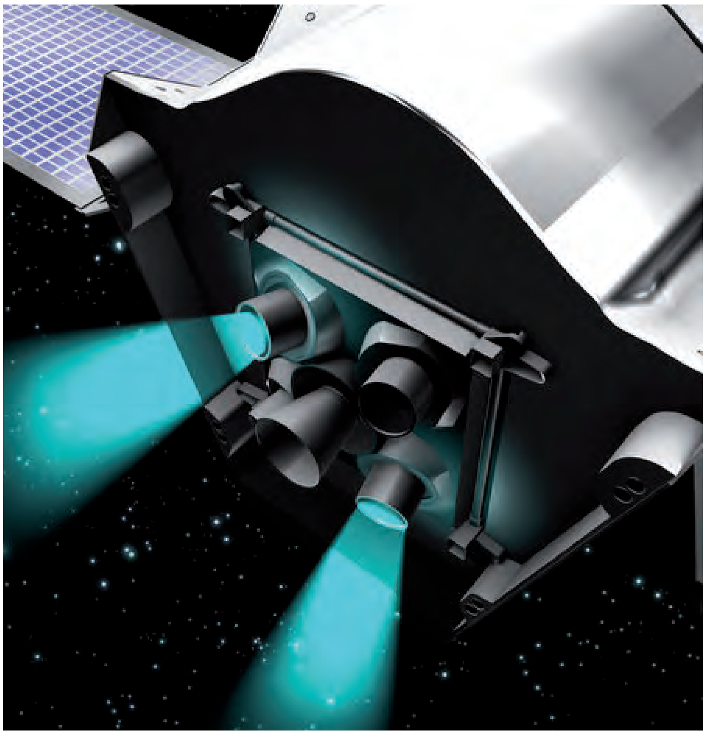
EcosimPro
Modelling and Simulation Software

EcosimPro · Newsletter Nº8 · July 2013

EDITORIAL



Bepi-Colombo satellite (ESA courtesy)

In this issue of the EcosimPro/PROOSIS newsletter, we intend to present some subjects that many of our users have had questions about and which we sometimes assumed were well understood from our courses and documentation but which we now see may actually benefit from a simpler, less technical explanation.

For example, the three types of IF used in the simulation language and that allow doing different things are presented: from the classic IF of traditional imperative programming that changes the flow of a program, to the IF that allows an equation to be changed at simulation time to the IF-INSERT that allows users to model components of variable complexities.

We also take a look at parallelism and simulation, an important subject. The EcosimPro/PROOSIS team is currently taking the first steps to introduce different parallelism techniques into the software that will enable using the full potential of current multi-core and multi-thread machines. Until now, practically the entire process ran on a thread, but the idea now is to begin exploiting the capacity to execute multiple calculations in parallel, which will reduce simulation times.

We also present here the evolution of the C++ compilers used in EcosimPro/PROOSIS over the years, showing how the latest versions are able to reduce simulation times enormously, because they generate code that is much more optimized. For example, we have noted improvements of up to 40% in simulation time between using the Microsoft Versión 6.0 compiler and its 2010 version using the same machine. We are thus recommending to all our customers that they move to this new version. You will notice the difference.

We have also taken a look at a complex subject that is familiar to anyone who has modelled systems with algebraic loops: how to best model an algebraic box with N variables. Minimizing algebraics, choosing them and selecting the residue equations area all fundamental to making a robust model: if they are not well chosen, the user is doomed to having convergence problems. EcosimPro/PROOSIS offers "help" tools in this field that are essential knowledge for any user working in this area.

Also included is a summary prepared by Alex Alexiou of the University of Athens of a paper he presented at last year's ASME TURBO conference and which received the "Cycle Innovations Committee Best Paper Award". The article describes a study on the performance of a counter rotating turbofan aircraft engine.

Finally, the newsletter contains a short guide to the information the tool offers about the mathematical models that are generated in partitions. We believe that it is very important to be clear about concepts like what are the inputs and outputs of a mathematical model area as well as on all the equations and the order of calculation of the model. Few tools give as much detailed information about their internal mathematical models as EcosimPro/PROOSIS do.

Pedro Cobas Herrero (pce@ecosimpro.com)
Head of the EcosimPro/PROOSIS Development Team
Empresarios Agrupados Internacional S.A.

LATEST NEWS ECOSIMPRO

SPACE PROPULSION WORSHOP AT ESA-ESTEC

A space propulsion workshop of the EcosimPro/ESPSS (European Space Propusion System Simulation) simulation platform was held from 21st-23rd May 2013 at ESA-ESTEC in Noorwijk (The Netherlands). Over 30 engineers from various European companies that work in space propulsion attended the meeting. The workshop was organised by the ESA (European Space Agency) Propulsion Department and Empresarios Agrupados (EA). ESA is the owner and manager of the ESPSS libraries and EA is the owner of the EcosimPro simulation tool.

Presentations by participants from the different companies attending were given on May 21st and 22nd. Pedro Cobas, Fernando Rodriguez and José Moral took part on behalf of Empresarios Agrupados, presenting the most recent novelties of both the EcosimPro simulation tool and the set of ESPSS libraries for modelling space propulsion systems. Of special note was the official presentation of the first version of the new steady state propulsion library, which complements the existing transient libraries. Now, modellers can build both steady state and transient models using EcosimPro/ESPSS. A new optimization tool integrated into EcosimPro was also presented. This means that users no longer have to export their models to a different, external optimization tool.

Photo courtesy of ESA

Christophe Koppel of Kopoos presented a satellite library that allows simulating the propagation of orbits with EcosimPro/ESPSS while modelling the propulsion system. This is another milestone in the platform's development, as this calculation previously required the use of an external tool.

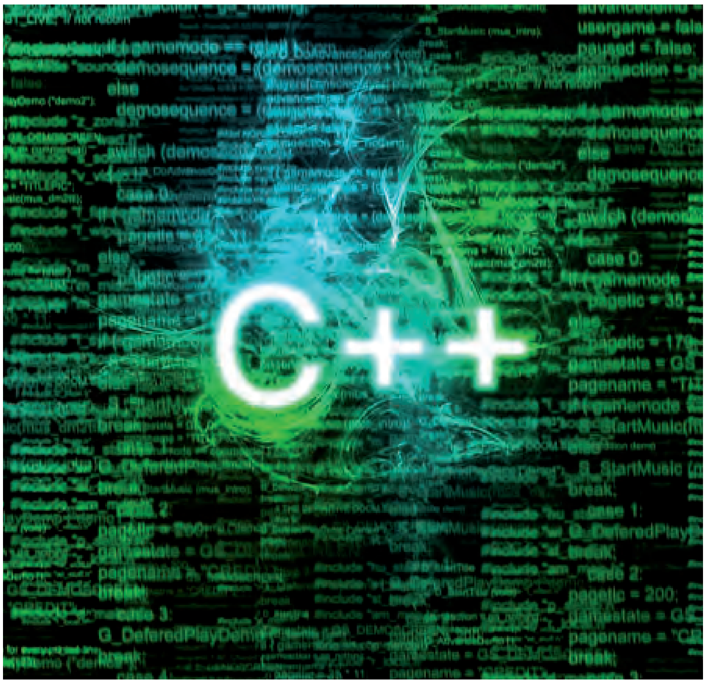
Apart from these initial presentations, others were given by a long list of companies, including Astrium (D&F), Von Karman Institute (Belgium), Deutschen Zentrum für Luft- und Raumfahrt, DLR (Germany), Centro Italiano Ricerche Aerospaziati CIRA (Italy), Snecma (France), European Space Agency ESA (Netherlands) and La Sapienza (Italy). In addition, several sessions were held on May 22nd to discuss problems encountered by users and proposals for further improvements to future versions of the EcosimPro/ESPSS platform. The idea is to consolidate this tool as a referent in ESA that will serve companies as the basis for modelling rocket and satellite propulsion systems of varying complexities.

Lastly, May 23rd was devoted to an introductory course on the use of the new Steady State library of the ESPSS. The course was given to 15 participants by José Moral and Fernando Rodriguez from Empresarios Agrupados. The new library is expected to be available shortly.

ESA aims to hold this workshop once a year. Once every two years it will coincide with the International Space Propulsion Conference organised by ESA, CNES and the space organizations of each ESA member state. The most recent conference was held in Bordeaux (France) in 2012, while the next one is set for 19th-22nd May 2014 in Cologne (Germany).

C++ COMPILERS IN ECOSIMPRO: CURRENT STATUS AND PROSPECTS

The first commercial version of EcosimPro was released in 1999. That was more than a decade ago. During this time, C/C++ compilers have undergone an important evolution. Some of the compilers available then have ceased to exist, while others have continued to evolve into new versions and other products from other companies have appeared.



EcosimPro generates models for simulation in C++ that need to be compiled with a C++ compiler prior to the simulation. This compilation is done automatically upon a request from the end user from the program. The first commercial version used the Microsoft Visual Studio C++ (MVS C++) 6.0. It was the right decision because it has been one of the most-used compilers in companies in the world. In fact, it is still used in some, although there is a tendency towards modernisation.

Later, EcosimPro incorporated support for the MVS 2003 .NET and GNU GCC compilers. The latter is an important C++ compiler that uses open code and is free, and it originates in the UNIX environment. Since this was expected to be a recurring scenario, version 4.8 of EcosimPro (2010) featured a scalable system that could support many future compilers. Thanks to this architecture it has been possible to add support for MVS C++ 2008 and MVS C++ 2010, as well as new versions of GNU GCC, simply and homogenously. There are currently few simulation tools that allow such a versatility of support to compilers.

In terms of performance we have made a comparison between MVS C++ 2010 vs MVS C++ 6.0 and the first is up to 40% faster than the second. We encourage all users to move to this new compiler since is much more powerful and they do not need to change anything on their libraries.

Over time, both the range of GNU GCC compilers and Microsoft ones have improved on both a programming environment level (speed, robustness, configuration options) and an implementation level for the various versions of the C++ language standard.

In addition, whenever mention is made of a compiler it is almost always necessary to speak about the hardware architecture for which binary code is to be created. In this sense, there has been a revolution in the field of computer processors since the second half of the past decade. The proposal of the manufacturers involved has been to leave the race to increase the frequency of the processors to one side and to focus on the calculation capacity, taking the number of cores in the processor as the scaling factor. Years ago, this technology was only available in special processors from certain companies, but they are now easily available in the market for computers, graphics cards and even cell phones, so companies in this sector are enjoying a great level of success.

Riding on the wave of this technological revolution, the programming libraries and compilers have benefitted from all this potential and generated highly-advanced codes that in most cases accelerate the execution speed. In some cases, the simple fact of compiling with them makes the C++ source code yield faster programs.

In line with its commitment to its clients, the EcosimPro/PROOSIS development team in Empresarios Agrupados expects to include support for the new Microsoft compiler MVS C++ 2012, in upcoming versions. In addition, the last few months have seen the completion of several tests designed to allow the range of Intel compilers to be used, both for C++ and for FORTRAN. These compilers perform well with regard to both simulation time and consistency of results, and they are widely used in academic and professional circles.

STATE OF THE ART OF PARALLEL COMPUTING WITH ECOSIMPRO

Attempts to improve the performance of programs have been ongoing for decades. Operating systems base a large percentage of their technology on the capability to allow several programs to run simultaneously in the best possible way. From the point of view of computer architecture, techniques to speed up access to the different types of memories, to improve pipelining, the execution of instructions that can actuate on a set of (vector) data, etc, have been incorporated.

There has been a revolution in the field of computer processors in recent years. The tendency is to stop increasing the frequency of the processors and to focus on the calculation capabilities, taking the number of cores in the processor as the scaling factor. These technologies were only available in special processors years ago, but they are currently easily accessible in the market for computers, graphics cards and even cell phones.

There are some parallels on some levels, in the sense that the evolution in hardware has gone hand in hand with the evolution in software to benefit from it. In recent years there has also been a revolution in the technologies that implement parallelism in applications. Historically, parallelism was achieved by using specific libraries that served to create a 'virtual parallel unit' by means of communication libraries for parallelism based on message transfers. The standard technology that was used in practice for this purpose is Message Passing Interface (MPI). This technology is very widespread in calculation centres and computer clusters in scientific and technological circles of companies and universities. With the dawn of multi-core processors, where memory is shared between the cores, the technology used to obtain parallelism has extended to that market too, and new programming environments or libraries that exploit this new capacity have appeared. OpenMP and Intel Threading Building Blocks are examples of this type of technology.

There is a tendency to mix this kind of technology with MPI to obtain parallelism in applications, both at the level of the multi-core processors where they are run and as regards the parallel execution of the application within a computer cluster. Applying these technologies is generally extremely complex, and it depends greatly on the problem to be solved. In the case of EcosimPro, having a large part of the simulation model generated dynamically presents an added challenge. Applying this type of solution to real models that establish a highly-specific and optimised calculation order is therefore very complex.

The upcoming versions of EcosimPro and PROOSIS will incorporate new capabilities of parallelism technologies, such as the simultaneous execution of simulation cases (e.g. parametric studies), that will greatly reduce calculation times, as well as other techniques to establish as much parallelism as possible between tasks of the same process. Furthermore, the new versions of C/C++/FORTRAN compilers already generate certain tasks in parallel, so the program does not need any more changes and simply changing compiler will improve calculation times.

SOME RECOMMENDATIONS FOR CREATING ROBUST NON-LINEAR EQUATIONS SYSTEM

One of the ways to simulate a system is by implementing and solving the equations that govern its behaviour. This is precisely what allows EcosimPro/PROOSIS to represent a process by means of algebraic differential equations and any potential events that could appear.

EcosimPro/PROOSIS can then be viewed as a pure solver of a series of previously-defined equations. There is a possibility that, when the equations are ordered, a set of them cannot be solved explicitly and need to be solved simultaneously. In turn, these equation systems can be linear or non-linear.

Whenever the equation system is linear, the problem is transparent to the user, that is, EcosimPro/PROOSIS has a solver that can solve the box without asking the user.

However, in non-linear equation systems EcosimPro/PROOSIS uses a particular heuristic technique based on the classical Equation Tearing algorithm. It identifies a set of variables on which the iteration is based to solve the non-linear box (tearing variables).

These non-linear boxes are common in models where only algebraic equations are used (steady simulation tools that are typical in engineering design and analysis)

The tearing variables and the residue equations need to be adequately selected to ensure the robust resolution of the non-linear equation systems. The aspects that a simulation engineer needs to take into account for this choice are following:

• the difficulty in solving certain equations: thus, for instance, equations where the tearing variable is dividing can create problems of divisions by zero. Selecting it would therefore not be appropriate.

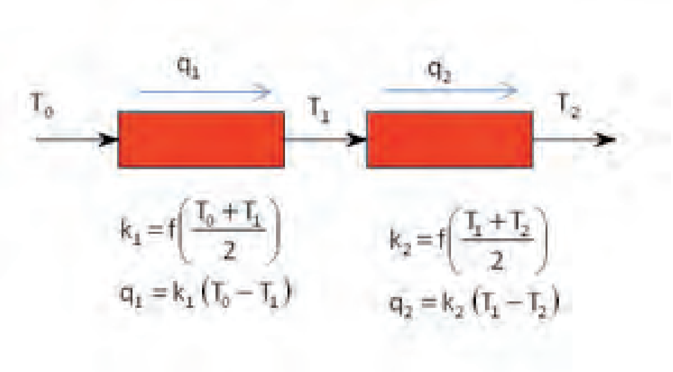
• the correct behaviour of the physics of the problem will always help to improve the selection.

A good choice of tearing variables and implicit equations has a direct bearing on the convergence properties of our model, hence its importance.

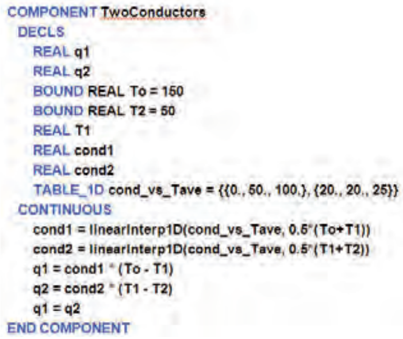
Normally a good choice is done automatically by EcosimPro/PROOSIS through the Equation Tearing algorithm. However, the modeller has a chance to modify it and improve it.

However, even better than the capability of changing this selection is possibility of 'aiming' the algorithm towards the solution we consider the most appropriate. This is done by means of the EL language (EcosimPro/PROOSIS language). The modeller can use certain directives to impose what variables are going to be tearing variables and what equations are going to be selected as implicit or closure equations so that it is easier to solve non-linear equation systems.

The following system can be used as a basis to explain how to obtain non-linear boxes that are more robust from the point of view of convergence:

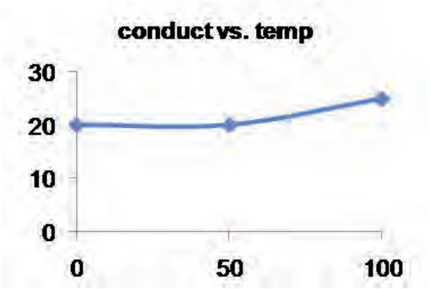

q_1
q_2
T_0
T_1
T_2
$k_1 = f\left(\frac{T_0+T_1}{2}\right)$
$q_1 = k_1(T_0 - T_1)$
$k_2 = f\left(\frac{T_1+T_2}{2}\right)$
$q_2 = k_2(T_1 - T_2)$

The implementation in EcosimPro/PROOSIS would be as follows:



```
COMPONENT TwoConductors
 DECLS
   REAL q1
   REAL q2
   BOUND REAL To = 150
   BOUND REAL T2 = 50
   REAL T1
   REAL cond1
   REAL cond2
   TABLE_1D cond_vs_Tave = {{0., 50., 100.}, {20., 20., 25}}
 CONTINUOUS
   cond1 = linearInterp1D(cond_vs_Tave, 0.5*(To+T1))
   cond2 = linearInterp1D(cond_vs_Tave, 0.5*(T1+T2))
   q1 = cond1 * (To - T1)
   q2 = cond2 * (T1 - T2)
   q1 = q2
END COMPONENT
```

These are two heat ducts in serial connection where the inlet and outlet temperatures are known (T0 and T2 respectively), and where the conductivity depends on the temperature.



Conductivity law vs temperature

The previous model has a non-linear box because the conductivity depends on the temperature at the ends of each duct. EcosimPro/PROOSIS suggests the following when it applies Equation Tearing to solve this box:

• tearing variable: T1

• residue equation: cond2 = linearInterp1D(cond_vs_Tave, 0.5 * (T1 + T2))

When the simulation is run, the solver fails because the equation that was selected as implicit or closure equation is not sensitive to changes in the tearing variable.

When the physics of the problem are analysed it is easy to determine that the correct choice would have been the following:

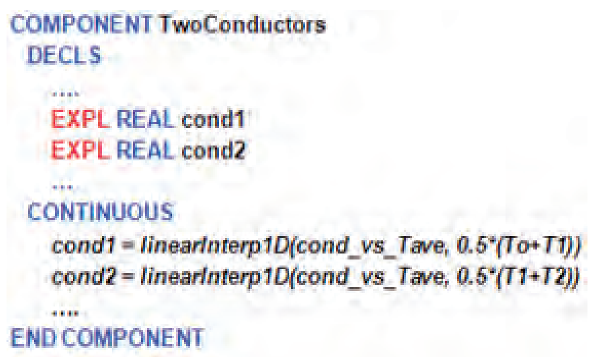
• tearing variable: T1

• implicit equation: 0 = q1 - cond2 * (T1 - T2)

How can this be achieved? New directives need to be introduced in the modelling. Below are some possible solutions.

Solution 1: EXPL directive

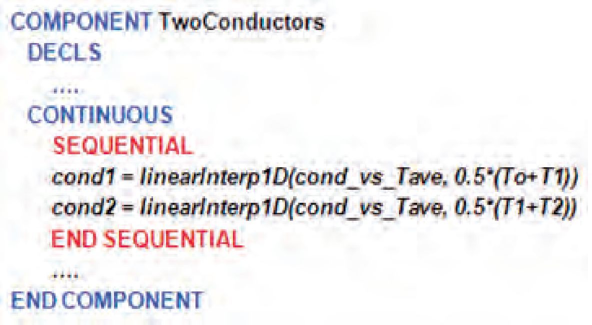
If a variable is declared as EXPL, it will be calculated explicitly by means of the corresponding equation. If this system is applied to the above system, the conductivities will be calculated directly with the table. The simulation is then run correctly.



```
COMPONENT TwoConductors
 DECLS
   ....
   EXPL REAL cond1
   EXPL REAL cond2
   ...
 CONTINUOUS
   cond1 = linearInterp1D(cond_vs_Tave, 0.5*(To+T1))
   cond2 = linearInterp1D(cond_vs_Tave, 0.5*(T1+T2))
   ....
END COMPONENT
```

Solution 2: SEQUENTIAL block

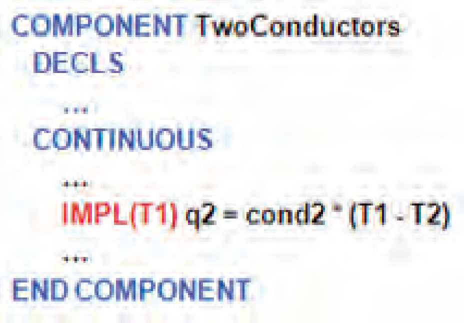
Sequential blocks in EcosimPro/PROOSIS are run sequentially. The process is therefore similar to running a function. If the calculation of the conductivity is done in this way, we will therefore avoid the possibility of selecting these equations as implicit equations, so that the required non-linear box will be obtained.



```
COMPONENT TwoConductors
 DECLS
   ....
 CONTINUOUS
   SEQUENTIAL
   cond1 = linearInterp1D(cond_vs_Tave, 0.5*(To+T1))
   cond2 = linearInterp1D(cond_vs_Tave, 0.5*(T1+T2))
   END SEQUENTIAL
   ....
END COMPONENT
```

Solution 3: IMPL (tearing variable) directive

Finally, the IMPL(variable) operator can be used. This operator 'sets' the equation that is to be implicit and the tearing variable at the same time. The following is obtained when it is applied to our case:



```
COMPONENT TwoConductors
 DECLS
   ...
 CONTINUOUS
   ...
   IMPL(T1) q2 = cond2 * (T1 - T2)
   ...
END COMPONENT
```

Again, using these operators has yielded a robust system of non-linear equations. The system simulation will therefore not have any convergence problems.

As a conclusion, it can be stated that the modeller should program the components to be as robust as possible so that the algebraic variables that are selected and the residue equations will be robust to allow convergence.

EcosimPro/PROOSIS currently selects automaticall a set of algebraic equations/residue equations to be selected that in many cases is good enough, but the upcoming versions will go even further and help minimise the number of algebraic equations and detect convergence problems with the residue before any study is done. This will allow more robust mathematical problems to be created and typical problems in these cases, such as non-convergence, to be predicted.

EcosimPro
Modelling and Simulation Software

GETTING INFORMATION ABOUT THE MATHEMATICAL MODEL

Solving of a system of 5000 equations has never been easy, not even with today's computers, capable of doing millions of operations per second.

The main problem is that these equations have to be processed symbolic transformations in order to obtain a sequentially operable system.

The EcosimPro simulation tool makes it possible to process and solve these systems almost automatically, with minimum effort on the part of the user.

What's more, thanks to the tool's flexibility, the user can also configure the system in the way that best adapts to his models, by selecting the inputs:

- DATA: values of the variables that act as constants and can only change discretely during the simulation

- BOUNDARY: boundary conditions when dealing with an indeterminate system, which can be defined as data or as temporal laws

- Jacobian Initialisation:

  o DYNAMIC: initial values of the dynamic variables that change over time

  o ALGEBRAIC: initialization of algebraic variables

The outputs of the equations system are:

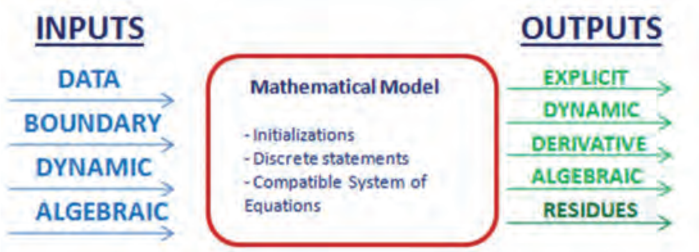
- EXPLICIT: variables calculated explicitly by the equations system

- DERIVATIVE: final values of the derivative variables

- DYNAMIC: final values of the dynamic variables that change over time

- ALGEBRAIC: final values of the algebraics

- Residues: final value of the residue equations of the Jacobian (normally the final value are close to zero when there is convergence).

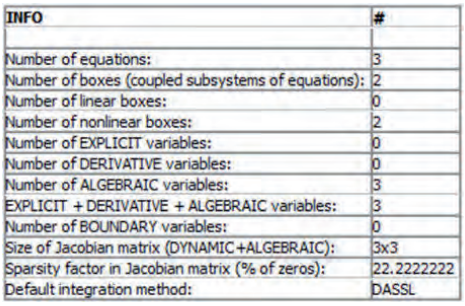


Input and output variables in the model

The system generated by EcosimPro receives the values configured by the user and performs the simulation, reporting the evolution of the variables calculated.While other simulation programs are rather opaque when it comes to information about the mathematical model generated, EcosimPro offers the user all the information he or she needs to carry out an in-depth analysis, simply and intuitively.

The .html file generated by EcosimPro for easy viewing of the model using the explorer has several sections:

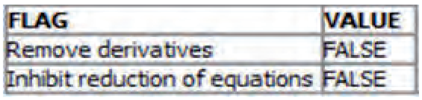
• General information: number and types of variables and equations, integration methods, information on the non-linear systems in the model, and default transient calculation method.



INFO	#
Number of equations:	3
Number of boxes (coupled subsystems of equations):	2
Number of linear boxes:	0
Number of nonlinear boxes:	2
Number of EXPLICIT variables:	0
Number of DERIVATIVE variables:	0
Number of ALGEBRAIC variables:	3
EXPLICIT + DERIVATIVE + ALGEBRAIC variables:	3
Number of BOUNDARY variables:	0
Size of Jacobian matrix (DYNAMIC+ALGEBRAIC):	3x3
Sparsity factor in Jacobian matrix (% of zeros):	22.2222222
Default integration method:	DASSL

General view of the mathematical model

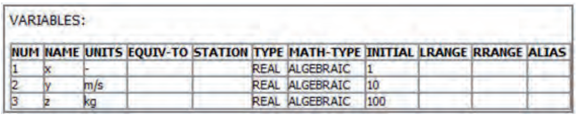
• Options configured for solving the system: elimination of derivatives in steady states or reduction of linear equations, among others.



FLAG	VALUE
Remove derivatives	FALSE
Inhibit reduction of equations	FALSE

Advanced flags

- Detailed information of the variables: initial values, equivalencies, type, alias, etc.

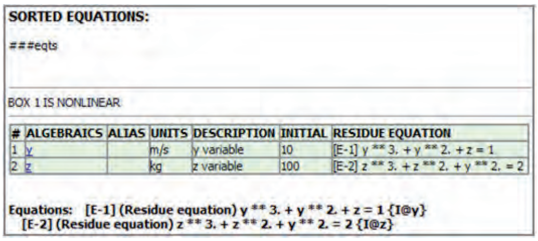


VARIABLES:

NUM	NAME	UNITS	EQUIV-TO	STATION	TYPE	MATH-TYPE	INITIAL	LRANGE	RRANGE	ALIAS
1	x	-			REAL	ALGEBRAIC	1			
2	y	m/s			REAL	ALGEBRAIC	10			
3	z	kg			REAL	ALGEBRAIC	100			

Variables List

- Order of solving of the mathematical model: system of equations shown in order of execution, including the path followed to break algebraic loops with all the necessary information of the iteration variables.



SORTED EQUATIONS:

###eqts

BOX 1 IS NONLINEAR

#	ALGEBRAICS	ALIAS	UNITS	DESCRIPTION	INITIAL	RESIDUE EQUATION
1	y		m/s	y variable	10	[E-1] y ** 3. + y ** 2. + z = 1
2	z		kg	z variable	100	[E-2] z ** 3. + z ** 2. + y ** 2. = 2

Equations: [E-1] (Residue equation) y ** 3. + y ** 2. + z = 1 {I@y}
[E-2] (Residue equation) z ** 3. + z ** 2. + y ** 2. = 2 {I@z}

Equations

In light of the above, the user will find in EcosimPro not only a tool focussed on the simulation of encapsulated and modular systems, but also a powerful tool for the analysis and solving of complex mathematical systems.

GRAPHIC MODELLING WITH ECOSIMPRO/PROOSIS

Graphic modelling is the capability of representing a model by visual means; however, beyond this formal definition, its usefulness for end users is twofold: it facilitates the construction of models and provides a better understanding of the finished models.

But, how does graphic modelling work in EcosimPro? What tools are available? What are the concepts that define it? We will now try and provide answers to all these questions

The EcosimPro/PROOSIS graphic editor

The design of graphics in EcosimPro is done with a vector editor for 2D graphics that has been entirely developed by the Empresarios Agrupados team.

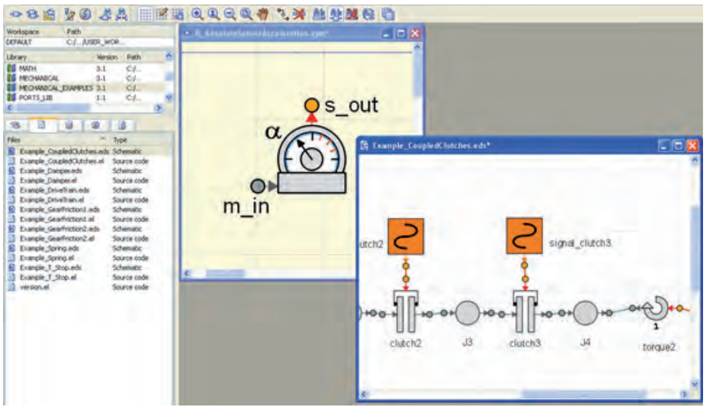


Fig. Aspect of the current EcosimPro graphics editor

The improvements since the first versions have been substantial in aspects such as management, the final quality of the graphics, the simplicity of the operations, etc. This evolution is ongoing, and new improvements and capabilities have been added in each version.

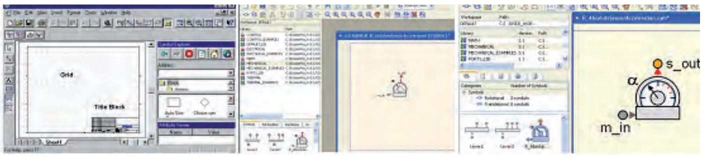


Fig. Evolution of the graphics editor (left to right): SmartSketch from Intergraph(year 2000), EcosimPro 4 (year 2004) and the current EcosimPro 5 (year 2011)

The ultimate goal is to create graphical representations of models based on simple geometric figures such as straight lines, curves, polygons, ellipses, etc. The geometry, outline and colour of all these figures will be editable.

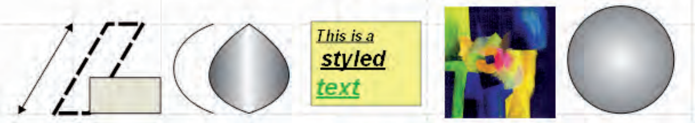
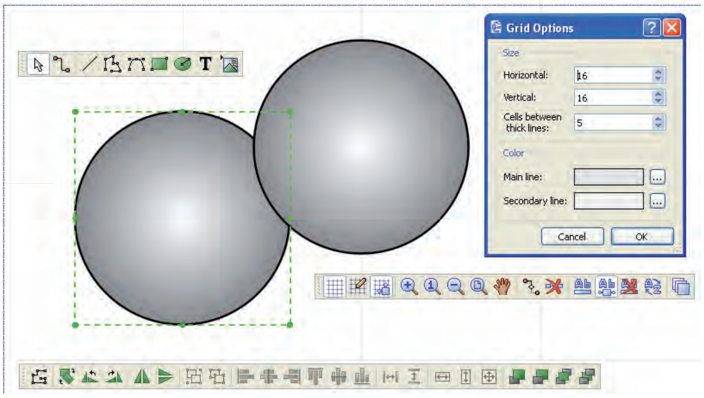


Fig. Different figures, texts and images created with the editor

Like any modern graphics editor, it has features that improve the display and printing of the graphics, the general management of the elements, such as grids for the distribution of the graphical elements, printing margins, dialog boxes to edit the styles, manage-

ment of the elements in layers, etc.



Grid Options

Size

Horizontal: 16

Vertical: 16

Cells between thick lines: 5

Color

Main line:

Secondary line:

Cancel OK

Fig. Elements of the editor: grid, control points, zoom, etc

Below are the two essential aspects for which the editor was conceived: the creation of symbols and schematic diagrams.

Symbols

EcosimPro is an object-oriented modelling tool. Thus, the modelling is based on establishing what parts form a system and then, within these parts, knowing the physical and mathematical behaviour that regulates them.

In EcosimPro jargon, analysing this 'individual' behaviour of the subsystems and describing it in equations with the EL language is known as creating a component.

Graphics modelling goes beyond the level of language because it is possible to create a representative symbol of a component and thus transfer the 'mathematical' level of the EL component to a 'visual' representation that is closer to the real system. This has certain clear advantages, such as understanding, reuse, etc.

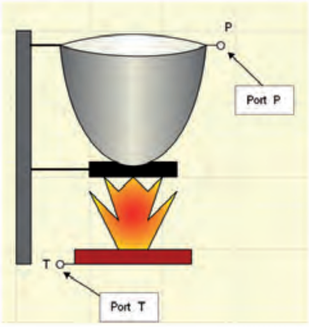
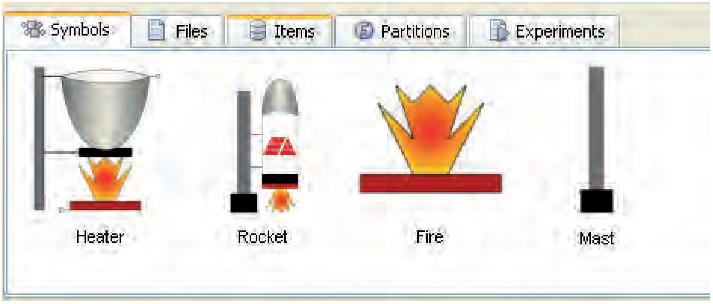


Fig. A symbol with its graphics and ports

All these symbols are stored in the libraries with the rest of the EcosimPro structures. A palette of reusable symbols therefore becomes available.

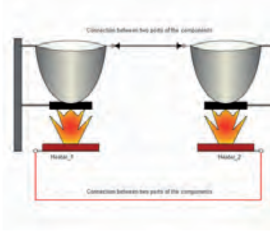


Symbols palette

Therefore, the symbols are the graphical representation of the components and the starting point of any modelling based on the connection of components.

Schematic diagrams

Once the symbols are completed, graphically creating new models is as easy as clicking and dragging to a new model that is known as a schematic diagram in EcosimPro.


Connection between two ports of the components
Heater_1
Heater_2
Connection between two ports of the components

Schematic diagram

The end result is a new component that is the topological interconnection of other simpler subsystems. In the next versions of the newsletter will present other graphics capabilities of the tool.

NEW PROOSIS AERONAUTICAL LIBRARIES

A set of libraries is being developed in PROOSIS to perform the integrated simulation of the engines and the main systems of an aircraft within a single simulation platform:

- Environment Control System (ECS)
- Pneumatic system
- Fuel system
- Electrical System
- Hydraulic system
- Nacelle ventilation

The AIRCRAFT_ECS library includes a set of components and subsystems for the simulation and modelling of the complete ECS system of an aircraft and of its bleeding system. The library allows models to be created with different aircraft cockpit configurations (several regions, different options of air distribution, etc). A model of two thermal nodes for passenger/crew is included. This model allows passenger comfort to be assessed.

The AIRFRAFT_LIQ library focuses on the 1D simulation of liquid flow networks and allows the aircraft fuel and lubricant system to be modelled. Phenomena such as thermal delay due to convective transport, thermal accumulation and expansion of the liquid inventory are taken into account in the formulation of the library.

The AIRCRAFT_ELEC library is used to simulate the electrical power supply network of a generic aircraft (generator, distribution network, loads of different characteristics, controllers, etc) coupled to the propulsion system.

Below are the symbol palettes of the AIRCRAFT_ECS, AIRCRAFT_LIQ and AIRCRAFT_ELEC libraries.

Palette of symbols of aeronautical libraries

Some examples have been developed to highlight the capabilities of the new libraries:

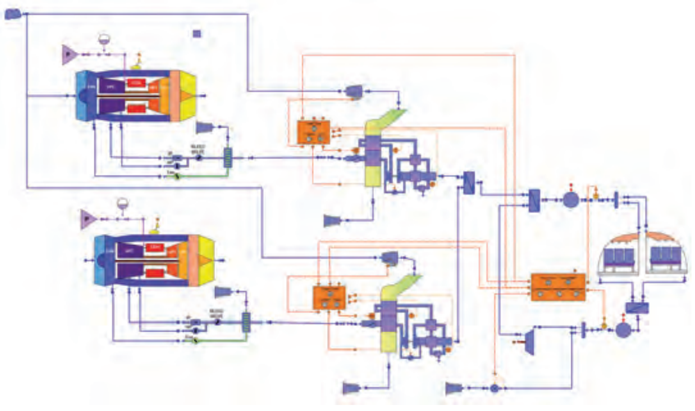
• Fuel loop connected to the GT motor and to the lube system

• Complete ECS system of a commercial airliner connected to a cockpit and to the engines

• Simplified model of the Electrical Power System (EPS) of a commercial airliner connected to an aircraft engine

Below is the diagram of the simulation model for the ECS system connected to the two aircraft engines (modelled with the TURBO library).

The different components that represent the main equipment can be identified in the model schematic diagram: the aircraft engines and its bleeding system, the air cooling packs, the cockpit model, etc.



Example of an aircraft model with 2 engines and ECS system

The complete aircraft simulation solution developed in PROOSIS allows users to easily evaluate the interactions between the various aircraft systems within the same simulation platform. It also minimises the analysis time and the associated costs and avoids co-simulation issues.

BEST AWARD FOR PAPER MODELLED WITH PROOSIS RELATED TO MODELLING THE PERFORMANCE OF A CONTRA-ROTATING TURBOFAN

The paper GT2012-69433 entitled "MODELLING CONTRA-ROTATING TURBOMACHINERY COMPONENTS FOR ENGINE PERFORMANCE SIMULATIONS: THE GEARED TURBOFAN WITH CONTRA-ROTATING CORE CASE" written by Alexiou, Roumeliotis, Aretakis, Tsalavoutas and Mathioudakis from the Laboratory of Thermal Turbomachines at the National Technical University of Athens was presented at the ASME TURBO EXPO 2012 Conference in Copenhagen, Denmark and received the Cycle Innovations Committee Best Paper Award.

The paper is also published in the ASME Journal of Engineering for Gas Turbines and Power (http://dx.doi.org/10.1115/1.4007197). The simulations described in the paper have been performed with PROOSIS v2.6.0. An extended abstract of the paper is included below.

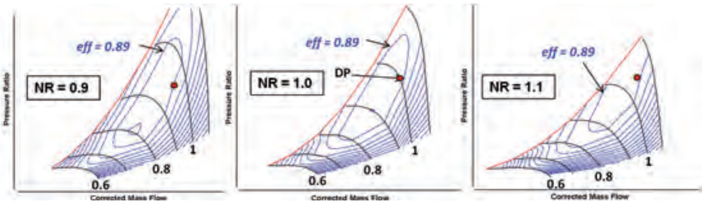
Some of the current and future engine configurations contain contra-rotating turbomachinery components (fans, propellers, compressors and turbines). because of certain advantages they have compared to the corresponding conventional ones (e.g. lower engine size and weight, increased efficiency and thrust-to-weight ratio, lower noise).

There are currently only a few studies available in the public domain dealing with the subject of performance modelling for engine configurations containing contra-rotating turbomachinery components. In these studies though either a basic design point analysis is presented or any off-design analysis is carried out assuming conventional component performance characteristics.

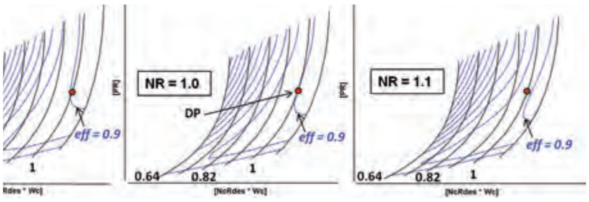
Hence, the objectives of the paper are to first derive physics-based performance characteristics of contra-rotating turbomachinery components then develop in PROOSIS component models that use these characteristics and finally, integrate these components in an engine performance model.

The first step for generating the performance characteristics is to obtain a rough estimation of the design parameters (e.g. blade angles) through a velocity triangles analysis. Then modified 1-D mean line analysis codes are employed that take into account flow angle deviation and pressure losses through empirical correlations from cascade experimental data.

Finally, the flowpath and blade geometry are further refined by applying basic design rules (e.g. flow coefficient, pressure rise coefficient, De Haller number) and previous experience from existing conventional compressors and turbines.



Contra-Rotating Compressor Map For Different Values of Speed Ratio

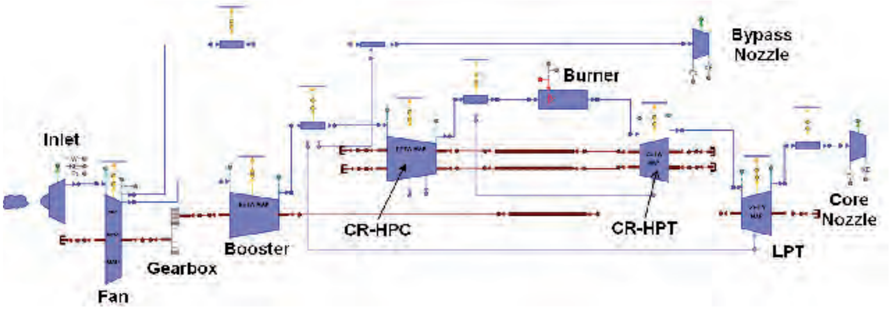


Contra-Rotating Turbine Map For Different Values of Speed Ratio

Having established basic geometric data for the contra-rotating components, the corresponding performance maps are generated in the form of three-dimensional tables (maps). Compared to conventional turbomachinery component maps, the speed ratio (NR) between the two shafts is included as an additional map parameter and the torque ratio as an additional table. The maps corresponding to the 7-stage contra-rotating compressor and two-stage contra-rotating turbine of the test case configuration are presented in the following figure for three different values of speed ratio (NR = 0.9, 1.0 and 1.1).

Dedicated component models are then developed in PROOSIS that use these maps to simulate design and off-design operation at component and engine level.

For configuration and version control purposes, the contra-rotating components are developed in a dedicated library which acts as an extension of the PROOSIS standard TURBO library since it is allowed to use all its components and ports. As the basic



Turbofan model in PROOSIS

compressor and turbine functionality is common for conventional and contra-rotating components and taking advantage of the abstraction and inheritance properties of the object-oriented simulation environment, only the code related to the contra-rotating part needs to be described in the new library.

Using these components, a performance model of a geared turbofan with a Contra-Rotating Core is created. This configuration was investigated in the context of the European program NEWAC (http://www.newac.eu/). In this arrangement, the compressor comprises an inner rotor (like in a conventional case) and an outer rotor so that successive blade rows can rotate in opposite directions. The rotors are driven by a contra-rotating turbine through two concentric drive-shafts, as shown in. the schematic diagram below:

The engine model is then coupled to an aircraft performance model and a typical short-range mission is carried out. The results are compared against those of a similar configuration employing a conventional core and identical design point performance. For the given aircraft-mission combination and assuming a 10% engine weight saving when using the Contra-Rotating Core arrangement over the conventional one, a total fuel burn reduction of 1.1% is predicted. If the engine weight saving is used to increase payload then a total of 3.2% fuel burn reduction per 100 passenger kilometre can be achieved. Further benefits could have been predicted if a lower amount of turbine cooling flow was assumed for the Contra-Rotating Core since there are reduced cooling and sealing requirements in this configuration as there is no stator between the two turbine rotors.

The methodology described in the paper is not application specific and is also relevant to contra-rotating turbofan and contra-rotating open rotor cases. Higher fidelity codes or experiments can be used to generate more accurate physics-based performance characteristics for the contra-rotating components – the integration of the maps can be performed in the same way.

Please feel free to send us any suggestions you may have for the EcosimPro newsletter, whether these have to do with content, articles or interviews that you would like to see published in coming issues to info@ecosimpro.com.

We look forward to receiving them. With your collaboration, we can continue to improve and expand the newsletter sections that are most relevant to you.

SUGGESTIONS

FAQS: DIFFERENCES BETWEEN THE 3 TYPES OF IF AND EXAMPLES

EMPRESARIOS AGRUPADOS

In EcosimPro's EL Language, different options are available to the user for marking the conditionality of an expression when modelling components. Doubts sometimes arise about these different options and how to use them.

EL possesses the classic sequential type structure IF-THEN-ELSE that can be used in sequential blocks such as the INIT of the components and the BODY of the functions defined in EL. Its purpose is to select a sequence of assignments for execution, depending on the value of a boolean. The syntax is traditional:

```
IF condition THEN
seq_stm_s
(ELSEIF condition THEN seq_stm_s) _
(ELSE seq_stm_s)
END IF
```

where **condition** is the condition being evaluated and **seq_stm_s** are the instructions that are going to be executed. During the run, the conditions are evaluated sequentially and if their value is TRUE, the corresponding instructions are executed. These kinds of structures can be nested and the conditions can contain logical expressions that use the logical operators AND, OR or expressions of the type less than or equal to (<=), greater than or equal to (>=), etc.
Below is a example of its use:

```
IF (signa l == SIN) THEN
    value = s in (2 * 3.14159 *x )
ELSEIF (signa l == SQUARE) THEN
   value = isqr
ELSE
   value = 1 .
END IF
```

A second type of IF is used in the CONTINUOUS block of the components and ports. In this case, an equation can be changed dynamically according to a condition. The mathematical model considers all of the branches defined and depending on the value that the condition takes during the simulation, will use one or another during the time that the condition is matched.

```
COMPONENT if_example
DECLS
  REAL x= 3
  ENUM modes switchMode= DESIGN
CONTINUOUS
    x= IF (switchMode == DESIGN) 5 . 6
         ELSEIF (switch Mode== TRANSIENT) sin (TIME)
         ELSE sin (TIME) + cos (TIME)
END COMPONENT
```

There is a third type of IF in the continuous block that allows introducing one set of equations or another (in the previous IF, this was done on the individual equation level). This is IF-INSERT. A typical case here would be a component with a parameter that determines whether a simpler or more complicated mathematical model should be used: the parameter determines which equations are taken for the simulation and which are discarded.

The variables that can be checked in this IF have to be construction parameters, because the configuration of the component has to be known beforehand.

In the following example, IF-INSERT is used to decide whether the x and y variables are calculated analytically or are read from a table depending on the value of the use Interpolation boolean parameter. As a function of the value of the condition, one block or another of equations will be introduced in the mathematical model.

```
COMPONENT example (BOOLEAN useInterpolation= TRUE)
DECLS REAL x
  REAL y
  TABLE_1D t1 = { { 0 , 1 } , { 0 . 2 3 , 0 . 8 9 } }
CONTINUOUS
   IF (use Interpolation == TRUE) INSERT
        x= linear Interp1D (t1, TIME)
        z= linear Interp1D (t1 , TIME+1)
   ELSE
        x= cos (y) **2
        z= sin (y) **2
   END IF
   y = sqr t (TIME)
END COMPONENT
```

This clears the way for the creation of components of variable complexity depending on the type of analysis to be performed.

EA Internacional S.A.
Magallanes, 3 Madrid
28015 Spain
E-mail: info@ecosimpro.com
URL: http://www.ecosimpro.com
Phone: +34 91 309 81 42
Fax: +34 91 591 26 55

EcosimPro
Modelling and Simulation Software

PROOSIS
Propulsion Object Oriented Simulation Software

EMPRESARIOS AGRUPADOS