasis is placed upon the active role of the learner: they must *do* rather than *watch*. In this regard, merely engaging the students in a few guided simulations will just not be enough: in the *guided simulations* approach, the student merely uses a software tool to produce some data and *watches* (following some guided steps from the instructor so that a certain software plots some results in the screen is not much different from just *watching* the instructor plot those same results in a projector for the whole classroom). And hence there is a close link, and some mutual necessity, between using computers as a cognitive tool and implementing active inquiry-driven, self-guided learning; between ‘learning with’ computers and PBL, in which the learner *does* actively engage in solving real-world, “ill-structured” problems.

According to [4], software tools can be used to alleviate the ‘cognitive burden’ of solving huge systems of equations, so that the student focuses on gathering data and analyzing the results rather than on the mathematical apparatus needed to solve such problems; and thus to encourage scientific

inquiry and self-guided learning. Although not every author agrees with the inquiry-driven learning paradigm [5], we share the view of many [6, 4] who consider *inquiry* not as a mere *tool* to learn something else, but as a scientific *goal* in itself, worthwhile teaching in its own right.

Active learning strategies in engineering have been thoroughly discussed, among others, by Prince et al. in [7, 3]. In the *Rocket Engines and Space Propulsion* subject, there is undoubtedly a need for at least some theoretical lectures, so that the fundamentals of these complex systems are explained, and hence pure *discovery learning* without guidance from the instructor is not recommended [3]. Still, inquiry-driven learning shows generally good results [7], and PBL, at the very least, encourages better attitudes from the students [7].

In other scientific areas, highly dependent on data gathering to infer theories or draw conclusions, computer modeling has already proved itself as a valuable tool for increasing the students' understanding of the matter [e.g. 8]. Even from the mere fact of engaging students in modeling activities, other, more general, cognitive benefits —such as the ability to scan data for patterns or make evidence-based predictions— that are intrinsic to science and not specific of a particular field, are acquired by the students [8, 9, 10].

In teaching physics, for instance, the usage of such modeling tools is widespread. We take from [11] the subtle difference between *modeling* and *simulating*: in the former, the learner can modify the *model* itself, according to the variables or effects that they think are relevant for the studied phenomena; while in the latter the learner can only modify the *value* of the input variables, but is unable to define or change the *relations* between those variables. This is a fundamental concept that has guided the development of our tool. In [12], the benefits of including computer modeling activities as a means to enhance the just-in-time teaching (JiTT) methodology is reported. Not only do the students get a more active role in the classroom by means of trying by themselves to answer the questions that the instructor poses in a mostly *guided-inquiry* teaching process, but also the professor can identify the common errors and misconceptions that the learners may have, and tailor subsequent modeling activities to address these misconceptions in a timely manner. In fact, as a well-known YouTuber [13, 14] proposes, and the 5E instructional model supports [15], addressing the students' *misconceptions* about a topic is paramount in teaching science. In this regard, computational *modeling* tools, as the vehicle for PBL in engineering, enables the instructor to tailor the problems and questions posed to the students so that they can find, and explain, their own misconceptions.

For the reasons stated above, we have developed a new tool, and set the methodologies that will enable us, to conduct a pilot experience on implementing a computer-supported more active teaching of the subject of rocket engines to the students of the Space Systems MSc program taught at the UPM, in which the aim is to evaluate the impact —both as the students' *subjective* perception of the knowledge they acquire as well as by *objectively* contrasting their grade of retention conducting tests to measure their performance— of scientific inquiry-driven learning processes on the academic success of the learners.

The contribution of this work is, thus, on the one hand, a computer tool that addresses the lacks of currently available software in the field of rocket engines modeling and simulation specifically targeting the engineering classroom; and, on the other, guidelines for the development of such educational software. A computer tool, developed with almost the sole goal of facilitating the implementation of active learning strategies in teaching rocket engines to engineering students. And guidelines, provided as a list of ‘desirable features’ —that remain valid in principle irrelevantly of the field, not only within the subject of rocket engines— that these tools should have, based on the available literature and upon our own developing and teaching experience, in order to be suitable to realize the full potential of active learning *with* technology.

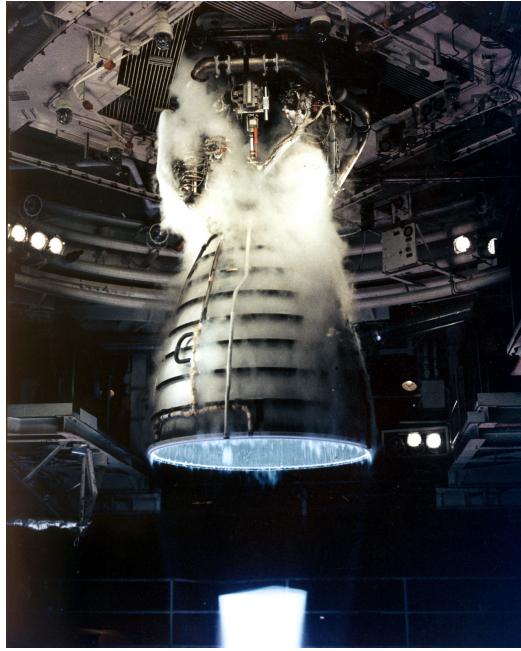


Figure 1: SSME hot-firing test

3 Tool Development

Prior to the beginning of the tool development itself, it is paramount to systematically establish concisely *what* is it that we want to develop. From the literature review in Section 2, and a closer look into the readily available tools in the sector of rocket engines modeling, key features to have implemented in our tool as well as the general “coding philosophy” to abide by in the development process must be precisely defined.

Liquid-propellant rocket motors (LPRMs) are efficient systems for aerospace propulsion and, as such, their study constitutes a substantial part of the academic curriculum of any MSc program taught within the aerospace sector. LPRMs can be configured in many different ways, depending on the application; but, in any case, their performance is the result of the collective work of a significant number of specialized elements (pumps, turbines, burners, valves, etc.) organized by hydraulic networks arranged in a complex architecture with a large number of forks and junctions. Moreover, the fluids have properties that depend on the temperature conditions, they may undergo phase changes and be composed of mixtures originating in combustion processes. Indeed, the whole system is meant to feed the stored propellant into a combustion chamber and nozzle. An example of such a LPRM can be found in the Space Shuttle main engine (SSME), shown in figure 1. The SSME delivers up to 2.3 million Newton in thrust, and was used to propel the US Space Shuttle. The propellants were cryogenic liquid hydrogen and liquid oxygen, following a complex cycle (see figure 2) with a significant number of elements: five turbopumps, driven by one liquid turbine plus three gas turbines (two of which run on hot gases from the pre-burners, and hence need to be cooled), two heat exchangers, three combustion chambers, five hydraulically-actuated valves to control engine’s output, etc. Thus it is made clear the enormous complexity of the system under study and the challenges and difficulties that trying to analyze the performance of such engine “by hand” would pose.

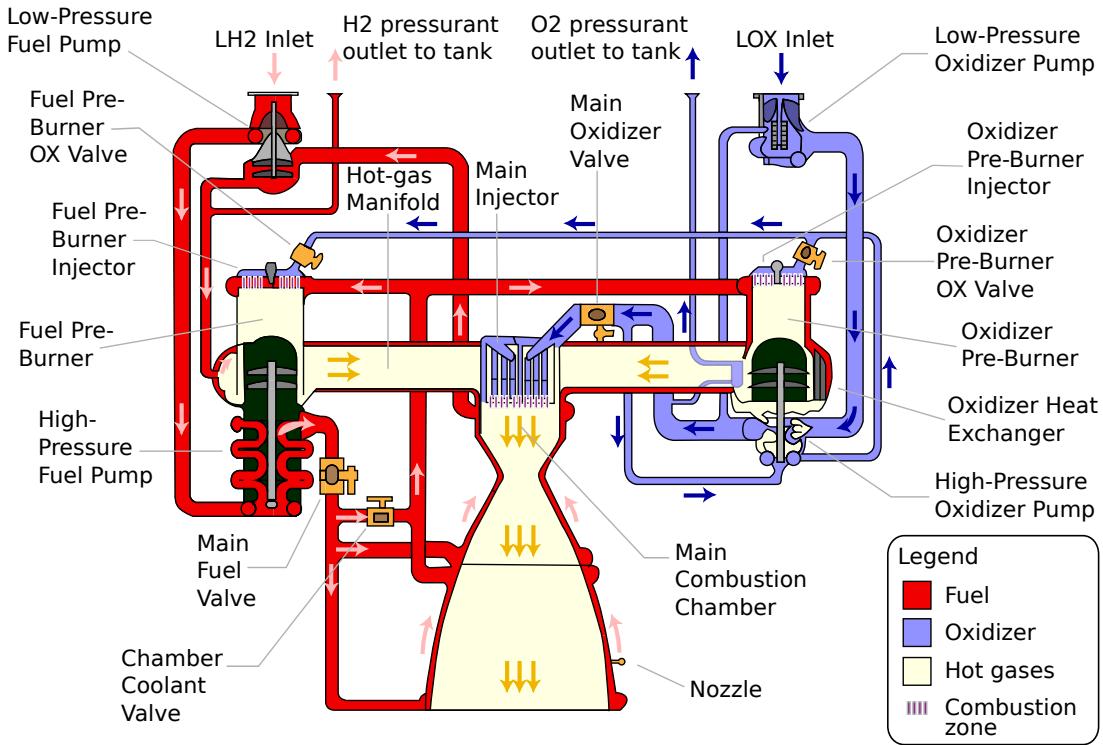


Figure 2: SSME schematic diagram

From the teaching point of view, the practical activities carried out in the traditional classroom related to these problems often consist of simple exercises that can be solved with “pen and paper”. The intention is to apply the knowledge and skills acquired by the learners to solve a real case, but the simplicity of the tools undermines the purported goal, and the student, because of the contrast with the reality facing him, may have the perception that what they are doing is trivial, oversimplified and non-worthwhile for their training. The present project aims to address this issue by making use of computer tools specifically tailored for the classroom environment—i.e. with a flat learning curve, easy to grasp its fundamentals in a short period of time, and implementing models and concepts similar to the ones the student would use in solving the classroom problems “by hand”. There are two types of benefits to be had with this approach: one is to improve the perception that the student has on what they do in the classroom, bringing their activity closer in line to their idea of what they will do in the real working world, which creates a positive predisposition to assimilate the concepts that these simple models can help to make clearer; the other is to allow the learner to take on more complex problems that could not be done otherwise “by hand”, avoiding oversimplifications, which helps them to better judge and understand the behavior and performances of the systems simulated.

Implementing any sort of computer-supported active learning strategy in teaching LPRMs, as discussed in Section 2, means overcoming a few challenges, the first of which is to find the right software tool for *modeling* (and not merely *simulating*) these systems. As mentioned, most of the industry use their own in-house tools, inaccessible for the students, and specifically tailored for that particular company’s very own products, but unable to tackle ‘generic modeling’ problems in a flexible way. A few instances of different tools created for the simulation of LPRMs have been reported in the open literature, but these are mostly *ad hoc* models, created specifically for the

study of a particular effect in a very particular engine configuration. Among the initiatives to create flexible tools for the modeling of LPRMs, the LiRA project [16] and REDTOP-2 [17] stand out. However, in spite of the efforts made, it has to be remarked that both these tools are intended for *simulation* of some configurations only, rather than allowing for actual *modeling* of complex systems: both tools include a relatively extensive set of pre-defined configuration options through which different LPRM power plants can be analyzed, but constructing your own system, outside of those pre-defined configurations, is not allowed [a more in-depth literature review on the different tools available can be found in 18].

The few generic-purpose modeling tools commercially available, among which probably Simulink[®] and EcosimPro[®] are the most well-known, are often times used alongside their own in-house codes by companies in the aerospace sector and are, therefore, an option to consider. From the two, Simulink[®] is more of a model-anything tool, whilst EcosimPro[®] is the tool specifically tailored for the modeling and simulation of LPRMs by means of the ESPSS libraries—particularly, the European Space Agency (ESA) has chosen EcosimPro[®] as its recommended tool for simulation in several fields, including space propulsion, environmental control and life support systems (ECLSSs) and energy systems [see, for instance, 19, 20]—and hence is the most widely used within the industry, so that it would seem more natural to teach the students of the Space Systems MSc program the tools that are specific to their sector, namely EcosimPro[®]. Nevertheless, the ESPSS libraries for EcosimPro[®] are a very complex set of models, which require specialized training to be able to use them, and have a steep learning curve that makes them not suitable at all for a classroom environment.

For these reasons, we have decided to develop a new tool, called Liquid-Propellant Rocket Engine Simulation (LPRES), a simplified library for EcosimPro[®] presented in [18]. The tool is, therefore, a simplified version of the ESPSS professional libraries, developed mainly for educational purposes. The approach of implementing a ‘simplified’ version of a more complex software tool for the classroom is not new: there are examples of similar strategies in other fields where accurately modeling the phenomena involved can result in highly demanding software tools [21]. But in our case, it has the additional benefit of providing a platform to get the students of the Space Systems MSc program to familiarize themselves with an environment —i.e., EcosimPro[®]— widely used in companies within the aerospace sector, which will likely prove to be a valuable *technological literacy* [1] for the learners to acquire during their training, at the same time that gives them the feeling of being ‘connected’ to current standard practices in the industry, so that they perceive what they are being taught as useful and meaningful for their future professional careers. Furthermore, the intuitive “drag and drop” interface used by EcosimPro[®], along with its powerful symbolic and numeric solver, removes the burden from the student of having to deal with complex input configuration files and/or programming sophisticated methods for reaching numerical convergence of said models, so that the emphasis can be placed upon the *modeling* task itself; much in line with the idea of creating a software tool that alleviates the ‘cognitive burden’ from the students and lets them focus on the data gathering and interpretation processes [4].

Moreover, developing such a tool as LPRES with an educational mindset has other benefits to highlight. For starters, making the code available for the students (rather than having a palette of blocks functioning as ‘black-boxes’ to choose from when building up the model) enables the learners to actually understand the relations and phenomena that govern the performance of the LPRM. The capability to choose from several degrees of pre-implemented model fidelity, or even to modify the code by themselves to account for new effects that they may consider relevant, allows the students to understand the *limitations* and *applicability* of the equations and models that they are dealing with, a fundamental goal of teaching *modeling* according to [9, 10]. At the same time, it provides a meaningful platform in which to compare the results of the more simplified equations against the greater accuracy obtained by means of more complex modeling, showcasing the great value of being able to make quick *estimates* about a particular system, a valuable ability for the engineer to have from the design point of view according to Dym et al. [22]. Furthermore, the possibility to re-

arrange the hydraulic network of components of a particular LPRM, or to even create their very own system from scratch exploring novel configurations, provides the learners with the *systems thinking* holistic view and the understanding of *system dynamics* of the LPRMs that an engineer must have [22, 1]. Last, but not least, the way the EcosimPro® modeling environment works, by means of setting up (*virtual*) experiments, teaches the students how to *design experiments* [22] so that they provide meaningful data, and teaches them how to interpret the results and infer conclusions from gathered data, enhancing their *data literacy* [1].

A relevant note on the benefits of developing our own tool must also be made herein. One of the shortfalls of the JiTT methodology is that requires extremely competent instructors, who are experts on the topics covered so that they can tailor the contents taught to the needs of the classroom *on the fly* [3]. Much the same applies when the JiTT is made use of within the context of computational modeling, as in [12]. In this regard, having our own tool ensures that we are experts on its usage and know its inner functioning in and out, which has allowed us to be effective in tailoring the modeling activities to address learners' misconceptions by guiding the inquiry-driven learning process. Making use of other available tools for which the instructors do not possess this kind of deep knowledge poses a risk to the effectiveness of such concepts as the JiTT and PBL, since the instructor may not be able to know if (or how) a certain model to address a specific topic could be carried out with such a tool.

Lastly, it is worthwhile mentioning that developing LPRES not as a stand-alone tool, but rather as a library of components for EcosimPro®, and putting the development of the code in hands of the students rather than being done by the professors, has facilitated over the past couple of years the *cooperative education* concept by Aoun [1], in which the university and the industry both cooperate and are responsible for the student's training. Empresarios Agrupados, the company in charge of developing EcosimPro® embraced this project as theirs, and provided the opportunity for the students working on the LPRES library to be immersed in the professional world, by means of semester-long remunerated internships, so that the students could develop the library within an actual professional team of developers and had the chance to actually see, and get familiar with, the working environment by themselves, which will hopefully help them in making the best decisions and drive their future careers towards their goals with a more educated view on the topic.

3.1 The EcosimPro® Computer Program

EcosimPro® is a powerful simulation environment capable of modeling any type of dynamic system represented by differential algebraic equations (DAEs), ordinary differential equations (ODEs) and/or discrete events. It is based on numerical and symbolic methods that can process complex systems of algebraic and differential equations. It is endowed with an intelligent system for problem analysis that lets the user pose tough problems and adopt optimal solution strategies. EcosimPro® has a very complete and intuitive graphical user interface where all the work is carried out, from building the model by dragging and dropping component icons from several palettes within pre-implemented component libraries into a schematic window, to programming the behavior of any element by means of its very own non-causal language, called EcosimPro Language (EL).

EcosimPro® is used in companies in the energy and aerospace sectors, and particularly the ESA has chosen EcosimPro® as its recommended tool for simulation in several fields [see 19], including space propulsion [see 20], ECLSSs and energy systems.

3.2 ESPSS Professional Libraries

The purpose of the ESPSS libraries is to provide a standard set of components and functions to simulate jet propulsion systems and space propulsion systems. The ESPSS also provides a standard database for propellants, pressurization substances and other fluids. The ESPSS libraries solve the continuity, momentum and energy equations in a discretized mesh that can vary depending

on the component, but that in general terms can be understood most of the times as a 1D-CFD. It is capable of handling multi-phase flow, it retains all the transient terms in the equations, and certainly the working fluids are treated as real gases/liquids, with properties depending both on the temperature and pressure conditions. Needless to say, the associated cost with this fine level of simulation detail is the vast amount of information that the user must provide as an input: the full geometry of the whole system must be known beforehand, so that volumes in the fluid network can fill up or empty out; construction materials for pipes or tanks must be specified, so that heat transfer calculations and elasticity/elongation of certain elements can be accounted for; etc. This sort of approach, useful in the industry, is not appropriate for a classroom environment, where the students might feel overwhelmed by the sheer amount of knowledge about a particular system that is required from the user in order to be able to build up a single model.

Figure 3 shows a typical schematic created with ESPSS for simulating an engine. In this case it is a closed-cycle turbo-powered bi-propellant rocket motor. More specifically, it is a staged-combustion cycle. The program can carry out direct calculations of the steady-state equations for an engine [see 23], perform transient analyses regarding the startup, thrust control and shutdown of the system [see 24], and even be programmed to do optimization tasks [see 25]. This type of work is customary in designing and analyzing these complex engineering systems.

Companies usually have their own software for analyzing their products, and modify it depending on the changes they make to their components, the innovations they add to their calculation methods and the types of simulation they end up doing on a project. Some of the advantages of the ESPSS libraries over the in-house software specific to a product include the ease for developing new models (or improving existing ones), the flexibility for adapting the same tool to a large number of different configurations simply by using standard components from a palette, the evolution and improvement of these libraries thanks to projects carried out every year by the industry, and the ease for exchanging models between companies (or, for that matter, between centers within the same company) by having a common tool.

3.3 The LPRES Educational Library

LPRES is conceived to imitate the capabilities of the professional ESPSS libraries. As the ESPSS, the LPRES library provides a set of standard re-usable components, with which the user can model virtually any imaginable jet propulsion or space propulsion system (ranging from aeroengines to turbocharged LPRMs) by mere “drag and drop”, or upon which the user may further expand its capabilities by implementing models for new components or refining the modeling of currently existing ones. The LPRES library has been developed to make use of the powerful solver of EcosimPro®, which enables the learner to focus on the *modeling* aspect, leaving the numerical hassle to be dealt with by the program.

It must be remarked herein that the intention is *not* to code an exact replica of (nor any sort of replacement for) the ESPSS libraries, but rather something else. The main goal is the same: to model and simulate jet propulsion systems and space propulsion systems. But the means are substantially different, given that the LPRES library is conceived for the classroom environment, with the emphasis placed upon the educational usage of the tool, rather than on the accuracy of the numerical results. In this regard, one of the main objectives is to reduce as much as possible the amount of information that the user must input into the models, so that the modeling activity seems affordable for the students at the same time that makes up for a much smoother learning curve. The characterization of the components is thus reduced —by means of simplified formulation based on first principles only, very similar to the formulation that one would use when solving problems with “pen and paper” (neglecting effects that can be considered of second order, and focusing only on the steady-state operation of the components, for instance)—, most of the times, to a mere characteristic length (or radius) and/or a free-flow cross-sectional area. The need for a discretized mesh is therefore eliminated, and the formulation is reduced from the 1D-CFD of ESPSS

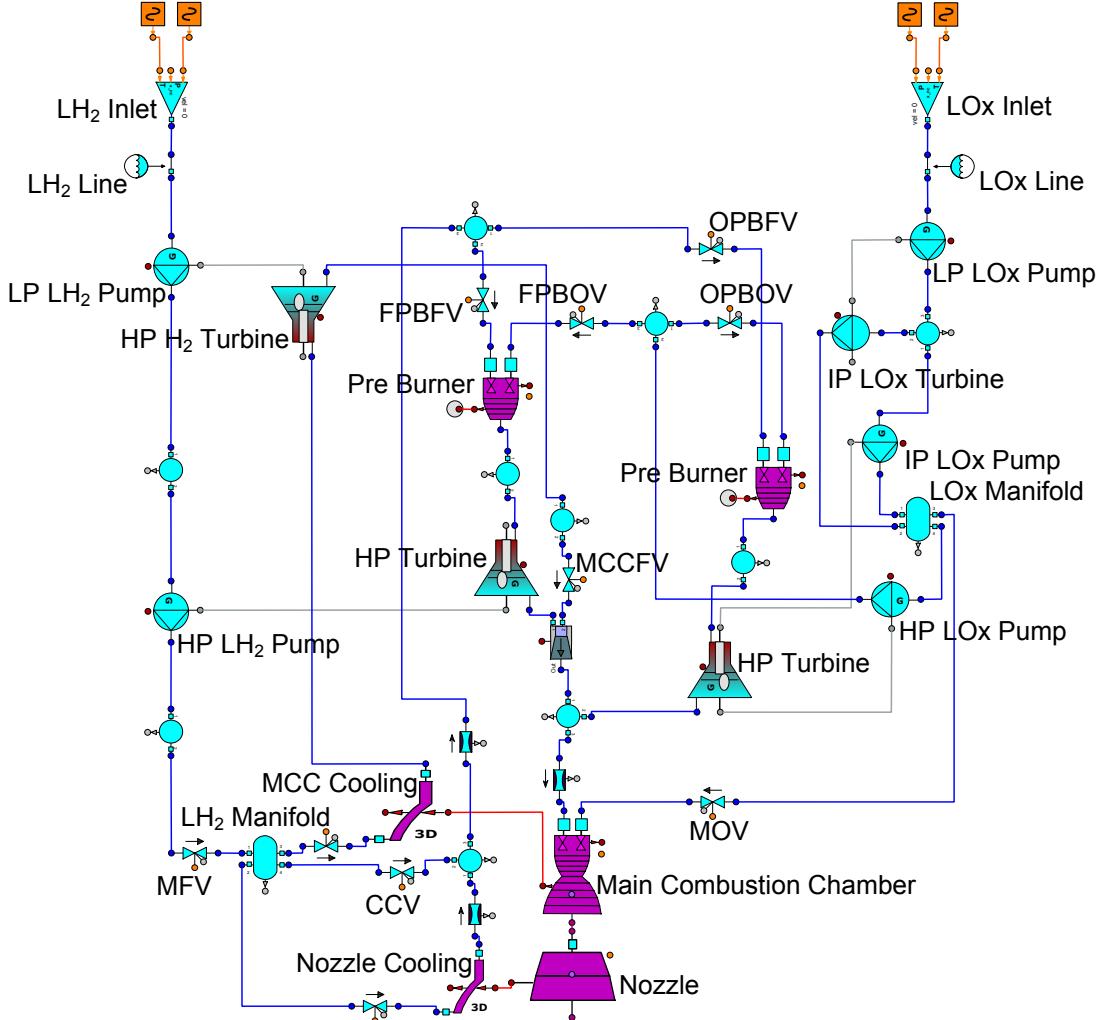


Figure 3: Example of LPRM modeled with the ESPSS libraries.

to just a fluid network of 0D components in LPRES. Furthermore, as the intention is just to obtain “qualitatively good results”, useful maybe for the very preliminary design phase of a LPRM cycle, but not 100% accurate numerical values, the formulation of the components is simplified even more by considering ideal liquids and ideal gases only, with ideal combustion products, so that complex calculations on the chemical equilibrium of the substances involved are avoided altogether.

To make up for these strong simplifications, the components include a number (this can vary from every component to one another) of degrees of freedom, mostly in the form of *efficiencies*, that allow the user to fine-tune a model to replicate the performance of any known system. Hence, the modeling process comprises two steps: firstly, the user *aligns* the model with the actual performances of the system they want to replicate, in what is called the *design* step; then, they can proceed forward to simulate that system at any *off-design* operating condition that they may so desire. The *design* step serves both to determine the *efficiencies* that would make a particular model with the simplified formulation perform closely enough to the available real data, and to perform the basic *sizing* of the geometry (this is, to calculate the characteristic lengths, radii and free-flow cross-sectional areas

of the different components) so that *off-design* calculations (for which both the efficiencies and the geometry are kept constant) can be tackled subsequently. More details about the formulation used within each component can be found in Sierra et al. [18].

With this approach, the two main challenges stated in Section 2 have been successfully addressed: on the one hand, the expected degree of realism for describing the different elements is achieved by means of the aforementioned efficiency parameters, that enable the simplified components to be fine-tuned to replicate the behavior of a real system; and, on the other, the complexity—understood as the presence of many interrelated components—of the systems to analyze is dealt with by the powerful symbolic-numerical solver of EcosimPro®.

Figure 4 shows an example of modeling with LPRES in which a closed-cycle bi-propellant rocket engine has been modeled. Specifically, it is an expander cycle belonging to rocket engine RL10A-3-3A [26]. The propellants are stored in two tanks: one for the oxidant and the other for the fuel. Two pumps are in charge of separately pressurizing the propellants. Then, the fuel goes through some channels used for cooling the nozzle. While circulating through the cooling system, the fuel heats up, thereby increasing its enthalpy enough to move a turbine located at the outlet of the nozzle cooling system. This turbine is the one in charge of powering the pumps. After passing through the turbine, the fuel is injected into the combustion chamber along with the oxidant. Finally, there are several metering elements to calculate characteristics achieved by the whole assembly, such as the thrust and the specific impulse.

Figure 4 illustrates how intuitive the modeling of complex systems becomes with LPRES for the learners, so that introducing modifications into the model, or building up new models from scratch, does not pose a problem for implementing PBL strategies in the classroom. It also comes to show the similarities between the usage of the ESPSS professional libraries in figure 3 and the LPRES educational libraries in figure 4, both having a very similar appearance, so that the students do get the feeling that they are working with a professional tool, part of the standard practices within the industry today. The relevant documentation produced for the LPRES library does also follow current industry standards, much in the line of the ESPSS manuals, to reinforce this perception.

3.3.1 Components of the LPRES Library

The LPRES library consists of several models representing the basic components making up a LPRM. However, unlike the professional ESPSS library, since the LPRES library uses simplified models taught in the classroom, it is steady-state only, which means all its components have null derivatives with respect of time in their equations.

Figure 5 shows the components that the LPRES library makes available for the user. Ports are essential elements, since they are in charge of connecting any component to one another, by means of the right interface that specifies which variables are passed on. The library relies heavily on having the right definition for the connection ports to allow for versatile and efficient modeling. Different types of tanks and the components needed to perform the fluid dynamic connections between elements are made available, too. Components that represent turbomachinery as well as those from the mechanical union between turbines, compressors and pumps are included within this library. The remaining components shown represent combustion chambers, nozzles (convergent and convergent-divergent), nozzle cooling circuits and injectors, as well as the measuring components used to predict how the rocket engine will act as a whole. They can all be used to create models for more complex systems by means of simple “drag and drop” actions from the palette of re-usable components.

3.3.2 Examples with the LPRES Library

Several examples were built with LPRES in EcosimPro®, mainly to test the robustness of the library, and are now bundled within the same package. Making sure that the code is robust enough (so

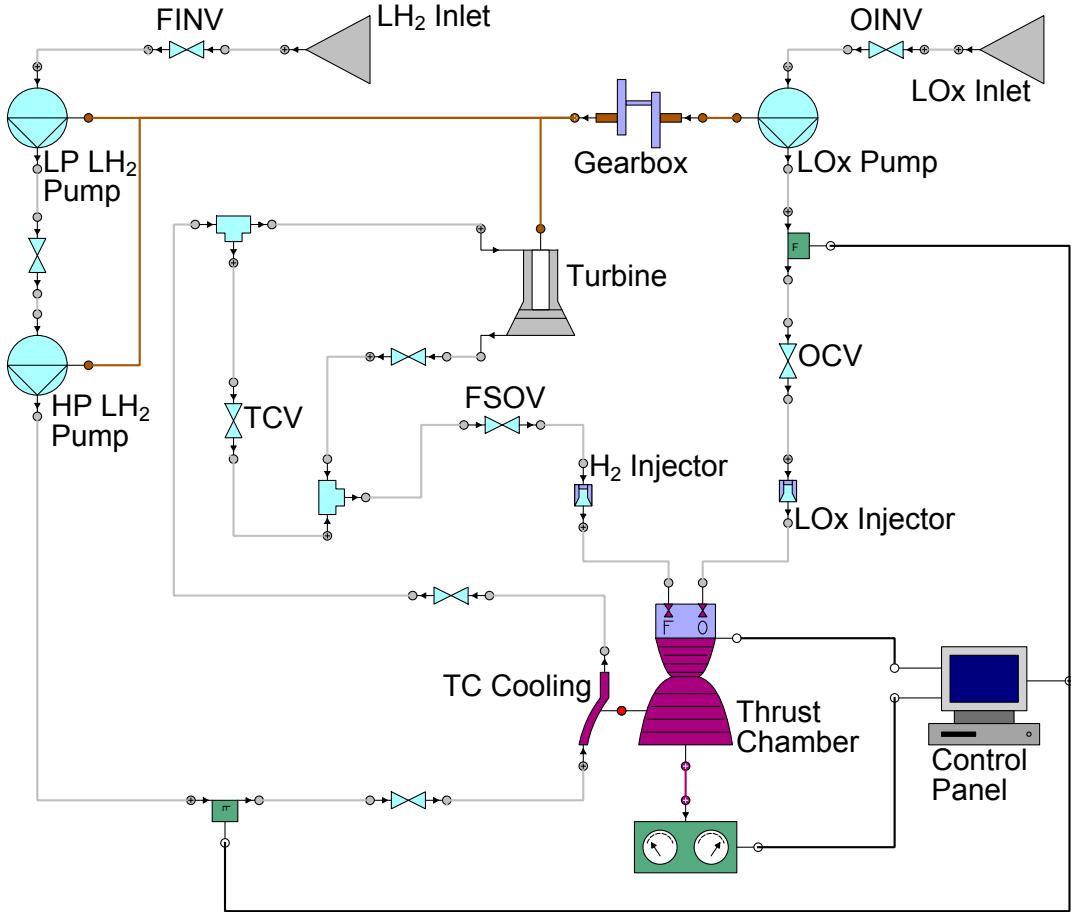


Figure 4: Example of a model with the LPRES library. Simulation of the RL10A-3-3A rocket engine [26].

that modeling novel configurations, and getting meaningful results from these, without having issues with the numerical aspects of solving complex systems of equations) is paramount in facilitating the learners' autonomous usage of the tool. However, this is not their only purpose, since these examples also help to ease the user's learning process by a great deal, given that they are all readily available for the students and are fully working and error-free models that can be used to showcase the capabilities of the library.

There are two types of models within the package: what we have called the (simpler) “test cases” and (more complex) “whole-engine examples”, all of them inspired (or closely resembling) actual “real-life” rocket engines. We take from Roache [27] the succinct definition of *verification* as “solving the equations right” and *validation* as “solving the right equations”. Simple test cases (modeling, for the most part, sub-systems of actual LPRM cycles) were used to verify the components. Furthermore, as these test cases contain just a few components within each model, they can also prove highly useful to the learner in understanding the modeling equations and behavior of some of these components when they are isolated from everything else (or just connected to very few other elements in simple fluid networks). Whole-engine examples, on the other hand, feature much more complex network arrangements and have many interrelated components within each

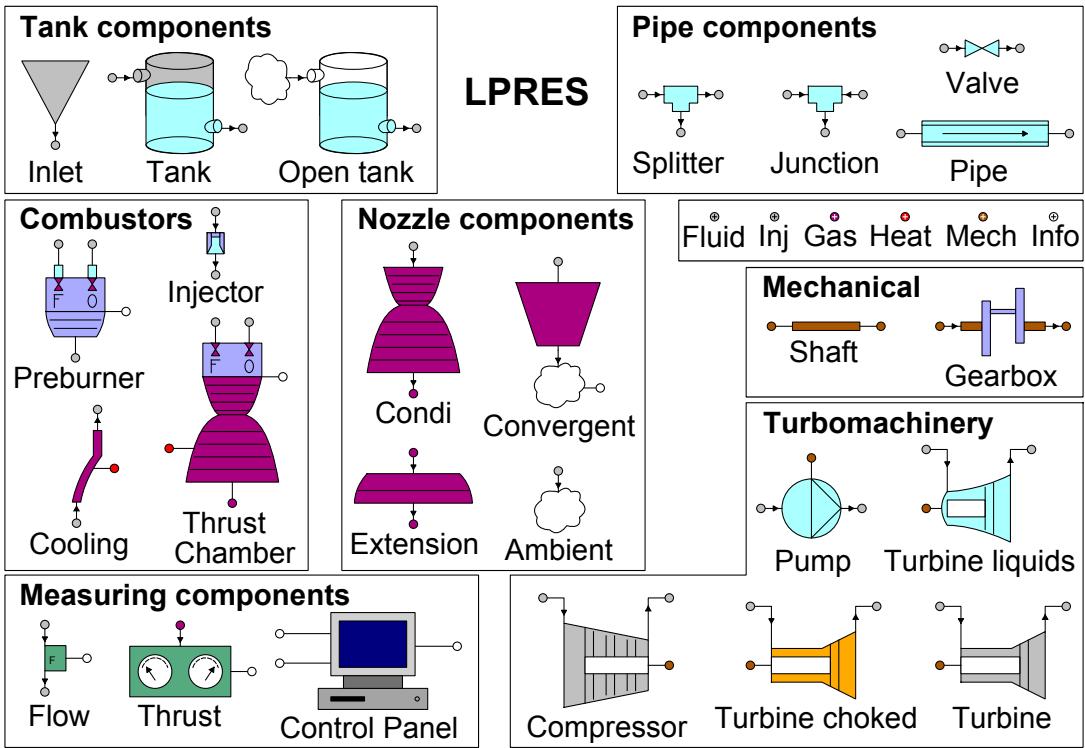


Figure 5: Available components within the LPRES educational library.

model, and hence were used to validate the library. These whole-engine examples comprise some of the most representative cycles that can be modeled with the LPRES library: a gas generator cycle, an expander cycle (such as the one shown in figure 4), a pressurized rocket engine, and even a jet engine, are all included within. Moreover, some of these whole-engine examples were used to validate the library directly, against real publicly-available data, or indirectly, against results from other computational tools that had been previously validated themselves (such as the ESPSS). As a matter of fact, the expander cycle shown in figure 4 was one of such whole-engine models used for validation (in this case, indirect validation against data from simulations with the ESPSS professional libraries). The most representative results from this validation case, which also serves to illustrate how the ESPSS professional libraries and the LPRES educational tool compare with each other, are summarized in table 1.

The data shown in table 1 correspond to the *design* calculations for the RL10A-3-3A rocket engine [26], and compares the results obtained with the LPRES simplified models, against data from simulations with ESPSS. In ESPSS, the full geometry of the rocket engine has been provided as an input, as well as the boundary conditions, and the performances of the rocket engine have been calculated. In LPRES, on the contrary, some of the operational data of the system that we want to replicate are supplied as input data by the user (namely: oxidizer and fuel mass flow rates, pumps pressure leaps, turbine expansion ratio and rotational speed, etc.), and these data are used to *align* the simplified model with the original system by means of adjusting certain efficiency parameters and calculating the characteristic geometry (characteristic lengths and/or radii, and free-flow cross-sectional areas) for the main components. Table 1 shows that the objective of getting qualitatively good results, in decent agreement with reality, has been achieved. Relative errors for the vast majority of calculated parameters is kept below 5%, and only certain specific elements —as might

Table 1: Validation data for the design model of the RL10A-3-3A rocket engine, comparing the results obtained by means of the LPRES simplified formulation with the more accurate simulations performed by ESPSS.

	LPRES	ESPSS	rel. error
Fuel injector discharge area [cm ²]	17.8	17.7	0.6%
Oxidizer injector discharge area [cm ²]	5.8	6.1	4.9%
Turbine mass flow rate [kg/s]	2.1	2.2	4.5%
Combustion chamber temperature [K]	3181	3242	1.9%
Cooling jacket outlet total temperature [K]	164	174	5.7%
Cooling jacket outlet total pressure [bar]	69.6	76.1	8.5%

be the cooling jacket, that involves more complicated heat-transfer calculations— inherently hard to model by means of 0D components, yield errors slightly higher (see, for instance, the temperature and pressure conditions at the outlet of the cooling jacket reported in table 1, which both accumulate errors nevertheless well below 10%). This sort of discrepancies are usually considered acceptable for the very preliminary design stages (configuration proposals) of any rocket engine, and are definitely considered sufficient for an educational tool aimed at developing the *systems thinking* and *design thinking* competencies of the students.

It must be noted, though, that these relative errors —i.e. in the calculated operational parameters— might increase slightly when performing off-design simulations (generally speaking, the further the operating condition is from the set design point, the greater the errors that tend to accumulate), but are in any case kept below reasonable margins (typically well below 10%) and, more importantly, the trends of the *system dynamics* are always captured accurately from a qualitative point of view, which makes the tool again very useful to help the students develop their *systems thinking* competencies.

4 Pilot Experience

Once a sufficiently stable version of the LPRES library was developed, the next step was to implement a pilot experience in the classroom and to evaluate the initial results. The workshop carried out within the context of the MSc in Space Systems taught at the UPM consists of two different parts. First, the students are asked to attend a total of 8 hours of “practical lectures” in the computer lab; then, the students are given a short assignment, with an estimated overall dedication of an additional 8 hours on their own. Overall, the whole pilot experience, including both parts, spans over a total duration of 6 to 8 weeks.

The 8 hours of computer-lab “practical lectures” are used to gradually build up from a model of *structured inquiry* [3] at the beginning, in which the instructor outlines not only the problem but also some guidelines for solving the questions posed, so that the learners get familiar with the tool first; to a model of *guided inquiry* [3] towards the end, in which the students no longer need guidelines for the solution of the problem, since they are already confident in the usage of the software, and they can take on answering the questions posed by the professor on their own. The assignment follows the concept of PBL [3], formulating an open-ended “real-world” problem that requires the students not only to investigate a certain phenomenon or system, but also to make some ‘design decisions’ and to learn how to deal with, and reason about, the *uncertainty* associated with those decisions along the way [22].

Table 2: Set of abilities learned by the students through the EcosimPro® | LPRES workshop.

1	Advanced usage of the EcosimPro® environment and capabilities.
2	Development of new models and components by means of coding in EL within EcosimPro®.
3	Modeling and simulation of complex LPRM cycles by using LPRES reusable components.
4	Programming and running experiments to gather relevant data in EcosimPro® LPRES.
5	Deep understanding of the operation and control of LPRMs and their essential sub-systems.

Table 3: Main competencies developed by the learners through the EcosimPro® | LPRES workshop.

1	<i>Systems thinking</i> [as defined by 1], both in terms of <i>systems design</i> as well as <i>system dynamics</i> [as per 22].
2	<i>Experiments design</i> [as per 22] & <i>data literacy</i> [as per 1].
3	<i>Technological literacy</i> [as defined by 1].

The set of major abilities that the students are expected to learn from this workshop is summarized in table 2. From these abilities, and as previously described, the competencies that the learners will be able to develop thanks to this sort of academic training are gathered in table 3.

5 Results and Discussion

Properly speaking, the evaluation of the impact on learners' performance of the usage of the developed tool in teaching the subject of rocket engines should be able to show how much better the learning objectives are achieved by comparing at least two groups: one in which this pilot experience is put into practice (the 'experiment' group) and the other in which it is not (the 'control' group). Due to the small number of students enrolled in the Space Systems MSc every year, though, strictly limited to 20, it was impossible to ensure that the two groups would not be biased, and thus the decision was made to try the 'experiment' condition with the whole class.

As a means to objectively evaluate the impact of the usage of our developed tool in the classroom, a pre-workshop test was run, asking the students to answer a set of questions dealing with the topics that had already been covered during the theoretical lectures. This pre-test had a two-fold aim: *a)* to serve as a starting point to measure the retention from the students by means of only attending theoretical lectures in a mostly 'passive' role, and the gains achieved by means of the practical computer modeling activities imparted thereafter; and *b)* to help the instructor identify common error or misconceptions within the class, and tailor the problems posed to be investigated specifically to tackle those misconceptions, apart of the more general learning objectives established for the course. A post-test was also carried out, essentially asking the students to answer very similar

Table 4: Summary of results from the pre- and the post-test for the ‘experiment’ condition ($N = 17$; marks on a /100 scale).

	mean	st. dev.
pre-test	28.83	12.64
post-test	61.51	15.95

questions about the same topics, following already established methodology [see, for instance 8, 28], to objectively measure the gains in retention.

Results for both the pre-test and the post-test are summarized in table 4. As shown, the average score achieved by the students more than doubled from the pre-test to the post-test. Even though the samples are small, the Welch’s unequal variances t -test yields a p -value $< 0.1\%$ with a significance level $\alpha = 0.1\%$ (power of the statistic test using the assumption of equal variances for both samples close to 99%), which showcases that the null hypothesis is extremely likely to be rejected and the gains in retention achieved by the students from the practical workshop can indeed be treated as significant for the purposes of this study, even for such a low value of α .

In addition to the above, a short survey was conducted among the participants after the workshop (detailed results from this survey can be found in table 5), to evaluate their subjective perception of the overall impact of this training on their whole academic program. Psychometric scale surveys (also known as Likert scale surveys) are commonplace in assessing the perceived importance that the students give to a particular item of the questionnaire [e.g. 29, 30, 31].

In particular, this questionnaire summarized in table 5 aims to evaluate the degree of achievement that the students perceive related to the abilities described in table 2. Namely, Q5 in table 5 tries to evaluate the degree of achievement of ability 1 in table 2, Q10 and Q12 are related to ability 2, Q11 is related to ability 3, Q14 to ability 4, and Q3 and Q8 are related to ability 5 in table 2 (some of the statements have been posed in negative form, to try to avoid the acquiescence bias). The rest of questions in table 5 evaluate the structure and organization of the workshop and the practical lectures, or are in general broader questions that try to assess whether the students perceive this sort of active learning strategies (and, more specifically, this workshop in particular) as a positive experience overall, with a significant impact in the context of their whole academic training.

It is worthwhile highlighting a few conclusions drawn from the aforementioned survey:

- ✓ The students highly value the teaching of computational tools relevant to the space sector as having a very positive impact on their academic training (see Q1 and Q4 in table 5), also benefiting their future professional career (see Q2 in table 5).
- ✓ It is shown that there is a fairly common agreement among the students in considering the teaching philosophy of LPRES, using simple models to simulate complex systems, as quite useful to be able to better understand both the operation and performances of the whole system of a liquid-propellant rocket engine, as well as the behavior of its isolated sub-systems (see Q3 and Q8 in table 5).
- ✓ The students consider that the practical lectures in the computer lab have indeed accomplished their intended purpose of teaching the fundamentals of modeling with LPRES (see Q5 and Q10 in table 5), although they are a bit more hesitant as to whether they could make use of the aforementioned tools without having the assistance from the professor (see Q11 in table 5).

In addition to the above, the final report that the students must submit on the proposed assignment they were given is used by the professors to evaluate the level of realization that the students have achieved, not only of the abilities gathered in table 2, but also of the competencies described

Table 5: Answers to the survey conducted among the students enrolled in the Rocket Engines and Space Propulsion course (averaged). Q1–Q14 are represented on a 1 to 5 Likert scale (being 1 = “completely disagree” and 5 = “completely agree” with the statement).

Q1. Having taken part in the workshop on the usage of EcosimPro® LPRES has significantly contributed to the overall quality of my academic training.	4.8
Q2. Knowing the usage of EcosimPro® LPRES will prove useful for pursuing my future career in the space sector.	3.9
Q3. The use of simple models to simulate complex systems has helped me to understand the operation and performances of liquid-propellant rocket engines.	4.3
Q4. Learning computational tools relevant to the space sector is beneficial for expanding and developing the set of skills gained during my academic training.	4.7
Q5. The guided tutorials in the computer lab have been useful for their intended purpose of teaching me the usage of EcosimPro® LPRES.	4.3
Q6. Learning EcosimPro® LPRES has proven too difficult in the short amount of time provided by this course.	3.2
Q7. The pace of the guided tutorials in the computer lab was too quick, making these practical lectures difficult to follow.	3.2
Q8. The models chosen for the development of the guided tutorials in the computer lab were interesting to me, and significantly helped me to understand the performances of essential sub-systems of the liquid-propellant rocket engines.	4.4
Q9. The guided tutorials in the computer lab were posed in such a way that made these practical lectures dynamic and engaging for the students.	4.0
Q10. The way that the professor guided us during the practical lectures in the computer lab, building up models of gradually increasing complexity, has helped me in grasping the fundamentals of modeling with EcosimPro® LPRES.	4.6
Q11. I have learnt to use EcosimPro® LPRES thoroughly enough so as to be able to make use of the aforementioned computational tools on my own.	3.8
Q12. The number of sessions in the computer lab were just too few, and would have preferred a longer, more in-depth course on the matter, to delve deeper into the details of modeling with EcosimPro® LPRES.	3.8
Q13. The number of sessions in the computer lab were just enough so as to be able to tackle the proposed assignment about the off-design control and performances of liquid-propellant rocket engines on my own, without excessive difficulty.	4.0
Q14. The proposed assignment was challenging and interesting, and has encouraged me to investigate on my own aspects of the functioning of liquid-propellant rocket engines that have helped me in further understanding their control and performances.	3.9

in table 3. The distribution of grades obtained by the students in this assignment, based upon that submitted report, is summarized in table 6. These grades showcase that the vast majority of the participants in this modeling workshop did indeed achieve the expected levels of understanding and satisfactorily fulfilled all the evaluated learning objectives, with only 3.7% of the students (excluding those who did not submit) not reaching the minimum score of 5.0 necessary to pass.

For the purposes of calculating the overall grade of the course for the students, the marks they obtained in this assignment were averaged with the marks obtained in a series of other practical assignments (all targeting towards a more PBL approach) developed during the whole semester on other topics covered in the syllabus of the course, to account for as much as 40% of their overall grades

Table 6: Distribution of grades obtained by the students attending the workshop in their final submitted report.

Students' grade (on a /10 scale)	% of students
Did not submit	15.6 %
Unsatisfactory (4.9 and under)	3.1 %
Average (between 5.0 and 7.4)	43.8 %
Good (between 7.5 and 8.9)	25.0 %
Outstanding (9.0 and over)	12.5 %

(the remaining 60% corresponding to the final written exam). In 86.7% of the cases, the overall grades of the students were improved by the marks obtained in this series of practical assignments proposed during the whole semester, which also supports the idea that this sort of complementary workshops and practical activities greatly help the students in understanding the different concepts and acquiring the different learning objectives, and thus improve their overall development in the whole academic program.

5.1 Lessons Learned

The development of the LPRES library is an ongoing process and, as such, the features that we demand from the tool and the requirements that we have imposed are ever-changing as well. Nevertheless, the current state of maturity of the library has already allowed us to establish a core set of the most critical—or ‘desirable’—features that an educational tool should have in order to be suitable for implementing active learning strategies in the engineering classroom. The list provided herein comes from a thorough review of the public literature, as well as by our own experience in this development process and subsequent usage of the library, and can be used either as a checklist to assess the suitability of any readily available computational tool when considering its usage within the engineering classroom, or as a list of requirements to guide the development of new computational tools, irrelevantly of the field considered:

- The tool must have a ‘professional look’, aligned with current standard practices within the industry, which reinforces the perception from the students of acquiring *technological abilities* meaningful for their future professional careers, as shown by the students’ responses to the survey in table 5.
- Although the instructor should be in charge of leading and coordinating the different steps of the development process (so that they become *experts* in the usage of the tool, thus enabling JiTT strategies to be implemented successfully), the coding *per se* should be carried out by the students. The concept of a tool developed *by the students, for the students* increases the involvement of the classroom in the subject, and triggers greater interest to delve deeper into learning *modeling abilities*, as demonstrated by the unprecedented interest exhibited by the students in willing to engage in long-term projects with the Department to be able to contribute to the development of the tool.
- Whenever possible, the development of the tool as a part of an already established modeling environment (and one that is considered within the industry as the current standard practices), allows for long-term collaboration between the industry and the university in the concept of *cooperative education*, greatly benefiting the learners.

- The general philosophy of using “simplified models” to simulate “complex systems” has proven successful in facilitating the students’ better understanding of the phenomena governing the operation and *system dynamics* of rocket engines.
- Additionally, the fact that the code is made available for the learners allows them to see and, far more importantly, to modify the relations between the variables in a given model (including effects or considering new modeling strategies that they might think are relevant for the phenomena in study), providing them with a thorough comprehension of the models’ limitations and range of applicability.
- The tool should provide the learners with the means to intuitively model any system they may devise, and be flexible enough to cope with the mathematical burden of the solving process, so that the learners can focus on improving their *systems thinking* and *design thinking* competencies in the context of implementing PBL strategies in the classroom.
- Robustness is paramount, both from the numerical stability standpoint as well as by ensuring that the tool yields results with a sufficient degree of fidelity when compared against real-life rocket engines performance data. This provides for the students a solid basis to understand the importance of making quick *estimates* and dealing with *uncertainties* in any engineering design process.

6 Conclusions

The first phase of the project concluded by developing a professional-looking tool, the LPRES library for EcosimPro®, that has the same operating capabilities as the ESPSS professional libraries, but incorporating simple models such as those used to solve practical exercises in the classroom with “pen and paper”. In consequence, the learning curve for the LPRES library is shorter, making it possible to do several engine models and calculations in the time and space available in just a few class sessions within the course. The LPRES library can be used to simulate systems with many elements, which would otherwise be tedious and time-consuming to solve “by hand”, thus alleviating the ‘cognitive load’ for the learners, who may instead focus on the development of their *systems thinking* and *design thinking* competencies.

The most important features for a software tool to have in the context of implementing active learning strategies in the engineering classroom were summarized and discussed in Section 5.

Despite the fact that not every student managed to become absolutely confident in the usage of the tool, most of them acknowledged to be autonomous in the tasks of modeling and simulating, with only minor issues —typically related to numerical convergence problems (which many times are straightforward to solve for a more experienced user) and not with the modeling itself— occurring from time to time, and hence the two important conclusions from this study remain valid: *a)* students’ perception of the quality of their training improved, they felt more motivated and engaged in the subject more willingly, as shown by the survey carried out; and *b)* their grade of retention and understanding of the matter increased very significantly, as revealed by the pre- and the post-test scores achieved by the students, as well as by the degree of acquaintance with the learning objectives evaluated with their final submitted report.

Given the positive impact observed on the learners, both in their understanding and in their involvement within the subject of *Rocket Engines and Space Propulsion*, the practicum has been definitely incorporated into the standard program for the course, and is nowadays considered an essential part of the curriculum.

We expect that the results presented will encourage instructors in other fields with similar challenges for implementing active learning strategies to develop their own computer tools, and we firmly believe that the list provided of required features identified for such a tool to meet the

demands of its intended usage can be made extensible to any other discipline, surpassing just the engineering (and, more precisely, the rocket science) scope, as reported in this paper.

Acknowledgements

We would like to thank the Universidad Politécnica de Madrid (UPM), Spain, for funding this project, as part of the 2014 edition of the “Ayudas a la Innovación Educativa y a la Mejora de la Calidad de la Enseñanza” program; and Empresarios Agrupados, for the long-lasting collaboration that we have established over the last several years, as well as for their continuous support in the usage of their tools.

References

- [1] J.E. Aoun. *Robot-Proof. Higher Education in the Age of Artificial Intelligence*. The MIT Press, Cambridge (MA), US, August 2017. ISBN: 978-0-262-03728-0.
- [2] D.H. Jonassen and T.C. Reeves. Learning With Technology: Using Computers as Cognitive Tools. In D.H. Jonassen, editor, *Handbook of Research for Educational Communications and Technology*, pages 693–719. Simon & Schuster, New York, 1996. ISBN: 978-0028646633.
- [3] M.J. Prince and R.M. Felder. Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases. *Journal of Engineering Education*, 95(2):123–138, April 2006. DOI: <https://doi.org/10.1002/j.2168-9830.2006.tb00884.x>.
- [4] T. Campbell, S.K. Wang, H.-Y. Hsu, A.M. Duffy, and P.G. Wolf. Learning with Web Tools, Simulations, and Other Technologies in Science Classrooms. *Journal of Science Education and Technology*, 19(5):505–511, October 2010. DOI: <https://doi.org/10.1007/s10956-010-9217-8>.
- [5] J. Settlage. Demythologizing Science Teacher Education: Conquering the False Ideal of Open Inquiry. *Journal of Science Teacher Education*, 18(4):461–467, 2007. DOI: <https://doi.org/10.1007/s10972-007-9060-9>.
- [6] A. Johnston. Demythologizing or Dehumanizing? A Response to Settlage and the Ideals of Open Inquiry. *Journal of Science Teacher Education*, 19(1):11–13, 2008. DOI: <https://doi.org/10.1007/s10972-007-9079-y>.
- [7] M.J. Prince. Does Active Learning Work? A Review of the Research. *Journal of Engineering Education*, 93(3):223–231, July 2004. DOI: <https://doi.org/10.1002/j.2168-9830.2004.tb00809.x>.
- [8] C.C. Carey and R.D. Gougis. Simulation Modeling of Lakes in Undergraduate and Graduate Classrooms Increases Comprehension of Climate Change Concepts and Experience with Computational Tools. *Journal of Science Education and Technology*, 26(1):1–11, February 2017. DOI: <https://doi.org/10.1007/s10956-016-9644-2>.
- [9] C.V. Schwarz, B.J. Reiser, E.A. Davis, L. Kenyon, A. Achér, D. Fortus, Y. Shwartz, B. Hug, and J. Krajcik. Developing a Learning Progression for Scientific Modeling: Making Scientific Modeling Accessible and Meaningful for Learners. *Journal of Research in Science Teaching*, 46(6):632–654, August 2009. DOI: <https://doi.org/10.1002/tea.20311>.
- [10] C.V. Schwarz and B.Y. White. Metamodeling Knowledge: Developing Students’ Understanding of Scientific Modeling. *Cognition and Instruction*, 23(2):165–205, 2005. DOI: https://doi.org/10.1207/s1532690xci2302_1.

- [11] V. López and M.I. Hernández. Scratch as a Computational Modelling Tool for Teaching Physics. *Physics Education*, 50(3):310–316, May 2015. DOI: <https://doi.org/10.1088/0031-9120/50/3/310>.
- [12] Y. Rodríguez, A. Santana, and L.M. Mendoza. Physics Education Through Computational Tools: the Case of Geometrical and Physical Optics. *Physics Education*, 48(5):621–628, September 2013. DOI: <https://doi.org/10.1088/0031-9120/48/5/621>.
- [13] D.A. Muller. The Key to Effective Educational Science Videos. published at TEDTalentSearch YouTube channel, June 2012. URL <https://www.youtube.com/watch?v=RQaW2bFieo8>.
- [14] D.A. Muller. *Designing Effective Multimedia for Physics Education*. PhD thesis, School of Physics, University of Sydney, Australia, 2008. URL [http://www.physics.usyd.edu.au/super/theses/PhD\(Muller\).pdf](http://www.physics.usyd.edu.au/super/theses/PhD(Muller).pdf).
- [15] R.W. Bybee, J.A. Taylor, A. Gardner, P. Van Scotter, J.C. Powell, A. Westbrook, and N. Landes. The BSCS 5E Instructional Model: Origins and Effectiveness. A Report Prepared for the Office of Science Education, National Institute of Health (NIH); BSCS, Colorado Springs, 12 June 2006. URL <https://bscs.org/bscs-5e-instructional-model>.
- [16] R.R.L. Ernst. Liquid Rocket Analysis (LiRA) Development of a Liquid Bi-Propellant Rocket Engine Design, Analysis and Optimization Tool. MSc thesis, Faculty of Aerospace Engineering, Delft University of Technology, Netherlands, May 2014. URL <https://repository.tudelft.nl/islandora/object/uuid%3Ae2cda6d2-e3ff-4809-a62a-80c250146413>.
- [17] J. Bradford, A.C. Charania, and B.St. Germain. REDTOP-2: Rocket Engine Design Tool Featuring Engine Performance, Weight, Cost, and Reliability. In *Proceedings of the 40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Joint Propulsion Conferences*, Fort Lauderdale (FL), US, 11–14 July 2004. Paper no. AIAA-2004-3514 DOI <https://doi.org/10.2514/6.2004-3514>.
- [18] P. Sierra, J.M. Tizón, J. Vilá, and J.F. Moral. Efficient Simulation of Liquid Propellant Rocket Engine Cycle. In *Proceedings of the 53rd AIAA/SAE/ASEE Joint Propulsion Conference, AIAA Propulsion and Energy Forum*, Atlanta (GA), US, 10–12 September 2017. Paper no. AIAA 2017-5007 DOI <https://doi.org/10.2514/6.2017-5007>.
- [19] *ESPSS: European Space Propulsion System Simulation — EcosimPro Libraries User Manual*. Empresarios Agrupados Internacional, March 2014. ESA document 4000103800/11/NL/CP – TN4130.
- [20] M. De Rosa, J. Steelant, J. Moral, Y. Elkouch, and R. Pérez Vara. ESPSS: European Space Propulsion System Simulation. In *Proceedings of the Space Propulsion 2008 – 2nd International Symposium on Propulsion for Space Transportation*, Heraklion (Crete), Greece, 5–8 May 2008.
- [21] L.N. Woolsey. Solar Wind Modeling: a Computational Tool for the Classroom. *Journal and Review of Astronomy Education and Outreach*, (3):B49–55, 2015. arXiv: <https://arxiv.org/abs/1505.00727>.
- [22] C.L. Dym, A.M. Agogino, O. Eris, D.D. Frey, and L.J. Leifer. Engineering Design Thinking, Teaching, and Learning. *Journal of Engineering Education*, 94(1):103–120, January 2005. DOI: <https://doi.org/10.1002/j.2168-9830.2005.tb00832.x>.
- [23] A. Isselhorst. HM7B Simulation with ESPSS Tool on ARIANE 5 ESC-A Upper Stage. In *Proceedings of the 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Nashville (TN), US, 25–28 July 2010. Paper no. AIAA-2010-7047 DOI <https://doi.org/10.2514/6.2010-7047>.

- [24] J. Moral, R. Pérez Vara, J. Steelant, and M. De Rosa. ESPSS Simulation Platform. In *Proceedings of the Space Propulsion Conference 2010*, San Sebastián, Spain, 3–6 May 2010. URL https://www.ecosimpro.com/wp-content/uploads/2015/02/SpacePropulsion2010_ESPSS_simulation_platform.pdf.
- [25] J. Amer, J. Moral, and J.J. Salvá. Adaptation of the ESPSS/EcosimPro platform for the design and analysis of liquid propellant rocket engines. In *Proceedings of the Space Propulsion Conference 2012*, Bordeaux, France, 7–10 May 2012. URL https://www.ecosimpro.com/wp-content/uploads/2015/02/SpacePropulsion2012.paper_faj_jal.pdf.
- [26] M. Binder, T. Tomsik, and J.P. Veres. RL10A-3-3A Rocket Engine Modeling Project. Technical Report NASA-TM-107318, NASA Lewis Research Center; Cleveland (OH), US, 1 January 1997. URL <https://ntrs.nasa.gov/search.jsp?R=19970010379>.
- [27] P.J. Roache. Quantification of Uncertainty in Computational Fluid Dynamics. *Annual Review of Fluid Mechanics*, 29:123–160, January 1997. DOI: <https://doi.org/10.1146/annurev.fluid.29.1.123>.
- [28] Jing Yan, Lin Li, Jianjun Yin, and Yujing Nie. A Comparison of Flipped and Traditional Classroom Learning: A Case Study in Mechanical Engineering. *International Journal of Engineering Education*, 34(6):1876–1887, 2018.
- [29] A. González González, A. Sánchez Ríos, C.J. Cruz García, and D. Rodríguez Salgado. A Proposed Methodology to Evaluate Educational Competences in Engineering Degrees Based on Electronic Devices and Open Access Software. *International Journal of Engineering Education*, 34(4):1150–1158, 2018.
- [30] L. Fernández-Samacá, J.M. Ramírez Scarpetta, Ó.O. Rodríguez, and E. Franco Mejía. PBL Model for Single Courses of Control Education. *International Journal of Engineering Education*, 33(3):963–973, 2017.
- [31] William Z. Bernstein, Devarajan Ramanujan, Fu Zhao, Karthik Ramani, and Monica F. Cox. Teaching Design for Environment through Critique within a Project-Based Product Design Course. *International Journal of Engineering Education*, 28(4):799–810, 2012.