POLITECNICO DI MILANO
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Ingegneria Chimica

3D IMMERSIVE DYNAMIC SIMULATION: THE AG2S™ PROCESS CASE

Relatori: Prof. Flavio Manenti, Prof. Simone Colombo

Correlatore: Dr.- Ing. Erik Esche

Tesi di Laurea Magistrale di:

Filandro Amoroso
Matr. 833674

Anno Accademico 2015-2016

A coloro i quali han sempre creduto in me, a tutti i docenti che hanno arricchito il mio bagaglio personale di conoscenze e competenze, ma anche insegnato importanti lezioni di vita. In particolare ai miei relatori di tesi, i professori Colombo e Manenti, che hanno creduto in me permettendomi di collaborare ad un progetto ambizioso ed interessante, ricco di nuove sfide e competenze da acquisire, offrendomi anche la possibilità di realizzare parte del lavoro in un'università estera, un'avventura da me tanto desiderata in passato ed ancor più apprezzata ora che si è conclusa. Un ringraziamento anche ad Erik Esche, persona fantastica che ha reso il mio lavoro alla TU Berlin fruttuoso e stimolante. Infine, un ringraziamento a tutta la mia famiglia che mi ha permesso di realizzare il mio sogno di bambino, quello di conoscere il mondo attraverso le sue leggi.

INDEX

FIGURE INDEX

TABLE INDEX

ABSTRACT

Abstract in italiano

La Simulazione Dinamica Immersiva in 3D può rappresentare un nuovo approccio nella formazione degli operatori in ambito industriale o di laboratorio. Tramite la formazione immersiva in 3D gli operatori possono infatti sperimentare differenti scenari di lavoro, incluse simulazioni di incidenti, con il valore aggiunto della perfetta riproducibilità delle simulazioni ed in completa sicurezza. Inoltre, sarebbe possibile introdurre all'interno delle simulazioni un sistema di valutazione delle performance degli operatori, in modo da avere diretto riscontro dell'efficacia della formazione. Questo lavoro di tesi si propone di gettare le basi per la realizzazione della simulazione dinamica immersiva di un esempio reale applicativo, producendo il modello ingegneristico in 3D e la simulazione dinamica del processo sperimentale di laboratorio *AG2S* (Acid Gas To Syngas) ed analizzando il potenziale dell'ambiente di simulazione MOSAIC (realizzato presso l'università TU Berlin) come fulcro di gestione del processo di virtualizzazione. Il processo *AG2S* sperimentato presso il Politecnico di Milano si propone di produrre Syngas, gas combustile commercialmente di valore, a partire da gas acidi quali la CO_2 e l'H_2S, prodotti di coda provenienti dalla desolforazione di Syngas da carbone, potenzialmente dannosi e/o tossici per l'ambiente e per l'uomo. La scelta di un esempio applicativo reale in scala di laboratorio simboleggia inoltre la volontà di realizzare il progetto al fine di consentire la formazione di operatori non solo di processi industriali, bensì anche di laboratorio.

Abstract in inglese

The 3D Immersive Dynamic Simulation can be a new approach for industrial and laboratory operators training. With an immersive 3D training the operators could experience different work scenarios, including accidents, with the added values of complete safety and perfect reproducibility of the simulations. Besides, it could be also possible to assess the operator performances during the simulation, in order to have a better evaluation of the training efficacy. This thesis work lays the basis for the 3D immersive simulation realizing of a real application, producing the 3D model and the dynamic simulation of the *AG2S* (Acid Gas To Syngas) lab scale process, and analyzing the potential of MOSAIC simulation environment (realized at TU Berlin University) as HUB of the virtualize process. The *AG2S* process wants to produce Syngas, combustible gas commercially valuable, starting from acid gases like CO_2 and H_2S, tail gases from the desulphurization of coal Syngas, potentially dangerous and/or toxic for humans and environment. The choice of the lab scale case of study symbolizes even the possibility to perform the 3D immersive simulations not only for the industrial operators, but also for the laboratory operators training.

1. IMMERSIVE DYNAMIC SIMULATION: instruments and goals

The final goal of this project is to realize an Immersive Dynamic Simulation of a chemical process. An immersive simulation of the process can be useful to reproduce the operators training in a virtual 3D environment. The potential of the immersive simulation is to reproduce an environment that gives similar ‘sensations’ of the real plant to the operators, in order to realize a training where they feel ‘immersed’ exactly in the plant.

With an immersive 3D training the operators could experience different situations as they are in the plant, but with the added values of complete safety and perfect reproducibility of the simulation. With this kind of training, the operators could be tested in all the possible scenarios of the plant, including also accidents with a proper simulator. Besides, it could be also possible to assess the operator performances during the simulation, in order to have a better evaluation of the training efficacy.

This work is focused on the creation of a first example of Immersive Dynamic Simulation. For this purpose a chemical process with a new experimental laboratory at the ‘Politecnico di Milano’ university is selected to be reproduced in the immersive 3D environment. The selected process is the *AG2S* (Acid Gas To Syngas), licensed by professor Flavio Manenti.

In this first chapter a brief introduction of the principal components to be used in order to realize the immersive dynamic simulation is presented.

1.1 HOW TO CREATE AN IMMERSIVE DYNAMIC SIMULATION

An immersive dynamic simulation is the result of a very complex harmonization between different instruments:

- 3D reproduction of the plant. This is the starting model which contains all the equipment in their real position in plant and with the same dimensions and engineering characteristics like temperature and mechanical resistance or construction material.
- Process Dynamic Simulation. The plant has to be dynamically simulated to reproduce, during the operator training simulation, its real behavior depending on the operator’s actions.
- 3D virtualization of the process. Thanks to specific software the 3D model and the dynamic simulation are used as inputs to ‘virtualize’ the plant. In this phase all the textures and lighting effects to the 3D are applied to produce a good immersive effect of the virtual plant.
- Generalizing language software. This work also tries to generalize the input that 3D virtualization software has to receive. As regards 3D plant models, there is a worldwide usage (more than 80%) of Intergraph® Smart3D; but in dynamic simulations contest, there is not prevailing software. This is why it could be very useful putting an ‘interface’ environment between the dynamic simulation software and the virtualizer which is able to ‘generalize’ the output of the dynamic simulation to produce a standard input for the 3D virtualization of the process, whatever the dynamic simulation software used.

This master thesis work is focused on the creation of all the elements needed for the 3D virtualization of the real application chosen, the *AG2S*™ process, creating the 3D plant reproduction and the process dynamic simulation; besides, in the last part of the study one possible way to generalize the dynamic simulation inputs to virtualization software is presented.

The most important software used in the thesis project are listed and described in the following.

1.1.1 Intergraph® Smart3D

The 3D model of the plant is realized with Intergraph® Smart3D.

This program is released on 2014, from the consolidation of SmartPlant® 3D, SmartMarine® 3D and SmartPlant® 3D Materials Handling Edition, and it’s a design software specifically tailored for plant, offshore, shipbuilding metals, mining and bulk material handling industries. It employs an engineering approach of the 3D model: every equipment, slab, piece of architecture has not only dimension characteristic, but even other engineering proprieties, like construction material, temperature and pressure resistance.

Every piece of the plant has a collocation in the huge database provided by Smart® 3D, in which there are piping, reactors, heat exchangers, typical chemical process equipment, and all the civil components used to produce the substructure of the plant, like girders, beams, columns, electric components, and also some choices of architectural buildings.

Intergraph® Smart 3D enables also to introduce new equipment in database if they are not already present.

1.1.2 Simsci™ Dynsim

The Dynamic Simulation of this work is made with Simsci Dynsim.

Dynsim is a dynamic process simulator that enables to design and operate a process plant including the control system design and improve process yield reducing the capital investment costs. It includes a big database for chemical compounds that could be even expanded, and models for the most important chemical process utilities, like reactors, heat exchangers, valves and process controls.

With Dynsim it’s possible to simulate the dynamic behavior of the plant, from the turn on to the shut off, with a manual or an automated control system. This software is able to calculate the behavior of fluids in piping, thermodynamical equilibrium in flashes, rate of reactions in PFR, CSTR and non-ideal reactors, it can simulate also furnaces, distillation columns and other complex equipment.

1.1.3 MOSAIC

Mosaic is the program chosen to try to convert the output of whatever dynamic simulator into a standard input to the virtualization software.

Mosaic is a modeling, simulation and optimization environment. Based on a LaTeX-style entry method for algebraic and differential equations, systems can be built and used for simulations. Besides, Mosaic provides an automatic code generation for numerous simulation and optimization environment, like Matlab, Ampl, Modelica, gProms, and solvers interfaced via C++, Fortran, Python and others.

Thanks to its ability to convert languages and its open structure to model systems of equations, Mosaic could be able to receive an output from the dynamic simulator and convert it in a generalized input for the virtualization software.

The general overview of the project is illustrated; in the next chapter few considerations are reported to better explain the thesis work objective but also to introduce the future completions to this first step toward the final goal.

2. THESIS OBJECTIVES AND FUTURE STEPS

The previous chapter presents the main instruments used in this thesis work, excluding with DSmoke which will be described in the following chapters. At the beginning of the thesis project the results to achieve have been fixed; the most important results to reach were the 3D modelling of the studied process, the dynamic simulation complete of a new simplified kinetic scheme for the reactions considered, and finally a deep analysis on the MOSAIC simulation environment and the study of its potential in the Immersive Dynamic Simulation project. All these goals are reached during the long path of this thesis work, with alternation of successes and failures, unexpected problems and sudden great progress. The final result of all the efforts and gratifications of this master degree thesis is presented in the next chapters. The chosen order of description reflects the effective chronology in which the different arguments have been faced: the first section of the thesis regards the 3D model of the process, in the second section all the kinetic regression and dynamic simulation problems are addressed, and in the last section the study of MOSAIC environment is reported using the *AG2S™* process as an example to show the most important characteristics of this program. In conclusion, the thread which connects all arguments and instruments used in this thesis work, very different each other, is exactly the *AG2S™* process, which acts as useful example to show all the most important characteristics of the software used, and provides the first real challenge for the Immersive Dynamic Simulation project, which could embrace not only the industrial chemical processes, but also the lab world, in a completely new perspective of which this thesis work is the perfect symbol. In fact, the usage of industrial software i.e. Intergraph® Smart 3D or Simsci Dynsim to perform a laboratory scale 3D model and dynamic simulation has been a big challenge, but the potential of Immersive Dynamic Simulation in the lab scale is as big as in the industrial scale, because the possibility to train the operators before going the laboratory is a real need, which in the university world is settle with the usage of global safety trainings; lots of times the training for the specific activity to do in the laboratory is very lacking, and it needs more consideration. Using the Immersive Dynamic Simulation for the laboratory activities means to ensure the safety and the right working of the operators, training them for their specific mansions and certifying the learning level before allowing the access in the work place, at the cost of producing the engineer 3D model and the dynamic simulation of the laboratory. The same concept can be extended to the industrial work, where even more attention is taken on the safety and performances of workers, and where industrial simulators and 3D models are still available to be used for the Immersive Dynamic Simulation, so the problem is focused mostly on the standardization of the simulator output and on the connection with the virtualizer. This thesis work has the presumption to guarantee the prerequisites for finding the right solutions in both the situations, with a double focus on the industrial scale and the lab scale. The perfect representation of the double valence of this project is the usage of industrial engineer software like Intergraph® Smart 3D and Simsci Dynsim to reproduce a lab scale process like the *AG2S*™ process. In this way it's possible to verify the difficulties and opportunities in the 3D immersive dynamic simulation in both the scenarios, as in the industrial case the need to standardize the output of dynamic simulation trying to use MOSAIC environment, but also the lab scale problems like the actual 3D model restrictions in terms of dimensions and availability of lab scale process components.

Despite the large number of topics addressed in this thesis work, the Immersive Dynamic Simulation project has to face lots of other problems; the first problem is the visual rendering of the 3D model in order to create a realistic texturing and lighting of the components. This operation can be made only after the 3D model creation, and probably inside the virtualizer in order to enable a dynamic changing of lighting and shadows depending on the operator position; at this point this kind of problems are not considered, but in the development of the project this will be a crucial aspect to guarantee the operator immersion in the simulation space, and people with other skills are needed to reach a good result. The 3D environment with the Smart3D structure has to be rendered in order to obtain realistic coloring, shading and reflection effects of the equipment; texturing must be accurate and reflect the real conditions of the components, including fouling and rusting, because the operator must perceive the immersive environment like the real one, with all its imperfections. This is necessary because the worker is trained to specific operations to be done in the workplace, and must associate the actions done during the immersive training as the real acts to do subsequently in the real plant, and the effectiveness of that association is strongly dependent on the correspondence of not only shape, dimensions and position of equipment, but also their textures.

Another problem to be faced hereafter is the linking between all the different software necessary to perform the immersive simulation. The connection between the 3D model and the dynamic simulation must be the task of the virtualizer, but how to send the information from this software and receive and manage them into the virtualizer is another important future object of study. The integration with MOSAIC can be a solution to standardize the output of the dynamic simulation, but even in this case the way to send and receive information must be deepen. In the MOSAIC section of this thesis work the potential but also the limits of this simulation environment will be presented, and it’s possible to anticipate already in this introductory chapter that the biggest problem at now it’s the real-time communication in MOSAIC, which is not expected in the current version of the software. Even without the mediation of MOSAIC, the connecting process between different software preserves its intrinsic difficulty, and people with specific informatics skills are required to get it possible.

The greatest charm and complication of the 3D Dynamic Immersive Simulation project is the necessity to really high and various competences, which go from the engineering world with dynamic simulation, 3D model and language flexibility knowledge, to 3D rendering and texturing, informatic design and data exchange complexity which are designer, architectural and informatic competences.

The current state of the project is focused on the engineering contributes, which are the preliminary necessary steps to provide all the required data to start including the other competences. In this case the engineering work is the fundamental substrate over which building the artifices to reach the final goal. But, even within only the engineering contribute of the project, the competences required are really various; the 3D model production, a completely new skill to be learned, the dynamic simulation of the process flanked to the generation of a regression code using a mathematic library, and finally the discovery of a totally new simulation environment, with specific language and model construction rules, are an incredible challenge to be performed by a single person, but also an amazing personal enrichment. The possibility to see the entire process with different points of view has been a beautiful opportunity which enables to further understand the complexity and beauty of the engineering world. Even if the huge amount of the new competences to be learned seemed prohibitive, each difficulty has been overcome step by step, and the thesis work results are very satisfactory.

These considerations conclude the introduction section of this work; in the next chapters the whole thesis work is presented and depth, from the beginning to the final results, starting from the 3D modeling of the process.

3. 3D MODELING: INTERGRAPH® SMART 3D

As written in the introduction, Intergraph® Smart 3D is the software chosen to produce the 3D modeling of the laboratory. The choice was made because of the ‘engineering approach’ of the program, with the possibility to insert not only geometric proprieties of the process components, but even lots of different characteristics which are representative of the equipment and which can be useful in the virtualizing phase. In the next chapter the principal components of the Smart 3D workspace are presented.

3.1 SMART 3D WORKSPACE

The workspace represents the model data. Smart 3D operates through ‘tasks’, which can be selected in the home screen.

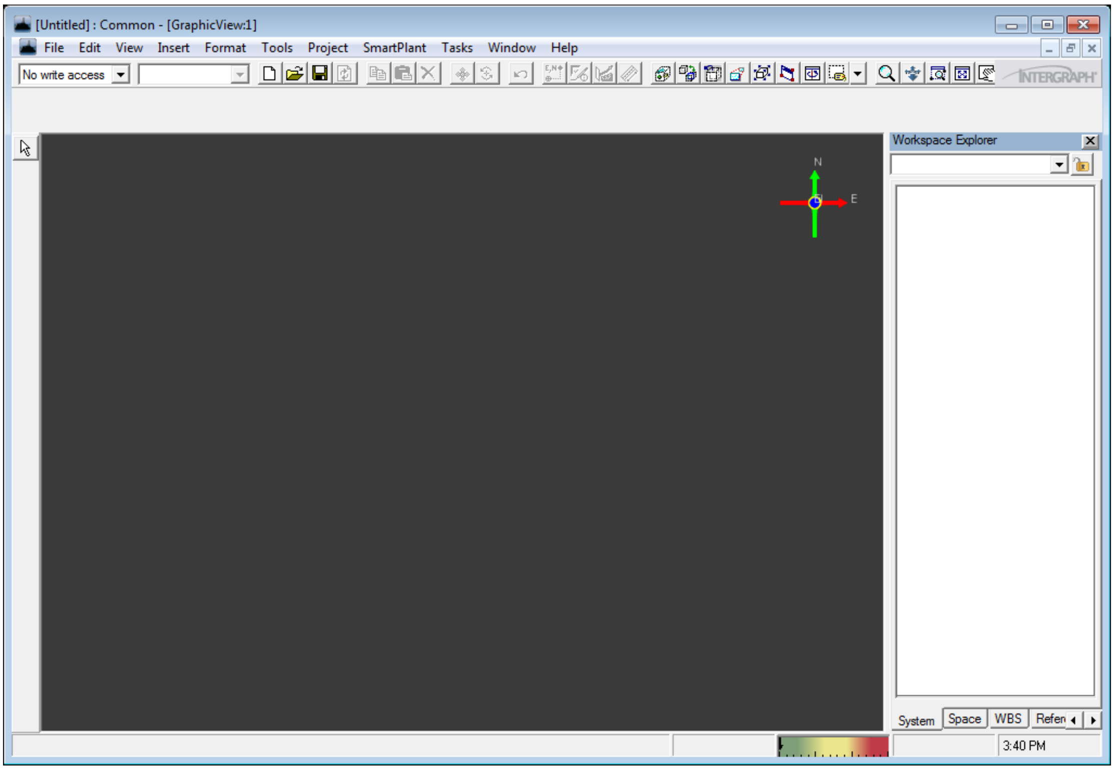


Figure 1 – Smart3D home screen

Different ‘tasks’ operates with different parts of the plant. The main tasks used in this work are:

- grid
- piping
- equipment
- structural system

3.1.1 Task grid

During the construction of the 3D model, there is the need of a reference system; this could be global (which follows original coordinate system) or relative (following a coordinate system determinate by the user).

For this scope the grid task is useful to create reference planes in the workspace. In this way it’s easier to place the objects in the right position, using a local reference instead of the global which can be uncomfortable in lots of cases.

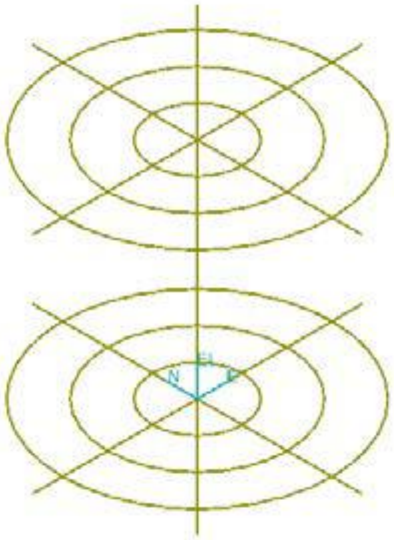


Figure 2 – radial grid

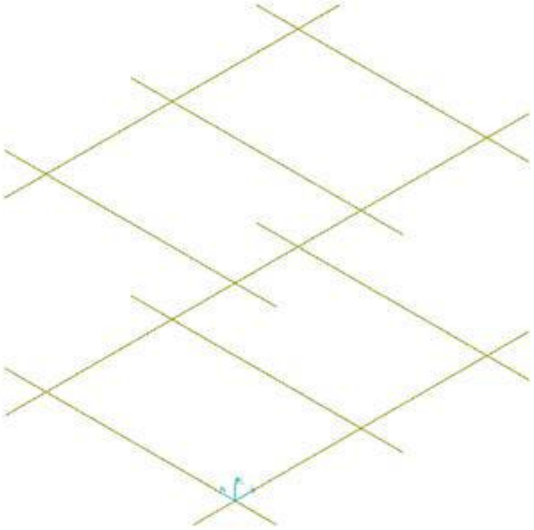


Figure 3 – rectangular grid

In this work grids are used to define the three planes of the chemical laboratory, i.e. the main plane with the fume hood base, and the two planes which support fluximeters and micro gc.

3.1.2 PinPoint

Another very important instrument of Smart 3D to change the local coordinate system is PinPoint command.

This command enables to select every point of the workspace defining it as the origin of new relative coordinate system. As Smart 3D recognizes crossing point between objects, axes of the structures and middle point of the planes, it's possible to choose one of these crucial points as the PinPoint. In this way it's possible to create complex equipment or architecture with beams in an easy way.

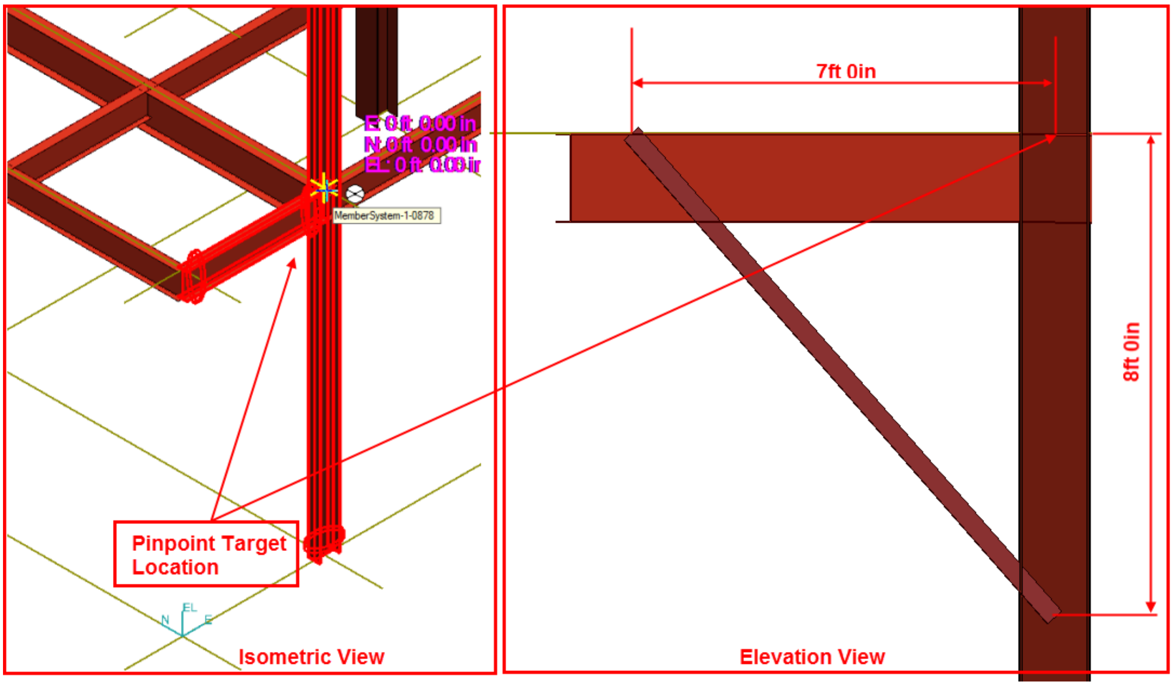
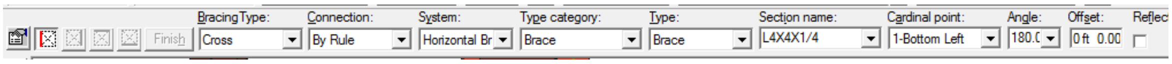


Figure 4 – example of PinPoint command usage to create a complex beam structure

3.1.3 Structural System task

The structural system task is dedicated to the creation of slabs, walls, columns, beams, and all the others objects with structural functions.

Smart 3D database contains many types of construction components, differentiating by section shape, dimensions and material proprieties with European and ASTM standards.



Bracing Type: Cross
Connection: By Rule
System: Horizontal Br
Type category: Brace
Type: Brace
Section name: L4X4X1/4
Cardinal point: 1-Bottom Left
Angle: 180.0
Offset: 0 ft 0.00
Reflec ☐

Figure 5 - structural system editor bar

With structural system task, it’s possible to create all the civil components of the plant. It is also possible to connect or split axis, to insert stairs and railings, in order to reproduce the real supporting system.

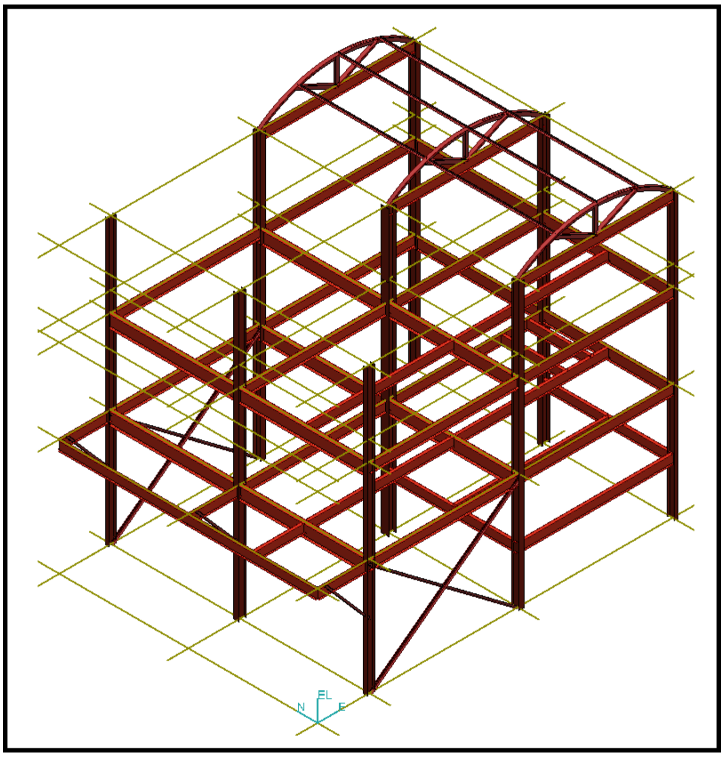


Figure 6 – example of structural system application with the presence of grids.

3.1.4 Equipment task

The equipment task is able to reproduce the chemical process components of the plant; in particular it models reactors, heat exchangers, pumps and compressors, distillation columns, tanks and vessels, but also safety showers and electrical or architectural components.

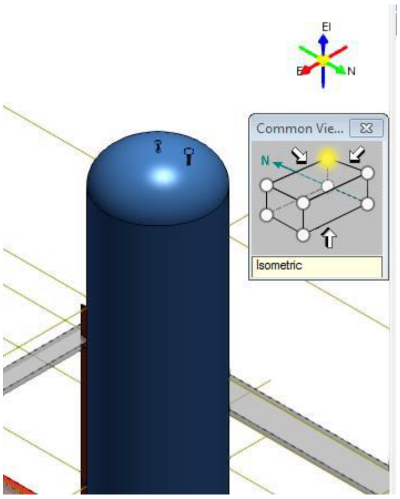
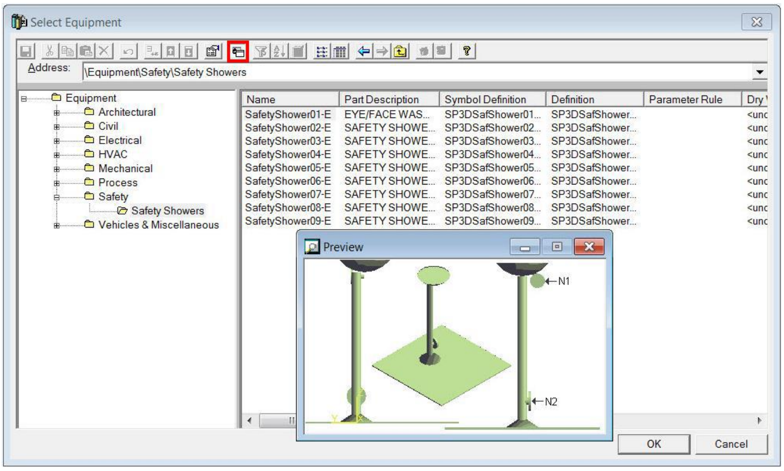


Figure 7 – example of vertical vessel equipment

It's possible to use defined equipment (symbol), which is present in database; to help to choose the right designed equipment from the many possibilities available in the system database, it's also provided a preview that contains a schematic illustration of the equipment reporting its most important components with an identifying symbol.



Address: \Equipment\Safety\Safety Showers

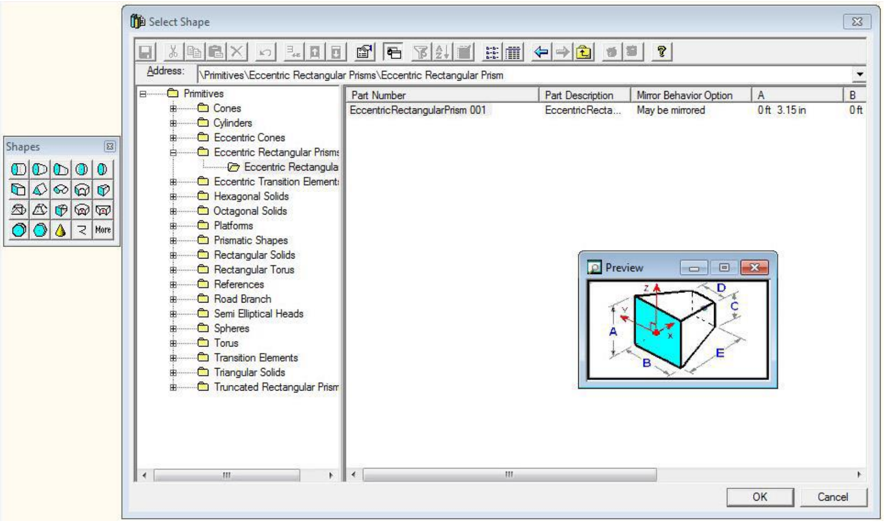
Name	Part Description	Symbol Definition	Definition	Parameter Rule	Dry
SafetyShower01-E	EYE/FACE WAS...	SP3DSafShower01...	SP3DSafShower...		<unc
SafetyShower02-E	SAFETY SHOWE...	SP3DSafShower02...	SP3DSafShower...		<unc
SafetyShower03-E	SAFETY SHOWE...	SP3DSafShower03...	SP3DSafShower...		<unc
SafetyShower04-E	SAFETY SHOWE...	SP3DSafShower04...	SP3DSafShower...		<unc
SafetyShower05-E	SAFETY SHOWE...	SP3DSafShower05...	SP3DSafShower...		<unc
SafetyShower06-E	SAFETY SHOWE...	SP3DSafShower06...	SP3DSafShower...		<unc
SafetyShower07-E	SAFETY SHOWE...	SP3DSafShower07...	SP3DSafShower...		<unc
SafetyShower08-E	SAFETY SHOWE...	SP3DSafShower08...	SP3DSafShower...		<unc
SafetyShower09-E	SAFETY SHOWE...	SP3DSafShower09...	SP3DSafShower...		<unc

Figure 8 – example of designed equipment with preview

It's often possible to modify dimensions of the designed equipment to create it like the real one, but this type of modeling has some restriction. In fact every nozzle dimension, length of a piece of equipment or diameter of the component, has to respect the limiting minimum and maximum programmed values.

For these reasons, in the equipment task it's also provided a 'place not designed equipment' section (symbol). Even in this case, the choice of the equipment in the database is mandatory, but it's only a way of the system to assign the non-designed component in a subgroup, for example in this thesis work we chose 'complex horizontal vessel' as subgroup of the laboratory reactor.

After the choice of the non-designed equipment subgroup, it's possible to start modeling the equipment. To produce a highly customized solid different shapes are available (symbol); they can be added or subtracted in a global solid (symbol) which could represent the entire equipment or only a part of it. Also for the available shapes there is a special catalogue in which is present their preview.



Address: \Primitives\Eccentric Rectangular Prisms\Eccentric Rectangular Prism

Part Number	Part Description	Mirror Behavior Option	A	B
EccentricRectangularPrism 001	EccentricRecta...	May be mirrored	0 ft 3.15 in	0 ft

Figure 9 – shapes catalog with preview

Differently from the designed equipment, shapes are completely modifiable in all their dimensions without any limiting value (but they must be real dimensions, negative numbers are not allowed). For some of the shapes present in the database, it's also possible to modify angles between sides to produce particular piece of equipment; during the laboratory modeling this special characteristic is used in particular pieces of the reactor.

Shape can be added (symbol) or subtracted (symbol) using the 'place a solid' tool, which is the way of the system to recognize as a unique solid the combination of single shapes. This type of system structure leads to the fact that shapes can have only dimension characteristic, while material proprieties, T and P resistance and all the others parameters are associated to the solid. This could be a problem when equipment is composed by pieces with different proprieties, like for example the tube furnace of this thesis work. In that case, it's necessary to produce different solids for the metallic base of the furnace and the refractory tube.

Thus, producing equipment with 'place a solid' tool enables to model a totally customized equipment, but it requires obviously much time in respect to 'place a designed equipment tool'.

3.1.5 Piping task

The piping task enables to create in the 3D model all the components concerning pipelines, i.e. not only piping but even valves, flanges, taps, nozzles and special components like fluximeters.

In the following the actions that can be performed in the Piping environment are presented:

- *Create and route a pipe run:* it's possible to create pipe run, extend an existing one, and route a pipe run to or from nozzles and equipment inlets or outlets;

- *Insert splits:* it's possible to insert splits into the pipeline. It could be useful to divide a pipe run into sections by placing a set of flanges, a union component, or to move one block of the pipeline without modifying the other parts.

- *Insert pipe components and custom instruments or specialty items:* it's possible to insert pipe components to create sophisticated pipe layouts that divide, branch and convey fluids between equipment. This is the case of fluximeters used in the laboratory modelling.

- *Add taps on pipe components:* it's possible to add taps to standard components such as caps, instruments, orifice flanges, and valves.

Besides pipelines can be characterized by lots of proprieties:

- Design and Operating Maximum and Minimum Temperature
- Design and Operating Maximum and Minimum Pressure
- Surface Treatment and Coating: interior and exterior surface treatment, cleaning requirement, auxiliary treatment requirement, interior and exterior coating area and type, coating color
- Insulation and tracing: insulation purpose, material, thickness, temperature, tracing requirement, type
- Responsibility: party responsible for cleaning, design, fabrication, installation, painting, requisition, supply, delivering, testing on

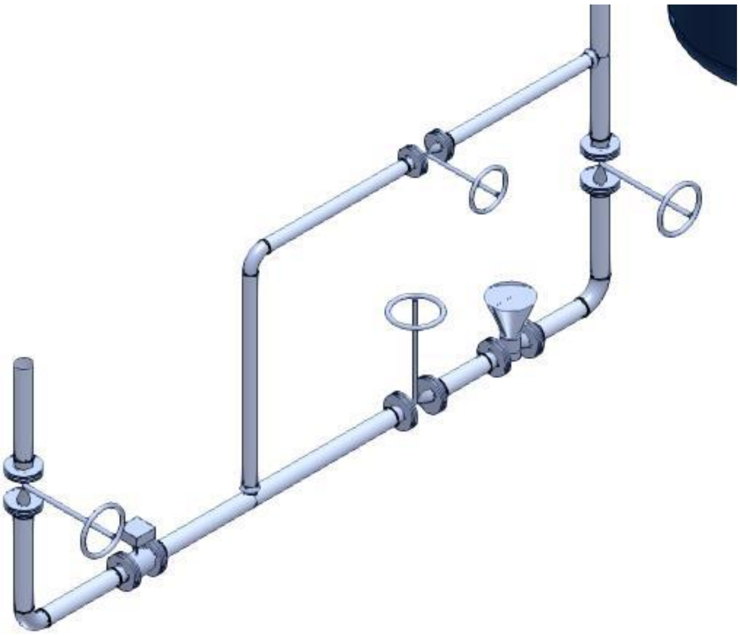


Figure 10 – complex piping example

3.2 SMART 3D MODELING OF THE LABORATORY

In this master thesis project Smart 3D is used to model the *AG2S*™ laboratory at the ‘Politecnico di Milano’ University.

The laboratory is an experimental unit for the *AG2S*™ process, which wants to produce syngas (CO and H_2) from tail gases containing H_2S and CO_2, undesirable products. A further analysis of the process is presented in the Dynamic Simulation Section.

3.2.1 Preliminary studies

The main components of the laboratory to be 3D modelled are:

Structural elements:

- all the support planes, specifically the fume hood base and the fluximeters and microgc supports;
- the wall behind the fume hood;

Equipment:

- base of the fluximeters;
- tubular furnace;
- glass reactor;
- microgc;
- metallic trap;
- flasks trap;

Piping:

- piping from the inlet of the fume hood to the fluximeter;
- piping from the fluximeter to the reactor;
- piping from the reactor to traps and to microgc;
- piping between traps;
- outlet piping which end in the fume hood top.

The 3D model is realized in collaboration with Intergraph® Company, the provider of Smart 3D software. The company organizes three tutoring sessions to help the laboratory 3D modeling. A preventively 2D CAD of the little plant is requested by Intergraph to provide specific help with the laboratory modelling. In the following the 2D CAD work realized for Intergraph is reported.

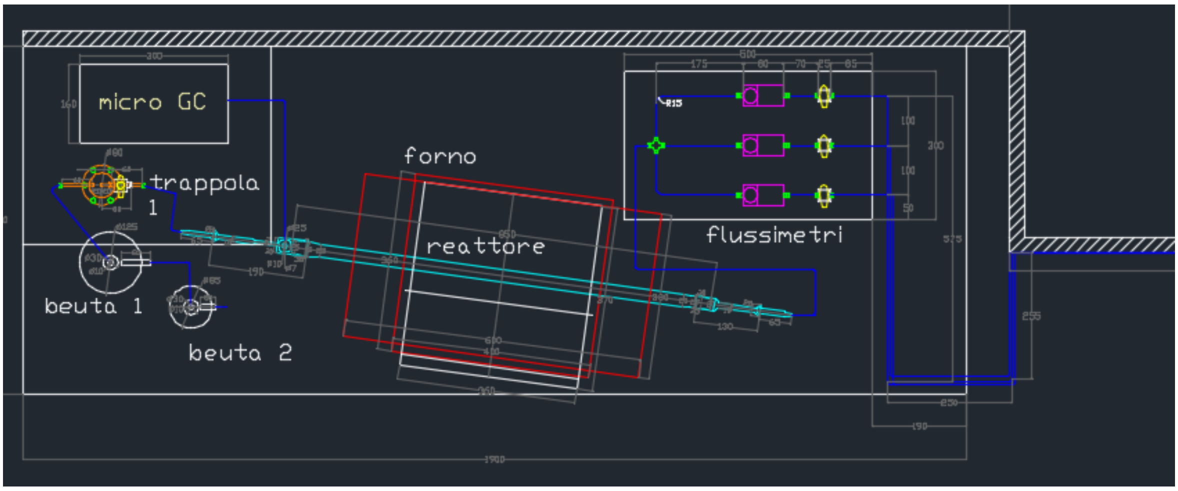


Figure 11 – plant of the laboratory realized in Autocad®

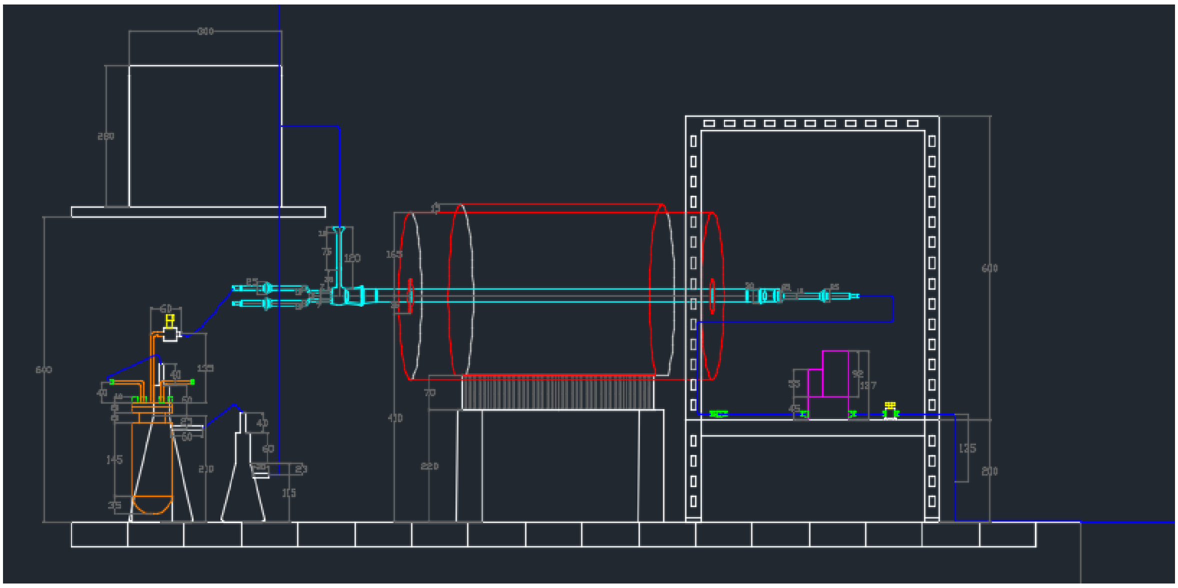


Figure 12 – frontal prospectus of the laboratory

It’s possible to see all the components listed previously. In particular the most complex pieces of equipment are reported with different colors to facilitate the comprehension of the CAD: fluximeters are realized in violet, the furnace in red, the first trap in orange and the reactor in light blue. All the pipelines are realized in dark blue while other components of the project are grey. It’s also possible to see all the quotes of the laboratory components, which are reported in [mm].

After the 2D CAD creation, the 3D modelling part can begin with a specific tutorial made by Intergraph.

3.2.2 Structural system realization

The first part of the 3D realization is dedicated to the structural scheme of the laboratory. The use of grids is useful to create three planes over which construct the slabs that represents the structural elements.

To create the elements, the 'Slab' command (symbol) is used, which enables to create horizontal supports with variable thickness and material proprieties. To position the slabs the grids previously created are used.

Also the wall behind the fume hood is created using 'Structural System' task. After the creation of grids and slabs, it's possible to choose a reference plane to create a wall of which position is linked to it using the 'Wall' command (symbol). This is perfect to produce the laboratory wall, because it must be placed in correspondence of the previously created slab. In the general overview section (3.2.5) it's better explained the way to reproduce walls in Intergraph® Smart 3D.

3.2.3 Equipment realization

After the structural system, equipment components are created. This is the main part of the work, because unfortunately Smart 3D database doesn't contain any equipment of the laboratory. In fact, Intergraph® Smart 3D is programmed for the creation of industrial plant 3D models, while in this work is used for a laboratory scale 3D modeling. In equipment section the designed components have a larger order of magnitude of 'Minimum and Maximum Values' to be assigned in respect to the laboratory real dimension; besides, the reactor section is highly customized to respect the experimental requirements. The only way to model equipment section too little and customized to be created with 'designed equipment' tool is to produce it using shapes and solids.

For every different diameter section of the reactor, it's necessary to create a couple of conical or cylindrical shape: the first shape represent the real section of the reactor, the second one has to be subtracted to create the effective void, otherwise the reactor would be like a full glass component without any correspondence to reality. Another difficult of the equipment, and in particular reactor, modelling, comes from its peculiar shape. The outlet part of the reactor is the most difficult to reproduce, because it contains very complex conical sections with rotation angles, the worst shape to reproduce in Smart 3D. It can be possible to notice them in the figures below:

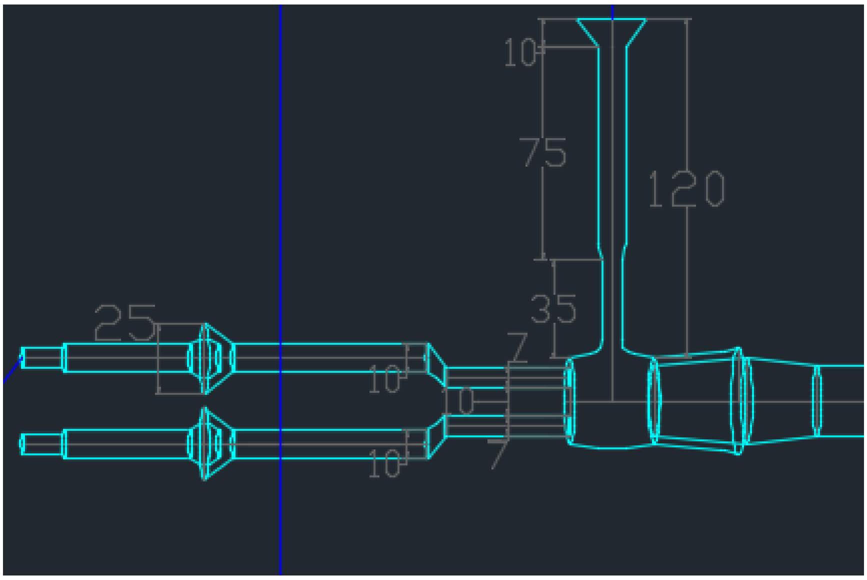


Figure 13 – particular of the reactor outlet in Autocad®

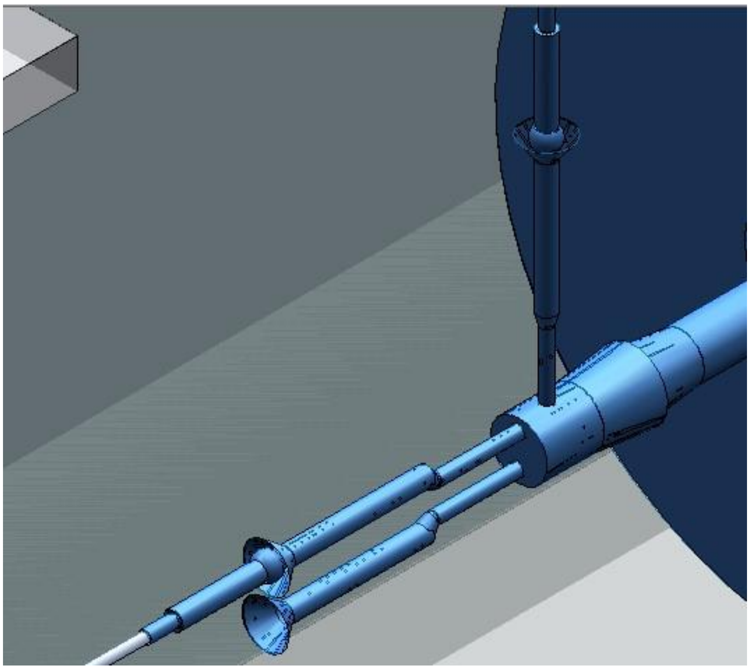


Figure 14- particular of the reactor outlet in Smart3D

In both the figures the outlet of the reactor is represented, including the junctions to the two pipelines exiting it: at the left, the pipeline that goes to the traps, on top the pipeline to the microgc analysis.

In the following figures the inlet part of reactor reproduced in AutoCAD and Intergraph® Smart 3D is reported. It’s possible to observe the same modelling difficulties of the outlet part, with lots of different diameters sized components which have to intersect themselves without interpenetration of corps. These specific fittings for the reactor is clearer in the AutoCAD™ representation, but they are perfectly reported even in the 3D model; in the virtualization software, it will be possible to ‘clothe’ the Smart 3D project with specific textures which can reproduce the reactor glass transparency. In this way, also the covered components in the reactor 3D model will be observable like in the AutoCAD™ project.

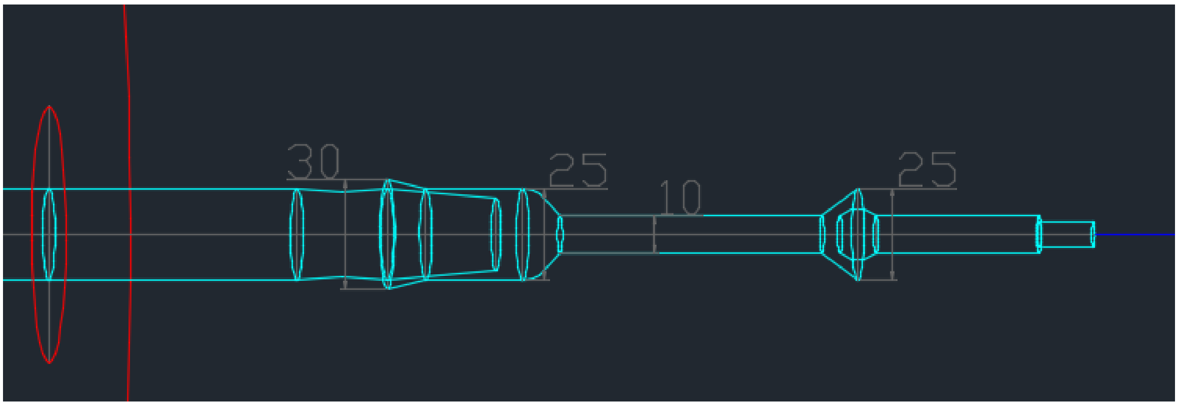


Figure 15 – particular of the reactor inlet in AutoCad™

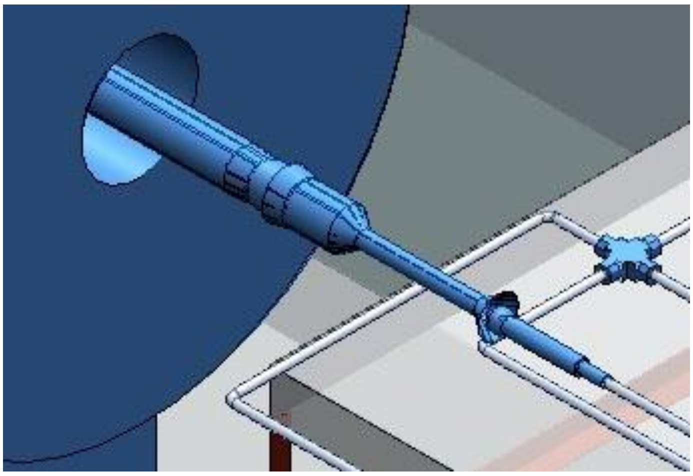
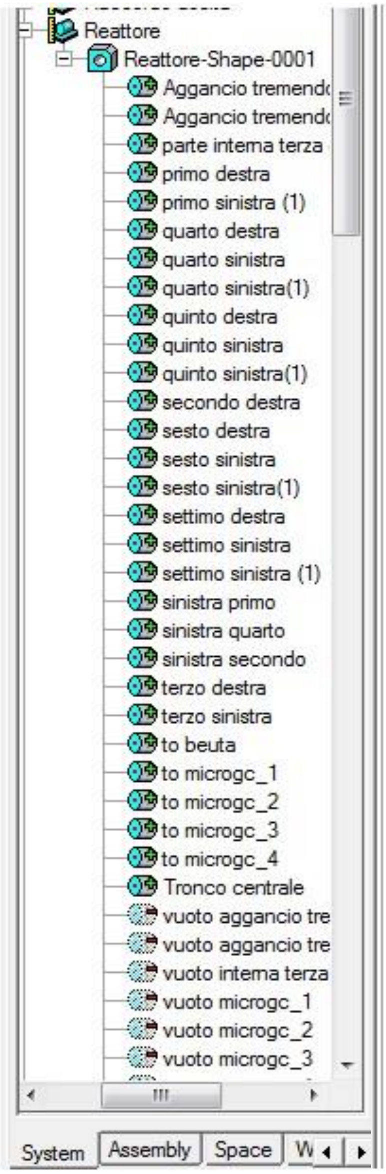
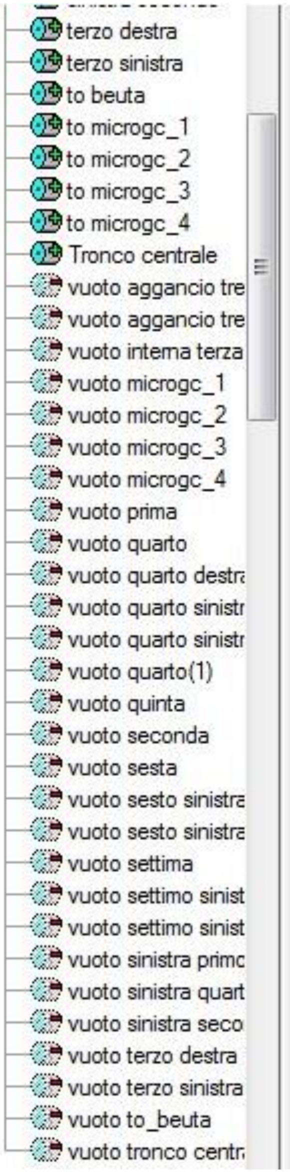


Figure 16 – particular of the reactor inlet in Smart 3D

Before passing to the trap equipment, it can be interesting to observe the model scheme database through the ‘workspace explorer’ tab. In particular in the figure below the model scheme database of the reactor is reported.



Reattore
Reattore-Shape-0001
Aggancio tremend
Aggancio tremend
parte interna terza
primo destra
primo sinistra (1)
quarto destra
quarto sinistra
quarto sinistra(1)
quinto destra
quinto sinistra
quinto sinistra(1)
secondo destra
sesto destra
sesto sinistra
sesto sinistra(1)
settimo destra
settimo sinistra
settimo sinistra (1)
sinistra primo
sinistra quarto
sinistra secondo
terzo destra
terzo sinistra
to beuta
to microgc_1
to microgc_2
to microgc_3
to microgc_4
Tronco centrale
vuoto aggancio tre
vuoto aggancio tre
vuoto interna terza
vuoto microgc_1
vuoto microgc_2
vuoto microgc_3



terzo destra
terzo sinistra
to beuta
to microgc_1
to microgc_2
to microgc_3
to microgc_4
Tronco centrale
vuoto aggancio tre
vuoto aggancio tre
vuoto interna terza
vuoto microgc_1
vuoto microgc_2
vuoto microgc_3
vuoto microgc_4
vuoto prima
vuoto quarto
vuoto quarto destr
vuoto quarto sinistr
vuoto quarto sinistr
vuoto quarto(1)
vuoto quinta
vuoto seconda
vuoto sesta
vuoto sesto sinistr
vuoto sesto sinistr
vuoto settima
vuoto settimo sinist
vuoto settimo sinist
vuoto sinistra primc
vuoto sinistra quart
vuoto sinistra seco
vuoto terzo destra
vuoto terzo sinistra
vuoto to_beuta
vuoto tronco centr

Figure 17 – workspace explorer: reactor

Figure 18 - workspace explorer: reactor_2

In these figures it’s easily recognizable the schematic structure of Smart 3D system. The first object to define is the ‘not designed equipment’, which in this case is: Reattore. Then it’s possible to create a solid, Reattore-Shape-0001, associated to the designed equipment. In this case it’s necessary only one solid because all the reactor components have the same proprieties. After the solid definition, shapes () can be created. Every shape can be added or subtracted from the solid to eventually create the reactor model; the plus or minus symbol near every shape describes the shape function within the solid. It’s interesting to observe that are necessary about 60 shapes only for the reactor model, a huge number derived from the highly customized form to be reproduced.

After the reactor realization other equipment has to be modelled. In particular, the most complex pieces to be created are the traps, which in the real laboratory are used to remove sulfur and H_2S residuals from the product flow. The first trap is a metallic cylindrical component with a tube section for the product flow, and a shell section for the cooling water that must lower the temperature of the process outlet. The reproduction of the first trap is illustrated in the figures below:

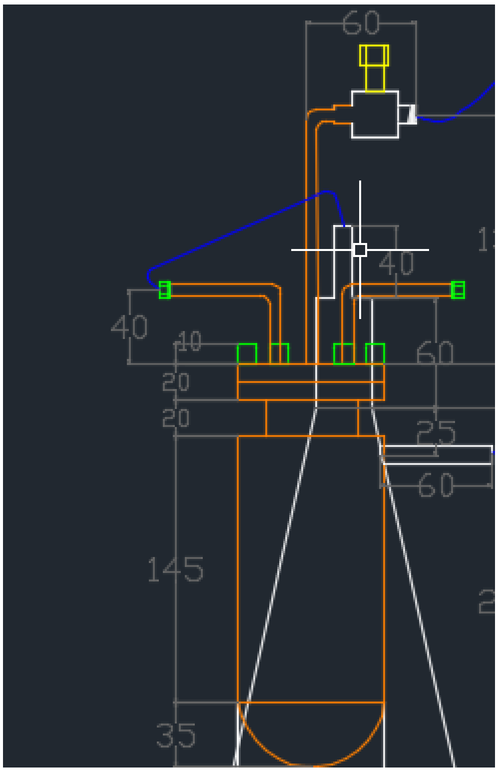


Figure 19 – metallic trap model with AutoCad™

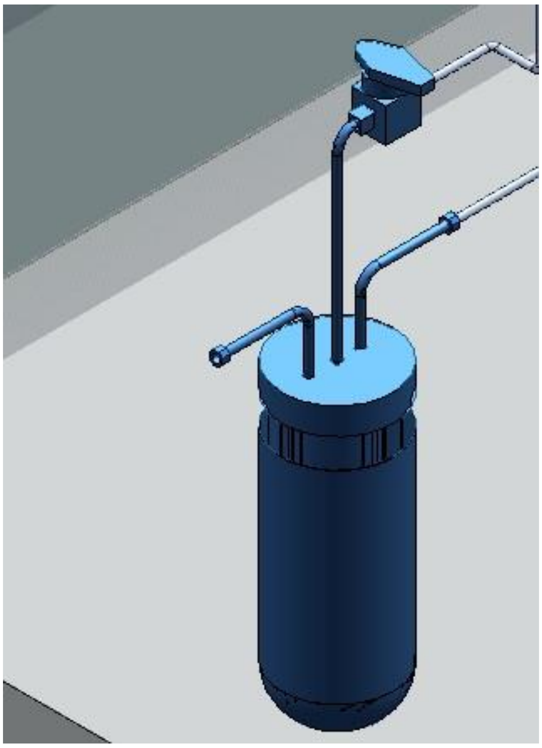


Figure 20 – metallic trap model with Smart 3D

It’s possible to distinguish, in particular in 3D model of the trap, the tube and shell sections realized with the adding and subtractions of different shapes to compose the concave part filled with water and the internal part for the process flow.

The other traps used in laboratory are simple flasks filled with a solution of NaOH and NaOCl used to separate H_2S by the product. The biggest one, which receives the outlet flow of the metallic trap, is reported in the following page. The other trap has the same construction modality but with different dimensions.

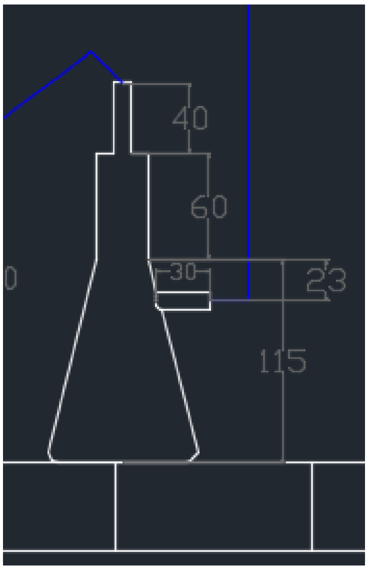


Figure 21 – flask trap model with AutoCad™

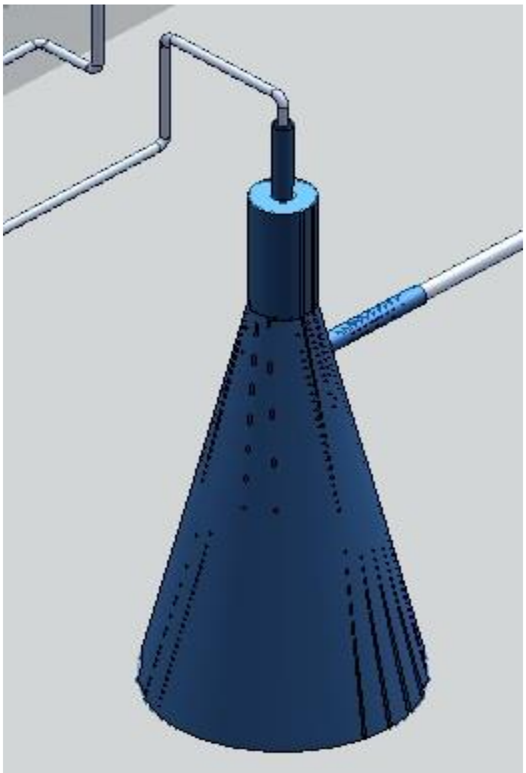


Figure 22 – flask trap model with Smart 3D

The equipment section is defined. There are other equipment components like the junctions of every piping, some little part of fluximeters and microgc, but they are created using few shapes and they can be observed in the final general view of the laboratory, without a specific dedicate section.

3.2.4 Piping realization

Piping is the final section to be produced. All the other components must be created and placed in their real positions; in this way it’s possible to create a realistic path of pipelines which should correspond to the real one.

Intergraph® Smart 3D, as seen before, provides dedicate piping section. It’s possible to produce pipelines with complex paths, choosing the right angle of curvature of tubes. The system can calculate if the imposed angle is feasible for the chosen pipeline, in particular in respect to its physical proprieties reported in database. If an unfeasible angle of curvature is chosen for the tube, Smart 3D suggests a welded section to reproduce the same trajectory of the original intent, but using three different tube parts in order to preserve the real feasibility of the 3D model.

Piping section is also used in this work to produce fluximeters, which are managed by the system as ‘special equipment’ linked to the pipelines. The link between piping and fluximeters is evident during the creation of these components. In fact, Smart 3D enables the creation of ‘special equipment’ only with the presence of a pipeline that can be connected to the fluximeter. It’s impossible to model a fluximeter without the creation of its referring pipeline.

In the laboratory 3D model, ‘Piping’ task is used to produce fluximeters that must control the process flowrate to the reactor, and all the pipelines associated to the fume hood, from the inlet of the fluximeters to the outlet to the top of the fume hood. In the case of AutoCAD™ project, it’s not necessary to model pipelines with their specific thickness, because that is only a preliminary work to decide together with Intergraph ® Company the best way to model the laboratory. Obviously it’s very important to model in AutoCAD all the equipment with their real dimensions and in their real positions.

In the figure below the fluximeters section and inlet reactor pipeline are reported. As previously mentioned, only Smart 3D pictures are illustrated.

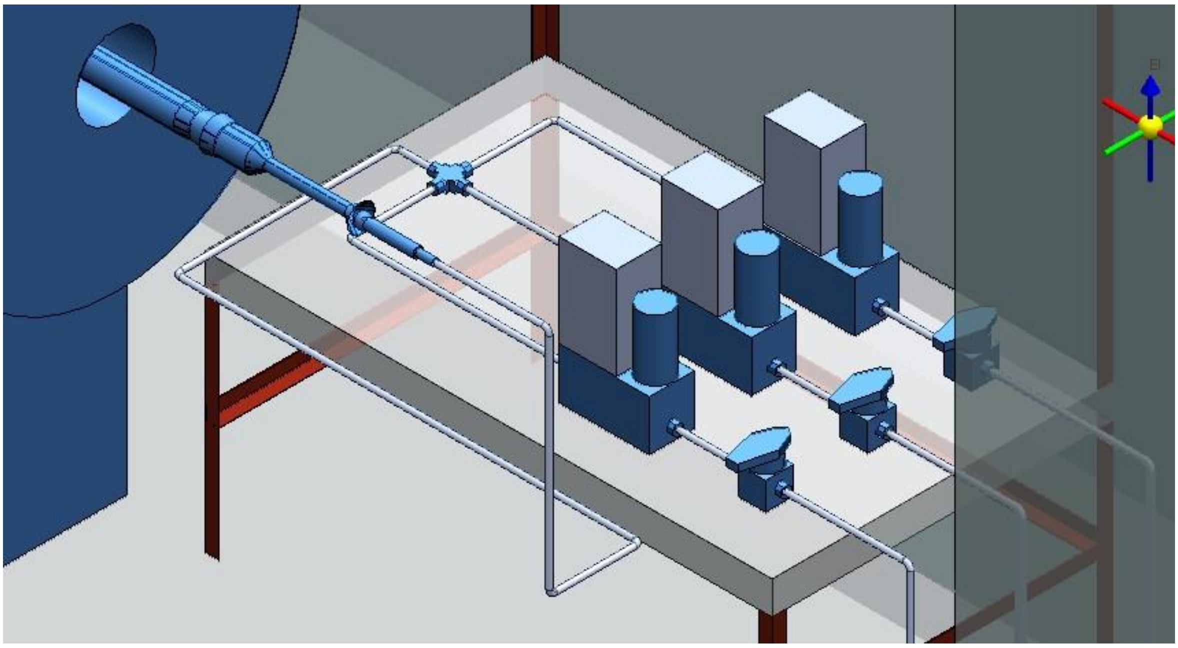
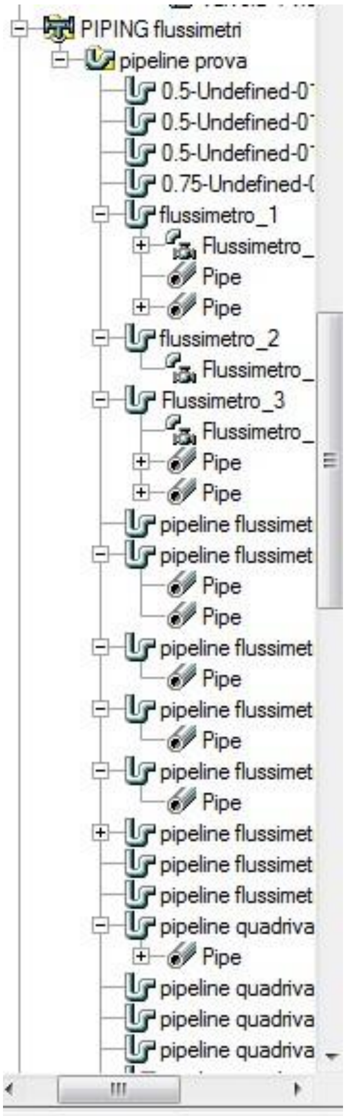


Figure 23 – fluximeters section and inlet of the reactor on Smart 3D

It’s possible to observe that, in this 3D project, different colors belong to different system sections. In fact, the red beams construction is produced with ‘Structural System Task’, while the grey pipelines and fluximeters are created with ‘Piping Task’; at last, the reactor, furnace, pipeline fittings, shut-off valves and also the support of the fluximeters are made by ‘Equipment Task’, so they’re represented in blue color. The only exceptions are the structural components like walls and slabs, which system produces in standard grey color.

It’s interesting to report a section of the workspace explorer of pipelines:



```
⊟ PIPING flussimetri
  ⊟ pipeline prova
      0.5-Undefined-0
      0.5-Undefined-0
      0.5-Undefined-0
      0.75-Undefined-(
    ⊟ flussimetro_1
      ⊞ Flussimetro_
        Pipe
      ⊞ Pipe
    ⊟ flussimetro_2
        Flussimetro_
    ⊟ Flussimetro_3
        Flussimetro_
      ⊞ Pipe
      ⊞ Pipe
      pipeline flussimet
    ⊟ pipeline flussimet
        Pipe
        Pipe
    ⊟ pipeline flussimet
        Pipe
    ⊟ pipeline flussimet
        Pipe
    ⊟ pipeline flussimet
        Pipe
    ⊞ pipeline flussimet
      pipeline flussimet
      pipeline flussimet
    ⊟ pipeline quadriva
      ⊞ Pipe
      pipeline quadriva
      pipeline quadriva
      pipeline quadriva
```

Figure 24 – workspace explorer: piping

In this way it’s possible to observe the system characterization of piping: in the main ‘Piping Flussimetri’, all the pipelines that go from and to fluximeters are included. Every pipeline is then decomposed to the single pipes that form the entire object. In this way Smart 3D is able to produce automatically pipelines with specific curvatures, assembling single pipes which compose the final result.

It’s also interesting to observe that, as said before, fluximeters are considered as a special piping object. In the figure above the three fluximeters ‘pipelines’ are reported: they consist of the special object (symbol), and the two pipes which are connected with the fluximeter.

3.2.5 3D model complete

With the piping realization the 3D model of the laboratory, as regards the fume hood, is complete. The environment around the fume hood, which includes the laboratory room, the external structure of the fume hood and all the elements linked to them, will be created in collaboration with Architectural Department of Politecnico di Milano using laser scanning technique to produce a points cloud. This cloud can be imported in Intergraph® Smart 3D to produce a substructure of the environment which enables to easily reproduce it in the workspace.

In this section it’s possible to appreciate the overview of the laboratory 3D model created on Intergraph Smart 3D.

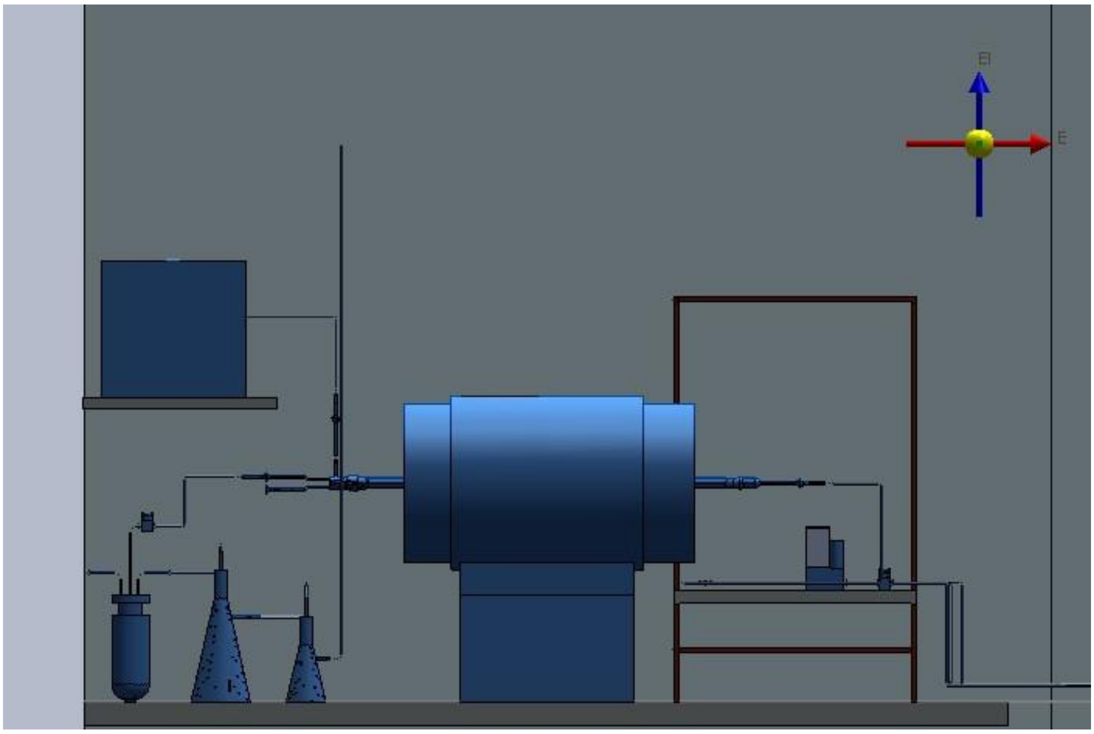


Figure 25 – front panoramic overview of 3D laboratory

In this front overview are visible the most important components of the laboratory. From the left to the right there is the microgc, under it there are the three traps, then the reactor, furnace and finally the fluximeters over their structural support. It’ also observable the ending pipeline, which starts from the last trap and go to the top of the fume hood, where the outlet flows can be aspirated.

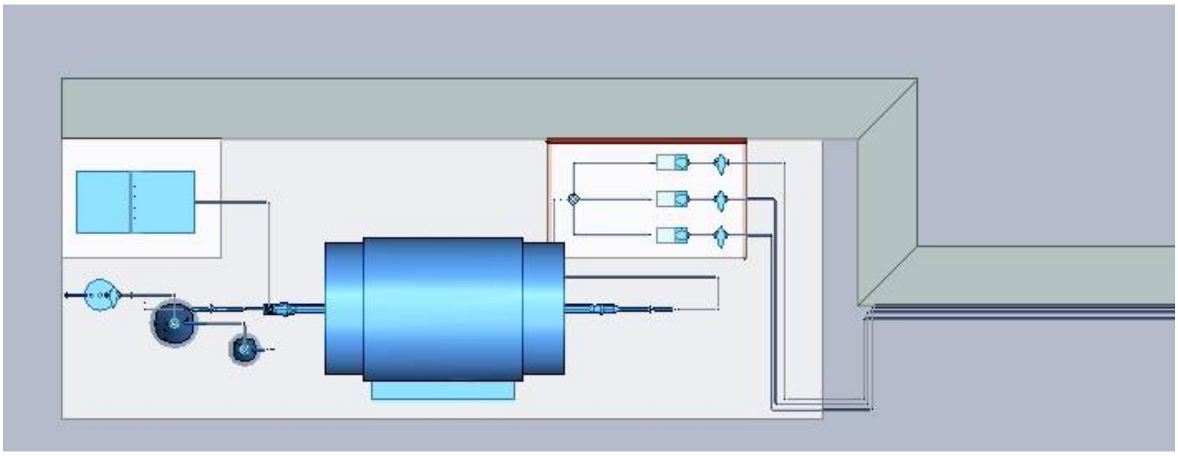


Figure 26 – Top panoramic overview of the laboratory on Smart 3D

In the top overview of the 3D model the positioning of the laboratory components is better observable. It's also possible to recognize the three different pipelines of H_2S, CO_2 and N_2, entering the fume hood and going to the fluximeters. In the panoramic overview is also highlighted the complex placing of all the equipment, due to the necessity to use an already available fume hood in the laboratory room chosen for the experimentation; in fact in this picture it seems that furnace interpenetrates the fluximeters structure, but it's possible to see in the front overview and even in the following pictures that furnace is above the fluximeters table without interpenetration problems. Because of the fact that laboratory is not completely assembled during the AutoCAD and 3D production, one of the crucial aspect of this work is to discuss and realize the positioning of all the laboratory components. This is the final result of the preliminary study, which satisfies space optimization and safety criteria.

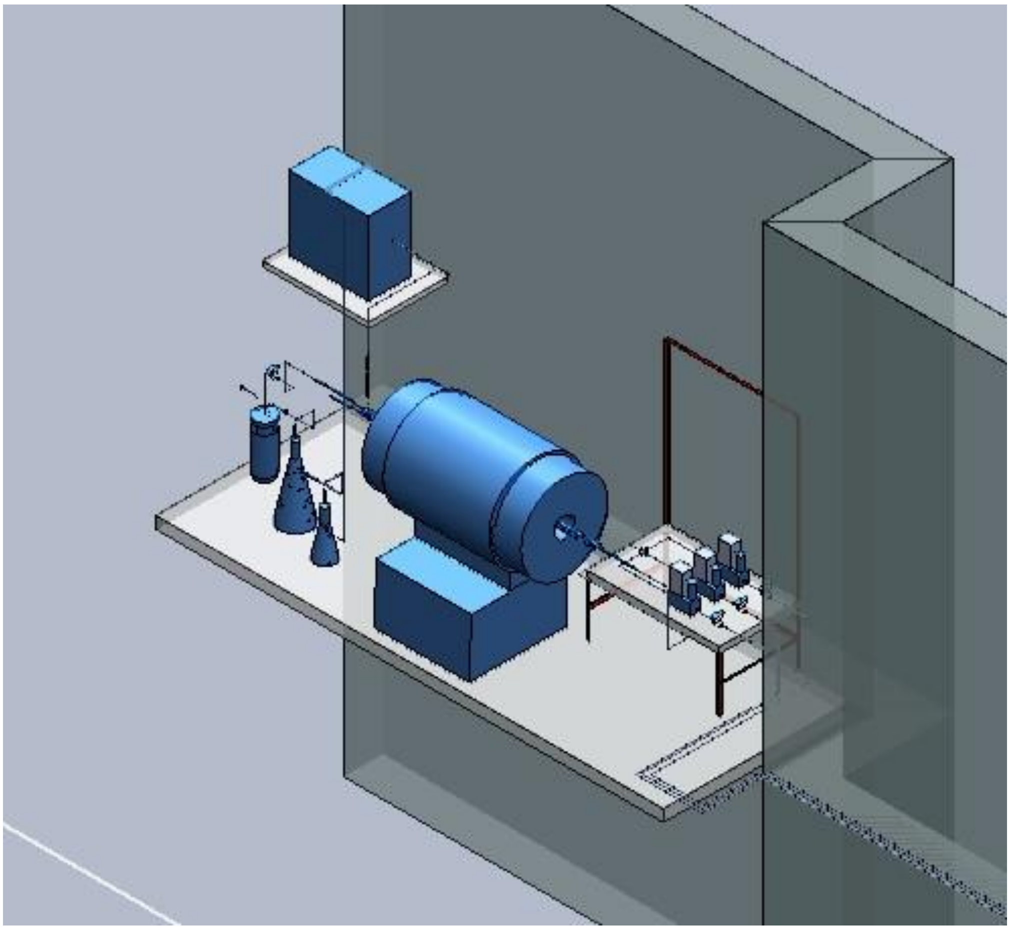


Figure 27 – right panoramic overview of the laboratory on Smart 3D

In the right panoramic overview is more evident the three-dimensionality of the model. Besides, it’s possible to better observe the wall behind the fume hood: because of the Intergraph® Smart 3D options for wall structures, it’s allowed to place a wall only under, above or in the middle of the reference plane chosen. In this case the reference plane is the base of the laboratory, and the wall has different heights in the below and above sections in respect to the table. In fact, as seen in the picture, the above section of the wall is taller than the below section. For this reason, to produce this part of structural system, a unique wall component isn’t sufficient, because none of the three alternatives to place the structural part is valid: choosing under or above the plane configuration, the entire wall would be placed in a wrong way, but also with the middle plane configuration, real dimensions would not be respected. The only way to reproduce faithfully the wall on Smart 3D is creating two wall parts, the first placed below the reference plane, and the second above it. Observing with attention the figure 25 is recognizable the particular structure of the wall.

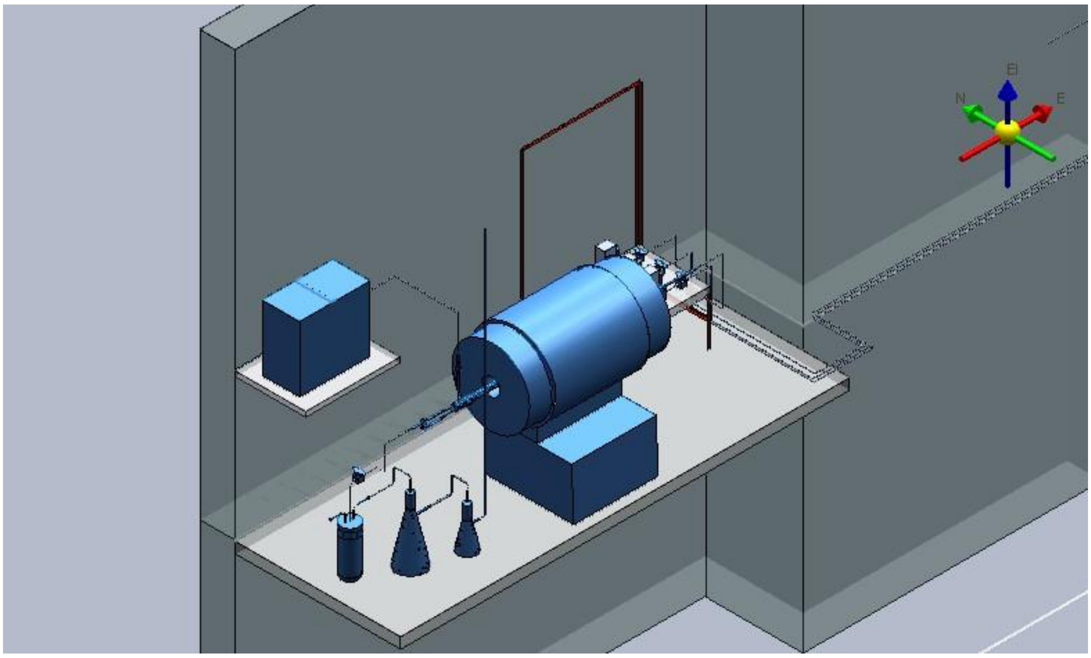


Figure 28 – left panoramic overview of the laboratory on Mosaic

At the end, the left panoramic overview is presented.

It results very similar to the previous one, but it’s better to appreciate the exiting section of the reactor and the three traps linked with piping.

4. BEFORE PROCESS SIMULATION: REACTION KINETIC

4.1 *AG2S* PROCESS DESCRIPTION

At this point, only the laboratory structure is presented because of its importance in the 3D modelling. Before approaching the dynamic simulation, it's instead necessary to deepen the description of the *AG2S*™ process, and in particular to the reactions involved, in order to find good kinetic data.

Having good kinetic data of the reactions is a crucial point to obtain good process simulation. Unlike the thermodynamical data, which are available with good reliability for the majority of the components, kinetic data for specific reaction are really difficult to find. The reason is that no global laws to theoretically find them are known, so experimental data regression is needed. In the next paragraph description of the reactions involved, and the way used in this project to find kinetic data are presented. This preliminary work occupied a good part of the whole project, due to the difficulty to obtain satisfying results.

4.1.1 Goal of the process

The *AG2S* process is an innovative system to threat carbon desulfurization tail gases. Collaboration between SOTACARBO and POLIMI (Politecnico di Milano) is started to study the oxi-reduction reaction between H_2S and CO_2 to produce syngas composed mainly by CO, H_2 and steam. The reaction is done at very high temperature in gas phase, and has the very interesting capacity of producing syngas, a valuable product, from H_2S and CO_2, which are respectively toxic and dangerous substances. This technology could be very useful in gasification plants where is still present a capture line of the acid gases, and so the *AG2S* process could be applied to reduce the pollutants emission and at the same time increase the syngas production.

4.1.2 Reaction description

Tail gases coming from desulfurization solvent regeneration consist of H_2S, the main component, CO_2, CO, COS and other organosulfur compounds. The *AG2S* process wants to convert these components into syngas, thanks to the oxi-reduction reaction below:

$$2H_2S + CO_2 = H_2 + CO + S_2 + H_2O$$

The reaction is then able to neutralize the H_2S present in the tail gases, as other conversion process currently used, but has the advantage to valorize the hydrogen potential of the molecule converting it into H_2. Besides, another crucial point is the oxidizing compound chosen: in this reaction CO_2, another dangerous acid gas, is used instead of the common air or oxygen technology.

The theoretical results of this process show a great potential for lots of applications, with very good yields at high temperature. Obviously, before starting an industrial application of a new technology, it’s necessary to have experimental validation of the theoretical results; only after very good experimental data collection it’s possible to begin the scaling up of the process until becoming industrial application. The laboratory is built for this reason, to validate the theoretical results and proceed to the scaling up of the process.

4.2 THEORETICAL RESULTS: DSMOKE

4.2.1 How to find kinetic data

In the introduction of this chapter the difficulty to find kinetic data is presented: there aren’t theoretical laws to find reaction velocities without experimental support, and it follows that only two ways are possible to obtain kinetic data: organizing an experimental laboratory to find them, or using the available literature.

Obviously, getting data from literature is extremely cheaper and faster than organizing an experimental session; otherwise, often literature is lacking of data, because the reaction kinetic is related to the components involved and the reaction conditions, which are not only temperature, pressure and inlet flowrates and compositions, but also the presence of catalysts and their composition. In this huge amount of possibilities, it’s really hard to find a relating process in literature of which use the kinetic data. This is valid expecially in the present case, where a new process is studied.

But, even in the case of experimental choice, a preliminary kinetic study is needed to prepare the pilot laboratory. It’s easy to understand that the experimental studies would need a series of equipment, at first the reactor, which has to be designed in all their dimensions to, in case, commission their construction or use those still available. However, focusing on the reactor, and expecially in a PFR (plug flow reactor) used in this project, the length needed to complete the desired reaction in the specific range of conditions chosen is extremely dependent on the rate of reactions involved, in other words, on the reactions kinetic. It’s so evident that to start a pilot work to find good kinetic data of the studied reactions, preliminary theoretical study of the reaction environment is necessary. In particular, good preliminary kinetic data are needed to have a good design of the laboratory which will be responsible for validating them with experimental results. Besides, having good preliminary reaction rates is very useful to study different possible new processes without the high cost of the experimental sessions, to discard the worst options and deepen the best ones. This is the case of *AG2S*™ process, studied only theoretically and now ready to an experimental validation of the results.

But, as written before, the reaction process is new, so there aren’t available data in literature, and the kinetic is no obtainable using a theoretical law like in thermodynamic. So how is it possible to obtain good preliminary reaction rates for the process studied? This is the role of DSmoke.

4.2.2 DSmoke introduction

DSmoke program is the software used to get the theoretical results of the *AG2S*™ process. Developed at the Politecnico di Milano, it allows the simulation of a sequence of different reactors eventually coupled with mixers or splitters. The program accepts only sequential schemes of reactors, i.e. in series or in parallel, but no recycle streams are allowed.

The structure of DSmoke is presented in the figure below:

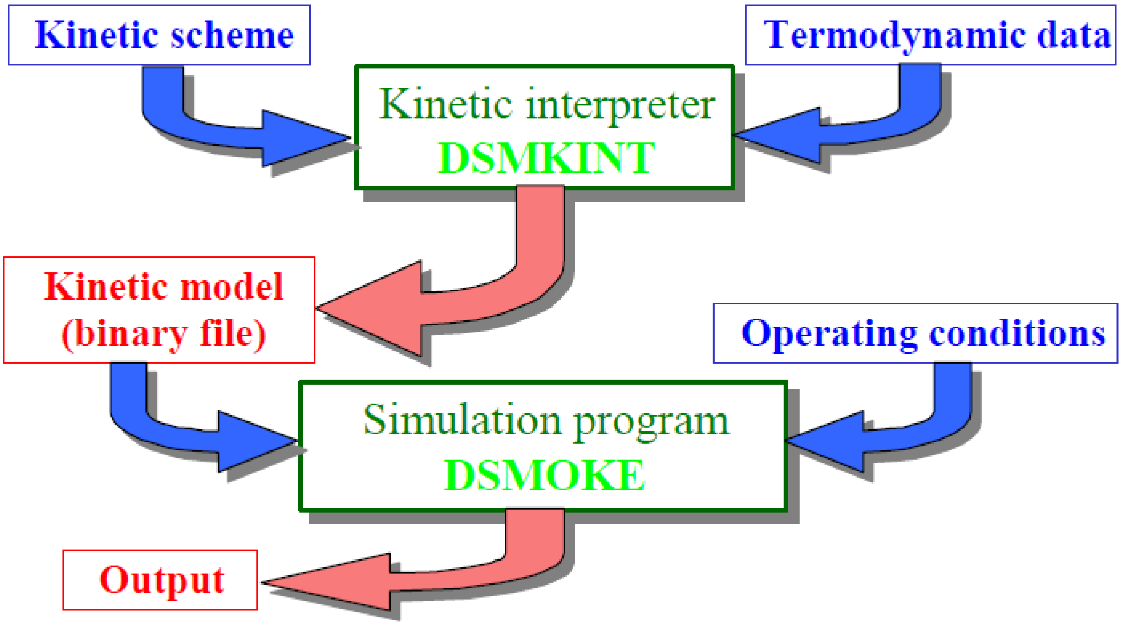


Figure 29 – DSmoke structure

The first step required by DSmoke is to provide as input data the kinetic scheme and the thermodynamic data of the system. In this way the internal DSmoke interpreter can generate the kinetic model used in the simulation program. At this point, the user has to specify the operating conditions of the process to proceed to the simulation. DSmoke is now ready to calculate the output values of the process.

Several parts of this package are directly taken from the experience developed in the pyrolysis and combustion modeling at Politecnico di Milano.

4.2.3 DSmoke general features of the kinetic model

As seen before, to provide the kinetic model necessary to the simulation, DSmoke requires thermodynamic data and the kinetic scheme. As regards the first requirement, the program has been developed to work with thermodynamic data in the form used in the NASA chemical equilibrium code: seven polynomial coefficients for each different temperature range to fit specific heat, enthalpy and entropy. Temperature ranges are two, low temperature (<1500K), and high temperature (>1500K).

The thermodynamic characterization of the system follows the expressions below:

$$\frac{cp}{R} = a_1 + a_2 T + a_3 T^2 + a_4 T^3 + a_5 T^4$$

$$\frac{H^0}{RT} = a_1 + \frac{a_2}{2} T + \frac{a_3}{3} T^2 + \frac{a_4}{4} T^3 + \frac{a_5}{5} T^4 + \frac{a_6}{T}$$

$$\frac{S^0}{R} = a_1 lnT + a_2 T + \frac{a_3}{2} T^2 + \frac{a_4}{3} T^3 + \frac{a_5}{4} T^4 + a_7$$

Therefore, for each species 14 coefficients must be introduced. Coefficients 1 to 7 refer to the upper temperature interval, while coefficients 8 to 14 are for the lower temperature interval.

In DSmoke program the thermodynamic data are already available for most of the chemical engineer interesting species, and in particular for the *AG2S*™ process. These data are taken from two different sources, the *'Chemkin'* (Kee et al., 1994a, 1994b) *'thermodynamic Data Base* and the *Benson's group additivity estimation method'* (Benson, 1976).

As for the thermodynamic data, DSmoke requires default structure of chemical reaction rate expressions. The forward rate constants are expressed in the following Arrhenius temperature dependence:

$$k_{f,j} = A_{f,j}\, T^{n_{f,j}} \exp(\frac{-Eact_{f,j}}{RT})$$

Where $A_{f,j}$ is the pre-exponential factor, $n_{f,j}$ is the temperature exponent and $Eact_{f,j}$ is the activation energy of the forward kinetic constant. These data must be specified directly in the kinetic input file.

The reverse rate constants are related to the forward rate constants through the equilibrium constants. They are calculated by the DSmoke interpreter using the kinetic input and the thermodynamical data without the need to specify them in the required files.

$$k_{r,j} = k_{f,j}/K_{eq,j}$$

This equation guarantees the thermodynamic consistency of the system at the equilibrium condition, where the global reaction rates must be zero.

The equilibrium constant ($K_{eq,j}$) is obtained from the thermodynamical proprieties of the components using the following expression:

$$K_{eq,j} = exp(\frac{\Delta S_j^0}{R} - \frac{\Delta H_j^0}{RT})$$

In DSmoke there is also the possibility to introduce third body and fall-off reactions.

With the thermodynamic input and the chosen kinetic, DSmoke generates the kinetic model thanks to the interpreter; in this way all the reactions will have the forward and reverse kinetic constants, ready to be simulated in the specific operating conditions chosen in the following step.

In the *AG2S*™ process hundreds of elementary reactions are reported in the kinetic input, to reproduce the kinetic model of the system. As written before, there is no possibility to theoretically forecast the kinetic data of the reactions; all the reactions data inserted in the kinetic input of the preliminary simulation of the process are present in the DSmoke database. This huge database, taken mainly from the experience developed in pyrolysis and combustion, contains lots of elementary steps kinetic data. The global reaction of the *AG2S*™ process is then divided in all the elementary steps which lead to the final products, considering also the alternative ways that all the compounds and radical species present in the reactive system could get. In this way the kinetic input can be produced using backdata of different reactions previously studied, choosing all the elementary reactions which could occur in the actual reaction system. This work mode offers also the possibility to increase the accuracy of the model subsequently including other elementary reactions not considered in the first time, without changing the other data. In lots of applications the results of this method are consistent to experimental data with high accuracy; however, the huge number of reactions considered during the simulation can be even a negative aspect where a fast response of the simulator is needed. This is the case of the immersive simulation, where a real-time dynamic simulation is necessary to provide a good experience to the user. In the Dynsim simulation realized for the *AG2S*™ laboratory, it can't be possible to introduce all the reactions considered in the DSmoke simulation. This leads to the necessity of a new simplified kinetic scheme production, which will be discussed in the next chapter.

4.2.4 The reactor models and *AG2S™* operating conditions

In DSmoke different reactor models are available to better simulate the studied system.

In particular, the program contains PFR reactor, CSTR reactor, batch reactor, dynamic CSTR transient, and finally the shock tube reactor and the NSI engine. This work is focused on the PFR reactor, which is the most similar to the real tube reactor in the laboratory. The equations for mass and energy conservation of the PFR used for the DSmoke calculations are here described, assuming that no mass crosses the boundary:

- mass balance for each of NC species:

$$\frac{d\omega_i}{d\tau} = \sum_{j=1}^{NR} \nu_{ij}\, r_j W_i$$

$$i = 1, \ldots, NC$$

- Energy balance:

$$cp\,\frac{dT}{d\tau} = \sum_{j=1}^{NR} -\Delta H_j\, r_j + \frac{U_{ex}S}{V}\left(T_{ext} - T\right)$$

- Momentum balance:

$$\frac{dP}{d\tau} = -2f\,\frac{G_s^3}{\rho D}$$

DSmoke requires as input of the simulation program the kinetic model generated from the interpreter and the operating conditions.

In the *AG2S™* process the reactor can be considered as an isothermal PFR with length of 200 mm, and the feed of the reactor is the gas stream composed by the 5% of H_2S, 5% of CO_2 and the 90% of N_2. The mass flowrate of the stream required by DSmoke is varied from 10 l/h to 30 l/h (from 3.3E-6 kg/s to 9.9E-6 kg/s), the reactor temperature is varied from 800°C to 1300°C. With these range of temperature and flowrates it's possible to cover all the possible process conditions of the real laboratory. The simplified reaction rates derived from those data are so capable to forecast the behavior of the system in all the possible simulation conditions. Obviously, the DSmoke results are used in this work as preliminary 'experimental data', but when the laboratory will start to provide stable experimental results, it will be possible to perform a regression using real data, which can at the same time increase the accuracy of the process dynamic simulation, and validate the preliminary theoretical results obtained from DSmoke.

4.2.5 DSmoke results

In this paragraph DSmoke results are presented. In particular H_2S and CO_2 conversions and mass fractions versus reactor temperature are illustrated for the three different mass flowrates considered.

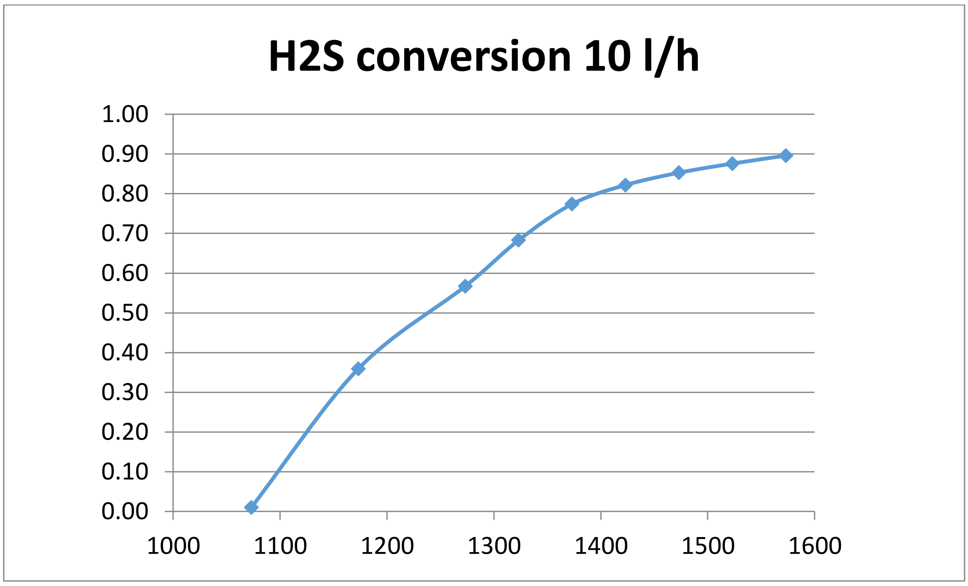


Figure 30 - conversion of H_2S for 10 l/h of volume flowrate

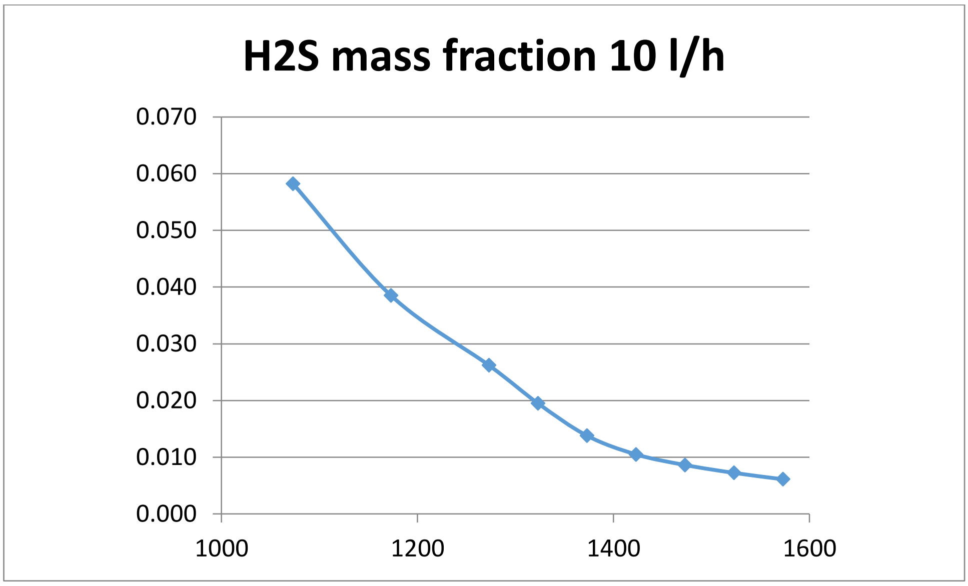


Figure 31 - mass fraction of H_2S for 10 l/h of volume flowrate

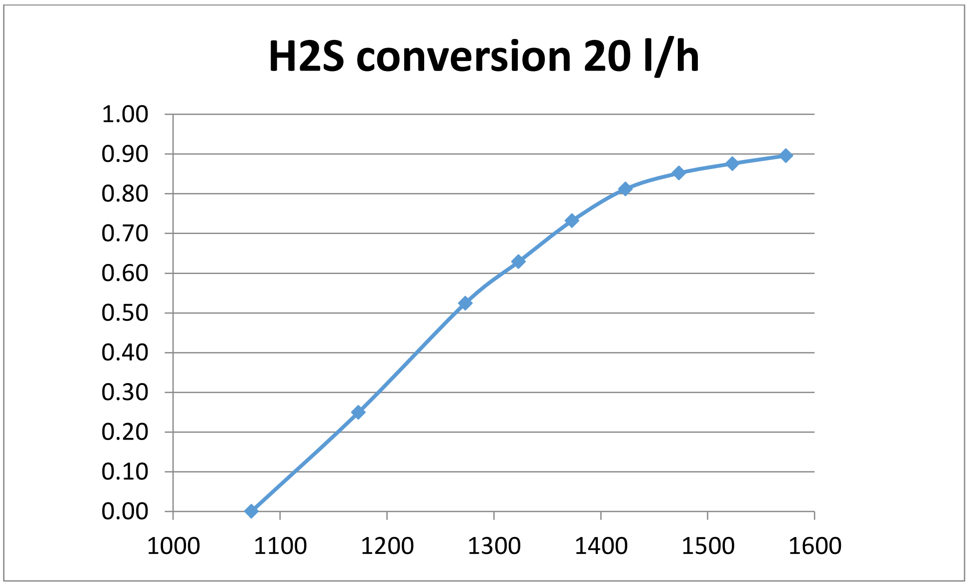


Figure 32 - conversion of H_2S for 20 l/h of volume flowrate

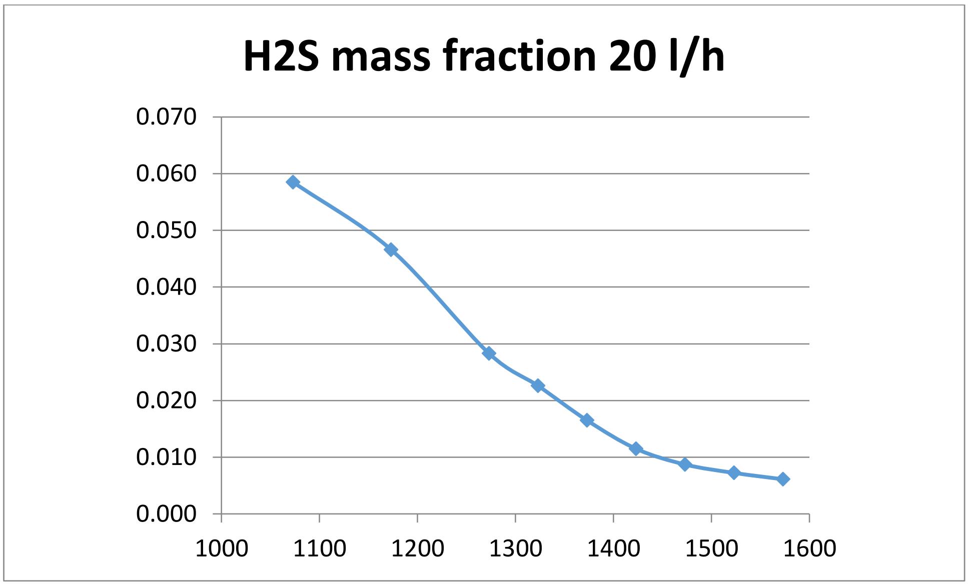


Figure 33 - mass fraction of H_2S for 20 l/h of volume flowrate

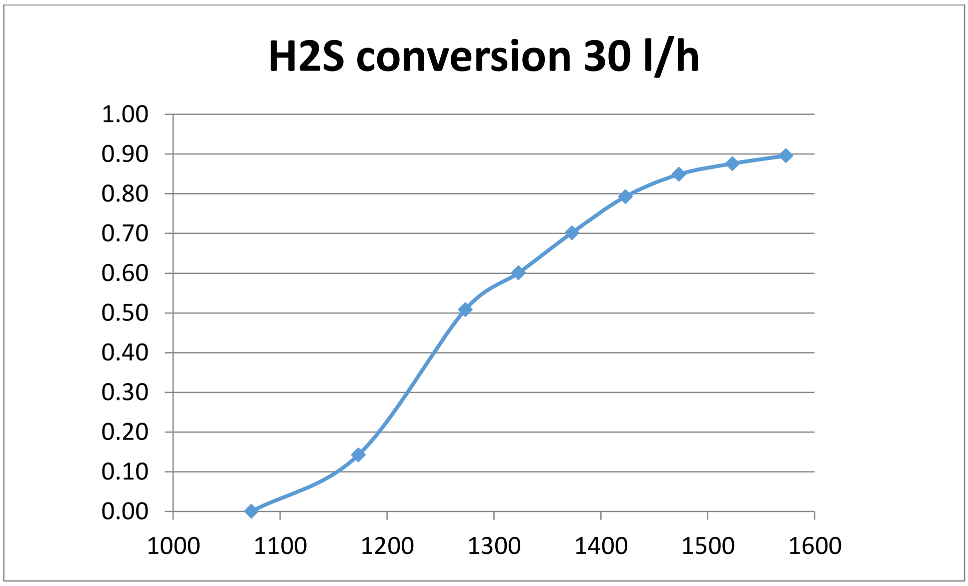


Figure 34 - conversion of H_2S for 30 l/h of volume flowrate

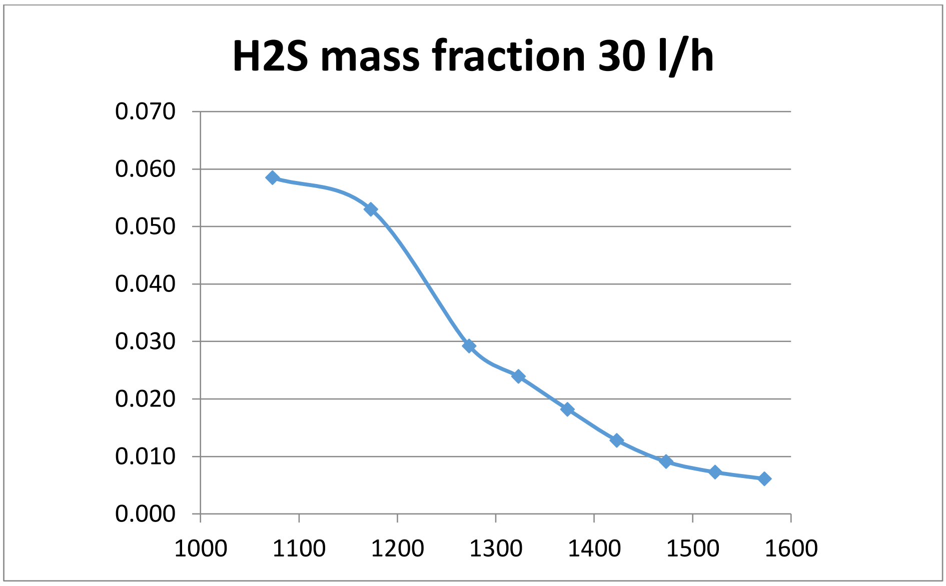


Figure 35 – mass fraction of H_2S for 30 l/h of volume flowrate

As regards the H_2S trends, it's possible to observe the differences in the behavior with changing temperature and flowrates:

- Temperature: conversion increases critically with the increase of temperature. There are no evidence of reaction at the lower temperature (1023K, corresponding to 800°C), while conversion reaches 90% value at 1573K (1300°C) for all the flowrates considered. Please notice that the inlet mass fraction of H_2S is 0.0585, derived from the 0.05 molar value.
- Flowrates: increasing the process flowrate leads to a decrease of residence time in the reactor of the species. This is why, at higher flowrates, the H_2S conversion trend shows a decrease in its value at low and mid temperature, where the reactions rates are not sufficiently fast to complete the reaction. The case at 1173K (900°C) is evident: at 10 l/h flowrate the conversion is 0.36, considerably less the conversions at 20 l/h and 30 l/h, respectively 0.25 and 0.14. When the temperature increases, the reaction rates are faster, so the equilibrium is reached within the 200mm of the reactor for all the three studied flowrates, so the conversion values are the same.

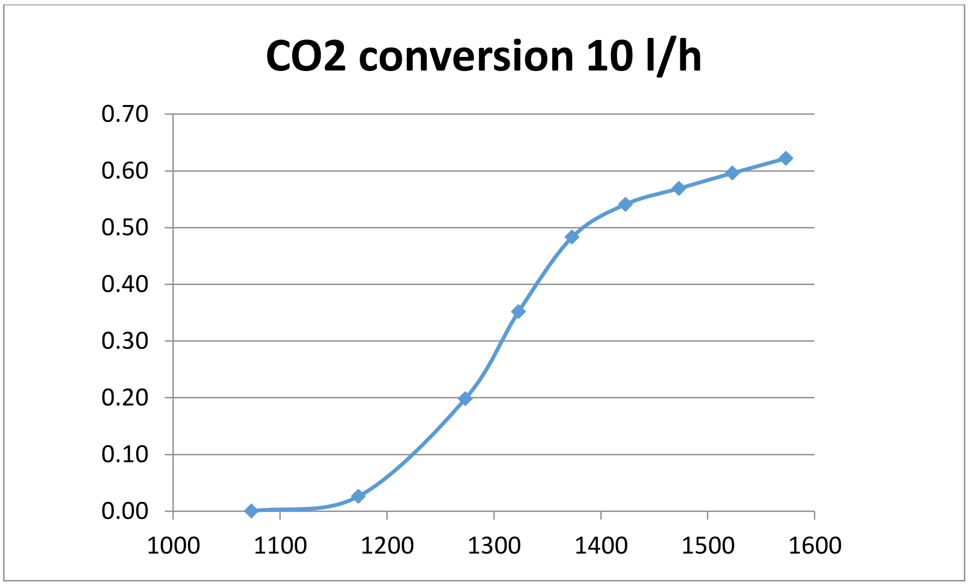


Figure 36 – conversion of CO_2 for 10 l/h of volume flowrate

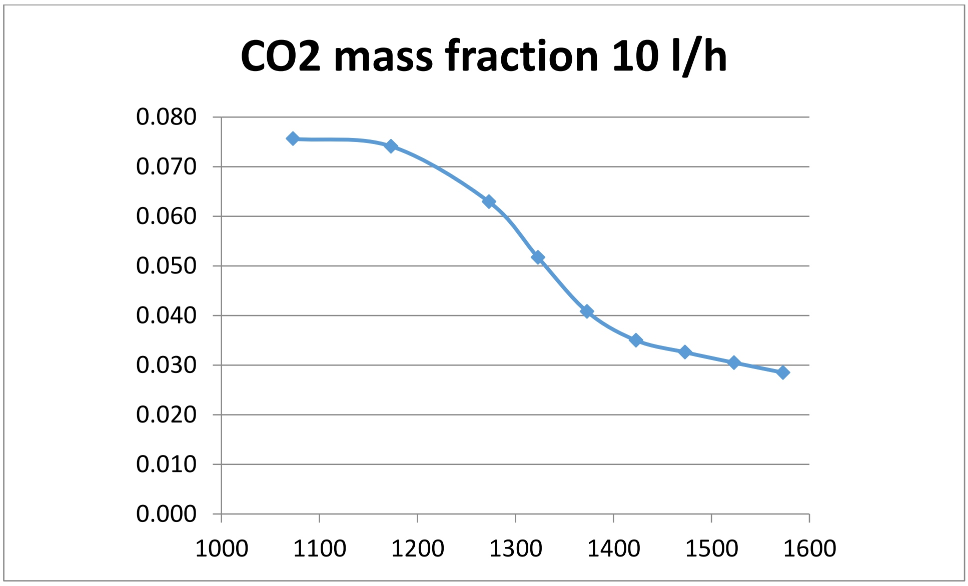


Figure 37 – mass fraction of CO_2 for 10 l/h of volume flowrate

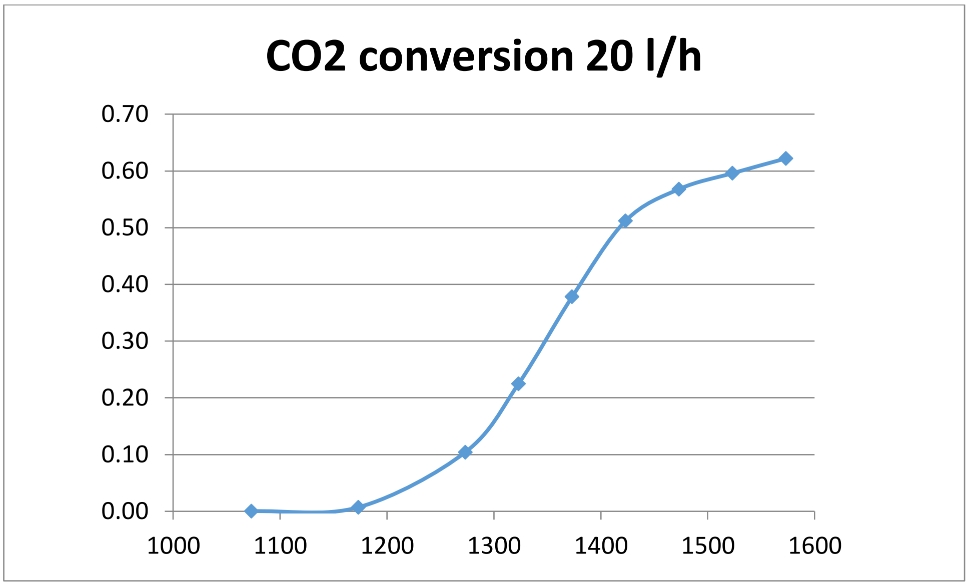


Figure 38 - conversion of CO_2 for 20 l/h of volume flowrate

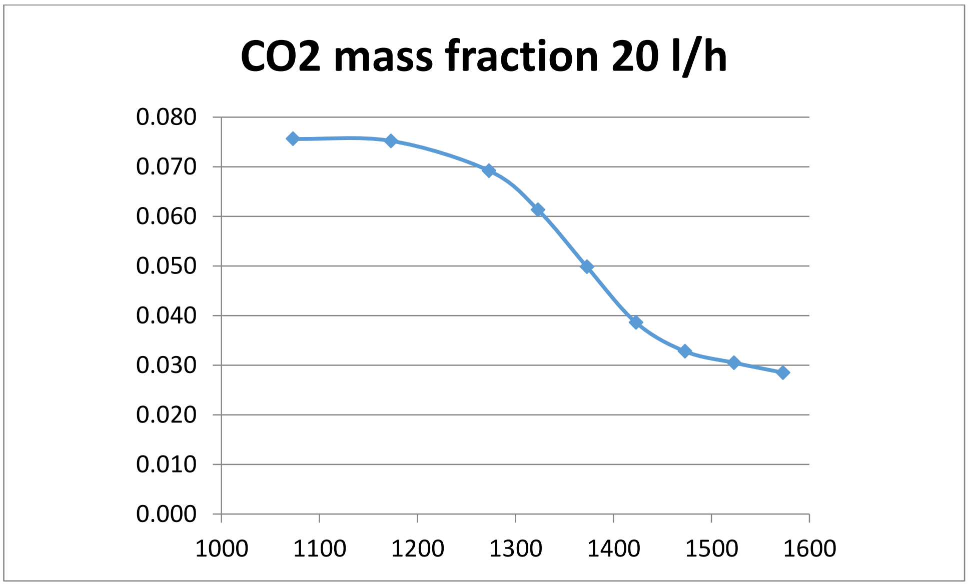


Figure 39 - mass fraction of CO_2 for 20 l/h of volume flowrate

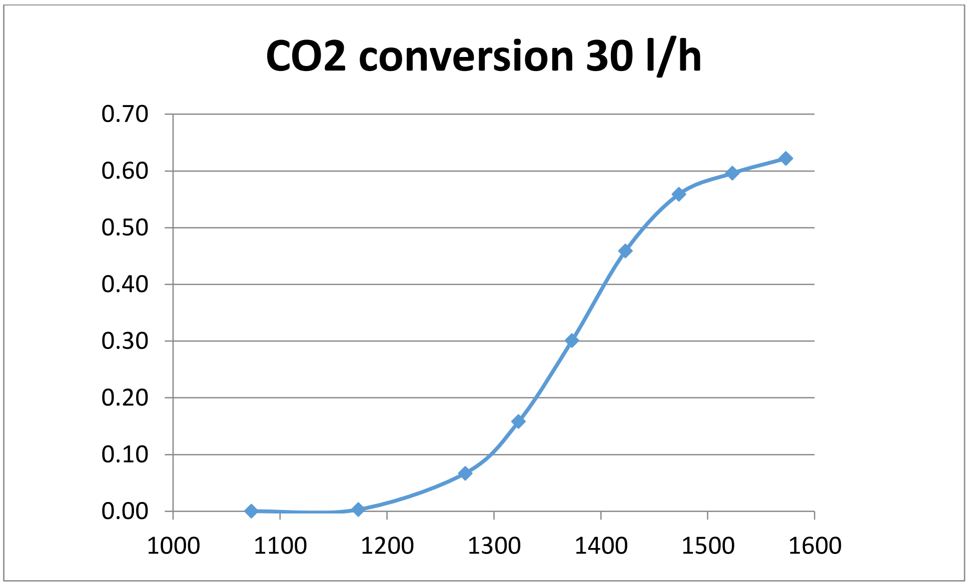


Figure 40 - conversion of CO_2 for 30 l/h of volume flowrate

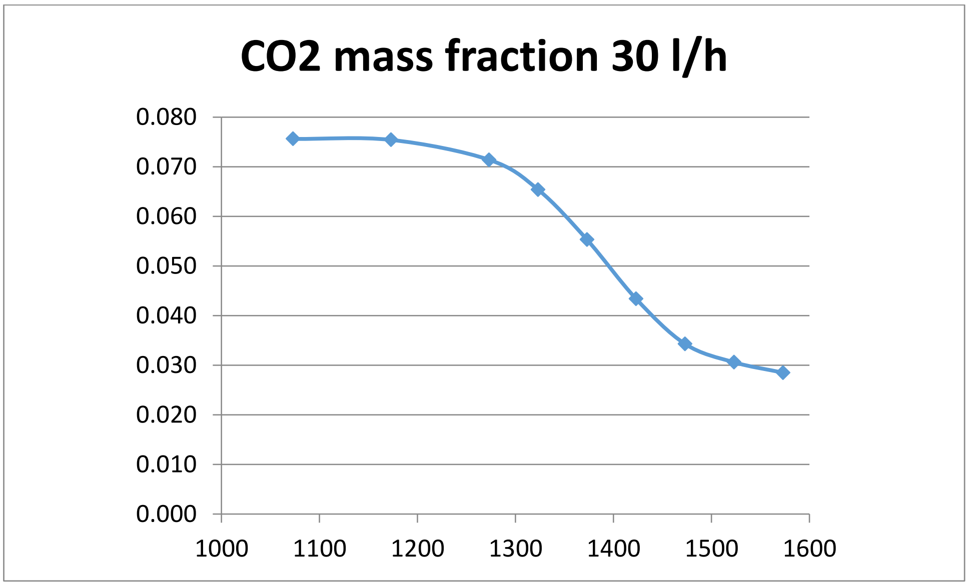


Figure 41 - mass fraction of CO_2 for 30 l/h of volume flowrate

The same global considerations of H_2S conversion and mass fraction trends can be done for CO_2 data. In this case CO_2 could reach a maximum conversion of 0.62, significantly lower than H_2S. The lower ability of the considered system to convert the CO_2 respect to the H_2S is also evident in the low temperature case, where the CO_2 is not reactive until 1273K (1000°C), where it anyway shows very low conversions (0.20, 0.10 and 0.07 for the three flowrates considered).

4.3 SIMPLIFIED KINETIC SCHEME FOR SIMULATIONS

4.3.1 Reasons and structure of the simplified kinetic scheme

After the system simulation with DSmoke, the work of data regression is presented.

As written before, the regression of the DSmoke results is necessary to find a simplified kinetic scheme that can be used in the process simulation. In fact, the two simulations realized in this thesis work require a simplified kinetic in respect to the hundreds of elementary reactions considered in DSmoke. Otherwise, the huge number of reactions would make really heavy simulations, which could be a problem for the accuracy and speed of the simulation, expecially if the final goal is to use the simulation results as a real time input of the virtualizer to perform the immersive simulation. For this reason the complex DSmoke reaction scheme is reduced to a simply one. But, what are the characteristic that the new reaction scheme should have to be used in place of the rigorous one?

The most important characteristics are that the simplified reaction scheme must be capable of reproducing as similar as possible the DSmoke trends of the key species, and obviously it must be very much simpler scheme respect to the rigorous one.

The first attempt to reproduce the DSmoke kinetic scheme in a simpler form was made using a unique reaction, i.e. the global *AG2S*™ process reaction:

$$2H_2S + CO_2 = H_2 + CO + S_2 + H_2O$$

Unfortunately, modeling the regression using only this reaction to fit the DSmoke data results in unsatisfactory outcomes, in particular in low temperature profiles where the new conversion trends were very far from the null value of the rigorous kinetic scheme. In the next paragraph kinetic scheme regression code will be explained, but even from now it can be clear that it works searching a good combination of reaction constants parameters to fit the DSmoke process output, starting from guess values chosen by the user. In the case of the global unique reaction scheme, this searching procedure could last hours, up to whole days, without finding a good solution for any combination of guess values proposed.

In order to obtain a better regression result, two different reactions are taken into account:

a) $2H_2S = 2H_2 + S_2$

b) $CO_2 + H_2 = CO + H_2O$

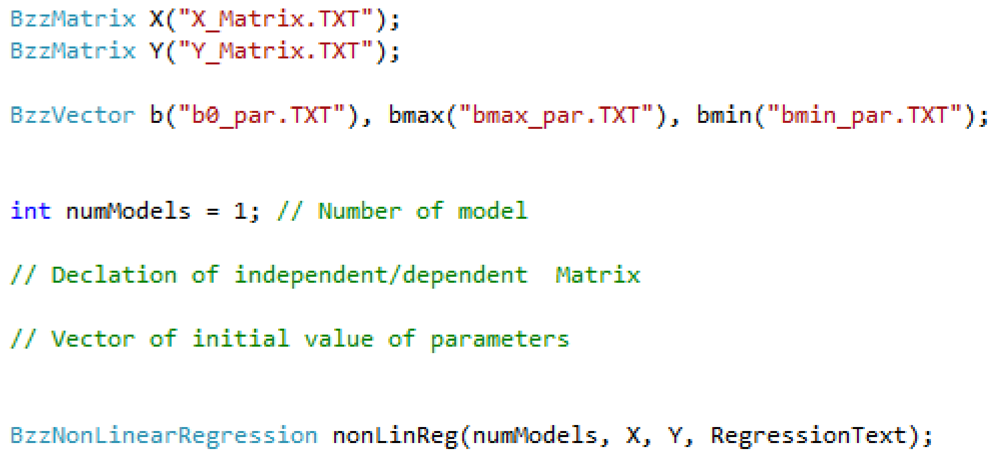
It’s evident that the summation of the two reactions is simply the global *AG2S*™ reaction written before, but in this way the regression code can vary two different sets of kinetic parameters, increasing the flexibility of the new reaction scheme. Once selected the reactions considered in the new simplified model, it’s possible to focus the attention on the regression code.

4.3.2 Regression code: C++ and BzzMath© library

To perform the kinetic data regression a C++ code is chosen exploiting the potential of BzzMath© library.

The BzzMath© library, realized at the Politecnico di Milano, is able to solve different mathematical issues like doing interpolations, operating with matrixes, solving DAE or ODE systems and doing linear and nonlinear regression to name a few.

In this case the nonlinear regression model is used. To perform the regression using the BzzMath© library, some fundamental components must be specified. At first, regression program requires the number of models present in the regression routine, and how much independent and dependent variables are present for each model. In this thesis work case, the model specified is one, with two dependent and two independent variables.

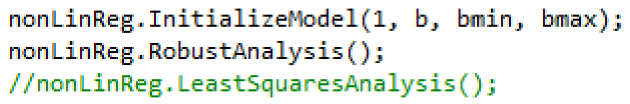


```
BzzMatrix X("X_Matrix.TXT");
BzzMatrix Y("Y_Matrix.TXT");

BzzVector b("b0_par.TXT"), bmax("bmax_par.TXT"), bmin("bmin_par.TXT");


int numModels = 1; // Number of model

// Declation of independent/dependent  Matrix

// Vector of initial value of parameters


BzzNonLinearRegression nonLinReg(numModels, X, Y, RegressionText);
```

Figure 42 – C++ code. BzzNonLinearRegression initial data

In the figure an extract of the C++ code is illustrated. It’s possible to see the declaration of the models number and the dependent and independent variables required by BzzMath. The variables are specified as matrixes in text files externally compiled: in this way is possible to clarify the dimensions and values of them without modifying the C++ code. This can be useful for using the same code changing the input data, i.e. in this case when the experimental data will be available they could substitute DSmoke values without changing the regression code.

Please notice also that in the initial part of the code even three vectors are declared: bmin, bmax and b0 BzzVectors. These vectors, represented as text files like the previous variable matrixes, are necessary to the BzzMath© regression routine to know respectively the minimum, maximum and initial guess values of the parameters which will be used to fit the data with the simplified model.



```
nonLinReg.InitializeModel(1, b, bmin, bmax);
nonLinReg.RobustAnalysis();
//nonLinReg.LeastSquaresAnalysis();
```

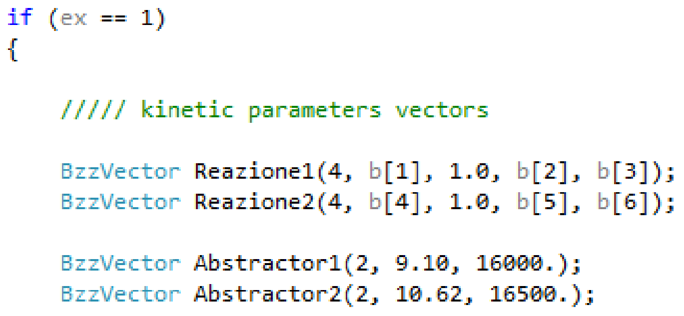
Figure 43 - C++ code. Regression command

After the parameters declaration, is sufficient to initialize the model and choose the way to perform regression, for example using least square analysis or a robust method.

The crucial issue of the regression code is how to link the kinetic parameters, i.e. pre exponential factor A, temperature exponent γ and activation energy Eact, to the DSmoke results which are conversions or mass fractions exiting the reactor. This is done using C++ to manage DSmoke.

4.3.3 Regression code: manage DSmoke using C++

To perform a correct regression, it’s necessary to compare the DSmoke results, obtained using the complex kinetic scheme, with the results obtained simulating the same process system but using the simplified kinetic scheme. To do this, DSmoke must be managed by the regression routine to iteratively change the kinetic parameters and perform the simulation, compare the results with the correct ones and proceed the cycle to find good kinetic parameters which are able to fit the Y data; this is the purpose of the main part of the code, ‘Regression text’. At the first iteration, the kinetic scheme is compiled:



```
if (ex == 1)
{

    ///// kinetic parameters vectors

    BzzVector Reazione1(4, b[1], 1.0, b[2], b[3]);
    BzzVector Reazione2(4, b[4], 1.0, b[5], b[6]);

    BzzVector Abstractor1(2, 9.10, 16000.);
    BzzVector Abstractor2(2, 10.62, 16500.);
```

Figure 44 – kinetic scheme initializing, regression parameters

In the figure are identifiable four kinetic parameters per reaction requested by DSmoke as kinetic input; the reason is that DSmoke needs the input kinetic parameters in this form:

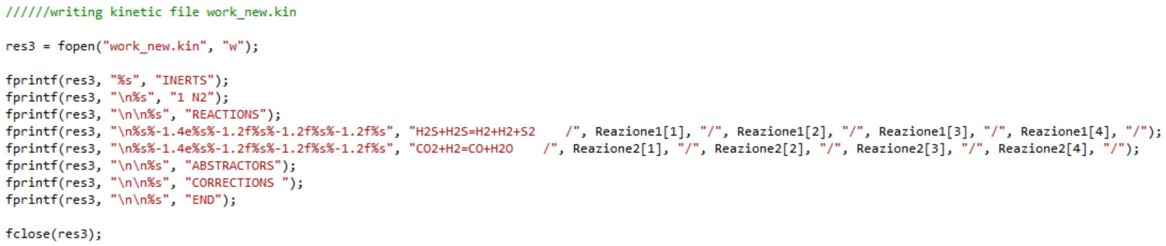
[A/b/n/Eact]

which correspond to:

$$k_{f,j} = \boldsymbol{A}_{f,j} 10^{\boldsymbol{b}}\, T^{\boldsymbol{n}_{f,j}} \exp\left(\frac{-\boldsymbol{Eact}_{f,j}}{RT}\right)$$

Substantially, the pre exponential factor is divided in two parts by DSmoke: the ‘A’ represents only its value, while the ‘b’ characterizes its order of magnitude. Notice that, to reduce the number of parameters to be iterated by the regression code, the pre exponential factor is formally reduced in one single data. In fact, while ‘A’ is recognized as ‘b[1]’ parameter, the ‘b’ values is fixed to 1. There are no differences in this type of kinetic parameters characterization for DSmoke, the only warning is to be careful in the minimum and maximum parameters value assignment; the value of A built in this way has no 0 to 10 interval, but must vary its value in a really much bigger range to include the order of magnitude previously represented by ‘b’.

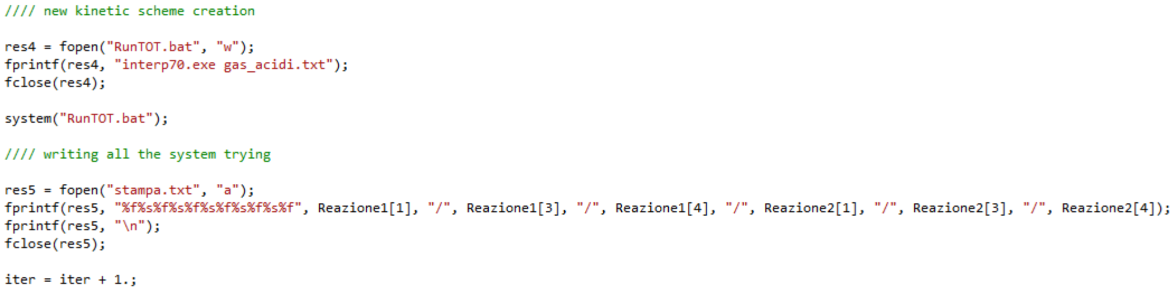
As the reactions considered are two, there are two groups of kinetic parameters to be initialized. Totally, there are six parameters to be iterated by the regression code.



```
//////writing kinetic file work_new.kin

res3 = fopen("work_new.kin", "w");

fprintf(res3, "%s", "INERTS");
fprintf(res3, "\n%s", "1 N2");
fprintf(res3, "\n\n%s", "REACTIONS");
fprintf(res3, "\n%s%-1.4e%s%-1.2f%s%-1.2f%s%-1.2f%s", "H2S+H2S=H2+H2+S2    /", Reazione1[1], "/", Reazione1[2], "/", Reazione1[3], "/", Reazione1[4], "/");
fprintf(res3, "\n%s%-1.4e%s%-1.2f%s%-1.2f%s%-1.2f%s", "CO2+H2=CO+H2O    /", Reazione2[1], "/", Reazione2[2], "/", Reazione2[3], "/", Reazione2[4], "/");
fprintf(res3, "\n\n%s", "ABSTRACTORS");
fprintf(res3, "\n\n%s", "CORRECTIONS ");
fprintf(res3, "\n\n%s", "END");

fclose(res3);
```

Figure 45 – kinetic scheme initialization, writing kinetic file

In the second step, the kinetic file is created. Inerts must be specified in the first part of the kinetic input file, thus considered reactions are written. DSmoke requires to write the stoichiometric reactions with separate components, for example using ‘H2S+H2S’ instead of ‘2 H2S’; the kinetic parameters are inserted after the respective reaction.

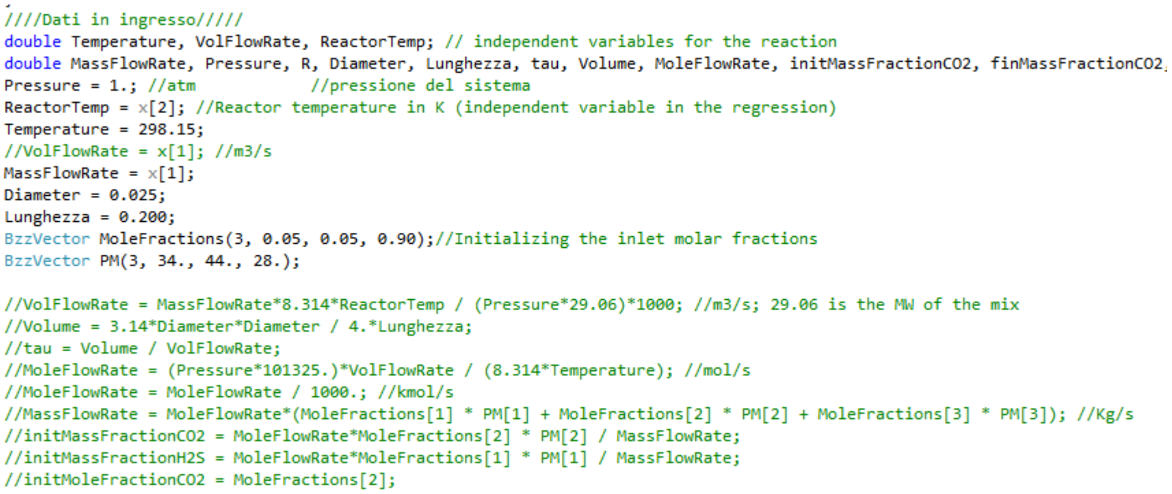


```
//// new kinetic scheme creation

res4 = fopen("RunTOT.bat", "w");
fprintf(res4, "interp70.exe gas_acidi.txt");
fclose(res4);

system("RunTOT.bat");

//// writing all the system trying

res5 = fopen("stampa.txt", "a");
fprintf(res5, "%f%s%f%s%f%s%f%s%f%s%f", Reazione1[1], "/", Reazione1[3], "/", Reazione1[4], "/", Reazione2[1], "/", Reazione2[3], "/", Reazione2[4]);
fprintf(res5, "\n");
fclose(res5);

iter = iter + 1.;
```

Figure 46 - kinetic scheme creation

After the kinetic file creation, it’s possible to create and execute the command to use DSmoke interpreter to build the kinetic scheme. The interpreter is the ‘interp70.exe’, while ‘gas_acidi.txt’ is the text file where are specified the components to be used by the interpreter. In particular, in ‘gas_acidi.txt’ kinetic and thermodynamic input file names are written. As regard the thermodynamic data, there is no need to write them in the C++ code, because the thermo input file is already provided and it obviously doesn't vary during the iterations.

In order to control the correct working of the regression code, in this section another text file is created, in which all the regression trying are registered, the 'stampa.txt'. Notice that 'Reazione1[2]' and 'Reazione2[2]', corresponding to the fixed parameters 'b' of the two reactions, aren't even registered.



```
////Dati in ingresso/////
double Temperature, VolFlowRate, ReactorTemp; // independent variables for the reaction
double MassFlowRate, Pressure, R, Diameter, Lunghezza, tau, Volume, MoleFlowRate, initMassFractionCO2, finMassFractionCO2
Pressure = 1.; //atm            //pressione del sistema
ReactorTemp = x[2]; //Reactor temperature in K (independent variable in the regression)
Temperature = 298.15;
//VolFlowRate = x[1]; //m3/s
MassFlowRate = x[1];
Diameter = 0.025;
Lunghezza = 0.200;
BzzVector MoleFractions(3, 0.05, 0.05, 0.90);//Initializing the inlet molar fractions
BzzVector PM(3, 34., 44., 28.);

//VolFlowRate = MassFlowRate*8.314*ReactorTemp / (Pressure*29.06)*1000; //m3/s; 29.06 is the MW of the mix
//Volume = 3.14*Diameter*Diameter / 4.*Lunghezza;
//tau = Volume / VolFlowRate;
//MoleFlowRate = (Pressure*101325.)*VolFlowRate / (8.314*Temperature); //mol/s
//MoleFlowRate = MoleFlowRate / 1000.; //kmol/s
//MassFlowRate = MoleFlowRate*(MoleFractions[1] * PM[1] + MoleFractions[2] * PM[2] + MoleFractions[3] * PM[3]); //Kg/s
//initMassFractionCO2 = MoleFlowRate*MoleFractions[2] * PM[2] / MassFlowRate;
//initMassFractionH2S = MoleFlowRate*MoleFractions[1] * PM[1] / MassFlowRate;
//initMoleFractionCO2 = MoleFractions[2];
```

Figure 47 – DSmoke input data for regression

After the kinetic scheme creation, DSmoke requires the operating conditions to perform the system simulation. For this purpose the input variables must be specified in C++. In the regression code two different reactor designs are tried:

- The first reactor design is simply the isothermal reactor with 200mm of length. Temperature and mass flowrates are assigned by the user in the X matrix file, and must be the same of the respective DSmoke rigorous simulation input data;
- The second reactor design includes the little heating section of few millimeters which furnace needs to bring the temperature of the reactants from ambient to the desired temperature. After this little section there is the same isothermal reactor of 200mm than before.

Even if the two reactor designs are really similar, the input data must change a bit. If there is the desire to assign as input the volumetric flowrate instead of the standard DSmoke request which is the massive flowrate, in the second reactor design is sufficient to calculate the massive flowrate starting from the input value and the initial components temperature, which is 298K (ambient temperature). Instead, in the first reactor design the input volumetric flowrate would not correspond to the values measured by the fluximeters (10 l/h to 30 l/h), but to the increased volume flowrate dependent from the reactor temperature. This is because in the first reactor design there isn’t the pre heating section of the furnace where the components start to the ambient temperature to reach the reactor one, but the inlet is already considered at the operative temperature. In the simpler reactor design is so more convenient to provide input flowrate in massive unit of measure which isn’t affected by the changing of temperature. In the figure above is evident the possible double choice: the commented code corresponds to the second reactor design, where volume flowrate can be easily assigned as input data, while the working code corresponds to the first reactor design, where mass flowrate input is more convenient.

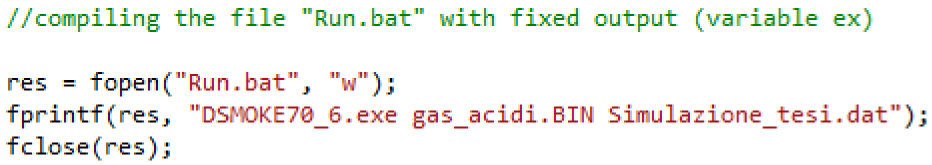
Actually, using DSmoke complex kinetic scheme, it’s been proved that the two reactor designs have really small differences in the results, so the simplest one is chosen to be replaced in Dynsim and MOSAIC, even because in the real laboratory there isn’t a separated preheating section, but only the furnace which uses few millimeters of its length to heat the inlet charge before reaching the isothermal conditions. The resulting regression parameters written in this thesis work refer then to the first reactor design, but they’ve been found even for the second one; confirming the practically total superimposability of the models, the resulting kinetic parameters are approximately the same for the two different designs. This is because the pre heating section of the furnace necessary to the scope is really small, and also because the kinetic of the reactions is very slow at mid/low temperatures. Then, the effective added part, i.e. the millimeters at high temperature to be added to the effective length of the reactor, are practically null. It’s so convenient in this situation choosing the simplest reactor design, to have a simpler simulation without losing in accuracy.

```
//compiling file.dat for Dsmoke

res1 = fopen("Simulazione_tesi.dat", "w");

fprintf(res1, "%s", "TITLE Simulazione_tesi");
fprintf(res1, "\n%s", "IVODE 3");
fprintf(res1, "\n%s", "SEQUENCE  REACT 1");
fprintf(res1, "\n\n%s", "STREAM 1");
fprintf(res1, "\n\n%s%f", "RATE= ", MassFlowRate);
fprintf(res1, "\n%s%f", "TEMP= ", ReactorTemp);
fprintf(res1, "\n%s%f", "PRES= ", Pressure);
fprintf(res1, "\n\n%s", "MOLE");
fprintf(res1, "\n%s%f", "H2S ", MoleFractions[1]);
fprintf(res1, "\n%s%f", "CO2 ", MoleFractions[2]);
fprintf(res1, "\n%s%f", "N2 ", MoleFractions[3]);
fprintf(res1, "\n\n%s", "SPKEY H2S");
//fprintf(res1, "\n\n%s%s%f%s%f", "EQUIPMENT REACT 1 TYPE=2 ", "TEMP= ", ReactorTemp, " PRES=1.0 DIAM= ", Diameter);
//fprintf(res1, "\n%s%f", "LENG= ", 0.080);
//fprintf(res1, "\n%s", "INPUT=1 OUTPUT=2 COSV=1");
//fprintf(res1, "\n%s", "TOLR=1.E-12 RTOL=1.E-5 TOLT=1.E-4 TOLM=1.E-6");
fprintf(res1, "\n\n%s%f", "EQUIPMENT REACT 1 TYPE=1 DIAM= ", Diameter);
fprintf(res1, "\n%s%f", "LENG= ", Lunghezza);
fprintf(res1, "\n%s", "INPUT=1 OUTPUT=2 COSV=1");
fprintf(res1, "\n%s", "TOLR=1.E-12 RTOL=1.E-5 TOLT=1.E-4 TOLM=1.E-6");
//fprintf(res1, "\n\n%s%s%f%s%f", "EQUIPMENT REACT 3 TYPE=2 ", "TEMP= ", ReactorTemp, " PRES=1.0 DIAM= ", Diameter);
//fprintf(res1, "\n%s%f", "LENG= ", 0.080);
//fprintf(res1, "\n%s", "INPUT=3 OUTPUT=4 COSV=1");
//fprintf(res1, "\n%s", "TOLR=1.E-12 RTOL=1.E-5 TOLT=1.E-4 TOLM=1.E-6");
fprintf(res1, "\n\n%s", "RECORD  H2S CO2 H2");
fprintf(res1, "\n%s", "!SENS=1");

fclose(res1);
```

Figure 48 – compiling file.dat for DSmoke

The biggest file to be compiled in DSmoke is the .dat; in this file operating conditions are specified. As described before, two different reactor designs are tried, but the working one is the simpler configuration, with one isothermal reactor of 200mm length.

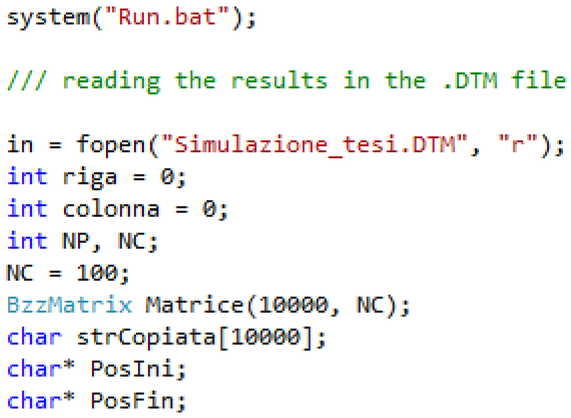
DSmoke wants first to know the number and the sequence of the reactors; if there were more than one reactors, they could be easily added in the sequence writing them after the 'React 1'.Then, mass flowrate, composition, temperature and pressure of the inlet charge are assigned in this file. In the considered case the temperature used is the 'ReactorTemp', in fact there isn't the preheating section; else, the 'Temperature' value would be used as input. Structure of the reactors sequence are now needed; notice that before the working code of the equipment 'React 1' another homonym equipment is specified in the commented part (used in the second reactor design simulation). DSmoke reads 'Type=1' as isothermal PFR, taking the operating temperature directly from its feed flowrate, while 'Type=2' is used for a PFR with assigned linear temperature profile, that is the case of including a preheating of the charge before the isothermal reactor. The input and output streams number for each reactor in the sequence is written, and the reactors description is complete. In order to decide the outputs of the simulation, the 'SPKEY' and 'RECORD' command are used: the first keyword is used to decide the key component which will be used for conversion and selectivity calculation, while the second gives the possibility to define species which will appear in the output files even in term of molar fractions vs the reactor length and in rate contributions.



```
//compiling the file "Run.bat" with fixed output (variable ex)

res = fopen("Run.bat", "w");
fprintf(res, "DSMOKE70_6.exe gas_acidi.BIN Simulazione_tesi.dat");
fclose(res);
```

Figure 49 – simulation Run file

It’s now possible to compile and execute the run file which requires the application to be run, the kinetic scheme ‘gas_acidi.BIN’ previously generated by the interpreter using the thermodynamic and kinetic inputs, and at the end the process operating simulation just written in ‘Simulazione_tesi.dat’.

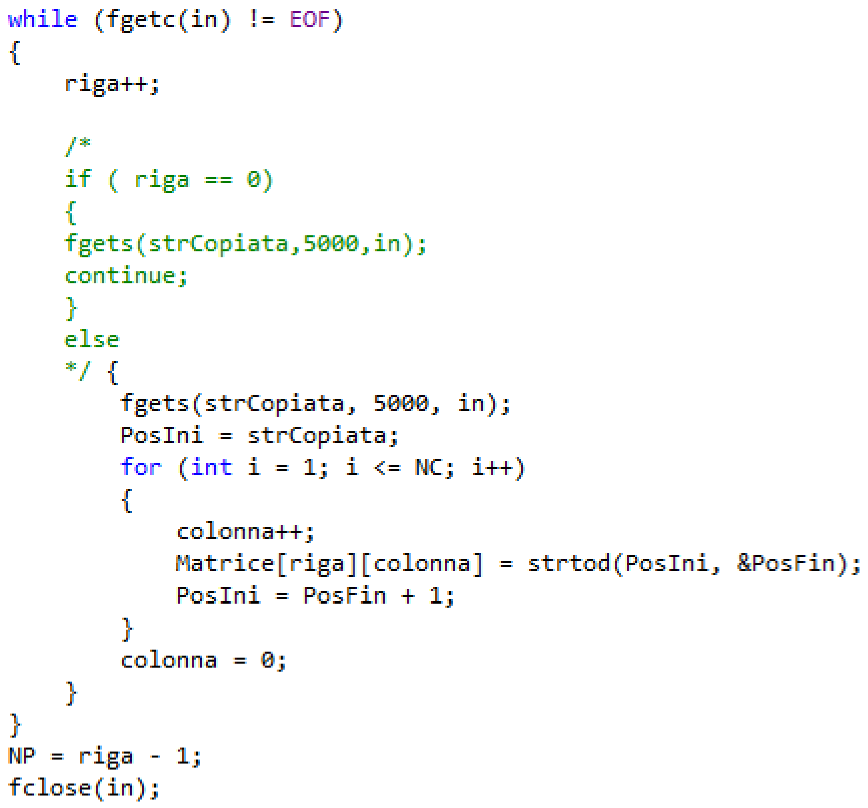


```
system("Run.bat");

/// reading the results in the .DTM file

in = fopen("Simulazione_tesi.DTM", "r");
int riga = 0;
int colonna = 0;
int NP, NC;
NC = 100;
BzzMatrix Matrice(10000, NC);
char strCopiata[10000];
char* PosIni;
char* PosFin;
```

Figure 50 – reading the results in .DTM file

After the running of the system lots of output files are provided by DSmoke. The most complete output file is the .DTM one; it provides, in this precise order, the trends of position, time, conversion of the key species, temperature, pressure, molar fractions of the species specified in ‘RECORD’ keyword, mass fractions of all the system species, and at the end selectivities and carbon selectivities of the species specified in ‘RECORD’.

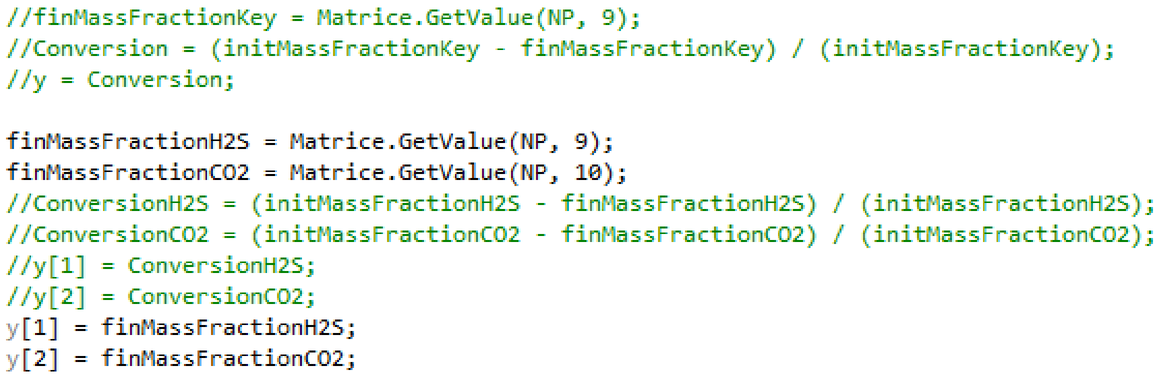
In this part of the code the objective is to read the output file and extract data which are comparable with DSmoke rigorous kinetic scheme results. This data can be the conversions or the output mass fractions of the components. For this purpose ‘Simulazione_tesi.DTM’ is opened.



```
while (fgetc(in) != EOF)
{
    riga++;

    /*
    if ( riga == 0)
    {
    fgets(strCopiata,5000,in);
    continue;
    }
    else
    */ {
        fgets(strCopiata, 5000, in);
        PosIni = strCopiata;
        for (int i = 1; i <= NC; i++)
        {
            colonna++;
            Matrice[riga][colonna] = strtod(PosIni, &PosFin);
            PosIni = PosFin + 1;
        }
        colonna = 0;
    }
}
NP = riga - 1;
fclose(in);
```

Figure 51 – finding the results of the .DTM file

This part of the code performs the reading of the file until finding the results row. To find it, the code reads the entire .DTM file searching a completely null row; in fact the first null row which can be found in the .DTM file is the subsequent of that where final results are collected. In this way, it’s only necessary to subtract one row to the empty one to find the precise position of the results. The row number is collected into ‘NP’ number, while the column position depends from the desired data, but can be easily determinate following the DSmoke output list rule written before.



```
//finMassFractionKey = Matrice.GetValue(NP, 9);
//Conversion = (initMassFractionKey - finMassFractionKey) / (initMassFractionKey);
//y = Conversion;

finMassFractionH2S = Matrice.GetValue(NP, 9);
finMassFractionCO2 = Matrice.GetValue(NP, 10);
//ConversionH2S = (initMassFractionH2S - finMassFractionH2S) / (initMassFractionH2S);
//ConversionCO2 = (initMassFractionCO2 - finMassFractionCO2) / (initMassFractionCO2);
//y[1] = ConversionH2S;
//y[2] = ConversionCO2;
y[1] = finMassFractionH2S;
y[2] = finMassFractionCO2;
```

Figure 52 – assigning y of the model

It's now necessary only to combine properly the desired data to their respective .DTM columns, and assign the regression routine y of the model. In the working code in the figure above output mass fractions of H_2S and CO_2 are used like y of the model, but also H_2S and CO_2 conversion can be used giving the same regression results. The only difference is in the Y matrix to be written: in the first case, from complex kinetic scheme simulations, output mass fractions must be extracted and then listed in the Y Matrix, in the second case the right data are the H_2S and CO_2 conversions. The X Matrix is conversely the same for both the cases, because it contains the mass flowrates and reactor temperatures of the respective simulations results written in the Y matrix.

4.3.4 Regression results

In this paragraph the regression results are shown.

Actually, after the choice of the two reactions as regression kinetic scheme, the first approach has been to use only H_2S conversion as Y data, i.e. using the same data of the regression code with the unique global reaction considered. In this way, the request to the regression is to find good trend in the H_2S behavior. The results obtained are effectively really near in terms of H_2S conversion, but observing the CO_2 kinetic it was evident that regression results were not realistic. In fact, the regression code with this first approach doesn't take care of the second reaction kinetic feasibility, and in particular of the behavior of CO_2. In order to obtain the peculiar trend of H_2S conversion, with a null value at 800°C and a 90% at 1300°C, the regression routine calculates the second kinetic constant with the only constriction of consuming the right quantity of H_2 to promote the first reaction conversion increasing at high temperature, but without any constraints in CO_2 conversion. The resulting kinetic was so good to explain the behavior of H_2S and sulfur in the system, but not for the other important components, first the CO_2.

For this reason in the final regression work, whose results are shown in this paragraph, as y of the model are considered both the H_2S and CO_2 trends (conversion or final mass fractions). In this way all the components in the system reach about the same final quantities of that resulting from the complex kinetic scheme system, because imposing the fitting of the H_2S and CO_2 trends, the other components are strictly related by the two key compounds by the reactions stoichiometry.

The final kinetic parameters found by the regression routine are listed in the table below:

Reaction	A [variable]	γ []	Eact [j/mol]
1 forward	2.206e-2	2.09	75548
1 reverse	1.456e-11	3.09	-104309
2 forward	6.956e-2	3.13	222034
2 reverse	1.535e-3	3.13	187259

Table 1 – kinetic parameters results after data regression

Actually, the regression routine as written before can find only 6 kinetic parameters, but in the table 12 values are reported. This is because of the internal structure of DSmoke; the kinetic parameters required to compile the kinetic scheme are only the forward rate parameters, while the reverse are calculated internally by DSmoke using thermodynamic inputs. In the C++ code the kinetic input is iterated in order to find the optimal correspondence of results, so the outputs of the regression routine are only the forward parameters. Using the interpreter of DSmoke, it’s then possible to find the reverse kinetic parameters assigning the thermodynamic data of the system.

DSmoke uses as standard units of measure kcal, kmol, m^3, so the kinetic results must be converted in the SI (International System of units). As regard the activation energy, it’s only necessary to convert the kcal/kmol in kj/kmol (equivalent to j/mol), and for γ values which are adimensional no transformations are needed. But the pre exponential factors have some more difficulty to be correctly transformed; in DSmoke the reactions written are considered as elementary reactions, so the order of the compounds in the kinetic expression are simply zero for the products, and the positive value of their respective stoichiometric coefficient for the reactants. To clarify, the simplified reaction scheme rate expressions as considered by DSmoke are reported:

- $2H_2S = 2H_2 + S_2$
    - $R_{forward} = k_{forward}c_{H2S}^2$
    - $R_{reverse} = k_{reverse}c_{H2}^2 c_{S2}$
- $CO_2 + H_2 = CO + H_2O$
    - $R_{forward} = k_{forward}c_{CO2}c_{H2}$
    - $R_{reverse} = k_{reverse}c_{CO}c_{H2O}$

The reaction rates must be always in [mol/(m^{3}s)]; it's obvious that the kinetic constants unit of measure varies depending on the order of the reaction rate. Since the only parameter which can change its unit of measure in the kinetic constant expression is the pre exponential factor, the various $A_{j,f}$ and $A_{j,r}$ will have different unit of measures.

As written before, DSmoke uses kcal, kmol, m^3 standard units, so the assigned unit of measures for the resulting A parameters are:

- for second order rate expressions: A in [m^3/(kmol s)];
- for third order rate expressions: A in [m^6/(kmol^2 s)];

The desired values of reaction rate are in [mol/(m^{3}s)], so it's necessary to convert the A DSmoke values in m^3, mol, s units. For the second order rate expressions, the resulting A must be divided from 10^3 to reach the correct units, while for the third order the dividing factor is 10^6. In this particular work, the first reverse reaction A is a third order reaction, and the huge dividing factor justifies the really small pre exponential factor (10^{-11} order of magnitude), else the other reactions are second order, so the 10^3 dividing factor must be applied.

In addition to the simplified kinetic parameters values, the regression routine of BzzMath library provides also a 'results' file, in which the assigned y, Y Matrix, and the model y, i.e. the DSmoke simulation results using the simplified kinetic scheme with the regression parameters, are compared. In the figure below are reported all the trends for H_2S and CO_2 conversion at the three flowrates considered (10, 20 and 30 l/h). In blue the DSmoke rigorous kinetic scheme results are presented, in red the simplified kinetic after regression results.

H_2S CONVERSIONS AND CONCENTRATIONS

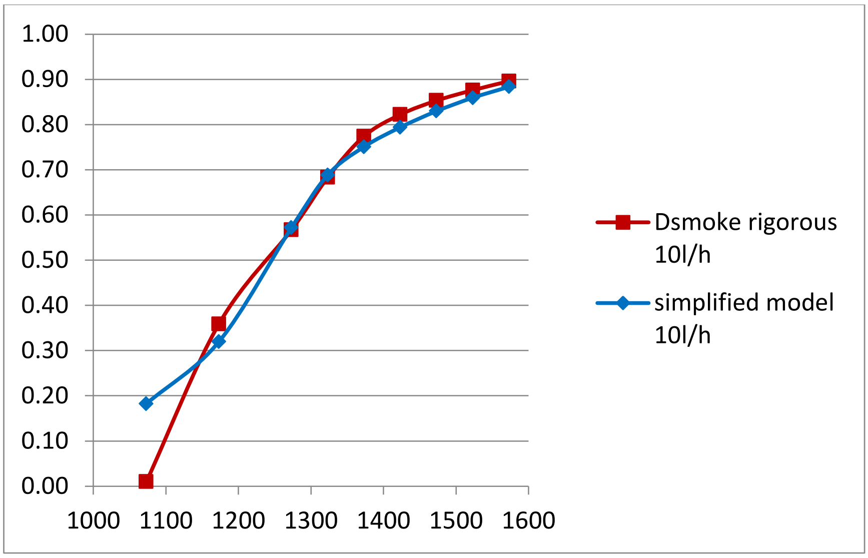


Figure 53 – H2S comparing conversion 10 l/h

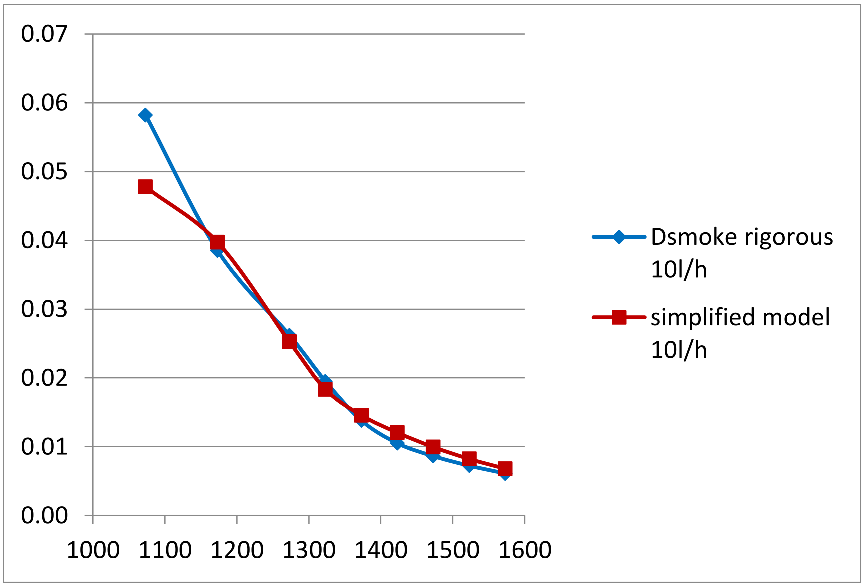


Figure 54 – H2S comparing concentration 10 l/h

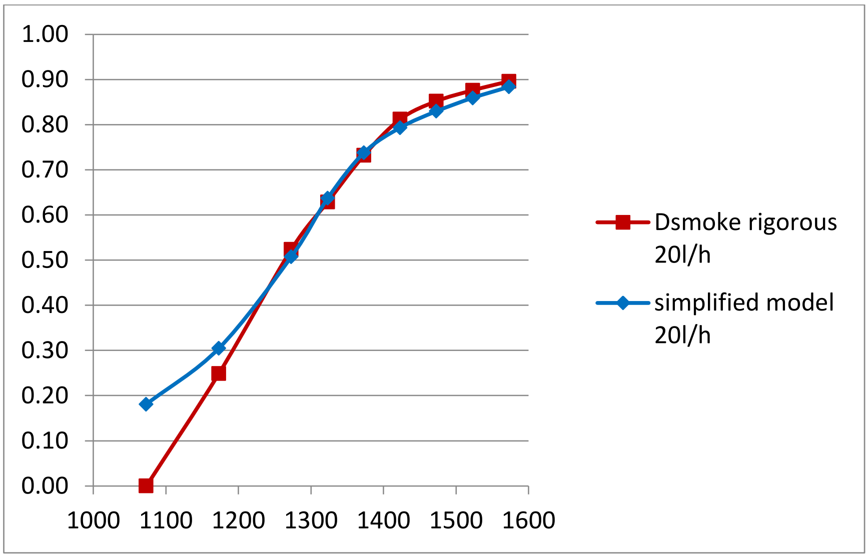


Figure 55 H2S comparing conversion 20 l/h

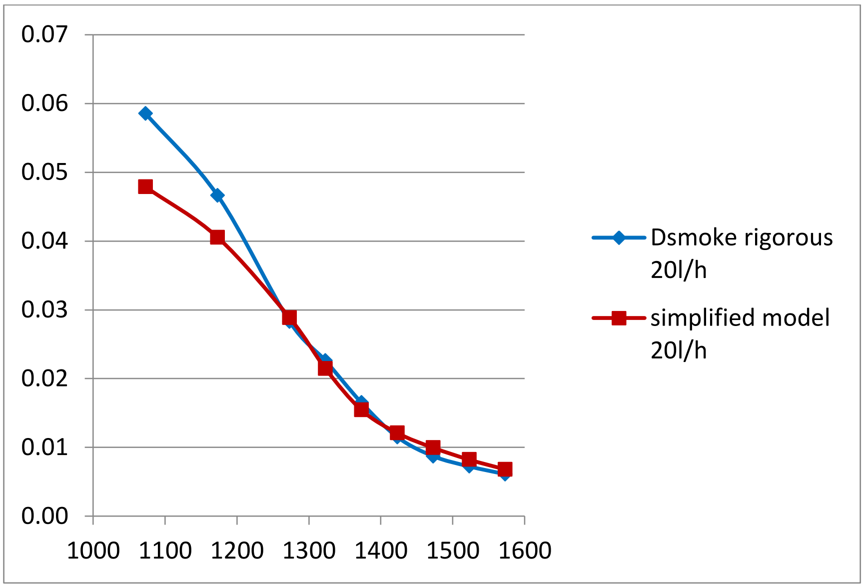


Figure 56 - H2S comparing concentration 20 l/h

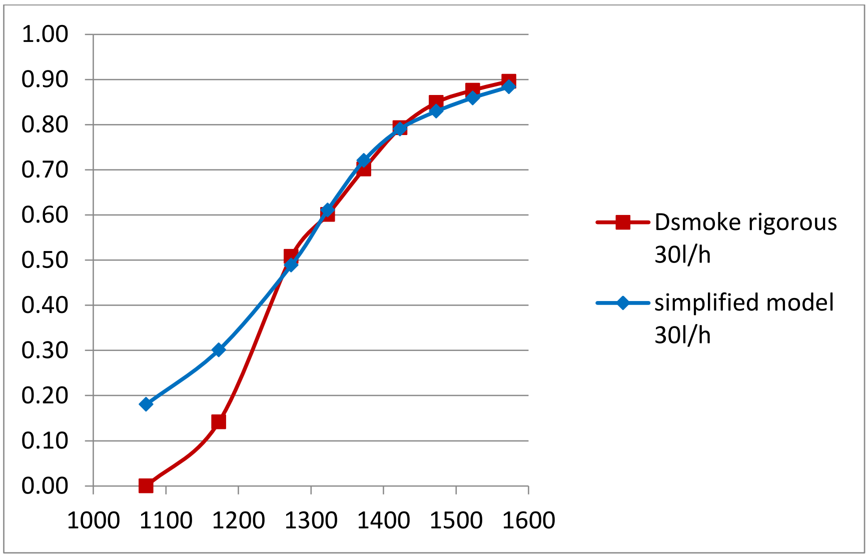


Figure 57 - H2S comparing conversion 30 l/h

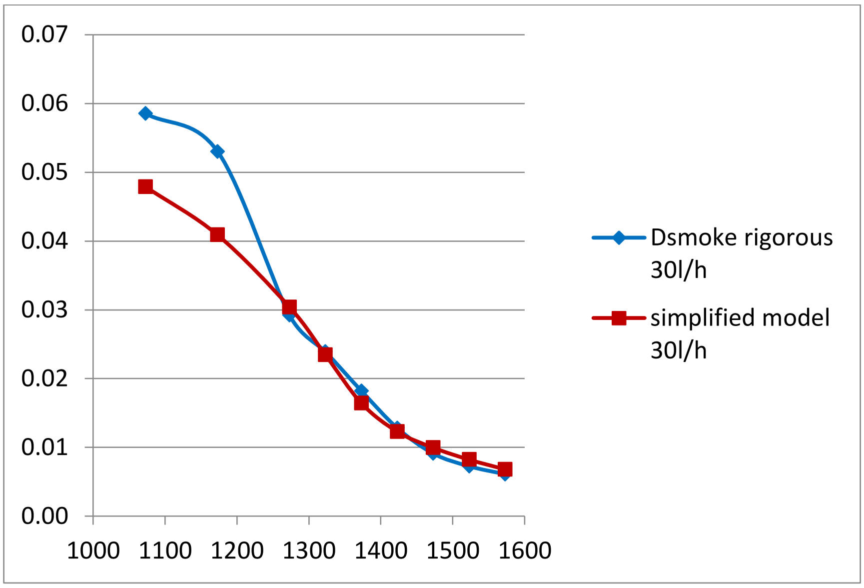


Figure 58 - H2S comparing concentration 30 l/h

It’s possible to see that the simplified model reach really good results in simulating H_2S behavior expecially from 1273K (1000°C), while at 1073K (800°C) simulation there is an important error between the two models. In the figures above the behavior at temperature lower than 800°C isn’t specified, because those data are not included in the regression work, but both the rigorous and the simplified model reaches an H_2S conversion value of zero below 700°C. This leads to the conclusion that the simplified model reaches comparable results with the rigorous model, except for the neighborhood of 800°C.

The distance of the results at lower temperature increases with increasing of the volumetric flowrates. Probably the simplified H_2S kinetic is faster than its real kinetic, so the effect of increasing the flowrate, which corresponds to a decrease of the residence time in the reactor, is smaller in the regressed kinetic scheme .While at high temperatures even the complex model has the speed to reach the equilibrium, with small or null changes with different flowrates, and in temperatures lower than 700°C both the models are slow enough to reach the zero conversion, at the mid-low temperatures around 800/900°C, the different speed of the two models leads to the detachment of the resulting values in this T range, expecially at higher flowrates. These results suggest to threat with care the simulation conditions, reaching very accurate outputs above 900°C and under 700°C. Fortunately, the interesting temperature range of the laboratory is not the problematic one, because obviously in the *AG2S™* process the goal is to reach high H_2S conversion: the operating temperature of the laboratory will surely be over 1000°C, where H_2S conversion starts increasing from 50% to 90%. In this range of temperatures, the accuracy of the simplified kinetic scheme, compared with the rigorous one, is really high at any considered flowrate, so the reaction parameters found satisfy the needs of the process simulation.

CO_2 CONVERSIONS AND CONCENTRATIONS

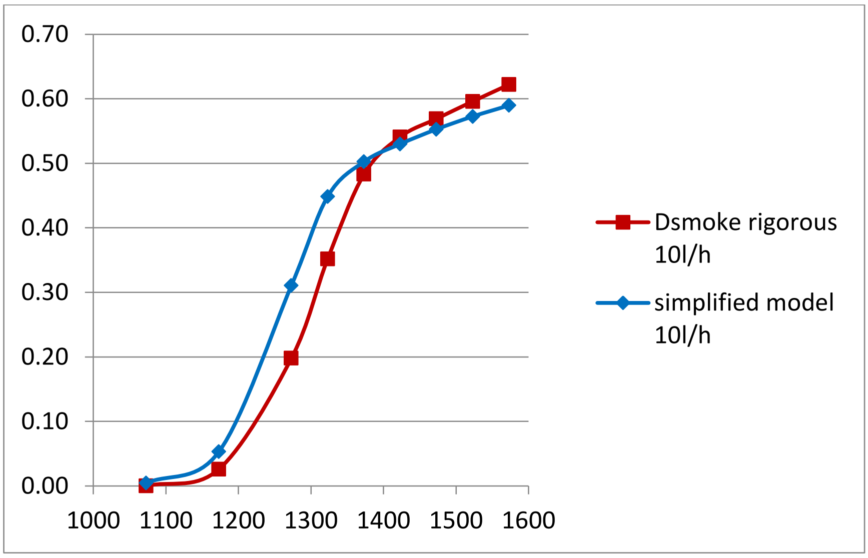


Figure 59 – CO2 comparing conversion 10 l/h

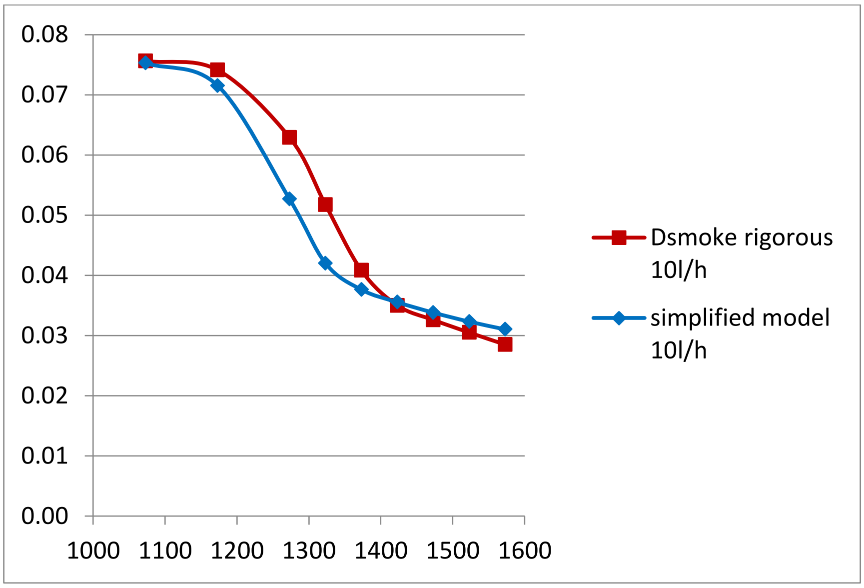


Figure 60 – CO2 comparing concentration 10 l/h

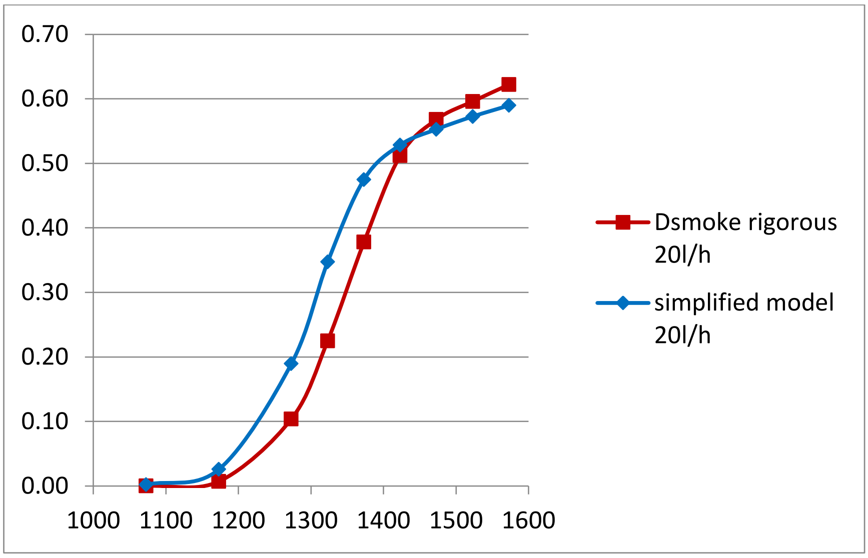


Figure 61 - CO2 comparing conversion 20 l/h

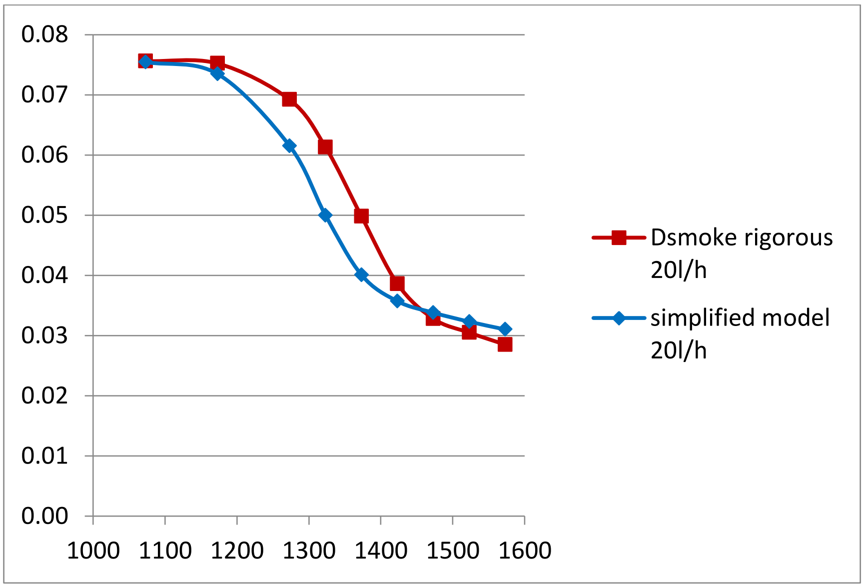


Figure 62 - CO2 comparing concentration 20 l/h

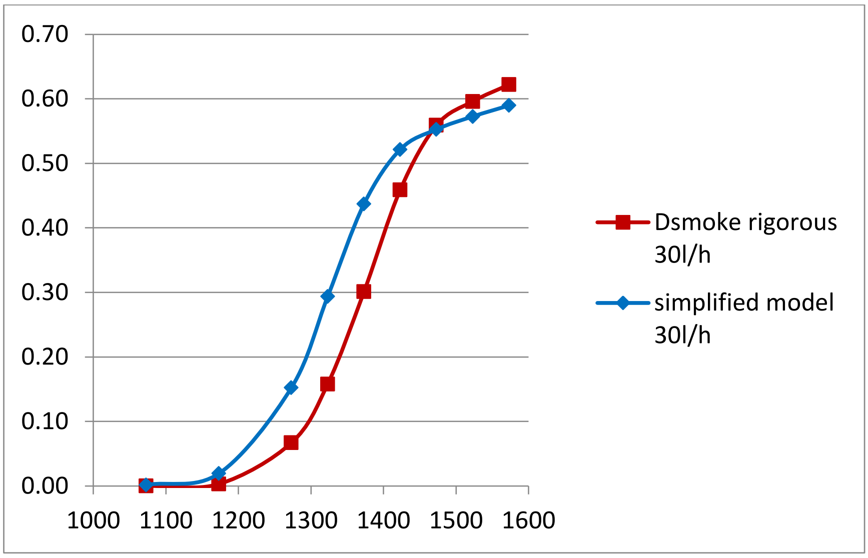


Figure 63 - CO2 comparing conversion 30 l/h

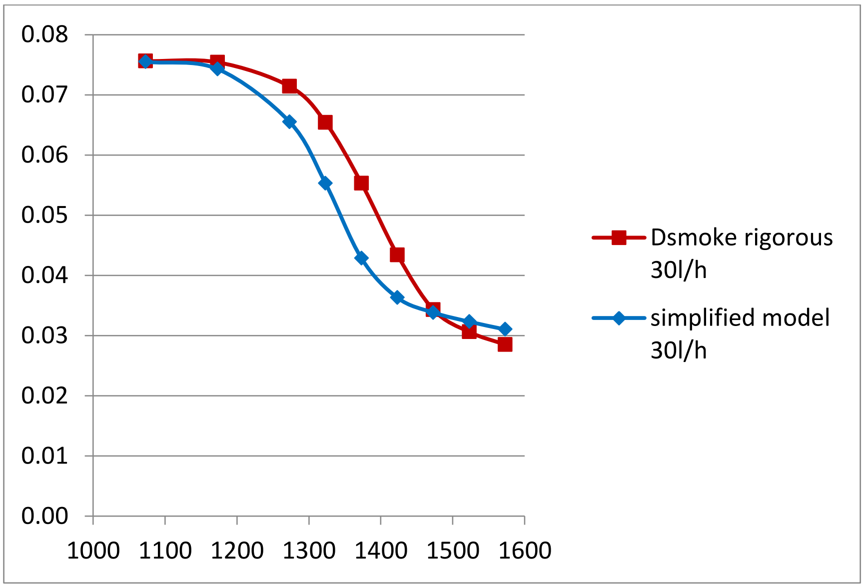


Figure 64 - CO2 comparing concentration 30 l/h

The CO_2 behavior figures show, in respect to the H_2S one, a better shape even at low temperature, where the CO_2 conversion approach the zero value for both the rigorous and the simplified kinetic scheme. However, in this case there is a more evident detachment at the mid temperature values, expecially for higher flowrates; so, also at process temperatures, could be observable errors in the CO_2 final conversions, even if the accuracy increases like in the H_2S case for very high temperatures, where both the CO_2 and the H_2S conversions increases significantly their values, so the results are better in those conditions where probably *AG2S*™ laboratory activities will focus their attention.

Globally, the regression results are comparable with the DSmoke rigorous simulations. Evidently, the simplified kinetic scheme shows some difference in the reactants behavior respect to the complex one, but the trends and final values, expecially within the interesting operating conditions range, are respected.

In the future, when the *AG2S*™ laboratory will start to produce reliable results, the kinetic parameters regression could take advantage of real experimental data and so the accuracy with the real compounds behavior will increase. At this moment, for the preliminary study, the obtained results satisfy the need of the process simulation, in order to have a first overview of the laboratory behavior.

5. DYNAMIC SIMULATION SOFTWARE: DYNSIM

5.1 STEADY STATE VS. DYNAMIC SIMULATION

Dynsim enables to perform the dynamic simulation of the *AG2S* laboratory. But why is it necessary to provide a dynamic simulation and not only a steady state simulation?

The critical difference is that in steady state simulations there isn't the possibility to study the effective behaviour of the system, but only its stationary trend. Conversely, the actual plant is normally moving towards or away a steady state. So the steady state approach disregards transient effects which are fundamental to simulate the reality of a process, and in fact they are detailed in the dynamic simulation.

Obviously, to provide a detailed description of the actual behaviour of the process, the dynamic simulation requires a huge amount of information respect to a steady state approach. For instance, dynamics require equipment sizes, metal masses, heat exchange data, desired holdups and process control schemes. In the dynamic simulation even simple intermediate equipment, like tanks which account for the dead volume in line must be incorporated to provide an accurate representation of the transient process. Viewing and interpreting the results can consume more time than steady state simulations due to the dramatic increase in the amount of data generated.

Finally, in particular for this project it's crucial to have an optimal representation of real behavior of the process, because the final goal is to provide real-time data of the simulation to the software responsible of the 3D virtualization of the laboratory. In this way, the operator that uses the immersive dynamic simulation could prove the effects of his/her actions observing real-time changing of variables like flowrates, temperatures, or tank holdups. In fact, with the immersive dynamic simulation it's possible to train the operators to work in the correct way using a direct proof of their actions. The 3D immersive environment should be well recognizable and apparently indistinguishable by the trainee (this concern to the 3D model and virtualization textures), and has to behave like the real one, in order to teach the causal relationship between the operator actions and the system response. It’s evident that all that real-time changes caused by trainee’s operations are transients which can be represented only with a dynamic simulation.

5.2 LABORATORY MAIN COMPONENTS

Before starting the description of the dynamic simulation, it can be interesting to describe the principal constituents of the *AG2S* laboratory built at Politecnico di Milano. They are also briefly described in the 3D model description, but in this case the overview will follow the real process flow.

5.2.1 Sources

Obviously the first components to be described are the reactant sources.

In the *AG2S* laboratory at Politecnico di Milano there are three different gas bottles for H_2S, CO_2 and N_2 (used as inert). As inlet of the process is used this fictitious mix of gases: 5% of H_2S, 5% of CO_2 and 90% of $N_{2.}$ At the moment so, the attention of the experimental tests is focused on the behaviour of the reactants of the desired reaction, without considering the influence of other species which are present in the real industrial tail gases like other contaminants or steam water.

The gas bottles are collocated at an external rack of the building because of safety reasons; in fact the gases are under pressure (6 bar), and the internal collocation of the bottles lead to specific precaution like armored cabinets which are not necessary with the external one following the safety legislation.

5.2.2 Fluximeters

As seen, there are three different sources. However, the feed provided to the reactor is a fixed mix of the three components. It’s then necessary to use a flowrate controller to have always the same composition entering the reactor. Fluximeters are used for this purpose. They globally consist of three digital flowmeters, which measure the flowrates from the sources and send the information to the controller, which elaborate the response to manipulate them; besides, there are also three plug valves, which can be manipulated manually to block the flowrates from the sources; the final goal is to feed the exact percentage of each compound into the reactor.

The flowmeters and regulators chosen for the *AG2S™* process laboratory scale realization are the elastomer sealed, digital thermal mass flowmeters, Brooks® Instrument SLA5800 series.

Figure 65 – Brooks® Instrument fluximeter used in the laboratory

They are able to reach accuracy with less than 1% of uncertainty, and a very fast response time linked to the set point controller Brooks® Instrument 0250 series.

Figure 66 – Brooks® Instrument set point controller

The plug valves used in the laboratory are Swagelok P4T series, represented in the figure below:

Figure 67 – Swagelok plug valve used in the laboratory

5.2.3 Reactor and furnace

The crucial components of the laboratory are the reactor and the high temperature tube furnace, which are necessary to produce the optimal condition for the *AG2S™* process desired reaction. The main characteristic that reactor must have, apart from its working length which is 800mm, is the resistance to the high temperature stress, and the possibility to provide three outlet streams, one for the microgc analysis, one for the thermocouple measures, and obviously the product flow exit. The first characteristic is ensured by the construction material chosen, which is quartz, while the second by the particular shape of the reactor, which is evident in the AutoCAD preliminary work figure below.

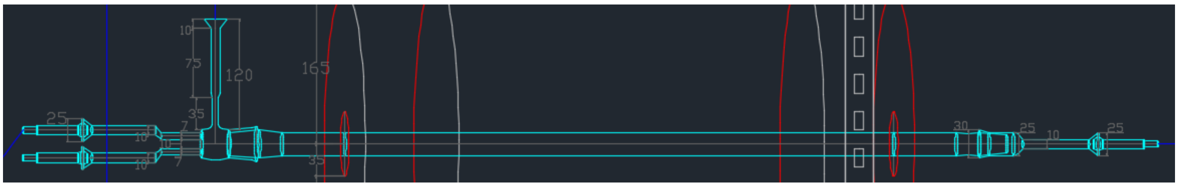


Figure 68 – reactor shape in AutoCAD

The outlet part of the reactor, which is the left section of the figure, shows the three exits described before. The upper tube connects with the microgc analyzer; the other two tubes connect with the rest of the process and the thermocouple (not present in the figure).

The tube furnace chosen is the Carbolite® Gero series, which is able to reach 1500°C of maximum temperature using with 180mm of isothermal section.

Figure 69 – tube furnace Carbolite® Gero series

5.2.4 First trap: sulfur recovery unit

After the reactor unit there is the separation section. The product exiting the reactor will be principally composed of the main products, CO, H_2, H_2O and sulfur, and the unreacted compounds, H_2S and CO_2. In the laboratory environment, the need is to remove from the process outlet the toxic H_2S and to recover the sulfur in gas phase at high temperature exiting the reactor. This last operation is made in the first trap.

The first trap is simply a jacketed steel drum with mains water, which enables to low the temperature of the products and so condense the S_2 gaseous phase of sulfur. The temperature can be taken under 100°C to perform also the product water condensation; in this case the sulfur, which is under its melting point in atmospheric pressure of 115°C, becomes solid.

5.2.5 Second and third traps: H_2S removal

In the following two traps, the goal is to remove the H_2S residuals from the outlet stream. This is realized with the use of two flasks filled with an aqueous solution of NaOH and NaOCl following the study of *Luke Chen, James Huang and Chen-Lu Yang (New Jersey Institute Technology and University of Taiwan).* The reactions involved realize the absorption of H_2S in NaOCl caustic aqueous solution:

$$H_2S_{(g)} + NaOH_{(aq)} = Na_2S_{(aq)} + 2H_2O_{(l)}$$

$$Na_2S_{(aq)} + 4NaOCl_{(aq)} \rightarrow Na_2SO_{4(aq)} + 4NaCl_{(aq)}$$

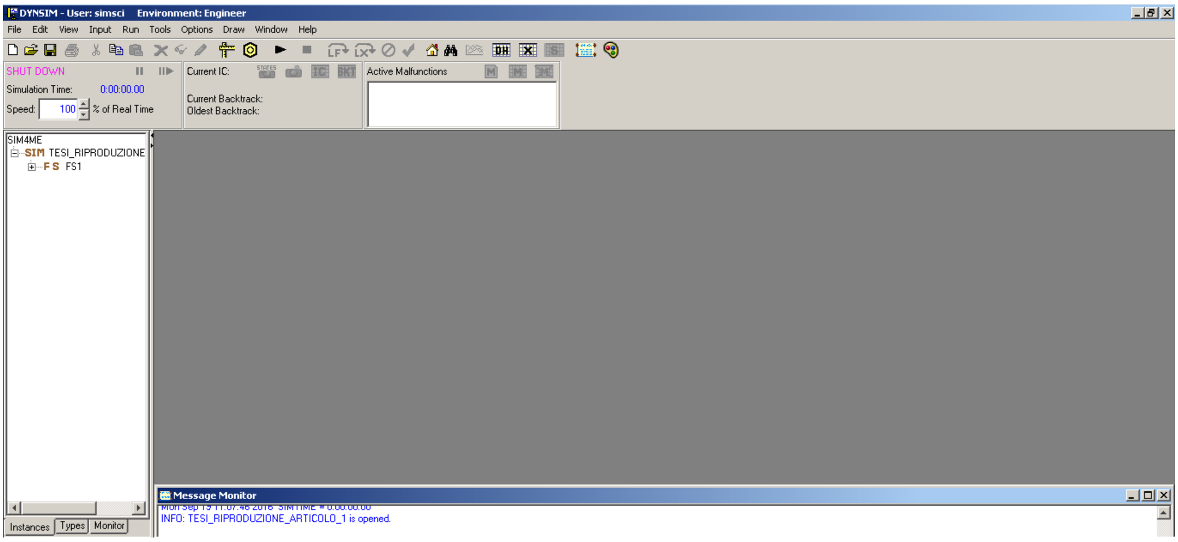
The first reaction is driven to the right as the pH of the solution increases, i.e. as more caustic is added, and to the left as the pH is decreased. This means that if the pH is allowed to decrease, the trapped H_2S starts to be released. The adding of NaOCl in the solution enables to perform an irreversible reaction in order to prevent this possible hazardous scenario.

The experimental conversion described in the article presents a minimum value of 80% (the worst condition) with a pH over than 10. This conversion enables, with the estimated quantities exiting the reactor, to perform a good H_2S removal, using two consecutive units.

The outlet of the last trap goes directly to the top of the fume hood, in order to have the least possible dispersion of the final products in the environment.

After the main laboratory component description, it’s possible to go deeper in the Dynsim environment and simulation analysis.

5.3 DYNSIM™ WORKSPACE



DYNSIM - User: simsci Environment: Engineer

SHUT DOWN
Simulation Time: 0:00:00.00
Speed: 100 % of Real Time

Current IC:
Current Backtrack:
Oldest Backtrack:

Active Malfunctions

SIM TESI_RIPRODUZIONE
FS FS1

Message Monitor

```
Mon Sep 19 11:07:46 2016  SIMTIME = 0:00:00.00
INFO: TESI_RIPRODUZIONE_ARTICOLO_1 is opened.
```

Figure 70 – Dynsim™ workspace

The workspace represents the model data and illustrates the software competences. Dynsim™ is organized in ‘Simulations’, which represent the entire plant project, consisting of different ‘Flow Sheets’, which are the plant sections. In this work it’s used only one flowsheet because of the small size of the plant.

The flowsheet is the real space where it’s possible to reproduce the desired process simulation, in this case the laboratory dynamic simulation.

5.3.1 Workspace components: Unit of Measure and Chemistry

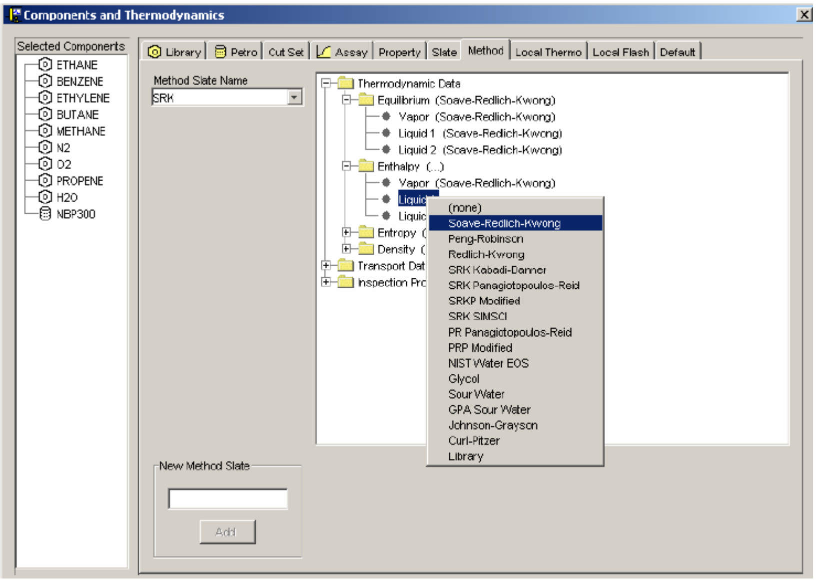
To provide a prediction of the time dependent behavior of the process, Dynsim requires some initial specifications. These specifications are:

- the units of measure chosen (symbol)
- the chemical components to be used (symbol)

These two specifications refer to the whole ‘Simulation’, and not the single ‘Flow Sheet’. In this way there is no need to specify them for all the flowsheets, but it’s necessary only once.

It’s also important to point out that it’s possible to change the reference units of measure for each specific equipment, so the preliminary choice is to assign default units of measure. As regard chemical components, there could be some compounds which are present only in a small area, and they aren’t influent in the other sections. In laboratory process for example the aqueous solution of NaOH and NaOCl is used in the last traps to remove H_2S from the outlet stream. In this case NaOH and NaOCl must be specified in the Chemistry of the plant; the system recognizes them as process compounds, and in the default initialization includes them in all the process equipment in an equimolecular fraction. It’s then crucial, due to the system structure, to make a good initialization of all the equipment in the process, in order to have the right compounds in the right laboratory components; in fact in this work example it’s realistically unfeasible to have traces of NaOH or NaOCl in the reactor, so this must be specified in the initialization tab that will be introduced in the following chapters.

In the ‘chemical components’ tab it’s also necessary to choose an appropriate thermodynamic for the considered system. Dynsim™ software contains lots of different thermodynamics; the user must choose the best one compared to the process to be modelled.



Method Slate Name: SRK

Thermodynamic Data
Equilibrium (Soave-Redlich-Kwong)
Vapor (Soave-Redlich-Kwong)
Liquid 1 (Soave-Redlich-Kwong)
Liquid 2 (Soave-Redlich-Kwong)
Enthalpy (...)
Vapor (Soave-Redlich-Kwong)
Liquid
Liquic
Entropy (
Density (
Transport Dat
Inspection Pro

Figure 71 – example of thermodynamic choose in Dynsim

5.3.2 Workspace components: Icon Palette

The ‘Icon Palette’ (symbol) is the most important component of the workspace used to create the Dynamic Simulation of the AG2S™ process. In the figure below it’s possible to observe the ‘Icon Palette’ structure:

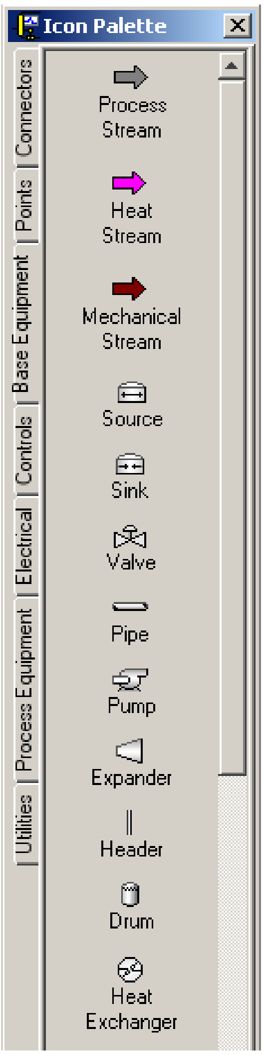


Figure 72 – icon Palette: base equipment

The ‘Icon Palette’ contains all the fundamental instruments to reproduce a chemical process.

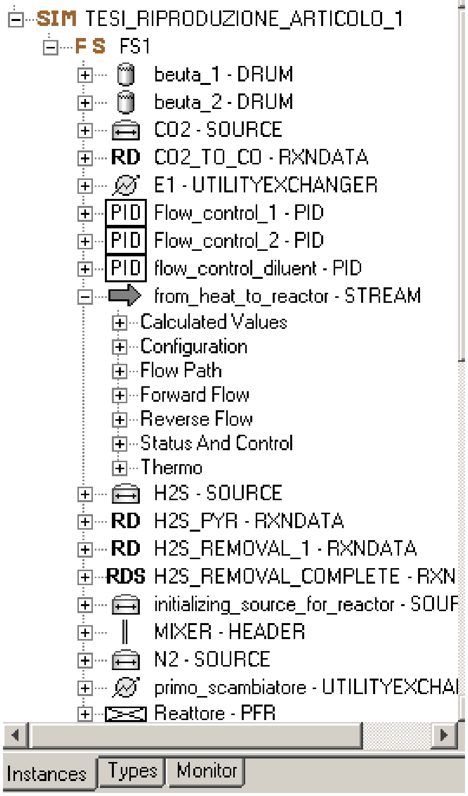
It’s divided in different sections, which correspond to particular categories of instruments of the plant:

- Connectors: used to create connections between parameters, i.e. controlled variables with controllers, or controllers with manipulated variables;
- Points: they are not used in this work, but they can be dynamical, static, graphic points with different proprieties, like floating point, integer or Boolean;
- Base Equipment: used to model the basic components of a chemical process. In particular in this section are included all the streams except for the electrical stream, which anyway is not necessary in this project, valves, sources, sinks and heat exchangers.
- Controls: used to create PID controllers but also all the other possible logic functions like counters, summations, balancer, and/or and also more complex logical functions.
- Electrical: used to insert electrical streams, motors and other electrical components in the flowsheet;
- Process Equipment: it contains advanced equipment like multi-exchangers, combustors, legacy columns and towers useful to model distillation columns, and also plug flow reactor model and reaction data sets;
- Utilities: enables to use useful functions to threat the process data like Gaussian distribution, custom malfunctions and process lag.

Then it’s evident that ‘Icon Palette’ is the instrument which contains all the process modelling possibilities. For this reason, a deeper description of the instruments contained in the ‘Icon Palette’ and used in this project is necessary. They will be presented in the next chapter, after the completion of the workspace description.

5.3.3 Workspace components: Instance tree

Another important component of the workspace, present in the home screen of Dynsim, is the ‘Instance tree’.



- SIM TESI_RIPRODUZIONE_ARTICOLO_1
  - FS FS1
    - beuta_1 - DRUM
    - beuta_2 - DRUM
    - CO2 - SOURCE
    - RD CO2_TO_CO - RXNDATA
    - E1 - UTILITYEXCHANGER
    - PID Flow_control_1 - PID
    - PID Flow_control_2 - PID
    - PID flow_control_diluent - PID
    - from_heat_to_reactor - STREAM
      - Calculated Values
      - Configuration
      - Flow Path
      - Forward Flow
      - Reverse Flow
      - Status And Control
      - Thermo
    - H2S - SOURCE
    - RD H2S_PYR - RXNDATA
    - RD H2S_REMOVAL_1 - RXNDATA
    - RDS H2S_REMOVAL_COMPLETE - RXN
    - initializing_source_for_reactor - SOUF
    - MIXER - HEADER
    - N2 - SOURCE
    - primo_scambiatore - UTILITYEXCHAI
    - Reattore - PFR

Figure 73 – example of Instance tree

It’s possible to see from the figure above that instance tree is the system scheme of the process. Like a tree consists of a principal body which is the simulation, then it’s divided in all the flowsheets present in the simulation project, and further in all the components which characterize the single flowsheet; finally, every component is ‘branched’ in all its proprieties. In the picture, the process stream ‘from_heat_to_reactor’ is highlighted: it’s possible to observe its calculated values i.e. the flowrate, the temperature and the composition, or the configuration, flow path or reverse flow proprieties.

Through the instance tree it’s also possible to directly observe the trend of a particular propriety of the process in the home screen during the simulation. In fact, it’s only necessary to choose the variable to observe within the instance tree and drag it into the desired flowsheet to see the variable behavior during the process dynamic. This can be done both in ‘Shut Down’ simulation phase and ‘Run’ simulation phase. It will be clear in the next chapters that these two phases are completely different each other, so only few actions are allowed in both of them.

5.3.4 Workspace components: Engineer toolbar

The engineer toolbar contains all the instruments to manage the start-up, the running time and the shut-down of the dynamic simulation. In particular the most important are:

- Start engines in freeze mode (► symbol): it stars the simulation in freeze mode. In this way all the equipment are stalled and can be managed in 'Dynamic mode'. It's now possible to manage manually the opening of the valves if necessary, and fulfill actions to the dynamic system without the effects of the running simulation.
- Stop the simulation (■ symbol): it shuts down the simulation. In this way the process is reset to the initial condition (time zero).
- Trends button (📈 symbol): it allows real-time plotting of point values against time during the simulation run.
- Simulation Status Panel: when the simulation is started by the 'Start engines' buttom, this panel indicates if it's in run or freeze mode. It also indicates how long the simulation has been running, and enables to temporarly freeze or resume it. It's also possbile to control the speed of the process running.

RUNNING : G2_Deethan...
Simulation Time: 0:01:48.00
Speed: 100 % of Real Time

Figure 74 – example of Simulation Status Panel

- Snapshot Panel: it includes buttons for saving and restoring Initial Conditions (ICs) and Backtracks. ICs and Backtracks can be started from a displayed list.

Current IC:
Current Backtrack: 1
Oldest Backtrack: 1

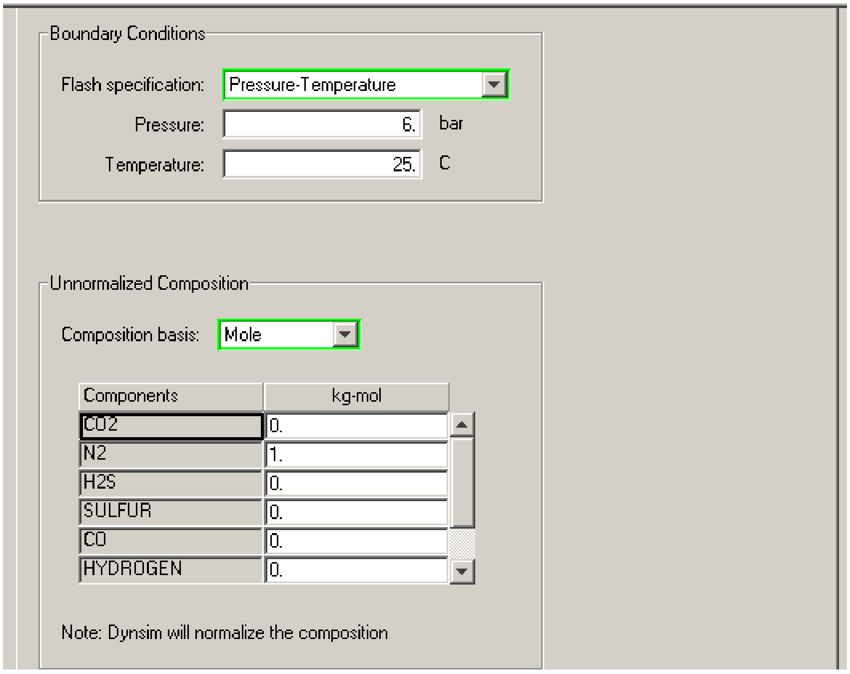
Figure 75 – example of Snapshot Panel

5.4 DYNAMIC SIMULATION REALIZATION: DYNSIM

Using the instruments presented in the last chapter, it’s possible to reproduce the *AG2S*™ laboratory structure and perform the dynamic simulation of its behaviour.

5.4.1 First section: before reactor components in Dynsim

Following the same description scheme of the last paragraph, the designing of the laboratory in Dynsim starts from the sources. Three sources must be created, like the real three different gas bottles.



Boundary Conditions

Flash specification: Pressure-Temperature

Pressure: 6. bar

Temperature: 25. C

Unnormalized Composition

Composition basis: Mole

Components	kg-mol
CO2	0.
N2	1.
H2S	0.
SULFUR	0.
CO	0.
HYDROGEN	0.

Note: Dynsim will normalize the composition

Figure 76 – N_2 source modelling

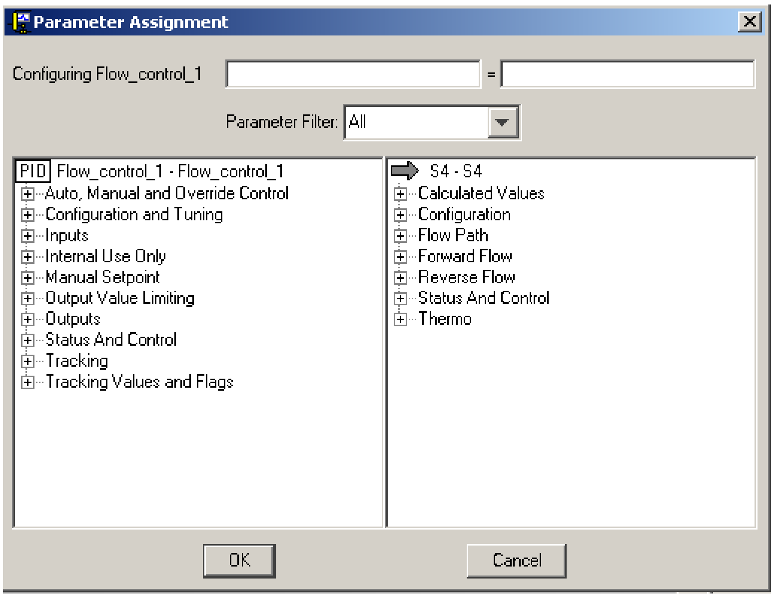
The source modelling is simple; it only asks to know the pressure, temperature and composition of the source. Composition and other proprieties can be specified in lots of different unit of measures, but the standard one are those chosen in the preliminary specifications.

After the sources modelling, the fluximeters section must be created. To reproduce the gas lines in Dynsim ‘Process Stream’ in the Icon Palette must be selected. The process streams must be connected to their inputs and outputs: for all the gas lines the input equipment is one of three sources created before, while the output is one of the three valves which represent the manipulated part of fluximeters.

The control system in Dynsim consists of three main components: the valve (manipulated object), the controller and the connectors between controller and the valve. To create the valve is used the ‘Valve’ symbol in the Icon Palette; similarly for what regards the controller creation, in this case using ‘PID’ button. To connect the controller and the valve, ‘connectors’ are needed; even the connectors are present in the Icon Palette, and they are used to specify the input variable to the controller and the manipulated variable of the controlled equipment. Obviously, the possible measured variables in a process stream, and manipulated variables in process equipment are many; for this reason Dynsim™ system eases the user providing an automatic choice of the right parameters in the connection phase for common applications. In fact two types of connectors are available in the Icon Palette: the ‘connectors’, and the ‘default connectors’, which enable Dynsim to choose automatically the measured and manipulated variables.

This is possible because the creation of the controller needs specific about the nature of it, i.e. temperature, pressure, flowrate, composition or volume flowrate controller. In this way, for common applications like a volume flowrate control using a manipulated valve to manage it, the ‘default connector’ between the process stream and the controller directly choose the volume flowrate as the measured variable associated to the set point of the controller, and the ‘default connector’ between the controller and the valve directly choose the valve opening as the manipulated variable submitted to the controller output.

In more complex situations, direct linking isn’t possible; in this case ‘connector’ must be selected in the Icon Palette. This enables to choose exactly the desired measured variable and its corresponding parameter in the connector, and the same for the manipulated variable and controller action. In the next page an example of the ‘connector’ display is reported.



Parameter Assignment

Configuring Flow_control_1 =

Parameter Filter: All

Figure 77 – choice possibility using ‘connector’

In the fluximeter section is sufficient to choose a volume flowrate controller and connect it with the process streams and manipulated valves using a ‘default connector’.

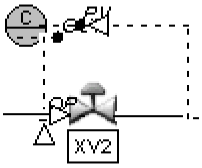


Figure 78 – fluximeter reproduced in Dynsim

The volume flowrates chosen for the experimental laboratory go from 10 l/h to a maximum 30 l/h. Fluximeters in the real process and flowrates control in Dynsim enable to change the inlet flowrates simply modifying the controller set point. The data of valves cv and controller parameters are adjusted to reach a good flowrate control in a small time. The Dynsim default values for valves, i.e. cv = 100, are huge respect to the final chosen values (cv = 0.11); the reason is the same of the Smart3D dimensional problems: Dynsim™ system is specialized in industrial process simulations, so its dimensional guess values are sized for a different application. Otherwise, in this case there are no limitations in equipment design, so there isn’t any problem to realize a dynamic process simulation in laboratory scale.

After the fluximeters section a mixer is required to converge the three different flowrates into the only input of the reactor. The mixer in Dynsim is realized with ‘Header’ base equipment, which allows the user to provide many inputs and one output calculating the resulting flowrate.

Following the mixer, in the first laboratory modeling a metal mass was put to pre-heat the charge. Substantially, the first reactor design tried in Dynsim simulations has been an adiabatic PFR, so the real isothermal reactor was modelled using two different unit operations: the heating metal mass to reach the desired temperature, and the PFR adiabatic reactor to perform the reactions; unfortunately, this reactor design provided not accurate results, because of the temperature decrease during the reactions in the adiabatic configuration. The choice to separate the reactor in smaller adiabatic reactors with heating metal mass in between has been excluded because of its excessive difference to the real reactor, so the final decision has been to model an isothermal reactor in Dynsim using only the PFR equipment, without fictitious pre-heating sections. The isothermal PFR model will be presented in the next paragraph.

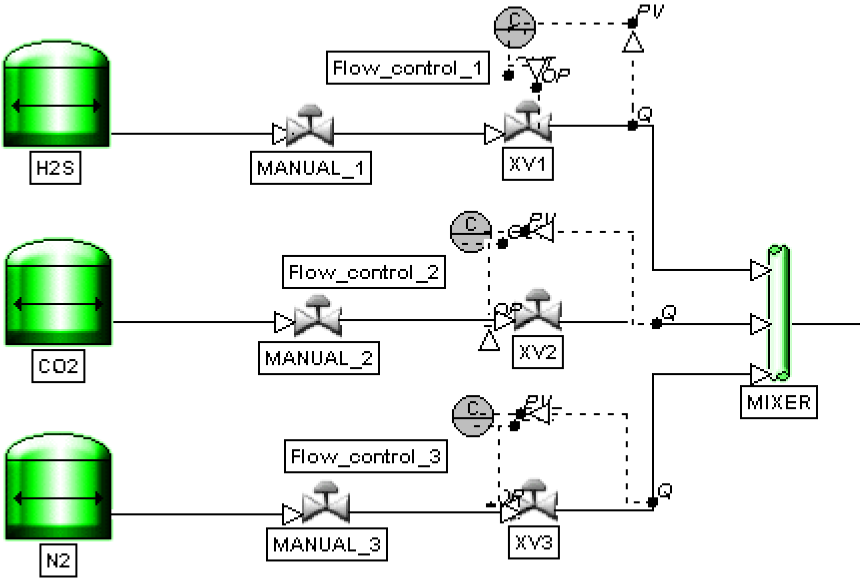


Figure 79 – pre-reactor section in Dynsim

In the figure is represented the final appearance in Dynsim of the pre-reactor section. It’s important to observe that before the fluximeters section manual valves are present, which are not manipulated from the controller. For those valves is sufficient to create them using ‘Valve’ command in the Icon Palette, and during the simulation phase they can be manipulated to guarantee a normal process running in safe conditions, either an immediate closing in an unsafe conditions simulation.

5.4.2 Reactor section introduction

The reactor section is the most crucial part of the simulation.

Dynsim provides the possibility to model PFR (plug flow reactor) using the ‘PFR’ task in the Icon Palette. The first choice to be made is the number of flow passes, reaction passes, wall and discrete elements per wall. All of these characteristic are fundamental to accurately simulate the real tube reactor, so they will be explained in detail in the following paragraphs.

5.4.3 Initial reactor design: PFR in Dynsim

Dynsim asks for preliminary proprieties of the reactor before passing to the specific modelling.

The first propriety is the number of flow passes: are considered as flow passes all the material fluxes not associated with the main feed straight one path. Flow passes can be the coolant flows, exchangers feed gases or even main feed annular flows.

The second propriety is the number of reaction passes, that can represent a catalyst bed or simply a fictitious pass in which the reactions occur.

The third propriety is the number of wall which separates the internal section of the reactor to the ambient. The number of walls increases with a jacketed reactor, in which there is not only the reactor shell, but even coolant/heating tubes or shells.

The last propriety is the number of elements per wall. As the *Dynsim process equipment user guide* explains: ‘PFR’s reaction pass sub-model includes a series of fluid holdups called elements. The total reaction pass volume is divided into equal elements to model heat transfer and reaction. All state variables within an element are assumed to be uniform.’

Practically, Dynsim solves the PFR like a series of CSTR with a volume defined by the number of elements per wall, applying to these reactors the mass, energy and momentum balances. The higher the number of elements per wall, the more accurate but also heavier the calculation.

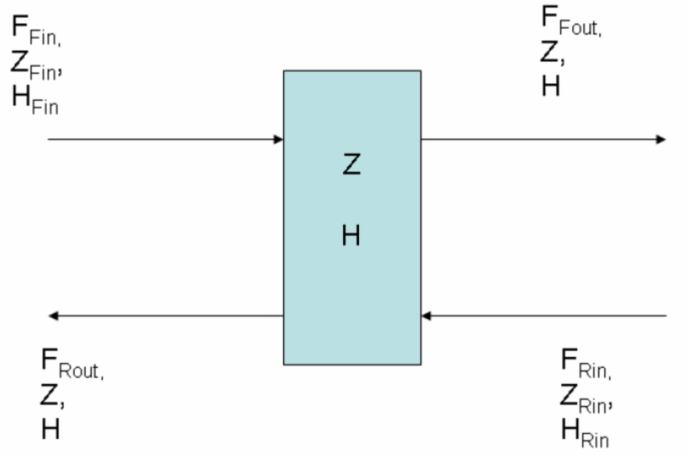


Figure 80 – example of one element of PFR reactor in Dynsim

In this work the initial proprieties chosen for the reactor are the following:

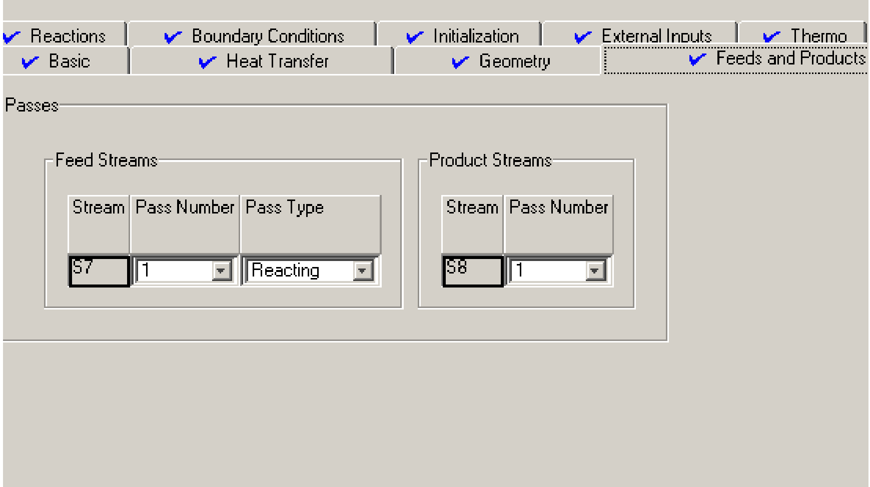
- Flow passes: 0. The reactor is heated by the silicon carbide heating elements, without any heating flow. The main feed flow is straight one.
- Reaction passes: 1. There is the *AG2S*™ process reaction set occurring in the reactor.
- Walls: 1. In this case only the reaction tube is considered; to take into account the external insulation a very low heat loss coefficient to ambient is used.
- Number of elements per wall: 10. This is the default value. In this case the reactor has a very fast kinetic, but even a low effective length (200 mm). Higher values of this propriety are not used to prevent the heaviness of the simulation.

5.4.4 Complete reactor design: PFR in Dynsim

After the initial design, the reactor is ready to be modelled in all its characteristics; the most important design tabs are illustrated in the following:

- Basic: includes reactor mass, volume and heat loss to the ambient; if present, proprieties of the reactor passes are also needed, like flow conductance, reverse flow, fluid volume and void fraction;
- Heat transfer: includes the reaction pass proprieties of heat exchange, i.e. forced and natural convection heat transfer coefficient.

- Feeds and products: depending on the initial choice of flow passes and reaction passes, it’s possible to choose a different number of streams passing through the reactor. In the following figure the configuration of the *AG2S™* process reactor is illustrated:



Passes

Feed Streams

Stream	Pass Number	Pass Type
S7	1	Reacting

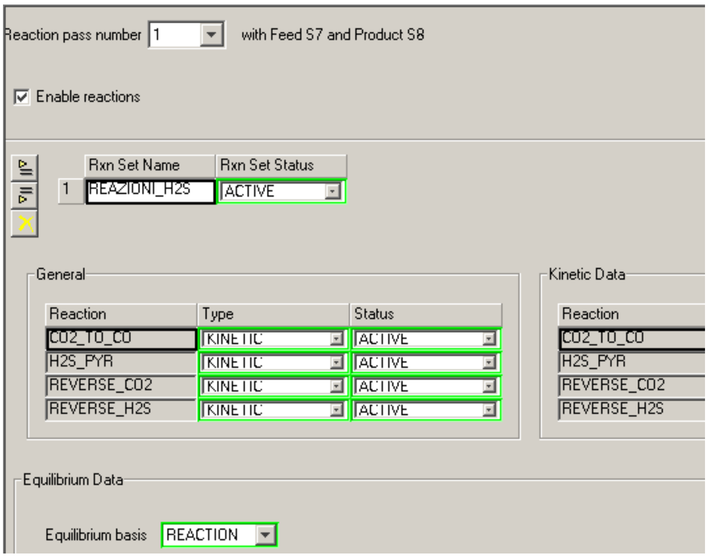
Product Streams

Stream	Pass Number
S8	1

Figure 81 – Feeds and products configuration of the reactor

The stream S7 is the inlet of the reactor, and it’s designed as reacting pass type, while the output stream is identified with the S8 symbol. The reacting pass type identifies the flow associated with the reaction specified in ‘Reactions’ tab;

- Reactions: in the ‘Reactions’ tab the reaction data set can be selected, and automatically all the reactions included are associated to the reacting pass type.



Reaction pass number 1 with Feed S7 and Product S8

☑ Enable reactions

	Rxn Set Name	Rxn Set Status
1	REAZIONI_H2S	ACTIVE

General

Reaction	Type	Status
CO2_TO_CO	KINETIC	ACTIVE
H2S_PYR	KINETIC	ACTIVE
REVERSE_CO2	KINETIC	ACTIVE
REVERSE_H2S	KINETIC	ACTIVE

Kinetic Data

Reaction
CO2_TO_CO
H2S_PYR
REVERSE_CO2
REVERSE_H2S

Equilibrium Data

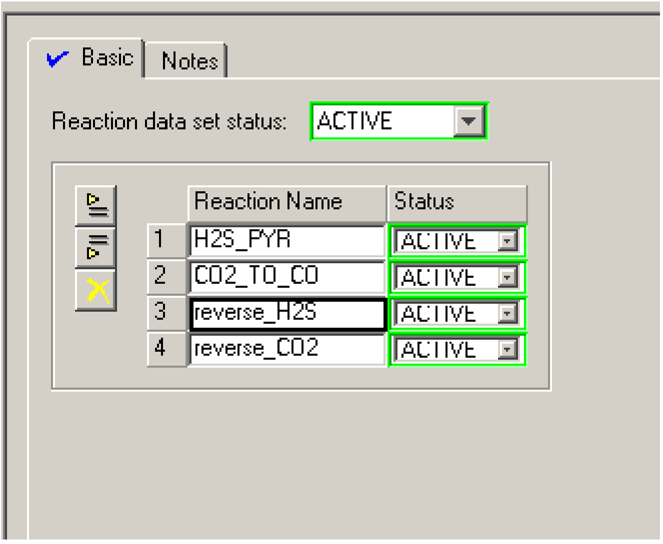
Equilibrium basis REACTION

Figure 82 – ‘Reactions’ tab in the *AG2S™* process

The reaction data set will be better explained in the following paragraph, but substantially it includes all the reactions involved in the considered process. In this case it includes the four reactions, two forward and two reverse, studied in the 'Before process simulation: reaction kinetic' chapter of this thesis work with their kinetic parameters values;

- Initialization: as explained before, to reach easier the steady state solution and mostly not to have wrong components in the equipment at the initial state of the process, it's possible to initialize the reactor (and all the other unit operations) with a specific condition of temperature, pressure and composition. In this case a good initial composition could be the inlet of the reactor one;
- Boundary conditions: in this tab boundary proprieties can be specified, like enthalpy or temperature; this type of condition is a constriction of the equipment to have the selected value of the chosen variable. To perform an isothermal reactor, the temperature boundary condition can be included in the equipment modelling, choosing the fixed value of temperature of the reactor in all its discrete wall elements.

The reactor design is then completed. The only instrument to be better described to fulfill the reactor section presentation is the 'Reaction Data Set', included into the 'Process Equipment' tab in the Icon Palette. It collects all the reaction data of specific reactive equipment. In this case the reaction data set of the reactor is shown.

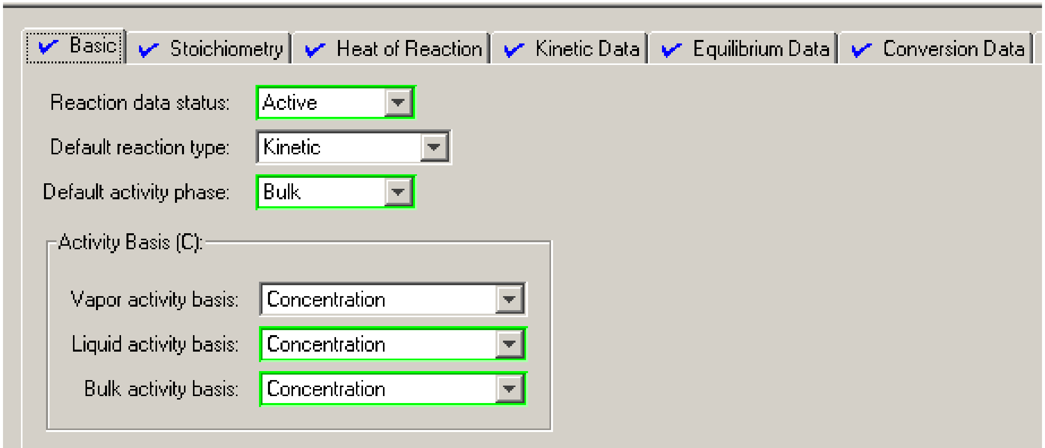


Reaction data set status: ACTIVE

	Reaction Name	Status
1	H2S_PYR	ACTIVE
2	CO2_TO_CO	ACTIVE
3	reverse_H2S	ACTIVE
4	reverse_CO2	ACTIVE

Figure 83 – Reaction Data Set

All the reaction names are associated to the respective ‘Reaction data’, which include the kinetic, conversion and stoichiometric data of the specific reaction.



Reaction data status: Active
Default reaction type: Kinetic
Default activity phase: Bulk
Vapor activity basis: Concentration
Liquid activity basis: Concentration
Bulk activity basis: Concentration

Figure 84 – Reaction data

In the ‘Reaction Data’ the reaction proprieties must be selected; in particular the ‘reaction type’, which represents the way to calculate the reaction products in Dynsim, can be ‘kinetic’, i.e. using the kinetic data inserted in the ‘Kinetic Data’ tab, or equilibrium, using a specific equilibrium constant equation, or directly a conversion equation depending on temperature and defining a key component. In this case the ‘kinetic’ type is selected for all the four reactions involved in the laboratory reactor, inserting the simplified kinetic regression results in the ‘Kinetic Data’ tab.

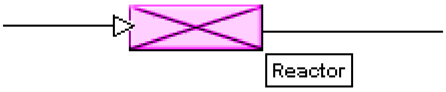


Figure 85 – PFR reactor in Dynsim

This figure shows how the PFR reactor looks like in Dynsim.

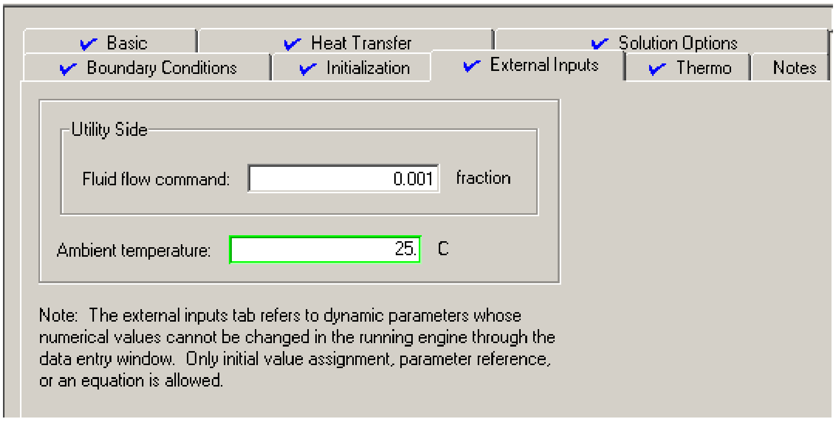
5.4.5 Last section: post reactor equipment

In the post-reactor section, the three traps for sulfur and H_2S recovery must be modelled.

The first trap is reproduced in Dynsim as a ‘Drum’ preceded by a ‘Utility Exchanger’. This is because the ‘Drum’ base equipment in Dynsim is a pressure node object that can be used as two or three phase separation vessel. In the first trap case the drum can perfectly simulate the separation between the vapor, the liquid and the solid phases. In the real laboratory the first trap is jacketed with a water cooling flow in order to reach the suitable temperature to condense sulfur at first, but the real goal is to reach temperature under 100°C to condense even the water product in the process flow. The jacket in Dynsim is replaced by the utility exchanger situated before the first trap. The ‘Utility Exchanger’ is the base equipment which requires only one input and one output process stream, and calculates the heat exchange between this process flow and a fictitious utility stream which doesn’t need to be created in the worksheet. Thus, respect to the ‘Heat Exchanger’, this equipment is used when the process stream exchange heat with an external source used for this purpose but without taking part of the process, exactly the case of the reactor outlet cooled by the laboratory line water.

As regards the utility exchanger, it is considered as base equipment by Dynsim, so the standard data are required to model it:

- Basic: in the first tab some important data are required for the process and utility side. In particular for the process the volume, metal mass and flow conductance must be inserted, while for the utility side the coolant proprieties like inlet temperature, fluid mass and maximum fluid flowrate are required.
- Heat transfer: in this tab the heat transfer proprieties must be compiled. The required data are divided in overall, process side and utility side heat transfer coefficients.
- Initialization: as in the other base and process equipment case, it’s possible to initialize the heat exchanger conditions;
- External inputs: the external inputs in this case are not only the ambient temperature (useful to calculate the heat loss to ambient) but also the fluid flow command.



Fluid flow command: 0.001 fraction

Ambient temperature: 25. C

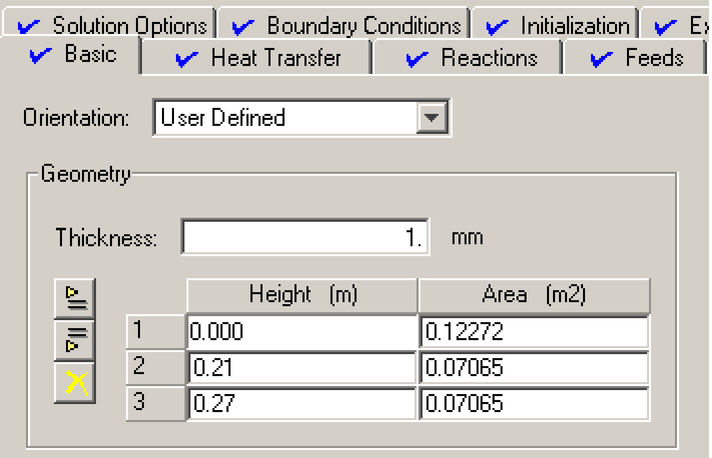
Note: The external inputs tab refers to dynamic parameters whose numerical values cannot be changed in the running engine through the data entry window. Only initial value assignment, parameter reference, or an equation is allowed.

Figure 86 – External inputs in the utility exchanger

The fluid flow command represents the opening of the utility fluid valve, which regulates the line water flowrate depending on the maximum fluid flowrate specified in the ‘Basic’ tab. The fluid flow command it’s a dynamic value which can be modified during the dynamic simulation; in this way the laboratory operator action of regulating the line water valve is perfectly reproduced in Dynsim dynamic simulation manipulating the external input. In the case of a PID controller is required to stabilize the outlet temperature at a fixed value, the ‘Connector’ must be used choosing as controller output exactly the fluid flow command, but in this case in the real laboratory a temperature controller is not expected, and the temperature is manually controlled by the user.

As regards the ‘trap_1’ equipment, being base equipment it has similar modelling characteristics with the reactor and utility exchanger, with ‘Basic’ data focused in this case on the geometry specifications like orientation, length and thickness, the heat exchange specifications like overall heat exchange coefficient and heat loss to the ambient information, the possibility to initialize it and even the ‘Reactions’ tab in case of a reactive drum. This last propriety of the Dynsim drums will be necessary in the case of the last two traps of the *AG2S*™ process, in which the H_2S is removed using NaOH and NaOCl with an absorption reaction.

The second and third traps are reactive drums, with a specific geometry due to the fact that they aren’t cylindrical drums but glass flasks in the real laboratory. To reproduce the shape of the two final traps in ‘Basic’ there is a special orientation called ‘User defined’. This special type of orientation enables the user to customize the shape of the drum selecting the area of the flask at different height levels; in this way Dynsim is able to create the model of complex drums. The ‘trap_2’ specifications are reported in the following together with an extract of the *‘Base Equipment User Guide’* of Dynsim in which the system interpretation of the user defined geometry is presented:



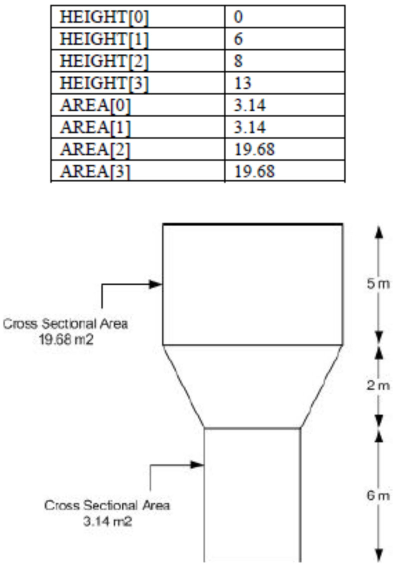
Orientation: User Defined

Geometry

Thickness: 1. mm

	Height (m)	Area (m2)
1	0.000	0.12272
2	0.21	0.07065
3	0.27	0.07065

Figure 87 – user defined orientation of drums in Dynsim



HEIGHT[0]	0
HEIGHT[1]	6
HEIGHT[2]	8
HEIGHT[3]	13
AREA[0]	3.14
AREA[1]	3.14
AREA[2]	19.68
AREA[3]	19.68

Figure 88 – user defined orientation example

As regard the reactions involved in the last two traps, the procedure to include them in the drum simulation is the same as the reactor. First the ‘Reaction Data’ must be created for each reaction involved in the two traps, then the ‘Reaction Set’ is prepared including all the ‘Reaction Data’, and at the end in the ‘Reactions’ tab of the two final traps the reaction set is selected to enable the reactions to be calculated during the dynamic simulation. The only difference between the reactor data set and the H_2S removal reaction set is that in the second case the ‘reaction type’ chosen is not the ‘kinetic’ but the ‘conversion’. This means that, on the contrary of the case of the reactor reactions, the products of the last two traps reactions are calculated by the system using the input conversion data and not the kinetic data. This is due to the fact that, while for the *AG2S™* principal reaction a good theoretical simulation program is available (DSmoke) to forecast the behavior of the compounds within the reactor, thus it’s been possible to perform a kinetic regression of the data in order to find the reactor kinetic, in the H_2S removal traps the reactions data are taken directly from an academic article, the ‘*Absorption of* H_2S *in NaOCl Caustic Aqueous Solution, Chen et Al., New Jersey Institute of Technology and Taiwan University, 2001*’, without the possibility to simulate them. In this case the kinetic data of H_2S removal aren’t available in the article, but are those of conversion which can be used in Dynsim to simulate the reaction. To use the conversion data to calculate the reaction products instead of the kinetic parameters as in the case of the reactor it’s necessary to select, during the ‘Reaction Data’ modelling, in ‘reaction type’ the ‘conversion’ one. In this way, completing the ‘Conversion Data’ tab choosing the base component and the parameters for the conversion equation: $\mathrm{X} = A + B\,T + C\,T^2$ with X representing the base component conversion and the three parameters which allow creating also a temperature dependency of the reaction, it’s possible to perform the dynamic simulation of the two final traps including the H_2S removal reactions in the system using the conversion data. In this case, the conversion is not affected by temperature which is always around the ambient temperature in the final traps, and it’s fixed at 80% for each trap. Actually, higher values of conversion could be reached by the reaction described in the reference article, but the choice is to be conservative in order to calculate the maximum value of H_2S leaving the final trap, being the H_2S a really toxic and dangerous compound. When the laboratory will be perfectly operative, experimental data could be collected in order to find a better correlation for the H_2S removal reaction conversion. In that case, it would be sufficient to change the ‘Reaction Data’ parameters in the Dynsim flowsheet with the new found data to update the reaction information, performing a more accurate simulation of the two final traps of the process.

5.5 COMPLETE PROCESS SCHEME AND RESULTS IN DYNSIM

The complete process scheme in Dynsim is observable in the figure below.

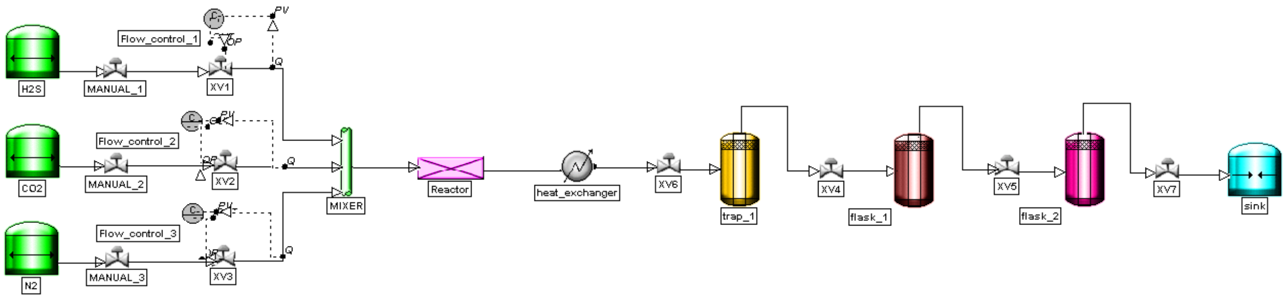
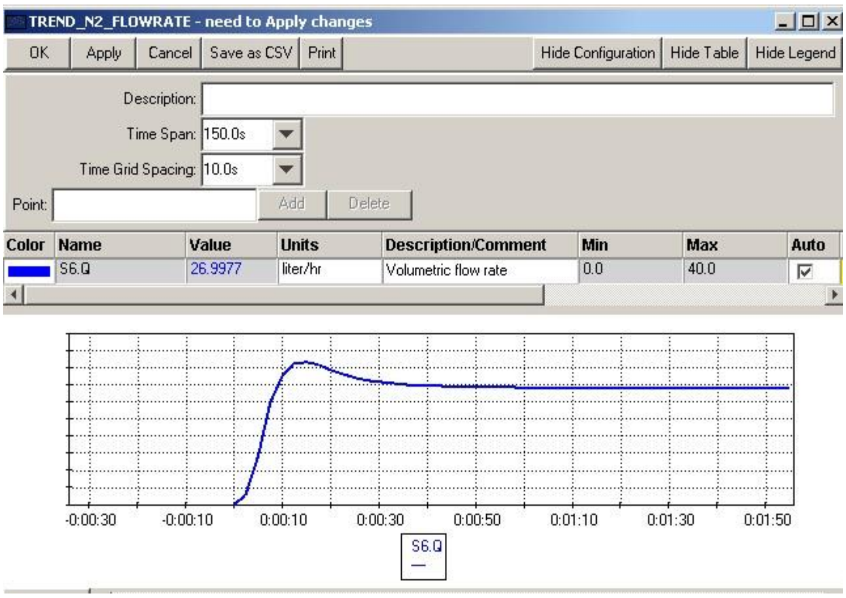


Figure 89 – complete process scheme in Dynsim

The dynamic simulation results can now be presented. At first, using the possibility to plot in real time the variables trend in Dynsim, it’s possible to see the exemplifying case of the volumetric flowrates control fixed at 30 l/h.



Description:

Time Span: 150.0s

Time Grid Spacing: 10.0s

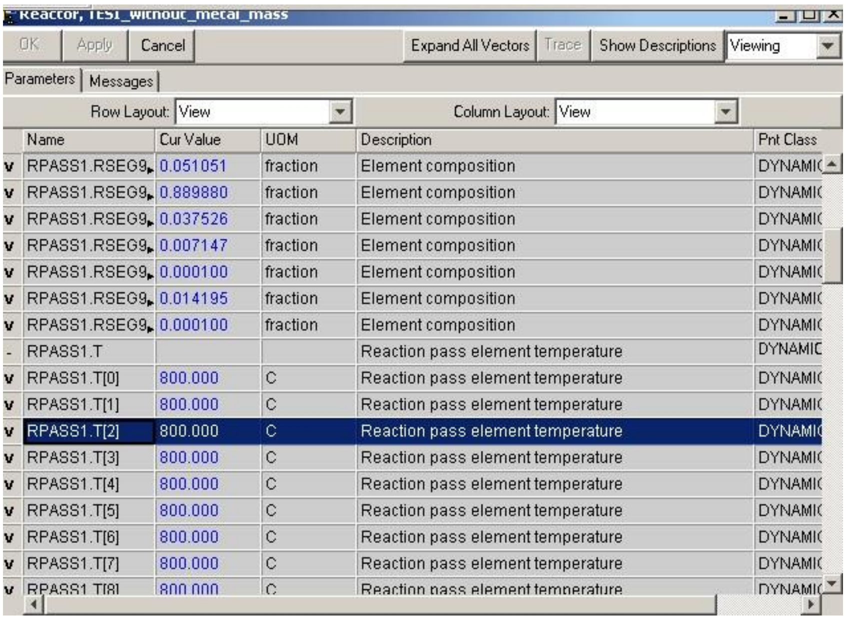
Point:

Color	Name	Value	Units	Description/Comment	Min	Max	Auto
🟦	S6.Q	26.9977	liter/hr	Volumetric flow rate	0.0	40.0	☑

Figure 90 – trend of N_2 flowrate in 30 l/h case

This is how the variables trends appear in Dynsim. The chosen variable value is reported in the plot in real time, and it’s possible to change the time span and the time of grid spacing of the figure. Besides, the current value is reported near the name and the identifying color of the variable. The trends can accept also contemporary variables, and be saved or printed with the commands in the upper part of the window. In this specific case the wanted value for N_2 volumetric flowrate is 27 l/h, and it’s possible to see that the set point value is perfectly reached by the measured variable.

As regards the reactor solutions, it’s possible to observe the changing of the molar fractions in trends like before, or to directly clicking on the reactor. The window that appears shows the dynamical changes in the reported variable; the figure below represents the dynamic window of the equipment:



Row Layout: View

Column Layout: View

	Name	Cur Value	UOM	Description	Pnt Class
v	RPASS1.RSEG9	0.051051	fraction	Element composition	DYNAMI
v	RPASS1.RSEG9	0.889880	fraction	Element composition	DYNAMI
v	RPASS1.RSEG9	0.037526	fraction	Element composition	DYNAMI
v	RPASS1.RSEG9	0.007147	fraction	Element composition	DYNAMI
v	RPASS1.RSEG9	0.000100	fraction	Element composition	DYNAMI
v	RPASS1.RSEG9	0.014195	fraction	Element composition	DYNAMI
v	RPASS1.RSEG9	0.000100	fraction	Element composition	DYNAMI
-	RPASS1.T			Reaction pass element temperature	DYNAMIC
v	RPASS1.T[0]	800.000	C	Reaction pass element temperature	DYNAMI
v	RPASS1.T[1]	800.000	C	Reaction pass element temperature	DYNAMI
v	RPASS1.T[2]	800.000	C	Reaction pass element temperature	DYNAMI
v	RPASS1.T[3]	800.000	C	Reaction pass element temperature	DYNAMI
v	RPASS1.T[4]	800.000	C	Reaction pass element temperature	DYNAMI
v	RPASS1.T[5]	800.000	C	Reaction pass element temperature	DYNAMI
v	RPASS1.T[6]	800.000	C	Reaction pass element temperature	DYNAMI
v	RPASS1.T[7]	800.000	C	Reaction pass element temperature	DYNAMI
v	RPASS1.T[8]	800.000	C	Reaction pass element temperature	DYNAMI

Figure 91 – Reactor dynamic window

It’s possible to observe the temperature at each reaction pass, in this case fixed to 800°C (1073K), and the elements compositions in the last section of the reactor. The same window would appear, with its characteristics dynamic variables, for each stream or equipment with double clicking their symbol. In fact, as described before, the dynamic and shutdown phases are completely separate in Dynsim, with the possibility to model the process in shutdown phase, while in dynamic phase the opportunity to follow the dynamic of the system and change some manipulating value, like the valve opening in manual conditions.

Finding the desired values of the compounds in the reactor window it’s possible to calculate the resulting conversions and to compare them with DSmoke with simplified kinetic results.

H_2S CONVERSION RESULTS

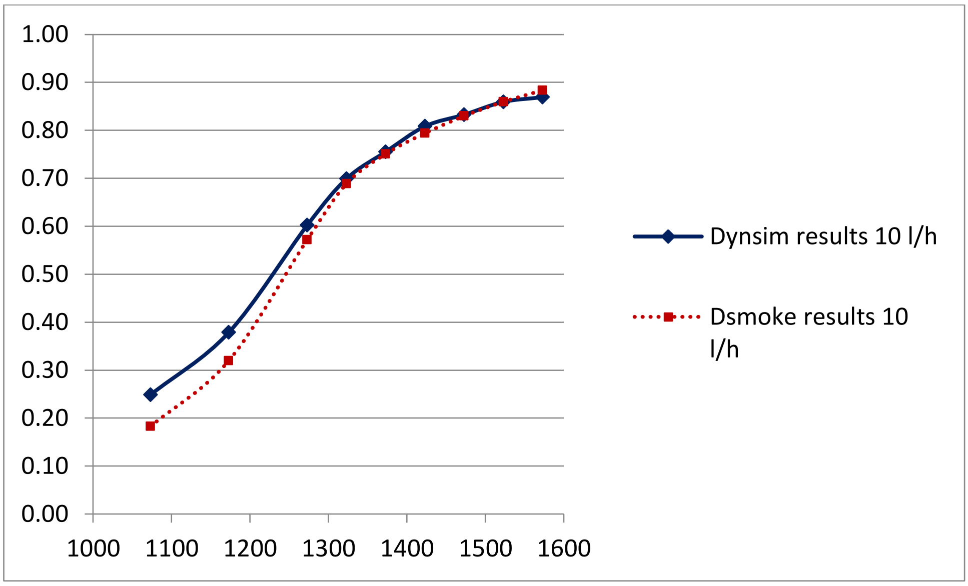


Figure 92 - H_2S conversion results at 10 l/h

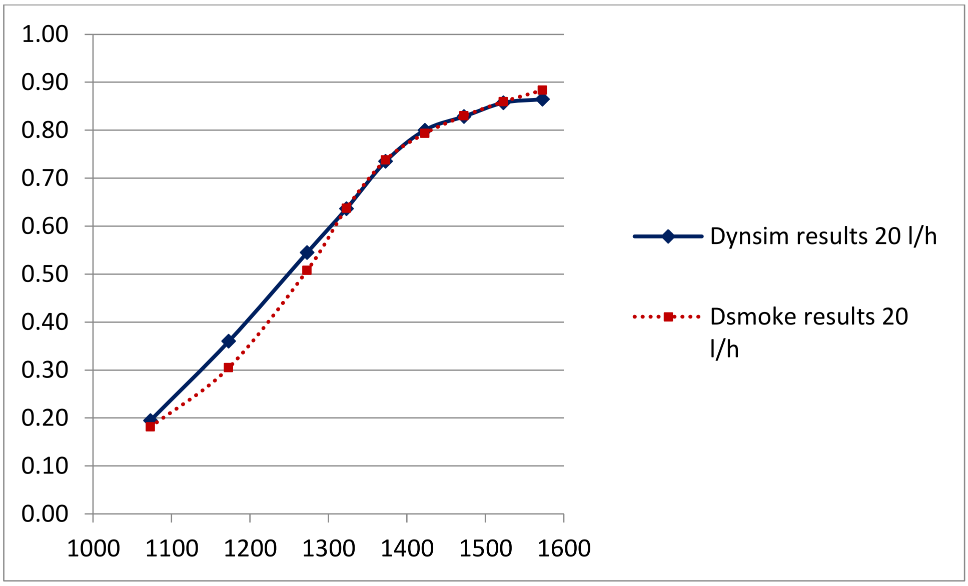


Figure 93 - H_2S conversion results at 20 l/h

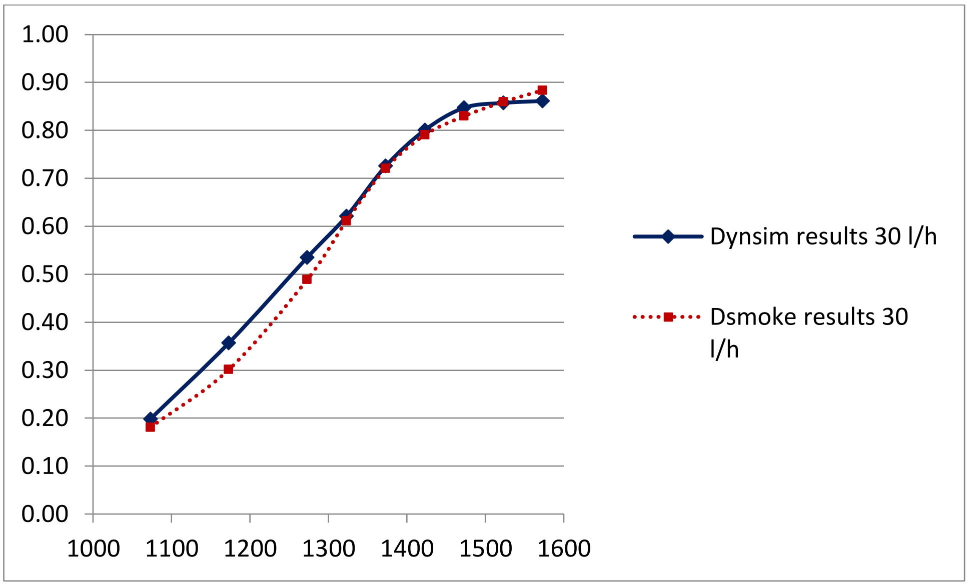


Figure 94 – H_2S conversion results at 30 l/h

CO_2 CONVERSION RESULTS

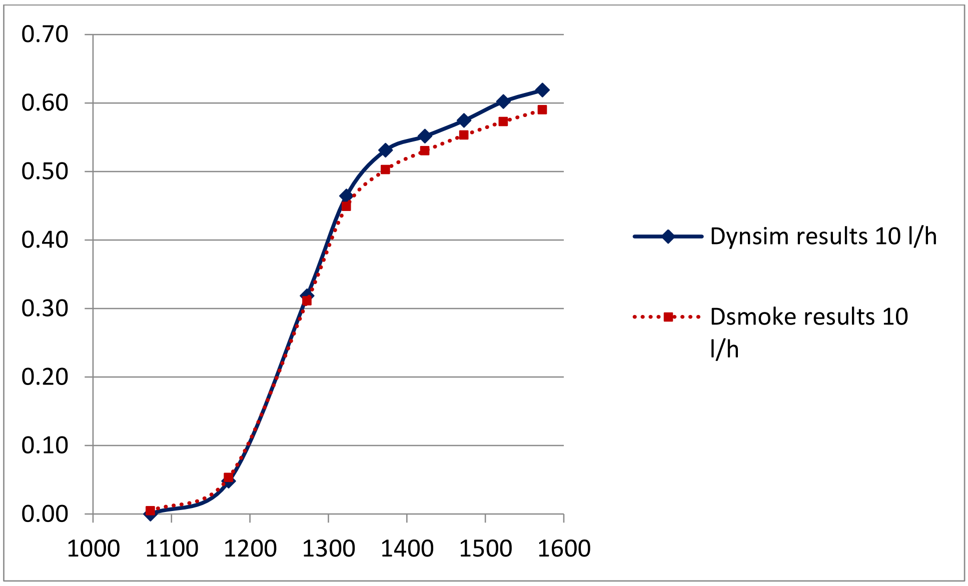


Figure 95 - CO_2 conversion results at 10 l/h

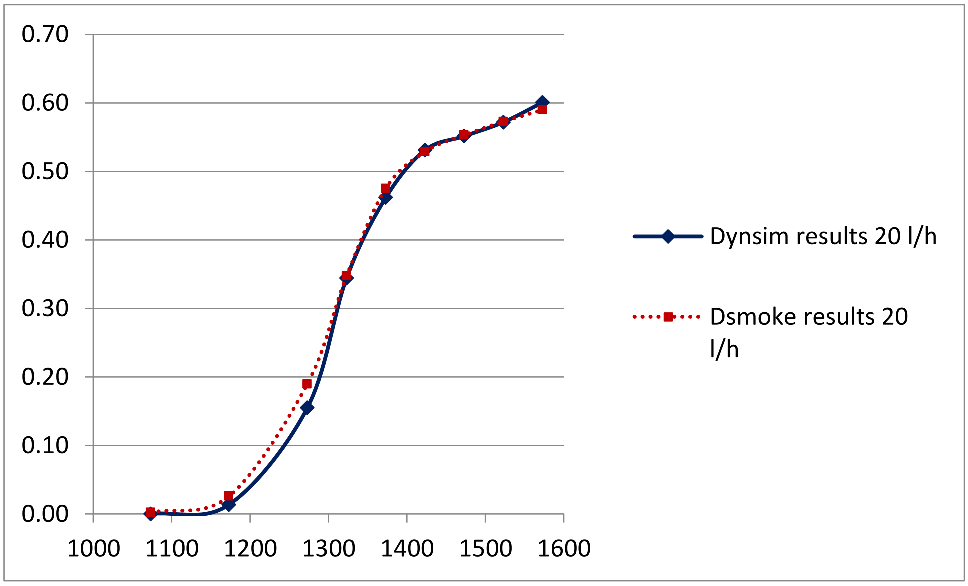


Figure 96 - CO_2 conversion results at 20 l/h

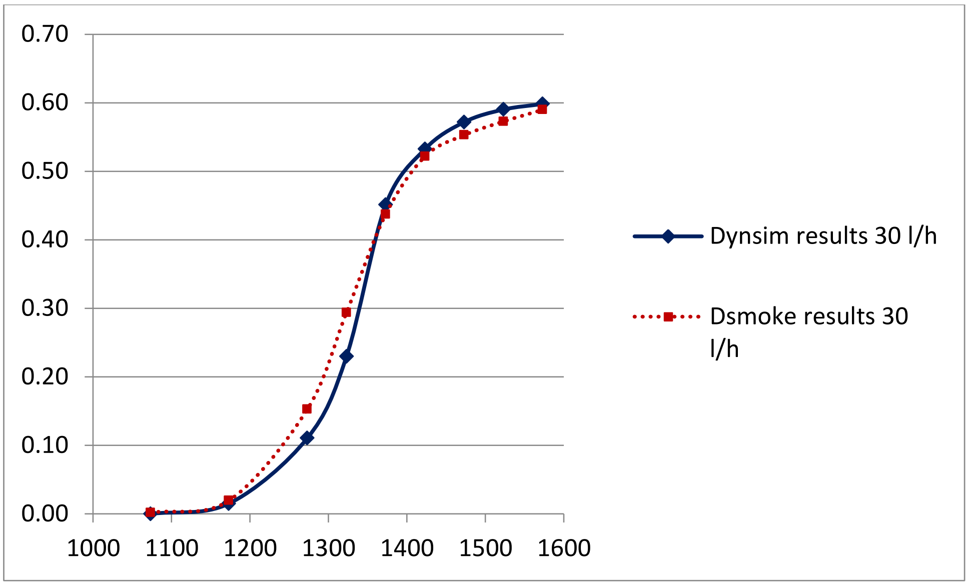


Figure 97 - CO_2 conversion results at 30 l/h

The results are evidently consistent with the DSmoke simplified model; the small variations in the output data can be attributed to different calculation method in the two programs and to dynamic changes during the Dynsim simulation which are not present obviously in the DSmoke calculations, but globally the simulation results are comparable each other. In general the H_2S conversion trends show slightly higher conversions at low temperatures and a maximum conversion at high temperature which tends to flatten near the 1523K value, while the CO_2 conversion is practically superimposable at low temperatures, while at high temperature there is a slight difference with higher values in Dynsim simulations.

As regard the subsequent unit operations, the heat exchanger can be manipulated to reach temperatures below 200°C (even under 100°C if required) to perfectly condense the sulfur, which has 0 molar fraction value at the ‘trap_1’ outlet for every condition. The second and third traps use a simple fixed conversion value; the H_2S outlet values are dependent from the chosen conversion in the reaction data set, which in this case is fixed at 0.8.

The dynamic simulation description is so concluded. The next chapter describes the last section of the thesis work, realized at the TU Berlin, which focuses on MOSAIC simulation environment.

6. MOSAIC: A NEW MODELING, SIMULATION AND OPTIMIZATION ENVIRONMENT

6.1 INTRODUCTION OF MOSAIC

Mosaic is a web-based modeling, simulation and optimization environment created at the TU Berlin (*Technische Universität Berlin*).

Based on LaTeX-style entry method for algebraic and differential equation, it’s possible to model the desired process using equation systems, which can be subsequently simulated or optimized by MOSAIC.

The best quality of this environment is the capability of working with equations or equation systems, fitting the languages to the user needs. In particular, with MOSAIC it’s possible to build equation systems using equations with different notations each other; besides, once modelled the system, MOSAIC enables the user to automatically generate the simulation or optimization code for numerous environment, such as AMPL, Aspen Custom Modeler, GAMS, gPROMS, MATLAB, Modelica, and for solvers interfaced via C++, FORTRAN, Python, etc. The simulation or optimization results can then be imported to be used in the MOSAIC environment. It’s also possible to perform directly in the environment an automatic discretization routine of the process system if required.

MOSAIC enables the user even to create new language specifications to connect other simulation or optimization environments to the platform, and to create an equation database to facilitate the equation systems building. Thanks to these proprieties, MOSAIC is constantly growing environment, with always more compatibility with simulation and optimization programs, and steadily increased library of existing models of chemical engineering applications.

6.2 DISCOVER THE POTENTIAL OF MOSAIC IN IMMERSIVE SIMULATION

Thanks to the characteristics illustrated before, MOSAIC could be a very good environment to be used in the immersive simulation.

In fact, as written in the thesis work introduction, one of the objectives of the immersive dynamic simulation project is the ‘generalized simulation’ approach. The final goal is to provide a package to be used from Companies for operators training. While the 3D models are mostly created with Intergraph® Smart3D, the simulation software which can be used from Companies is many, without a real predominant one. So, using Dynsim as official dynamic simulation software to create the immersive simulation, which means connecting directly the Dynsim outputs to the virtualizing software, would cause big problems with Companies which don’t use Dynsim simulation package: in this case, it would be necessary to create ‘ex novo’ the simulation with Dynsim before starting the immersive simulation, which is in terms of time and economic resources expensive.

For this reason, MOSAIC could be interesting for the immersive simulation. In fact, as described before, this simulation environment has the capability to import and export data with different languages, and also provides the possibility to create new language specifications. So it could be used like a hub between the dynamic simulation and the virtualizing software.

There are two possible ways MOSAIC could be used:

1) The most performing way, could be using MOSAIC to receive the dynamic simulator outputs importing them in the environment, and then exporting the data with a predefined language to be easily read from the virtualizing software. In this way whatever the dynamic simulator is used, once its outputs are imported in MOSAIC language, they can be managed by the environment and exported in a chosen language;
2) As MOSAIC is itself a simulation environment, it’s possible to use it to manage the simulation of the process and then provide the ‘generalized’ output to the virtualizer using the ‘language specifications’ capability of the environment. In this case, even if the desired generalized output is reached, the simulation has to be built

in MOSAIC software, so there is an additional passage. This possibility could be more useful where the system is characterized by non-standard equations which must be introduced manually, or no dynamic simulations of the process are still available. This last is obviously not the case of industrial chemical processes, but it could be the case of a laboratory process, in particular where the operator training regards university students approaching the laboratory works for academic projects. With immersive simulation support, before entering the laboratory, the laboratory workers should pass the immersive training, and in this case MOSAIC could be used not only as interface between the simulator and the virtualizer, but as the simulator itself. In fact the MOSAIC elasticity to manage equations even with different notations enables to perform simulation of chemical processes using the steadily increasing database of models for chemical engineer applications present in the environment.

6.3 REPLACING PROCESS WITH MOSAIC: SIMULATION ENVIRONMENT

In this thesis work it’s decided to follow the second way presented before. This choice has been influenced by lots of factors; at first, following the first way would mean explore only a little section of the environment. In fact, the ‘language specification’ tab used to set up new import/export rules would be the only necessary to the conversion work, but the whole simulation environment wouldn’t be studied and discovered as it deserves. Besides, while the trial import data could be taken by Dynsim outputs, the export language is at this point unknown, because the possible virtualizing software are many and the next step of the immersive simulation project will focus exactly on the virtualizer choice, so the MOSAIC export language will be defined and can be modelled by the ‘language specification’ section. It will be another subject of study the best way to extract and send the dynamic simulation data to the ‘generalizer’ software if the first way is chosen, mostly in industrial processes collaborations.

As previously established, this work thesis follows the ‘second way’ to use MOSAIC in an immersive simulation project, i.e. as a simulation environment itself. It’s then convenient to present the simulation environment before presenting the *AG2S™* process modeling and the obtained results.

6.3.1 MOSAIC environment: general overview

The modeling environment MOSAIC is a project that tries to improve the creation and the use of custom made models. Although lots of simulation software are available in chemical engineer world (AspenPlus, Dynsim etc) it’s often necessary to create new models that fulfill special needs. These custom models can be created using a large variety of programming languages (e.g. Fortran, C) and there are many environments able to use the custom models (e.g. Matlab, gProms, AMPL) to simulate them. The problem of the ‘standard’ approach is that the model creation requires long time, and at the end the built model would be ‘understandable’ only by the simulation environment for which it’s been created. MOSAIC is created to react to this situation, creating a mathematical environment in which building the model once, and then using the code generator simulating it in different cases, environments and languages.

The principal characteristics of MOSAIC environment are now presented:

- Mathematic expressions are written based on LaTeX method, so they are expressed in symbolical form in the environment just as they would appear in a publication or written on paper;
- The equation systems are created assembling the single equations and functions which must be individually modelled;
- Every function, equation or equation system must have a specific notation in which all the symbols used and their explanation are contained;
- It’s possible to build equation systems assembling not only equations, but also other equation systems;
- The equation system model doesn’t contain information about its calculation. In particular values of parameters, indexes maximum and variables, and specification of which are the design and the iteration variables are not present in the model. This leads to more reusability of the equation systems;
- All the calculation specifications are chosen in a specific section of MOSAIC named ‘Evaluation’.

From this brief overview of the MOSAIC environment, it’s evident the general modeling procedure:

a) create the *notation* containing all the symbols that will be used and their explanation;
b) create the *equations*, assembly them into equation systems which can be further added between them to create a superior equation system if necessary or convenient;
c) do special operations like creating *function* objects, *interfaces* and *connectors*;
d) create an *evaluation* object which contains all the calculations data needed, choosing the maximum index values, classifying the variables into design and iteration variables, assigning values to design variables and parameters and providing guess values (or initial values in the case of differential equations) to the iteration variables;
e) the system is now ready to be solved. MOSAIC can generate the solving program code in different languages; depending on the language chosen, it can be possible to execute the solving program and see the results directly in the MOSAIC environment, or exporting the code and executing it in the respective language simulating environment.

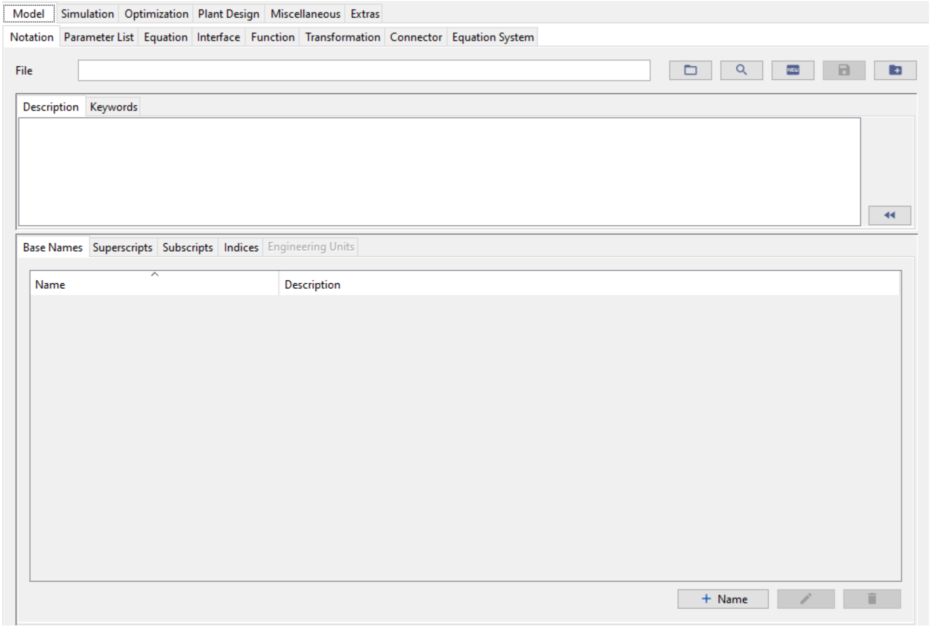


Figure 98 – MOSAIC workspace

In the previous figure the MOSAIC workspace is presented. It’s evident that, in the ‘Model’ section, the correspondence between the general modeling procedure and the tab sequence is perfect; the environment itself is built to follow the schematic procedure of model construction, and helps the user to proceed in his/her work.

The following paragraph will explain in more detail all the steps of the MOSAIC modelling procedure, starting obviously from the notation creation.

6.3.2 MOSAIC environment: notation

Notations are the fundamental modeling element in MOSAIC.

They are used to introduce all the symbols appearing in the equation systems, and to specify their meaning to the user throughout the entire modeling process. In fact, the description of the symbols written in the notation is always available to help the user modeling work. Notations in MOSAIC are a direct projection of notations found in the literature, where the symbols are systematically listed and described.

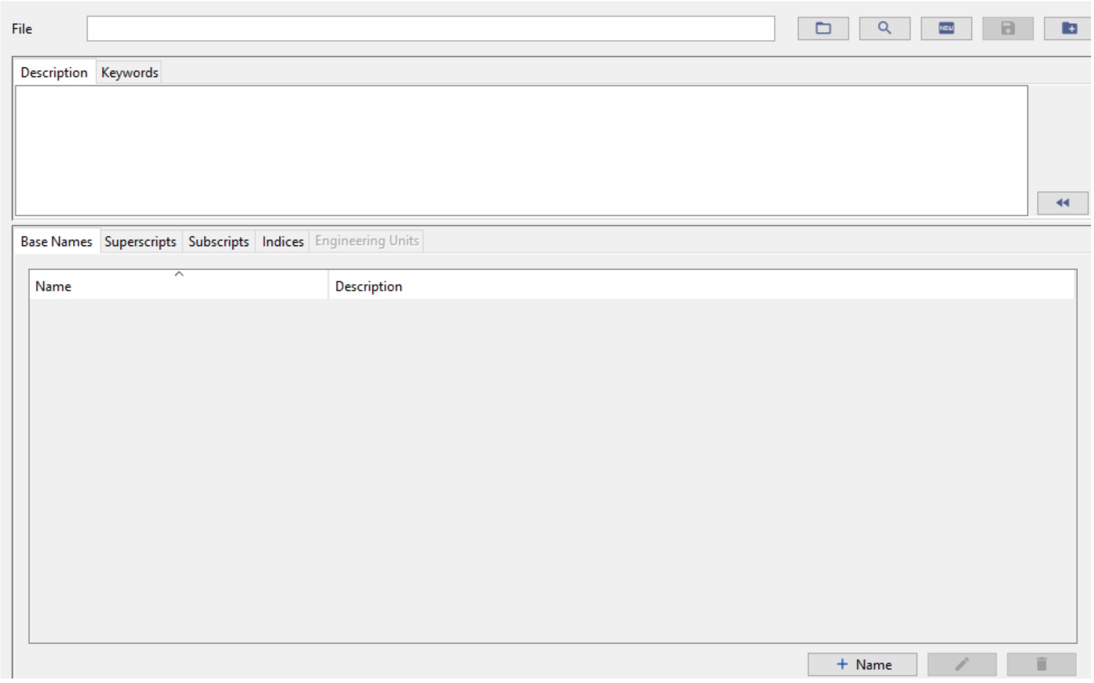


Figure 99 – notation tab structure

In the figure above the notation structure is illustrated. As in the literature, the notation symbols are divided into groups according to their position and their functionality: base names, superscripts, subscripts and indices. The base names appear on the base line of the written, the subscript and indices are lowered symbol while superscripts are raised symbols. The differences between subscripts and indices are that the first represent a fixed descriptive element, while the second are variable from 1 to their maximum value specifiable in the ‘Evaluation’ section. Notice that ‘Notation’ tab, but even all the other tabs present in MOSAIC, expects a ‘Description’ of the notation; this is mandatory, and it can be very helpful to the user to specify the characteristics of the model created.

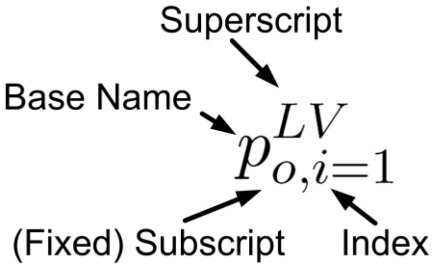


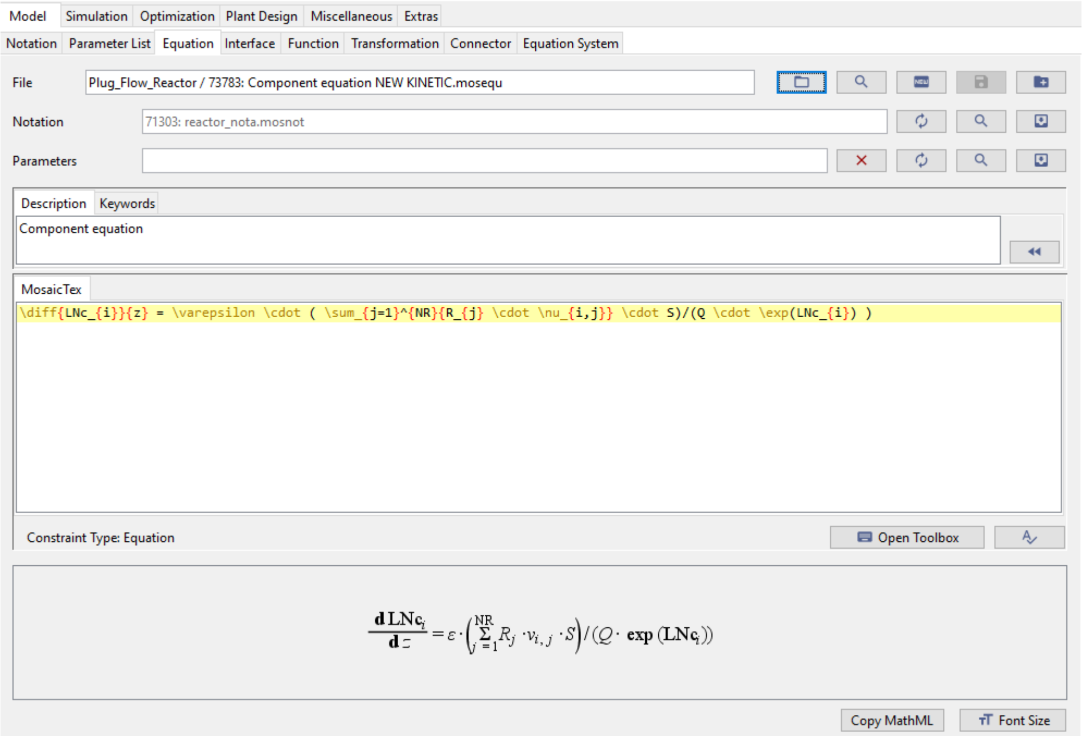
Figure 100 – MOSAIC different groups of symbols

In the example all the four symbols groups are represented. But why a so complex notation structure is chosen for MOSAIC models? The first reason is the proximity to literature. In chemical engineering the conservation laws of mass, momentum and energy are the most important equation, whence algebraic, ordinary differential or partial differential equations are derived. These equations are really frequent in the engineer literature with specific notations which add information to the model which are fundamental for the correct use of the equations. A notation based only to a direct sequence of characters like in the standard programming language (Matlab, FORTRAN…) loses important information in the model formulation and so the mathematical symbolic representation of the model isn’t possible; instead MOSAIC enables to create a perfect symbolic notation to write the equations and systems of equations as reported in a scientific article or paper.

Another important aspect of the MOSAIC language is the modularity; often some equations of a single model can be re-used in other systems, this because there are like ‘standard equations’ of the engineer world. When a new system is created, the users could re-use existing equations instead of re-writing the same equations, losing time and personal resources. This approach is possible in MOSAIC, thanks to the individual existence of the Equation, which will be deepen in the next paragraph, and of the Connectors, which are able to connect equations even with different notations, extending their re-usability.

6.3.3 MOSAIC environment: equations

The Equation is the simplest and smallest element in MOSAIC.



File: Plug_Flow_Reactor / 73783: Component equation NEW KINETIC.mosequ

Notation: 71303: reactor_nota.mosnot

Parameters:

Description: Component equation

```
\diff{LNc_{i}}{z} = \varepsilon \cdot ( \sum_{j=1}^{NR}{R_{j} \cdot \nu_{i,j}} \cdot S)/(Q \cdot \exp(LNc_{i}) )
```

Constraint Type: Equation

$$\frac{\mathbf{d}\,\mathrm{LNc}_i}{\mathbf{d}\,z} = \varepsilon \cdot \left(\sum_{j=1}^{\mathrm{NR}} R_j \cdot \nu_{i,j} \cdot S\right) / (Q \cdot \mathbf{exp}\,(\mathrm{LNc}_i))$$

Figure 101 – example of equation

The ‘Equation’ tab structure is reported above. It’s possible to see that the equation must use a reference notation (in this case the ‘reactor_nota), it has a description like all the MOSAIC elements, it may contain a list of global parameters and it’s characterized by a MathML expression which represents the mathematical content. The MathML expression is based on LaTeX style. The symbolic equation written in ‘MosaicTex’ is rendered automatically in MathML before saving it.

6.3.4 MOSAIC environment: interfaces

Interfaces are one of the instruments necessary to ‘connect’ in MOSAIC different unit operations. It represents an independent list of variable namings that are expected to be shared with another modeling element. Practically, the instances create a new notation used to share or pass information between modeling elements. They are used in ‘ports and streams’ (explained later) for this purpose, allowing the easy re-use of existing units.

6.3.5 MOSAIC environment: connectors

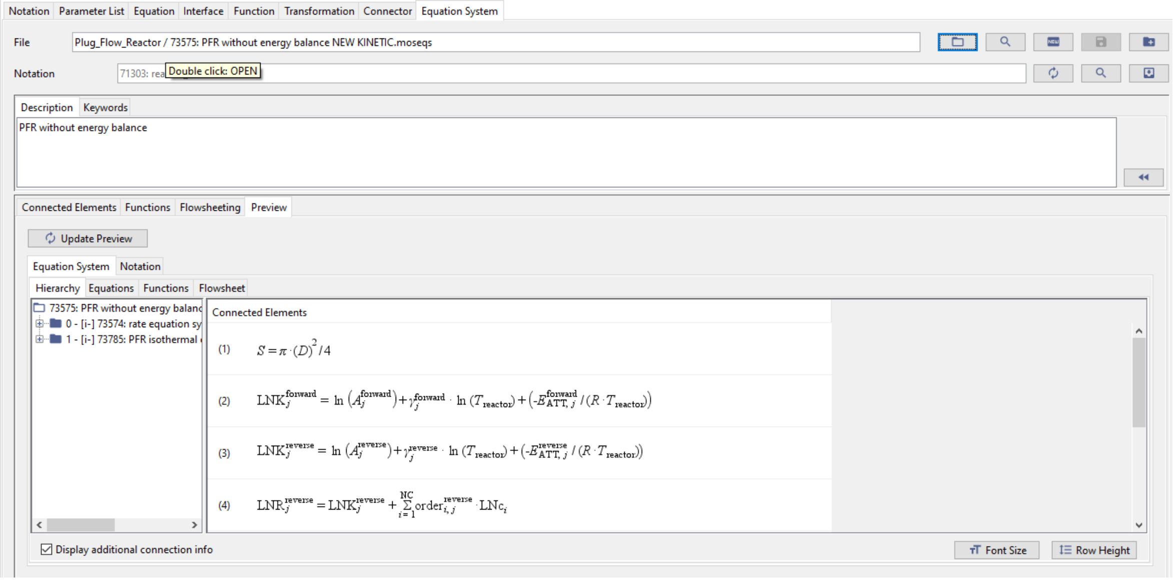
To reach the final goal of the interfaces, even the connectors must be used. They are used to translate symbols between two different notations. This is really useful in lots of different situations. At first, to extend the re-usability of the existing units; the ‘Connector’ objects enable to use an equation or equation system with notation ‘a’ in an equation system based on another notation ‘b’, connecting the respective symbols each other. In the case of passing information between modeling elements, it’s possible to connect the symbols of the unit operations with their equivalent of the interface, making possible the communication between ‘ports and streams’ which will be a topic of the next paragraph.

6.3.6 MOSAIC environment: ports and streams

Large equation systems can represent a unit operation in a chemical process. To perform the entire process simulation, it’s necessary to connect the different unit operations in their real sequence. To perform this linking between unit operation ports and streams are used. The ports belong to an Equation System and can’t exist independently. They are like a ‘door’ thanks to which communicate their output data or receive input data. They have a distinct name and use an interface. They can communicate with other ports provided elements using streams. Even the streams can’t exist independently from their equation system; their role is to connect exactly two ports, using an interface and an identification number which represent the global stream number and its listing number in the Output/Input group.

The ‘interfaces’, ‘connectors’ and ‘ports and streams’ topics will be deepen in the next chapters.

6.3.7 MOSAIC environment: equation systems



File: Plug_Flow_Reactor / 73575: PFR without energy balance NEW KINETIC.moseqs

Notation: 71303: rea

PFR without energy balance

73575: PFR without energy balanc
0 - [i-] 73574: rate equation sy
1 - [i-] 73785: PFR isothermal

Connected Elements

(1) $$S = \pi \cdot (D)^2 / 4$$

(2) $$\mathrm{LNK}_j^{\mathrm{forward}} = \ln\left(A_j^{\mathrm{forward}}\right) + \gamma_j^{\mathrm{forward}} \cdot \ln\left(T_{\mathrm{reactor}}\right) + \left(-E_{\mathrm{ATT},\,j}^{\mathrm{forward}} / \left(R \cdot T_{\mathrm{reactor}}\right)\right)$$

(3) $$\mathrm{LNK}_j^{\mathrm{reverse}} = \ln\left(A_j^{\mathrm{reverse}}\right) + \gamma_j^{\mathrm{reverse}} \cdot \ln\left(T_{\mathrm{reactor}}\right) + \left(-E_{\mathrm{ATT},\,j}^{\mathrm{reverse}} / \left(R \cdot T_{\mathrm{reactor}}\right)\right)$$

(4) $$\mathrm{LNR}_j^{\mathrm{reverse}} = \mathrm{LNK}_j^{\mathrm{reverse}} + \sum_{i=1}^{\mathrm{NC}} \mathrm{order}_{i,\,j}^{\mathrm{reverse}} \cdot \mathrm{LNc}_i$$

☑ Display additional connection info

Figure 102 – example of an equation system

Equation systems are the modeling object which contains the mathematical information of the model. They're created by combining equations, functions or other equation systems. The equation system can be used to insert 'ports and stream', so threated as a unit with desired input or output data, or to be evaluated specifying the designed variables and the other problem specific information. During the equation system creation it's also possible to perform peculiar operations to the equations which forms it, like the 'discretization of the system' (it will be explained in the next chapters) and other transformations, or to apply connectors to the equations to conform the languages.

Thus, the equation systems are the most important object which contains all the mathematical information and transformations of the model, allowing the calculation tasks to reach the desired results.

6.4 *AG2S™* PROCESS IN MOSAIC

To reproduce the *AG2S™* process in MOSAIC the procedure described in the previous chapter is followed. The first concept to underline is that MOSAIC environment works with Equations, while an industrial simulator like Aspen or Dynsim works with Unit Operations. The unit operations are like boxes representing the real components of the process. In Dynsim simulation it's possible to see the unit operations scheme: the valves, the controllers, the reactor and the traps, are all 'boxes' containing the models of their real respective. All the unit operations are then connected by 'streams', which link the inlets and outlets in order to obtain the process. In MOSAIC the work is based on equations modeling, so the goal is to written the models which are within the 'boxes'. These models are collected into the equation systems, so every superior equation system can represent a single Unit Operation. In this thesis work the principal units' models will be reported following the construction until reaching the results. The first unit to be described is the crucial one, the reactor, which has required the biggest effort in terms of modeling and solving.

6.4.1 The reactor in MOSAIC

As seen in DSmoke and Dynsim chapters, the reactor is an isothermal tubular reactor of 200 mm length. The reactor results the most difficult unit operation to be replaced in MOSAIC because of its intrinsic complexity, the presence of reactions. In this paragraph the intent is to present the whole 'story' of the reactor modeling, starting from the first tries until reaching the right results. The first reactor modeling test is now described. In the following mass balance equations are reported; obviously, the energy balance is not present in the model because of the isothermicity of the reactor.

1st equation system)

$$\frac{dN_i}{dz} = \sum_{j=1}^{NR} \nu_{i,j} R_j \, S$$

$$\frac{dN_{tot}}{dz} = \sum_{i=1}^{NC} \sum_{j=1}^{NR} \nu_{i,j} R_j \, S$$

Where:

- N_i = molar flowrate of the component i [mol/s]
- N_{tot} = total molar flowrate [mol/s]
- $\nu_{i,j}$ = stoichiometric coefficient of the i-th component in the j-th reaction
- R_j = rate of the j-th reaction [mol/(m^{3}s)]
- S = cross surface of the reactor [m^2]

This is the first approach used to solve the reactor: the model is written in the variation of molar flowrates. Obviously, this model is totally general and reaction rates definition are not specified in it, but the correct approach in MOSAIC is to maintain equation as general as possible, to preserve their re-usability in future. In this way, if the reaction rates expression should be changed, there wouldn't be variations in the component and global mass balances. Now the reaction rates and kinetic expressions are illustrated:

$$R_j = k_j \prod_{i}^{NC} c_i^{order_{i,j}}$$

$$k_j = A_j \exp\left(\frac{-E_{act_j}}{RT}\right)$$

Where:

- k_j is the kinetic constant of the j-th reaction [variable]
- A_j is the pre exponential factor of the j-th reaction [=kj]
- $order_{i,j}$ is the order of the component i-th in the kinetic expression of the j-th reaction
- Eact,j is the activation energy of the j-th reaction [j/mol]
- R is the gas constant [j/(mol K)]
- T is the reactor temperature [K]

Using the molar flowrate as equation base, a definition of the concentration is then necessary. Besides, to have as more generalize equation system as possible, even the cross surface is defined with a separate equation:

$$c_i = \frac{N_i}{Q}$$

$$S = \frac{\pi D^2}{4}$$

Where:

- Q is the volumetric flowrate of the charge [m^3/s]
- D is the reactor internal diameter [m]

The first approach it's been completely modelled in MOSAIC, including the variable specifications, i.e. choose the designed variables and insert their values, choose the iteration variable and insert their initial values. It has been tried to solve the DAE system in different environment, expecially Matlab and gProms, without finding the solution. The reason is found in the really fast kinetic of the system, which leads to an intrinsic difficult to solve the system in particular in the first section of the reactor.

To guarantee a better stability of the system, it's been decided to pass to a logarithmic formulation of the equations. In particular, the base of the mass balances chosen in the new formulation is no more the molar flowrate, but the logarithm of the concentration of the components in the system. All the elements which compose the equations are then converted in logarithmic form, to 'amortize' the strong variations at the first part of the reactor. The c_i becomes LNc_i, a new variable symbolizing the $\ln(c_i)$, and so all the other variables in the system. The new balance and kinetic equations are listed in the following:

2nd equation system)

1) $Q\ exp(LNc_i)\frac{d(LNc_i)}{dz} = \sum_{j=1}^{NR} \nu_{i,j} \exp(LNR_j)\, S$
2) $Q \exp(LNc_{tot})\frac{dc_{tot}}{dz} = \sum_{i=1}^{NC}\sum_{j=1}^{NR} \nu_{i,j} \exp(LNR_j)\, S$
3) $LNR_j = LNk_j + \sum_{i}^{NC} order_{i,j}\, LNc_i$
4) $LNk_j = \ln(A_j) - \frac{E_{act_j}}{RT}$
5) $S = \frac{\pi D^2}{4}$

The concentrations definition dependent on the molar flowrate is now not necessary because the base of the new equation system is the concentration itself. The desire to express all the equations in logarithmic form leads to following 'transformations':

$c_i \rightarrow \exp(\ln(c_i)) = \exp(LNc_i)$

and identically for all the other elements of the equations. In this way the differential equation:

$$Q\frac{dc_i}{dz} = \sum_{j=1}^{NR} \nu_{i,j} R_j\, S \qquad \rightarrow \qquad Q\frac{d(\exp(LNc_i))}{dz} = \sum_{j=1}^{NR} \nu_{i,j} \exp(LNR_j)\, S$$

So, finally:

$$Q\ exp(LNc_i)\frac{d(LNc_i)}{dz} = \sum_{j=1}^{NR} \nu_{i,j} \exp(LNR_j)\, S$$

for a simple derivation rule.

Actually, even if the system is now more stable and resistant to the fast kinetic of the reactions involved, the real definitive model has not been presented yet. This is because a further transformation has been necessary when the new kinetic with forward and reverse reactions has been implemented. In the case of equilibrium reactions, in fact, there is the need of writing different reaction rate equation to take into account even the reverse reaction rate. So, while the mass balance equations are left equal to the previous ones, the rate equations change their definition. The third evolved equation system is now presented:

3rd equation system)

1) $Q\ exp(LNc_i)\frac{d(LNc_i)}{dz} = \sum_{j=1}^{NR} \nu_{i,j}\ \exp(LNR_j)\ S$

2) $Q\ \exp(LNc_{tot})\frac{dc_{tot}}{dz} = \sum_{i=1}^{NC}\sum_{j=1}^{NR} \nu_{i,j}\ \exp(LNR_j)\ S$

$$3)\ LNR_j = \ln(\exp\left(LNk_j^{forward}\right)\exp(\sum_{i}^{NC} order_{i,j}{}^{forward}\ LNc_i)$$
$$-\exp\left(LNk_j^{reverse}\right)\exp(\sum_{i}^{NC} order_{i,j}{}^{reverse}\ LNc_i))$$

4) $LNk_j^{forward} = \ln\left(A_j^{forward}\right) + \gamma_j^{forward}\ln(T) - \frac{E_{act,j}^{forward}}{RT}$

5) $LNk_j^{reverse} = \ln(A_j^{reverse}) + \gamma_j^{reverse}\ln(T) - \frac{E_{act,j}^{reverse}}{RT}$

6) $S = \frac{\pi D^2}{4}$

In this third writing of the equation system the powerful modularity of MOSAIC is perfectly represented. The balance equations of the second try can be perfectly re-used in the third equation system, as also the cross surface definition, but not only them; the kinetic constant equations, even if they contain new characters (forward and reverse) and also a new temperature coefficient (γ) due to the formulation of the kinetic regression using DSmoke, it's not necessary to re-write them completely. This is because MOSAIC consider every equation like an individual element, so it's possible to open the 'Notation' tab to add the new superscripts 'forward' and 'reverse', and 'γ' base name to the 'reactor_nota', which is the reactor reference notation; then, going to the 'Equation' editor, it's sufficient to make the little changes and save the two new equations to include them in the third equation system to perform the new model.

Unfortunately, even this model is not the definitive one. This is because the definition of the global reaction rate is a logarithm of a difference. This difference is influenced by the kinetic constants and the concentrations of the compounds, but in general is evident that is not always greater than zero. In the case of a negative difference, obviously the solver would go in stall without finding a solution because of the try to find a logarithm of a negative number. This problem leads to a little reformulation of the equations, which results in the definitive equation system for the isothermal PFR reactor in MOSAIC:

Definitive system)

1) $Q\ exp(LNc_i)\,\frac{d(LNc_i)}{dz} = \sum_{j=1}^{NR} \nu_{i,j} R_j\, S$

2) $Q\ \exp(LNc_{tot})\,\frac{dc_{tot}}{dz} = \sum_{i=1}^{NC}\sum_{j=1}^{NR} \nu_{i,j} R_j\, S$

3) ${LNR_j}^{forward} = LNk_j^{forward} + \sum_{i}^{NC} {order_{i,j}}^{forward}\, LNc_i$

4) ${LNR_j}^{reverse} = LNk_j^{reverse} + \sum_{i}^{NC} {order_{i,j}}^{reverse}\, LNc_i$

5) $LNk_j^{forward} = \ln\left(A_j^{forward}\right) + \gamma_j^{forward}\ln(T) - \frac{E_{act,j}^{forward}}{RT}$

6) $LNk_j^{reverse} = \ln\left(A_j^{reverse}\right) + \gamma_j^{reverse}\ln(T) - \frac{E_{act,j}^{reverse}}{RT}$

7) $R_j = \exp(LNR_j^{forward}) - \exp(LNR_j^{reverse})$

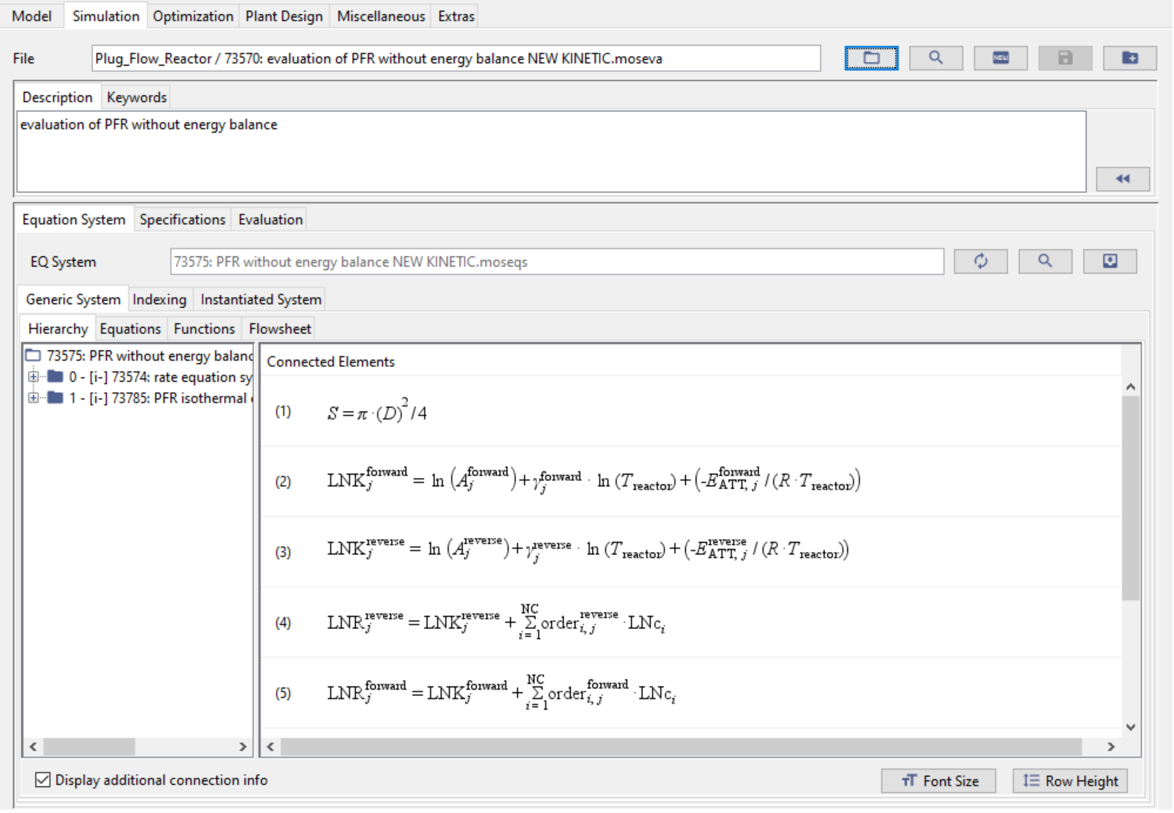
8) $S = \frac{\pi D^2}{4}$

Writing the LNR_j separately for the forward and reverse contribute there is no possibility to have a negative logarithm; in fact the global reaction rate is expressed in its original form instead of the logarithm one, and in this same form it is present even in the mass balance equations.

This is how the reactor DAE equations system appears in its definitive version. The obtained equation system is then solved in all the conditions used to find the simplified kinetic scheme with DSmoke, to compare the previously found results with the MOSAIC model solutions.

6.4.2 Reactor model evaluation

Before presenting the results of the reactor evaluation, it could be interesting to better illustrate the MOSAIC solving section.



File: Plug_Flow_Reactor / 73570: evaluation of PFR without energy balance NEW KINETIC.moseva

evaluation of PFR without energy balance

EQ System: 73575: PFR without energy balance NEW KINETIC.moseqs

(1) $S = \pi \cdot (D)^2 / 4$

(2) $\mathrm{LNK}_j^{\mathrm{forward}} = \ln\left(A_j^{\mathrm{forward}}\right) + \gamma_j^{\mathrm{forward}} \cdot \ln\left(T_{\mathrm{reactor}}\right) + \left(-E_{\mathrm{ATT},\,j}^{\mathrm{forward}} / \left(R \cdot T_{\mathrm{reactor}}\right)\right)$

(3) $\mathrm{LNK}_j^{\mathrm{reverse}} = \ln\left(A_j^{\mathrm{reverse}}\right) + \gamma_j^{\mathrm{reverse}} \cdot \ln\left(T_{\mathrm{reactor}}\right) + \left(-E_{\mathrm{ATT},\,j}^{\mathrm{reverse}} / \left(R \cdot T_{\mathrm{reactor}}\right)\right)$

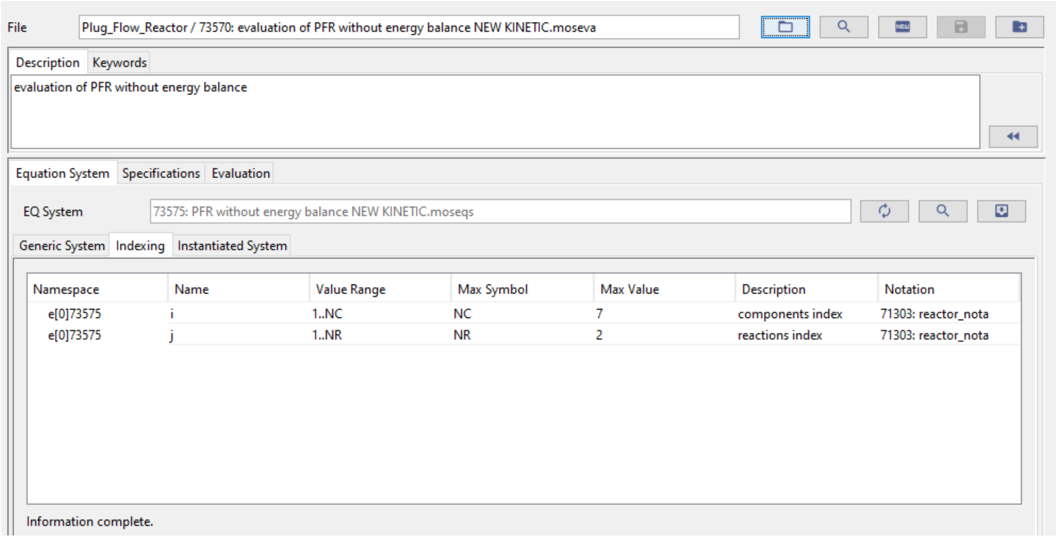
(4) $\mathrm{LNR}_j^{\mathrm{reverse}} = \mathrm{LNK}_j^{\mathrm{reverse}} + \sum_{i=1}^{\mathrm{NC}} \mathrm{order}_{i,\,j}^{\mathrm{reverse}} \cdot \mathrm{LNc}_i$

(5) $\mathrm{LNR}_j^{\mathrm{forward}} = \mathrm{LNK}_j^{\mathrm{forward}} + \sum_{i=1}^{\mathrm{NC}} \mathrm{order}_{i,\,j}^{\mathrm{forward}} \cdot \mathrm{LNc}_i$

☑ Display additional connection info

Figure 103 – ‘Simulation’ evaluation section

The evaluation section in MOSAIC is included in the tab ‘Simulation’. At first the ‘Equation System’ window appears; inside it the preliminary actions of the evaluations are made. At first, it’s necessary to select the equation system to be solved, in this case the ‘PFR without energy balance NEW KINETIC’ system. In ‘Generic System’ equations are listed and hierarchically divided into their provenance equation sub-systems; besides, the ‘Flowsheet’ tab can be used to create internal streams between two connectors. The internal streams usage will be explained better in next chapters. The next step in the ‘Equation System’ window is to choose the ‘Indexing’, inserting the maximum values of all indices present in the system.



File: Plug_Flow_Reactor / 73570: evaluation of PFR without energy balance NEW KINETIC.moseva

evaluation of PFR without energy balance

EQ System: 73575: PFR without energy balance NEW KINETIC.moseqs

Namespace	Name	Value Range	Max Symbol	Max Value	Description	Notation
e[0]73575	i	1..NC	NC	7	components index	71303: reactor_nota
e[0]73575	j	1..NR	NR	2	reactions index	71303: reactor_nota

Information complete.

Figure 104 – ‘Indexing’ in ‘Simulation’ section

It’s possible to see the maximum values of the indices in this evaluation:

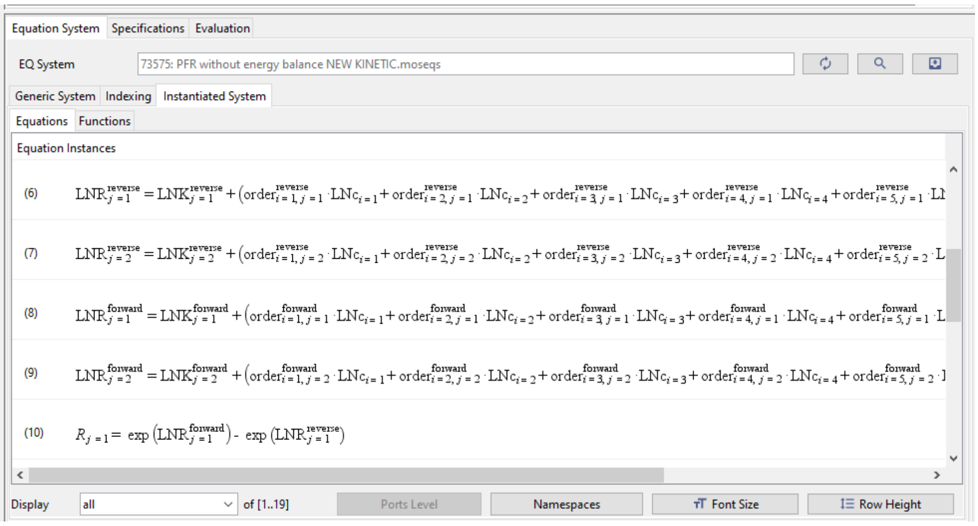
- 7 components, ordered as it follows:

1) H_2S
2) CO_2
3) H_2O
4) CO
5) N_2
6) S_2
7) H_2

- 2 reactions:

1) $2H_2S = 2H_2 + S_2$
2) $CO_2 + H_2 = CO + H_2O$

Selecting the indices, the equation system is now instantiated, and can be viewed completely in the 'Instantiated System' tab:



EQ System: 73575: PFR without energy balance NEW KINETIC.moseqs

Equation Instances

(6) $LNR_{j=1}^{reverse} = LNK_{j=1}^{reverse} + \left(order_{i=1,j=1}^{reverse} \cdot LNc_{i=1} + order_{i=2,j=1}^{reverse} \cdot LNc_{i=2} + order_{i=3,j=1}^{reverse} \cdot LNc_{i=3} + order_{i=4,j=1}^{reverse} \cdot LNc_{i=4} + order_{i=5,j=1}^{reverse} \cdot\right.$

(7) $LNR_{j=2}^{reverse} = LNK_{j=2}^{reverse} + \left(order_{i=1,j=2}^{reverse} \cdot LNc_{i=1} + order_{i=2,j=2}^{reverse} \cdot LNc_{i=2} + order_{i=3,j=2}^{reverse} \cdot LNc_{i=3} + order_{i=4,j=2}^{reverse} \cdot LNc_{i=4} + order_{i=5,j=2}^{reverse} \cdot\right.$

(8) $LNR_{j=1}^{forward} = LNK_{j=1}^{forward} + \left(order_{i=1,j=1}^{forward} \cdot LNc_{i=1} + order_{i=2,j=1}^{forward} \cdot LNc_{i=2} + order_{i=3,j=1}^{forward} \cdot LNc_{i=3} + order_{i=4,j=1}^{forward} \cdot LNc_{i=4} + order_{i=5,j=1}^{forward} \cdot\right.$

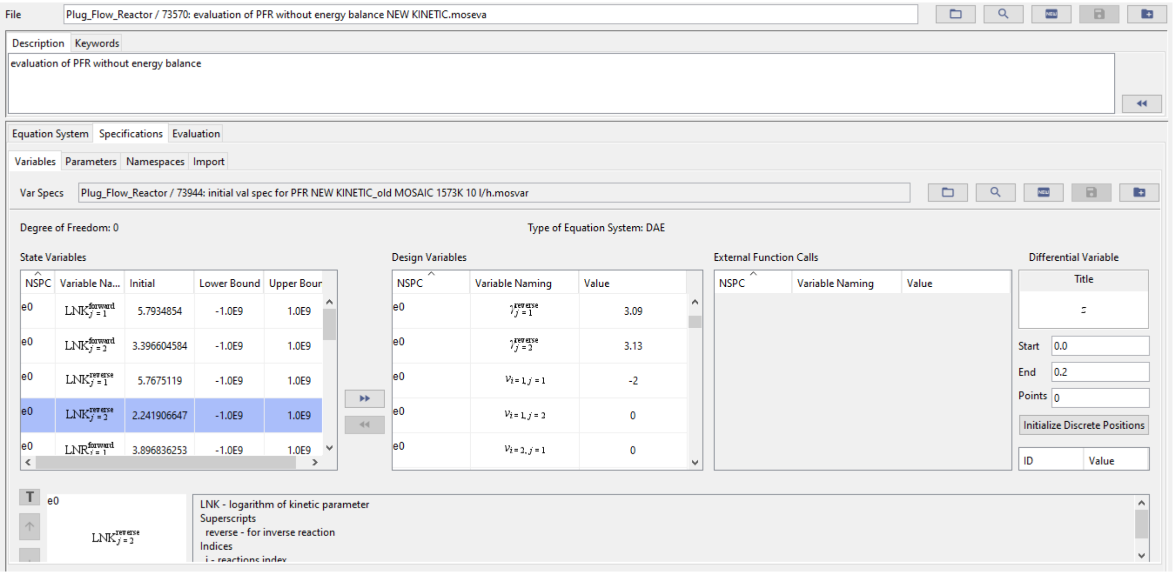
(9) $LNR_{j=2}^{forward} = LNK_{j=2}^{forward} + \left(order_{i=1,j=2}^{forward} \cdot LNc_{i=1} + order_{i=2,j=2}^{forward} \cdot LNc_{i=2} + order_{i=3,j=2}^{forward} \cdot LNc_{i=3} + order_{i=4,j=2}^{forward} \cdot LNc_{i=4} + order_{i=5,j=2}^{forward} \cdot\right.$

(10) $R_{j=1} = \exp\left(LNR_{j=1}^{forward}\right) - \exp\left(LNR_{j=1}^{reverse}\right)$

Display: all of [1..19]

Figure 105 – instantiated reactor system

Now the equation system is ready for the 'Specifications' window. In this section the design and iteration variables are chosen, and their values inserted by the user.



File: Plug_Flow_Reactor / 73570: evaluation of PFR without energy balance NEW KINETIC.moseva

evaluation of PFR without energy balance

Var Specs: Plug_Flow_Reactor / 73944: initial val spec for PFR NEW KINETIC_old MOSAIC 1573K 10 l/h.mosvar

Degree of Freedom: 0

Type of Equation System: DAE

State Variables

NSPC	Variable Na...	Initial	Lower Bound	Upper Bour
e0	$LNK_{j=1}^{forward}$	5.7934854	-1.0E9	1.0E9
e0	$LNK_{j=2}^{forward}$	3.396604584	-1.0E9	1.0E9
e0	$LNK_{j=1}^{reverse}$	5.7675119	-1.0E9	1.0E9
e0	$LNK_{j=2}^{reverse}$	2.241906647	-1.0E9	1.0E9
e0	$LNR_{j=1}^{forward}$	3.896836253	-1.0E9	1.0E9

Design Variables

NSPC	Variable Naming	Value
e0	$\gamma_{j=1}^{reverse}$	3.09
e0	$\gamma_{j=2}^{reverse}$	3.13
e0	$\nu_{i=1,j=1}$	-2
e0	$\nu_{i=1,j=2}$	0
e0	$\nu_{i=2,j=1}$	0

External Function Calls

NSPC	Variable Naming	Value

Differential Variable

Title
z

Start: 0.0
End: 0.2
Points: 0

ID	Value

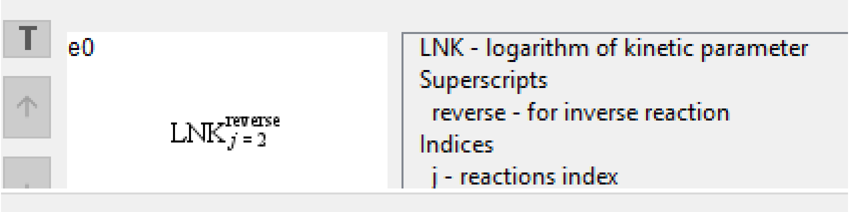
e0

$LNK_{j=2}^{reverse}$

LNK - logarithm of kinetic parameter
Superscripts
reverse - for inverse reaction
Indices
i - reactions index

Figure 106 – specification window

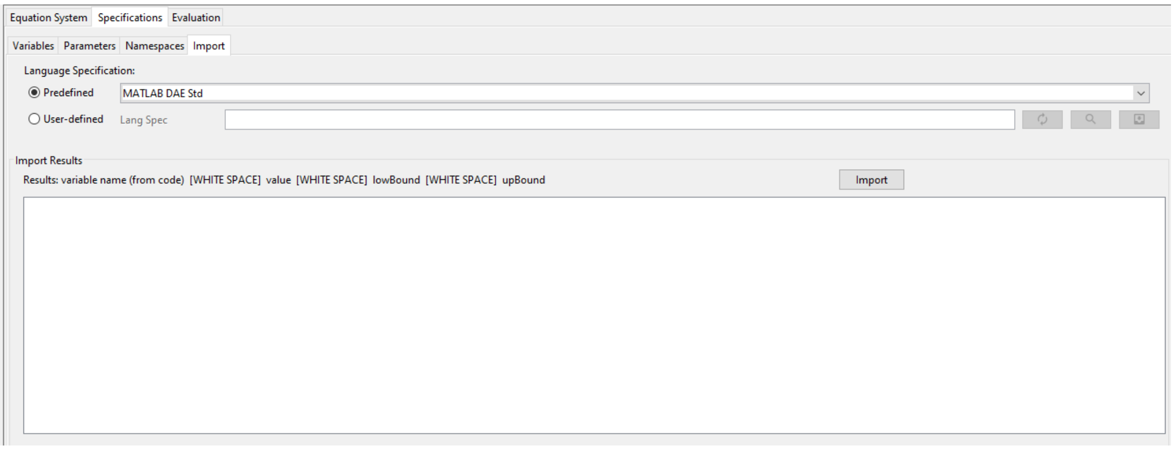
It's possible to see the division between the 'fixed' variables and the state ones. At the right the 'Differential Variable' is automatically recognized by the system, and it's asked to the user to select the differentiating interval. Other interesting thing is the possibility to insert not only the initial value of the iteration variables, but also their lower and upper bounds. At the end, the importance of the notation descriptions is perfectly explained in this figure: selecting one of the variables, in the window below the variable specifications, its description is reported. In this case, choosing $LNk_{j=2}^{reverse}$ all the descriptions regarding the base name, super- and sub-scripts are listed. The particular of the ‘notation description’ is shown in the figure below:



LNK - logarithm of kinetic parameter
Superscripts
reverse - for inverse reaction
Indices
j - reactions index

Figure 107 – particular of notation description in ‘variable specifications’

In ’Specifications’ window there is also a section dedicated to the import of the variables from external programs. The great potential of MOSAIC to manage different languages is represented in this tab, where it’s necessary only to choose the environment from which the data are taken to import them in the MOSAIC environment.

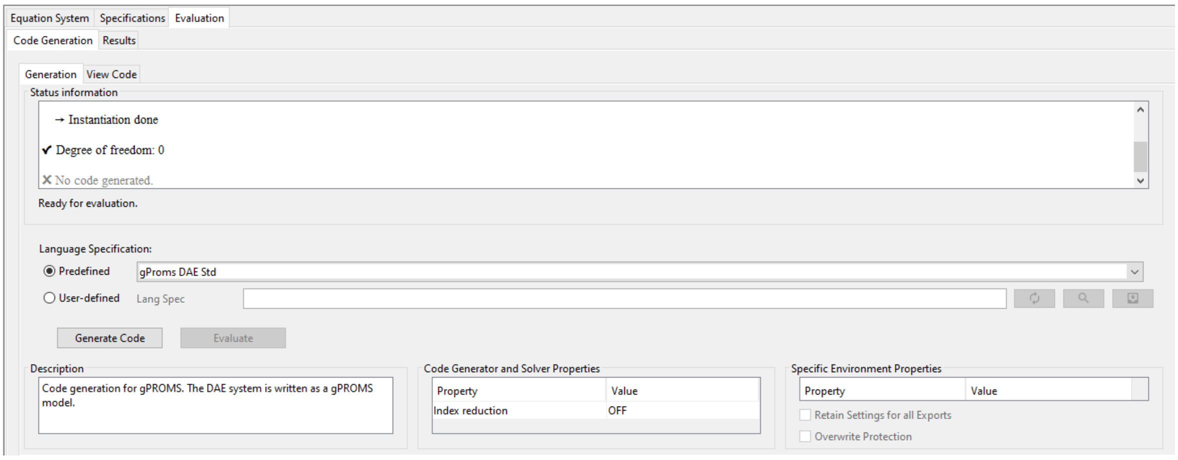


Language Specification:
● Predefined MATLAB DAE Std
○ User-defined Lang Spec

Results: variable name (from code) [WHITE SPACE] value [WHITE SPACE] lowBound [WHITE SPACE] upBound

Figure 108 – import section in ‘Specifications’ window

After the variables specification, the system is ready for the evaluation. In the evaluation window it’s possible to choose the desired environment in which solve or export the code to solve the equation system. Depending on the system proprieties, different environment and code generation are available; in this case for example, the equation system is a DAE (differential algebraic equation), so the code generations available regard this type of system.



Status information

→ Instantiation done

✔ Degree of freedom: 0

✗ No code generated.

Ready for evaluation.

Language Specification:

● Predefined gProms DAE Std

○ User-defined Lang Spec

Description

Code generation for gPROMS. The DAE system is written as a gPROMS model.

Code Generator and Solver Properties

Property	Value
Index reduction	OFF

Specific Environment Properties

Property	Value

☐ Retain Settings for all Exports

☐ Overwrite Protection

Figure 109 – code generation

During the language specification it's also possible, depending on the language chosen, to select some solver proprieties, like the possibility to use analytic derivatives or apply index reduction in gProms solver. After the language specification the code is ready to be generated; in the case of solvers executed directly in MOSAIC the 'Results' tab will show all the variables trends and results, else in 'View Code' there will be the 'Export Code' command, to export the generated code in the chosen language and execute it externally. The results and variables trends could then be observed in the chosen environment or imported in MOSAIC using the 'Import' command illustrated before.

6.4.3 Reactor model results

After this brief overview of the MOSAIC 'Simulation' section, the solving routine of the reactor could be clearer. The equation system is solved using gProms language, i.e. exporting the generated code from MOSAIC to the gProms solver. The variable specifications used are the same of the DSmoke simulations, to compare the results in the same conditions between the DSmoke software and MOSAIC modeling environment.

The following trends summarize the results for the three volumetric flowrates and the nine reactor temperature conditions chosen in DSmoke simulations. All the results are expressed in *massive fractions vs temperature*, as they are collected in DSmoke, so the MOSAIC concentration results have been converted before comparing them with DSmoke ones.

H_2S MASS FRACTIONS vs T RESULTS

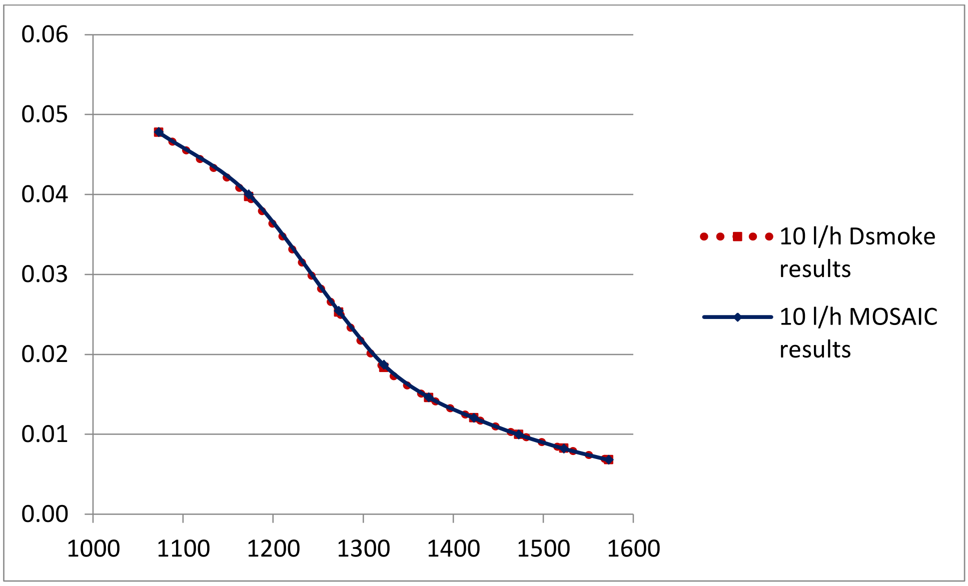


Figure 110 – massive fractions of H_2S vs reactor temperature, 10 l/h

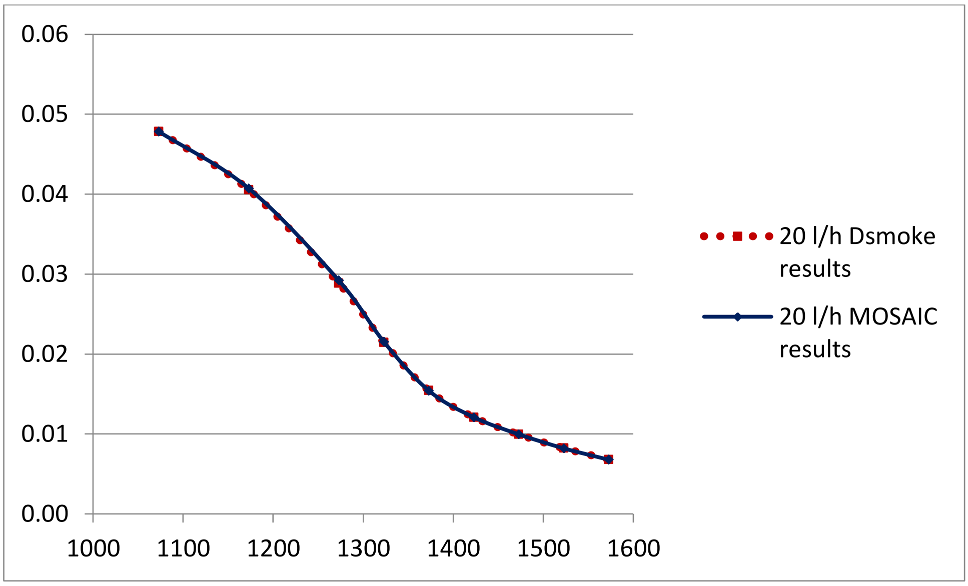


Figure 111 - massive fractions of H_2S vs reactor temperature, 20 l/h

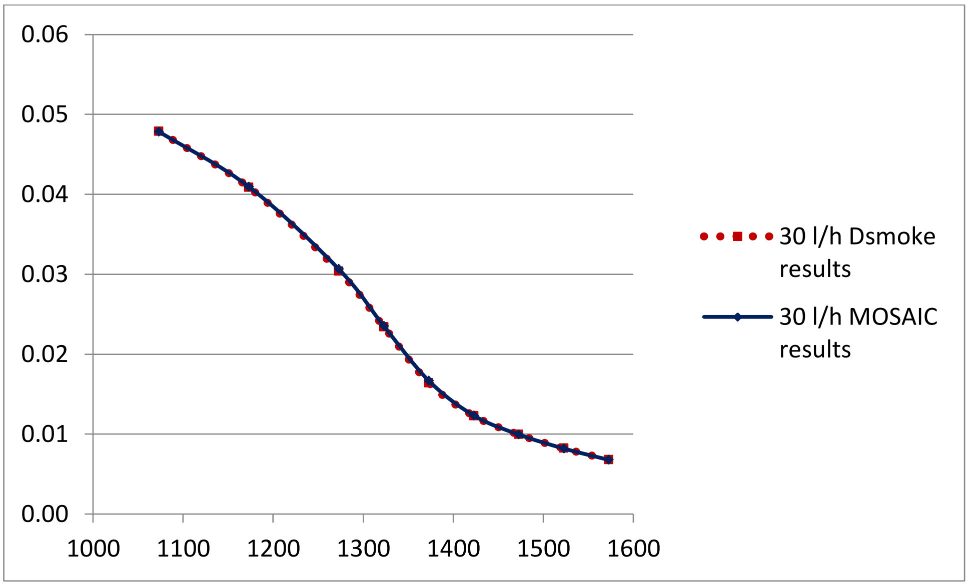


Figure 112 - massive fractions of H_2S vs reactor temperature, 30 l/h

CO_2 MASS FRACTIONS vs T RESULTS

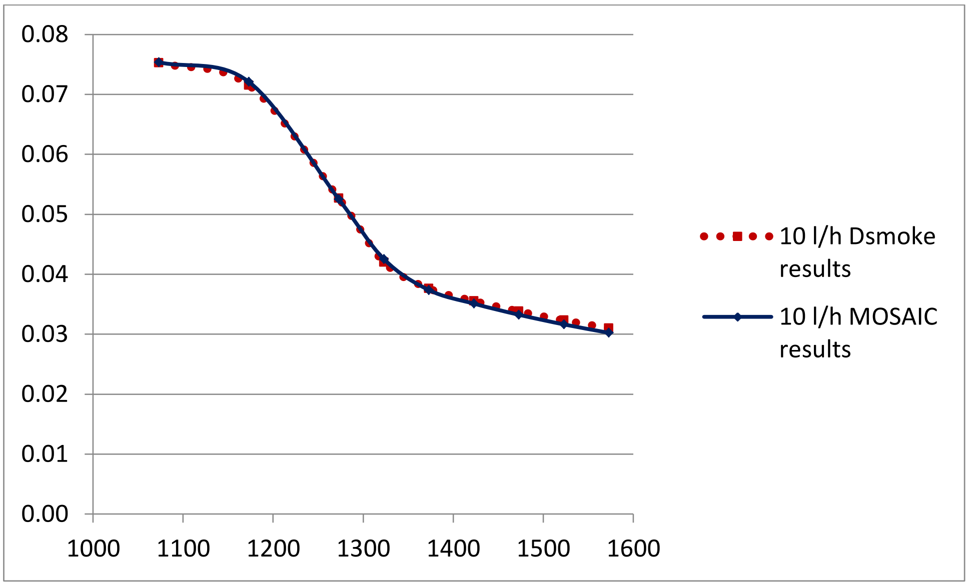


Figure 113 - massive fractions of CO_2 vs reactor temperature, 10 l/h

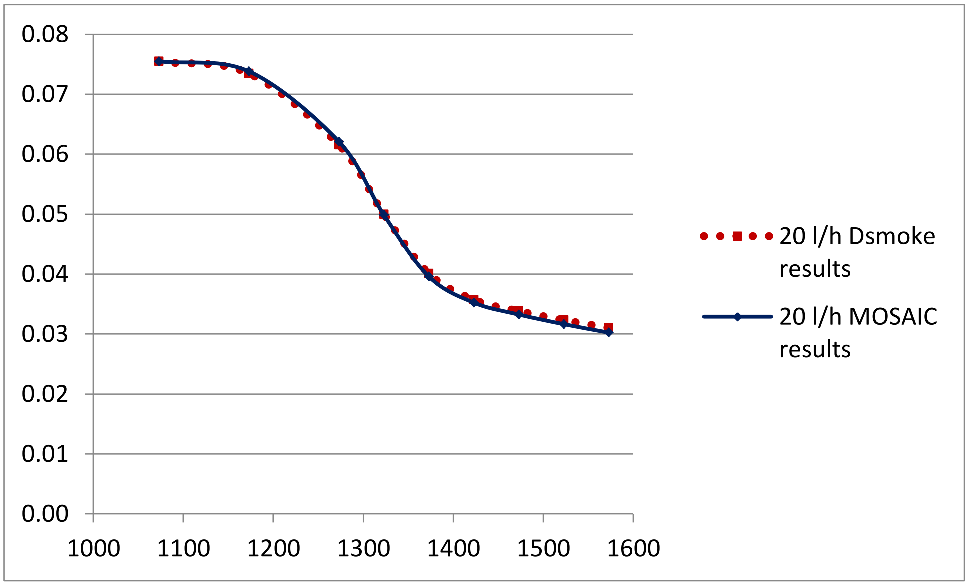


Figure 114 - massive fractions of CO_2 vs reactor temperature, 20 l/h

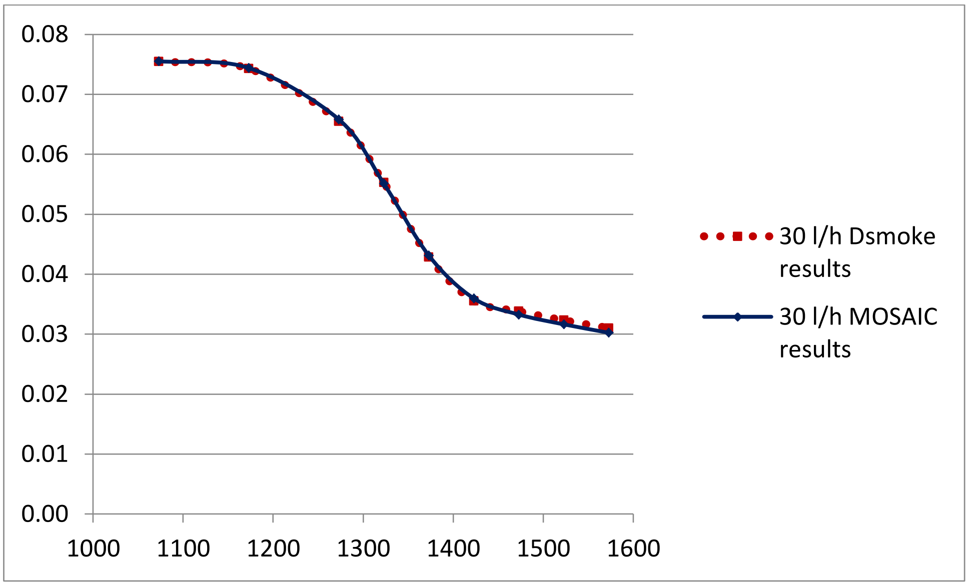


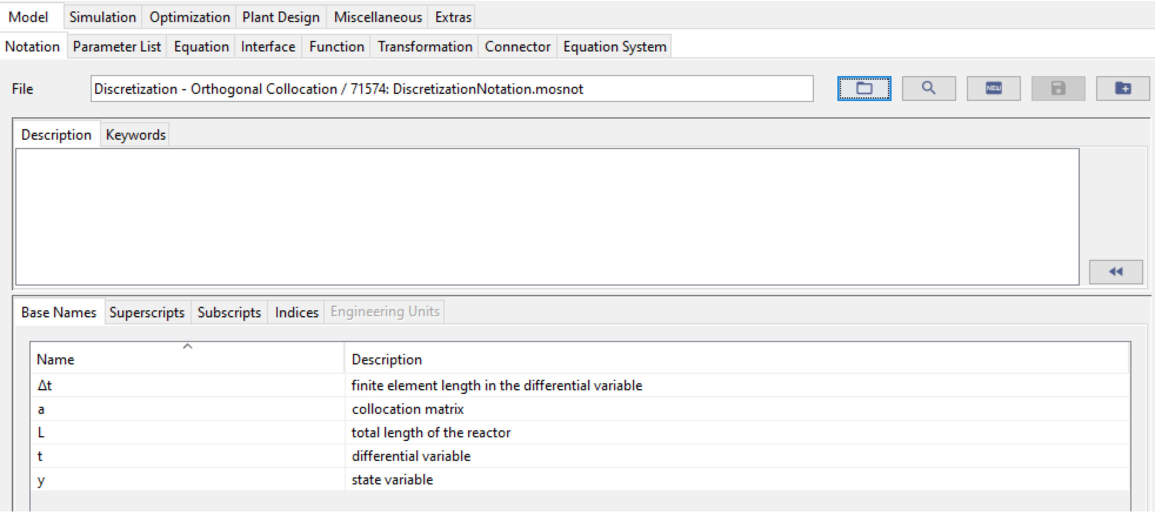
Figure 115 - massive fractions of CO_2 vs reactor temperature, 30 l/h

It's evident that the MOSAIC model results are perfectly comparable with the original DSmoke simulations. The results certify the correct writing of the reactor model in MOSAIC and the right interpretation of and usage of the kinetic data obtained from the regression using DSmoke simulations.

6.4.4 Reactor model: discretizing the equation system

In this work thesis another important instrument of MOSAIC is tried: the automatic discretization. In fact, in MOSAIC it's possible to fully discretize an equation system, choosing the variables to be discretized and even the thickness of the separations. The goal is to transform a differential system into an algebraic system, to facilitate the MOSAIC work of connecting variables in the flowsheeting (for better explanation, see 'Building the complete process in MOSAIC: creating ports'). Practically, the *dz* differential variable is transformed into a discrete position variable, with as many values as are the spacings chosen. In this work the discretization method used is the orthogonal collocation.

To perform the orthogonal collocation discretization in MOSAIC the tab 'Transformation' has to be used to declare the transformation equations to be applied on the studied equation system (in this case the reactor). First the notation of the orthogonal collocation transformation is created:

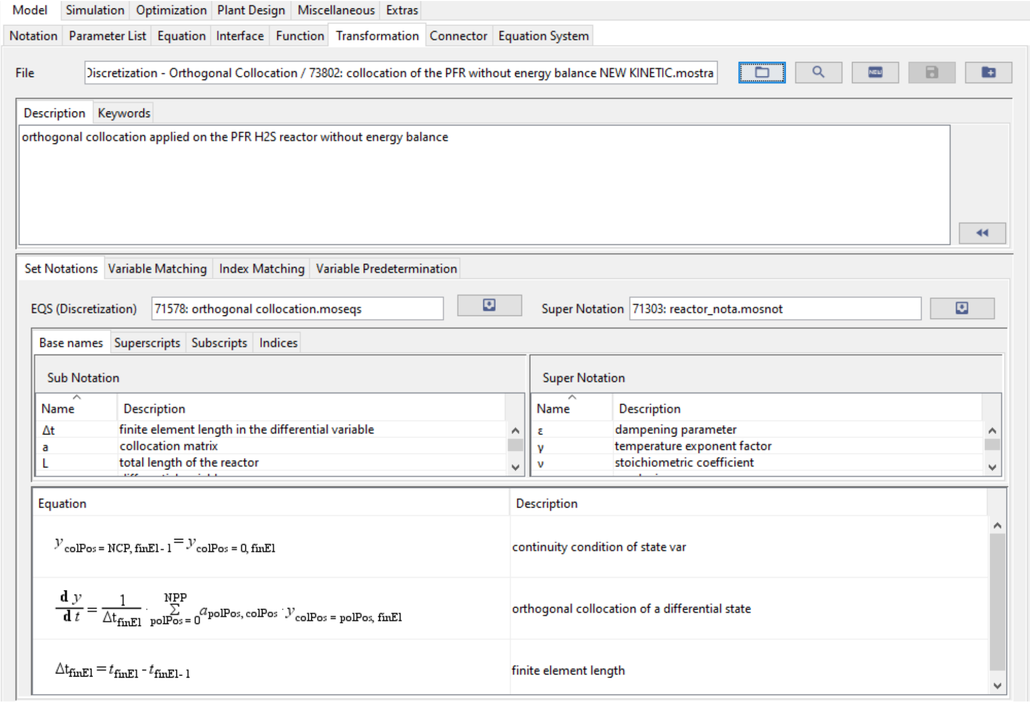


File: Discretization - Orthogonal Collocation / 71574: DiscretizationNotation.mosnot

Name	Description
Δt	finite element length in the differential variable
a	collocation matrix
L	total length of the reactor
t	differential variable
y	state variable

Figure 116 – orthogonal collocation notation

Then, the transformation equations are written choosing as a sub notation the transformation one, while as super notation that of the system to be discretized. The equation to be written in MOSAIC ‘Transformation’ section are shown in the following:



File: Discretization - Orthogonal Collocation / 73802: collocation of the PFR without energy balance NEW KINETIC.mostra

Description: orthogonal collocation applied on the PFR H2S reactor without energy balance

EQS (Discretization): 71578: orthogonal collocation.moseqs

Super Notation: 71303: reactor_nota.mosnot

Sub Notation

Name	Description
Δt	finite element length in the differential variable
a	collocation matrix
L	total length of the reactor

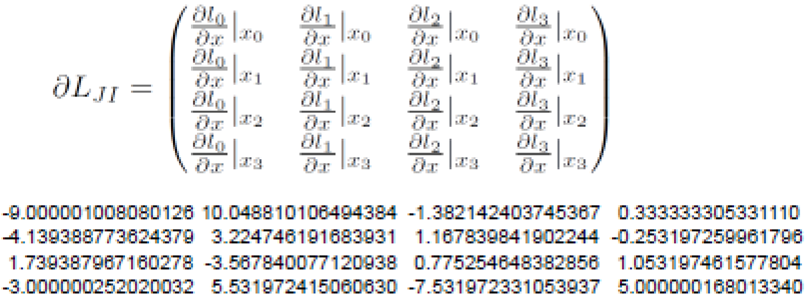
Super Notation

Name	Description
ε	dampening parameter
γ	temperature exponent factor
ν	stoichiometric coefficient

Equation	Description
$y_{\mathrm{colPos=NCP,\,finEl-1}} = y_{\mathrm{colPos=0,\,finEl}}$	continuity condition of state var
$\frac{\mathbf{d}\,y}{\mathbf{d}\,t} = \frac{1}{\Delta t_{\mathrm{finEl}}} \cdot \sum_{\mathrm{polPos=0}}^{\mathrm{NPP}} a_{\mathrm{polPos,\,colPos}} \cdot y_{\mathrm{colPos=polPos,\,finEl}}$	orthogonal collocation of a differential state
$\Delta t_{\mathrm{finEl}} = t_{\mathrm{finEl}} - t_{\mathrm{finEl-1}}$	finite element length

Figure 117 – orthogonal discretization equations

The system space is divided in order to obtain a series of finite elements subdivided in collocation positions. Thus the solver will apply the balance equations in algebraic form in every collocation position, finding the variables solutions for each collocation position in each finite element. Writing the continuity condition of the state variables, the last collocation position of each finite element is forced to correspond to the first collocation position of the next finite element, with a superposition which guarantees the continuity of the state variables in the calculation. The finite element length is simply the difference between two subsequent finite elements; notice that in this case the t value which is the differential variable length in the transformation notation corresponds to the z differential length of the reactor in the reactor super-notation. The matching between variables is done after the transformation equations writing. The last transformation equation, the crucial one, represents the orthogonal collocation of a differential state variable. In particular it explains how to transform the differential state variables into discretized forms. The derivative of y, which represents the state variable, in dt, which is the differential variable, is transformed into a sum of the product between the state variable in a determinate collocation position and finite element, and the value of the collocation matrix *a* which contains the polynomial derivatives, in a specific polynomial and collocation position, all divided by the finite element length (dt replaced by Δt). Practically, knowing the collocation matrix values $a_{i,j}$ far all the polynomial and collocation positions, it's possible to perform the discretization. The collocation matrix, which represents the first derivatives of the Lagrangian polynomials, is available in the Dr.-Ing. Erik Esche's *'Orthogonal Collocation Optimization in Process Science'*:



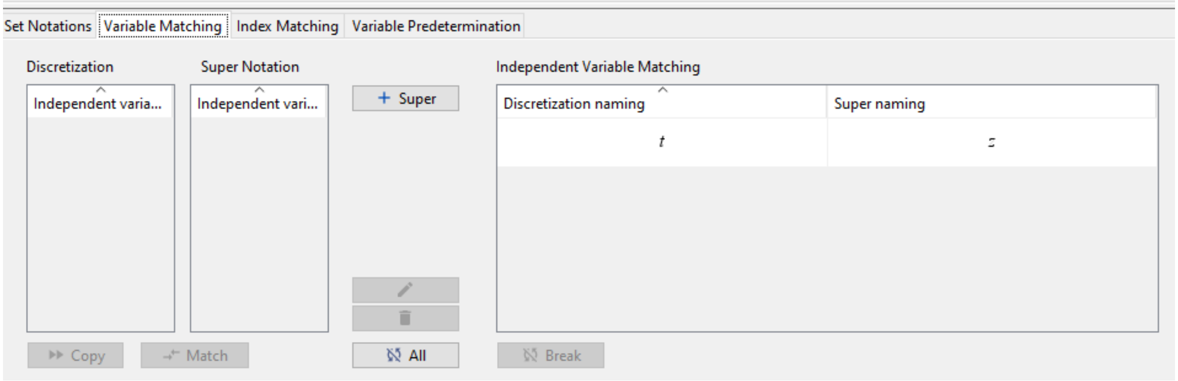
$$\partial L_{JI} = \begin{pmatrix} \frac{\partial l_0}{\partial x}\Big|_{x_0} & \frac{\partial l_1}{\partial x}\Big|_{x_0} & \frac{\partial l_2}{\partial x}\Big|_{x_0} & \frac{\partial l_3}{\partial x}\Big|_{x_0} \\ \frac{\partial l_0}{\partial x}\Big|_{x_1} & \frac{\partial l_1}{\partial x}\Big|_{x_1} & \frac{\partial l_2}{\partial x}\Big|_{x_1} & \frac{\partial l_3}{\partial x}\Big|_{x_1} \\ \frac{\partial l_0}{\partial x}\Big|_{x_2} & \frac{\partial l_1}{\partial x}\Big|_{x_2} & \frac{\partial l_2}{\partial x}\Big|_{x_2} & \frac{\partial l_3}{\partial x}\Big|_{x_2} \\ \frac{\partial l_0}{\partial x}\Big|_{x_3} & \frac{\partial l_1}{\partial x}\Big|_{x_3} & \frac{\partial l_2}{\partial x}\Big|_{x_3} & \frac{\partial l_3}{\partial x}\Big|_{x_3} \end{pmatrix}$$

```
-9.000001008080126 10.048810106494384 -1.382142403745367  0.333333305331110
-4.139388773624379  3.224746191683931  1.167839841902244 -0.253197259961796
 1.739387967160278 -3.567840077120938  0.775254648382856  1.053197461577804
-3.000000252020032  5.531972415060630 -7.531972331053937  5.000000168013340
```

Figure 118 – collocation matrix

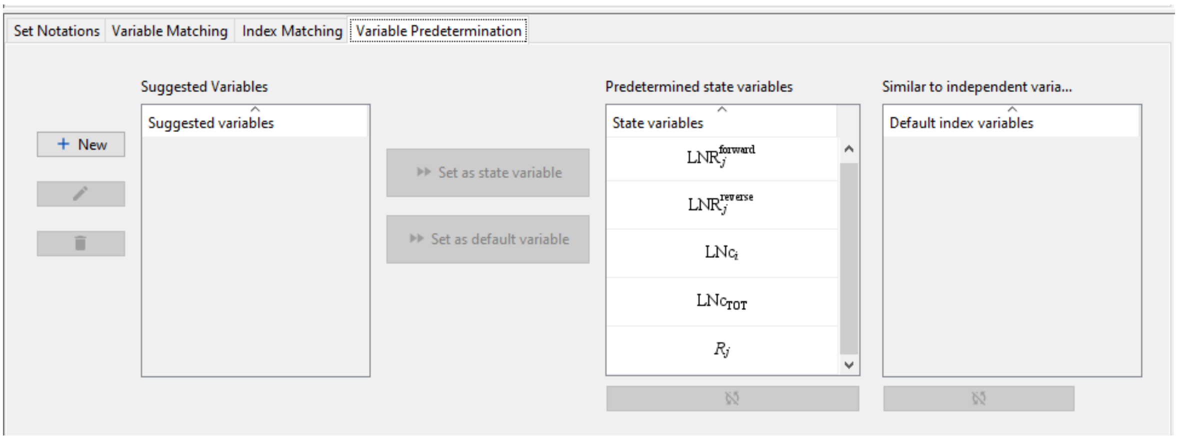
The collocation matrix shown is for a discretizing transformation with 3 polynomial positions and 3 collocation positions.

After the equations writing, it’s necessary to perform the index matching and to select the state variables to be discretized:



Discretization naming	Super naming
t	z

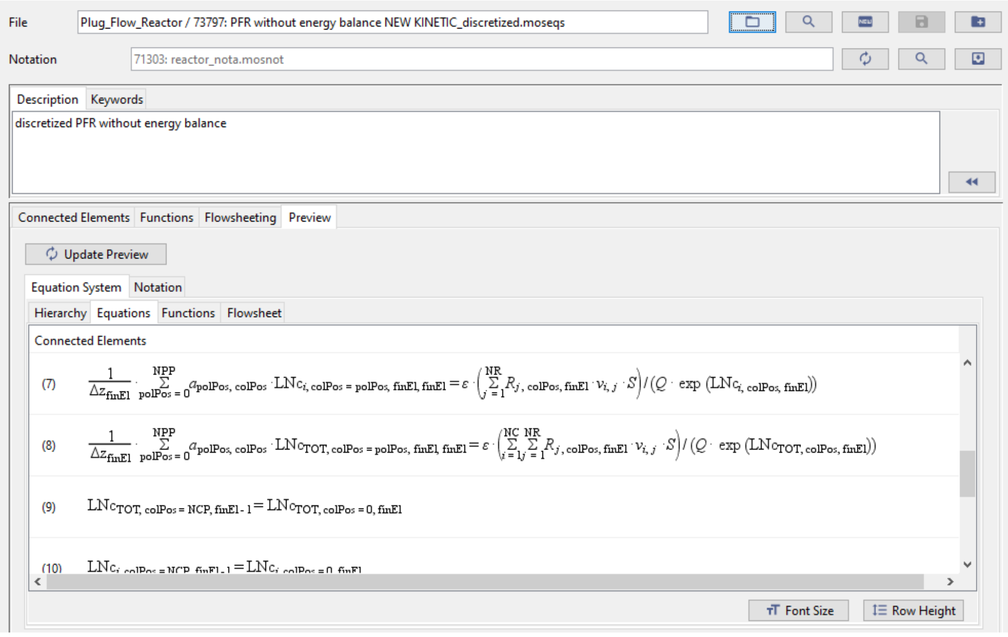
Figure 119 – index matching in discretization transformation



State variables
$\mathrm{LNR}_j^{\mathrm{forward}}$
$\mathrm{LNR}_j^{\mathrm{reverse}}$
LNc_i
$\mathrm{LNc}_{\mathrm{TOT}}$
R_j

Figure 120 - state variable determination in the discretization transformation

To perform the illustrated discretization, it’s only needed to select the equation system to be discretized and apply the transformation.



File: Plug_Flow_Reactor / 73797: PFR without energy balance NEW KINETIC_discretized.moseqs

Notation: 71303: reactor_nota.mosnot

discretized PFR without energy balance

Connected Elements

(7) $$\frac{1}{\Delta z_{\mathrm{finEl}}}\cdot\sum_{\mathrm{polPos}=0}^{\mathrm{NPP}} a_{\mathrm{polPos,\,colPos}}\cdot \mathrm{LNc}_{i,\,\mathrm{colPos}=\mathrm{polPos},\,\mathrm{finEl},\,\mathrm{finEl}}=\varepsilon\cdot\left(\sum_{j=1}^{\mathrm{NR}} R_{j,\,\mathrm{colPos},\,\mathrm{finEl}}\cdot \nu_{i,j}\cdot S\right)/\left(Q\cdot \exp\left(\mathrm{LNc}_{i,\,\mathrm{colPos},\,\mathrm{finEl}}\right)\right)$$

(8) $$\frac{1}{\Delta z_{\mathrm{finEl}}}\cdot\sum_{\mathrm{polPos}=0}^{\mathrm{NPP}} a_{\mathrm{polPos,\,colPos}}\cdot \mathrm{LNc}_{\mathrm{TOT},\,\mathrm{colPos}=\mathrm{polPos},\,\mathrm{finEl},\,\mathrm{finEl}}=\varepsilon\cdot\left(\sum_{i=1}^{\mathrm{NC}}\sum_{j=1}^{\mathrm{NR}} R_{j,\,\mathrm{colPos},\,\mathrm{finEl}}\cdot \nu_{i,j}\cdot S\right)/\left(Q\cdot \exp\left(\mathrm{LNc}_{\mathrm{TOT},\,\mathrm{colPos},\,\mathrm{finEl}}\right)\right)$$

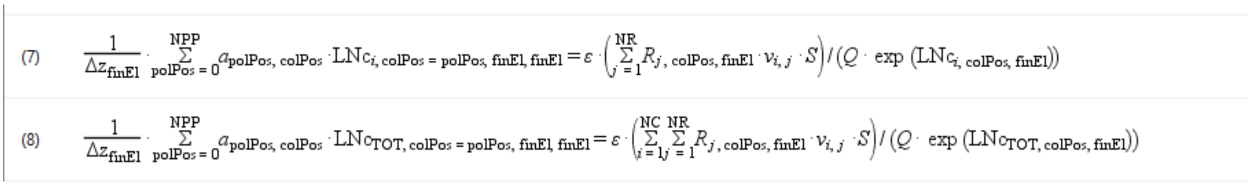
(9) $$\mathrm{LNc}_{\mathrm{TOT},\,\mathrm{colPos}=\mathrm{NCP},\,\mathrm{finEl}-1}=\mathrm{LNc}_{\mathrm{TOT},\,\mathrm{colPos}=0,\,\mathrm{finEl}}$$

(10) $$\mathrm{LNc}_{i,\,\mathrm{colPos}=\mathrm{NCP},\,\mathrm{finEl}-1}=\mathrm{LNc}_{i,\,\mathrm{colPos}=0,\,\mathrm{finEl}}$$

Figure 121 – reactor system discretized

In the figure above the aspect of the mass balances of the discretized reactor is shown. The derivative of the LNc_i is replaced by the transformation equation, and all the state variables have their value discretized, while the other variables like Q volumetric flowrate or S cross surface are left the same. MOSAIC is so able to perform the discretization of the system for the selected variables, allowing an easier connection between inputs and outputs in the flowsheeting.

The system can be solved with all the available packages in MOSAIC for algebraic systems. In this case the discretized reactor is solved using AMPL; the solution of an algebraic system, expecially in the case of the reactor, is often not simple. In particular, there can be problems with the initializing of the system and with its continuity. In this thesis work the solution methods for this problematic equation system is the adding of the dampening factor ε to the equation, and the finite elements length analysis. The dampening factor is used in the mass balances as a multiplier of the right side of the equations:



(7) $$\frac{1}{\Delta z_{\mathrm{finEl}}}\cdot\sum_{\mathrm{polPos}=0}^{\mathrm{NPP}} a_{\mathrm{polPos,\,colPos}}\cdot \mathrm{LNc}_{i,\,\mathrm{colPos}=\mathrm{polPos},\,\mathrm{finEl},\,\mathrm{finEl}}=\varepsilon\cdot\left(\sum_{j=1}^{\mathrm{NR}} R_{j,\,\mathrm{colPos},\,\mathrm{finEl}}\cdot \nu_{i,j}\cdot S\right)/\left(Q\cdot \exp\left(\mathrm{LNc}_{i,\,\mathrm{colPos},\,\mathrm{finEl}}\right)\right)$$

(8) $$\frac{1}{\Delta z_{\mathrm{finEl}}}\cdot\sum_{\mathrm{polPos}=0}^{\mathrm{NPP}} a_{\mathrm{polPos,\,colPos}}\cdot \mathrm{LNc}_{\mathrm{TOT},\,\mathrm{colPos}=\mathrm{polPos},\,\mathrm{finEl},\,\mathrm{finEl}}=\varepsilon\cdot\left(\sum_{i=1}^{\mathrm{NC}}\sum_{j=1}^{\mathrm{NR}} R_{j,\,\mathrm{colPos},\,\mathrm{finEl}}\cdot \nu_{i,j}\cdot S\right)/\left(Q\cdot \exp\left(\mathrm{LNc}_{\mathrm{TOT},\,\mathrm{colPos},\,\mathrm{finEl}}\right)\right)$$

Figure 122 – dampening factor in the discretized reactor

Because of the fast kinetic of the reactions, the right side of the equations tends to change sharply, and this cause a strong difficult in the solver because of the algebraic expression of the equations. In fact the solver must calculate distinct solutions for the discretized positions, without having information about what's going between them; in the case of a strong variations in the variables values between subsequent collocation positions, it would be really difficult for the solver to accurately calculate them. Using a dampening factor $\varepsilon < 1$ reduces the strong variations of the right side of the equations, allowing the solver to better calculate the variations. Practically, the routine to find the solution using a dampening factor is the following:

- choose an ε value < 1 which enables to solve the equation system (e.g. 10^{-12})
- solve the equation system (for example using AMPL solver)
- the solution found is not the right solution, but can be used as initialing source for the equation system calculation with a higher value of ε (e.g. 10^{-11}). In this way the guess values are closer to the solution respect to before, because they're calculated using the same equation system only with a higher dampening factor of the reactions effects. To practically do this with AMPL is sufficient to specify in the 'simulation.run' file to 'include results.dat' instead of 'include initial.dat' after the first routine
- repeat the routine until increasing dampening factor to 1
- the solution obtained with dampening factor $\varepsilon = 1$ is the real system solution

The routine has strong benefits, but cannot solve all the algebraic system problems alone. The other method to facilitate the system solution is the analysis of the finite elements. The finite elements number and length can be decided by the user. In this thesis work the finite elements is increased from 3 to the final value of 10. This choice, even if weights the calculations and the MOSAIC variable specifications adding a lot of new iteration variables, is really useful in the *AG2S*™ reactor. The complete reactor length is divided by the discretization in finite elements; this means that less the finite elements, bigger their dimension with the same reactor length. In the laboratory reactor, as said before, the reactions kinetic is really fast, so the calculation are really difficult because of the big variations in variables values between one position and its subsequent. The 'jump' between two subsequent discretized positions is higher with less finite elements, so the calculation becomes more difficult decreasing the number of finite elements. Another important point is that the higher variation in state variables corresponds to the first 10^{-5} meters of the reactor, while the reactions velocity decreases very quickly in the following centimeters. Thus, the need is to accurately calculate the equation system in its first 10^{-5} meters because of the strong variables variations, and then the step of calculation can be increased. Practically, the finite elements length has to be differentiated: for the first finite elements, the length must be really small, while for the last finite elements the length can be even higher because of the lower speed of variables variations. It’s obvious that with only 3 finite elements this type of differentiation is impossible, while with 10 finite elements it could be feasible to leave the first five finite elements with very short length, and the other five with greater length in order to reach the complete reactor length of 200 mm. To perform this method using MOSAIC and AMPL, the following routine is used:

- the first variable specification in MOSAIC is done using all the finite elements with length 10^{-5} meters
- AMPL results (using the dampening factor progressive increasing method to reach them) are then imported in MOSAIC with ‘Import’ tab, in order to use the first results as new variable specifications of the system
- with the new variable specifications taken by AMPL results, the last finite elements are increased a bit, to slowly approach the complete reactor length;
- a new AMPL solution is obtained, and the AMPL results with new finite element spacing are imported in MOSAIC
- the finite elements are progressively increased in MOSAIC starting from the last one and without touching the first five finite elements. The last finite elements are constantly increased more than the other. For each new finite elements spacing a new solution in AMPL is found, in order to have always good initializing values for the equation system
- the finite elements length is finally varied to reach the 200 mm reactor length, and the final AMPL solution is found.

Using this approach, which combines the dampening factor method and the finite elements dimension analysis, the reactor solution can be found. The final finite elements dimensions are shown in the following; the results of the discretized reactor are not presented because they correspond perfectly to the DAE reactor model results.

$\mathbf{Dz}_{\textbf{Finite El = 1}}$	1.0 e^{-5} [m]
$\mathbf{Dz}_{\textbf{Finite El = 2}}$	1.0 e^{-5} [m]
$\mathbf{Dz}_{\textbf{Finite El = 3}}$	1.0 e^{-5} [m]
$\mathbf{Dz}_{\textbf{Finite El = 4}}$	1.0 e^{-5} [m]
$\mathbf{Dz}_{\textbf{Finite El = 5}}$	1.0 e^{-4} [m]
$\mathbf{Dz}_{\textbf{Finite El = 6}}$	0.02 [m]
$\mathbf{Dz}_{\textbf{Finite El = 7}}$	0.03 [m]
$\mathbf{Dz}_{\textbf{Finite El = 8}}$	0.03 [m]
$\mathbf{Dz}_{\textbf{Finite El = 9}}$	0.06 [m]
$\mathbf{Dz}_{\textbf{Finite El = 10}}$	0.06 [m]

Table 2 – Dz finite elements of the complete reactor evaluation

The next equipment modelled in MOSAIC is the first trap, in which the sulfur condensation is realized with water cooling, but before it a ‘natural cooling’ section is presented.

6.4.5 Natural cooling before the first trap

Because of the high temperature exiting the reactor, it could be interesting to study the little section of tube between the reactor and the first trap, in which even a natural heat exchange between the ambient and the process flow could change significantly the temperature. As opposed to the previous equipment, in this case the material balances are not used to model the process section, but the energy balance is necessary. In fact, in this case the goal is to find the inlet temperature of the first trap, which could be different from the exiting reactor temperature because of the natural cooling effect of the external air to the process flow in the tube.

The balance equations are presented in the following:

1) $\frac{dT}{dz} = \frac{U\ S\ \Delta T}{W\ cp\ L}$
2) $\Delta T = T_{amb} - T$
3) $S = \pi\ Diameter\ L$
4) $cp = A + B\ T + C\ T^2 + D\ T^3 + E/T^2$
5) $U = (\frac{1}{h_e} + \frac{1}{h_i})$
6) $LNc_i = LNc_i^{in}$

Where:

- T is the temperature of the process flow [K]
- T_{amb} is the ambient temperature [K]
- U is global heat transfer coefficient [W/(m^2K)], in this case calculated using h_e and h_i which are internal and external heat transfer coefficients [W/(m^2K)]. The resistance of the steel tube is considered negligible
- W is the massive flowrate of the process flow [kg/s]
- S is the tubular exchange surface [m^2]
- Diameter is the tubular diameter [m]
- cp is the heat capacity coefficient of the charge [j/(mol K)]. The cp values (A, B, C; D and E) are taken from the NIST Chemistry WebBook
- LNc_i is the logarithm of the component i-th concentration

Actually, there wouldn't be the need to explicit the equivalence between the input and output concentration; this is necessary for the final step of MOSAIC evaluation, in which the entire process is solved assembling together all the unit operations. The results of this simple unit are listed in the graph below:

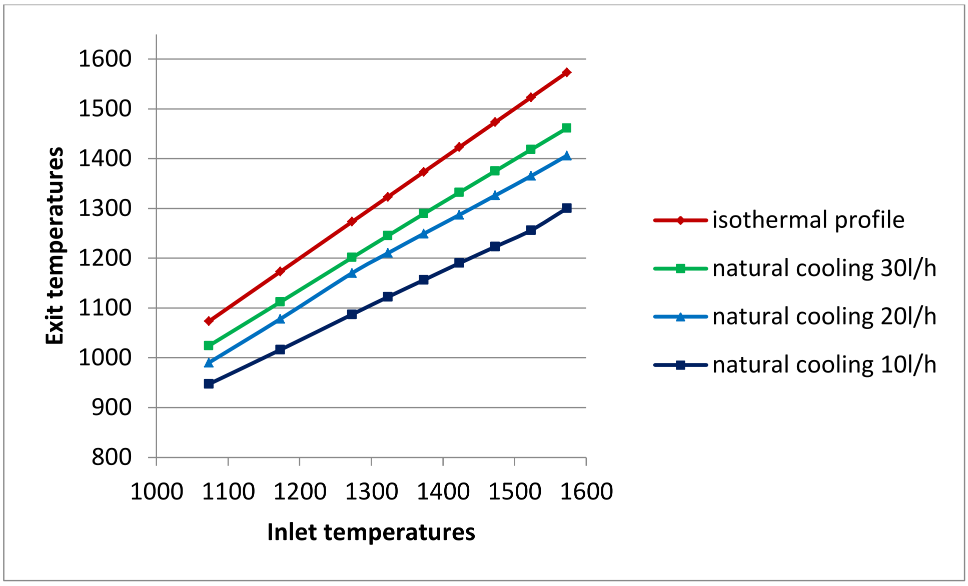


Figure 123 – temperature results after the natural cooling section

The results show that effectively there is a significant decrease of temperature because of the high temperature of the reactor outlet (high ΔT with air) and the high thermal conductivity (and small thickness) of the steel tube (316L $\lambda = 15$ [W/mK]) which doesn't oppose resistance to heat exchange. It's also interesting to see the temperature behavior changing the massive flowrate; even if with lower velocities the heat exchange is not favored because of the decrease of h_i (increase of the resistance $1/h_i$), the charge needs less exchanged heat to reduce its temperature (because of the term W decrease), so removing about the same heat from the process flow results in a bigger decrease of temperature for lower flowrates. The temperature exit values represent the inlet of the first trap, which is the second equipment modelled in MOSAIC.

6.4.6 The first trap in MOSAIC

The first trap, as seen even in the Dynsim section of the thesis work, it's a flash with heat exchange to ensure the sulfur to be condensed, so the temperature of the flash should be lower than the boiling temperature of the sulfur at atmospheric pressure (445°C). Actually, it can be possible in the same flash to remove also the water formed in the *AG2S*™ reaction, lowering the temperature under 100°C, but in this case the sulfur, which has a melting temperature of 115°C, becomes solid. In both cases, the temperature should go under 200°C to ensure the sulfur condensation and not to have hot charges entering the following traps. The flash has the double meaning to separate components and to reduce the process temperature. The first trap principal equations are shown below:

1) $F\,z_i = V\,y_i + L\,x_i$
2) $F\,H_F = V\,H_V + L\,H_L + Heat$
3) $H_F = DH_{form}^{gas} + intCP_F$
4) $H_V = DH_{form}^{gas} + intCP_V$
5) $H_L = DH_{form}^{liquid} + intCP_L$
6) $Heat = U\,S\,\Delta T_{MLN}$
7) $T_{out}^{water} = T_{in}^{water} + Heat/(W\,cp^{water})$
8) $y_i = K_i x_i$
9) $\sum_{i=1}^{NC} x_i - y_i = 0$

Where:

- F, V and L are the molar flowrate of the inlet, the vapor outlet and the liquid outlet respectively [mol/s]
- z_i, y_i and x_i are the molar fractions of the inlet, the vapor outlet and the liquid outlet respectively
- H_F, H_V and H_L are the specific molar enthalpies of the inlet, the vapor outlet and the liquid outlet respectively [j/mol]
- ΔH_{form}^{gas} and ΔH_{form}^{liquid} are the specific molar enthalpies of formation in gas or liquid phase of the components [j/mol]

  The enthalpy values are taken from the ‘Perry’s Chemical Handbook’.
- $intCP_F$ is the $\int_{298}^{T_{in}} cp_F\,dT$, where cp_F is the heat capacity of the inlet mixture [j/(molK)]
- $intCP_V$ is the $\int_{298}^{T_{out}} cp_V\,dT$, where cp_V is the heat capacity of the vapor outlet mixture [j/(molK)]
- $intCP_L$ is the $\int_{298}^{T_{out}} cp_L\,dT$, where cp_L is the heat capacity of the liquid outlet mixture [j/(molK)]

  All the cp values are taken from NIST Chemistry WebBook
- Heat is the heat removed from the system using the water cooling per second [W]

- T_{in}^{water} and T_{out}^{water} are the inlet and outlet cooling water temperatures [K]
- U is the global heat exchange coefficient [W/(m^{2}K)]
- ΔT_{MLN} is the logarithmic mean value of temperature between the flash and the cooling water [K]
- S is the exchange area [m^2]
- W is the cooling water mass flowrate [kg/s]
- K_i is the VL equilibrium constant of the i-th species; it represents the ratio between the liquid and the vapor fugacity of the component i. Because of the presence of supercritical components, the fugacity in liquid phase of the i-th component can be its vapor pressure (for subcritical components) or Henry constant for supercritical components, considering ideal mixture and negligible the Pointing correction. All the vapor pressure and Henry's constant values are taken from NIST Chemistry WebBook.

The coolant to perform the heat exchange, as expressed in the equations, is line water, so water at about 25°C. The design variables chosen for this simulation are the temperature at the inlet of both the process and the water flow and the massive flowrate of the line water which is, in the real laboratory, equivalent to increase or decrease the water flow manipulating the laboratory valve. The objective is to reach a temperature lower than 200°C, and under 100°C if possible to condense even water, and to solve the flash in order to obtain the concentrations of the compounds in liquid and gas phase. In the second case, with temperature lower than melting point of sulfur, the previous system cannot model the S solid phase. Actually, the system becomes a three phase system with VLS equilibrium. But, considering the real system, at temperatures around 80°C and atmospheric pressure only the sulfur can be in solid phase, so there is a VLE (vapor liquid equilibrium) between all the other species, while sulfur is considered in solid phase; the results of the presented model are then correct if temperature is over than 115°C; if the desired temperature is under this value, it's necessary to change the model to consider the sulfur in solid phase and the VLE between the other species. The results will vary depending on the process flowrate, the inlet temperature, composition and the water coolant flowrate. In this thesis work the illustrative solution for an inlet charge of 1400 K with a water flowrate of 10 l/h is presented, to discuss the principal aspects of the flash.

Components	xi	yi	T out	Twater out	L mol/s	V mol/s
H_2S	0.0184	0.0106	345 K	301.8 K	1.43e-6	3.39e-4
CO_2	0.0345	0.0240				
H_2O	0.8946	0.0209				
CO	0.0099	0.0260				
N_2	0.0263	0.9050				
Sulfur	0	0				
H_2	0.0153	0.0133				

Table 3 – example of first trap results

The sulfur content is considered in solid phase because the temperature reached in the first trap, 72°C, is lower than the 115°C melting point of sulfur. The solid sulfur has a fictitious molar flowrate of 8.04e-6 mol/s, equivalent to about 1 g/h of sulfur condensing in the trap; the flowrate value is fictitious because in the real laboratory the solid sulfur together with the condensed water are accumulated in the trap, considering their low quantities, so its real meaning is an accumulation of sulfur in the trap. With the chosen flowrate is observable that cooling water has a very low increase of temperature, while the process flow reaches the desired temperature thanks to the differences between heat capacities and mostly between the respective mass flows. It’s also possible to observe that in liquid phase there is a big quantity of water, but also the other components are present because of their solubility in water represented by Henry’s law constant. Actually, the liquid molar flowrate is very low, so the lost quantities of other compounds are really low (less than 0.2% for the produced syngas).

6.4.7 The second and third traps in MOSAIC

The two final traps used to H_2S removal are modelled, as in Dynsim simulation, using the resulting conversion data of the article '*Absorption of H_2S in NaOCl Caustic Aqueous Solution, Chen et Al., New Jersey Institute of Technology and Taiwan University, 2001*'. The extent of the reaction method is used to create the equation system. The extent of reaction j-th is called λ_j, and it represents the ratio between the reacted moles in the j-th reaction of a generic component i and its stoichiometric value in the reaction j-th; practically, it symbolizes the progress of the reaction j-th in the considered system. Because of the relation already written, in a reacting system like the two final traps, the exiting moles of the components can be easily determinate by using λ_j:

$$n_i^{out} = n_i^{in} + \sum_{j=1}^{NR} \nu_{i,j}\, \lambda_j$$

Where:

- n_i^{out} is the molar quantity (or molar flowrate) exiting the system
- n_i^{in} is the molar quantity (or molar flowrate) entering the system
- $\nu_{i,j}$ are the stoichiometric coefficients of the i-th compound in the j-th reactions
- λ_j is the extent of the j-th reaction

Knowing the extents of the considered reactions it's possible to obtain all the exiting molar values of the traps. Before presenting the complete equation system, it can be interesting to explain how to choose the right number of reactions to be written in order to find the solution. In fact, the right number of λ_j to be used isn't a fixed number, but it's an intrinsic characteristic of the system, so finding it requires a preliminary study with crucial importance; if a wrong number of λ_j is used to find the solution, the system problem becomes not well posed compromising its feasibility. The procedure to find the NR, i.e. the number of reactions to be written, equivalent to the number of extents of reactions to be used in the modeling system, is explained in the following.

The relation used to find NR is:

$$NR = NC - NA$$

Where:

- NR is the number of reactions to be written
- NC is the number of components
- NA is the rank of the atomic matrix

It’s evident that to obtain the NR value is necessary to find NA. The first step to find it is to build the atomic matrix of the system components, creating a table with in the first row the elements characterizing the process, and in the first column the compounds present in the system; the matrix is now completed inserting in every box thus created, the number of atoms of the corresponding row element included in the corresponding column compound. The atomic matrix of the current system is presented:

	H	**S**	**Na**	**O**	**Cl**
H_2S	2	1	0	0	0
NaOH	1	0	1	1	0
Na_2S	0	1	2	0	0
H_2O	2	0	0	1	0
NaOCl	0	0	1	1	1
Na_2SO_4	0	1	2	4	0
NaCl	0	0	1	0	1

Table 4 – atomic matrix for the second trap system

The NA value is the rank of the atomic matrix already created. In this case the rank of the atomic matrix is 5, thus $NR = 7 - 5 = 2$.

Two reactions are necessary to be written in order to fulfill the system. The reactions must include all the reactive components and be linearly independent each other. The two reactions chosen at this scope are the same illustrated in Dynsim simulation section:

$$H_2S_{(g)} + NaOH_{(aq)} \overset{\lambda_1}{\rightarrow} Na_2S_{(aq)} + 2H_2O_{(l)}$$

$$Na_2S_{(aq)} + 4NaOCl_{(aq)} \overset{\lambda_2}{\rightarrow} Na_2SO_{4(aq)} + 4NaCl_{(aq)}$$

All the molar balances are written using these two reactions. Now it's necessary to saturate the two degrees of freedom with two reactions data; these data are the reactions conversion taken by the article. For what concerns the first reaction, the conversion value resulting from the article has 0.8 value, while the second reaction, which is useful to fix in a stable form the Na_2S, has a complete running with a unitary conversion. These two equations are included in the equation system, together with the equations necessary to link the traps at the inlet and outlet (this passage will be explained in the following chapter). Thus, the final traps equation system is:

1) $n_i^{out} = n_i^{in} + \sum_{j=1}^{NR} \nu_{i,j}\, \lambda_j$
2) $\mathrm{X}_1 = (n_1^{in} - n_1^{out})/n_1^{in}$
3) $\mathrm{X}_2 = \ \lambda_2/\ \lambda_1$
4) $n_{tot} = \sum_{i=1}^{7} n_i^{out}$
5) $\Upsilon_i = n_i^{out}/n_{tot}$
6) $n_i^{in} = V\, y_i + charged_i$

Where:

- X_j are the molar conversions of the j-th reactions
- n_{tot} is the total molar flowrate exiting the trap [mol/s]
- Υ_i are the molar fractions exiting the trap
- y_i are the molar fractions entering the trap
- V is the molar flowrate entering the trap [mol/s]
- $charged_i$ is the charged quantity of NaOH and NaOCl

Assigning $X_1 = 0.8$ and $X_2 = 1$ the system can be easily solved. In this case the model is built in order to find not only the exit molar fraction and flowrate of the H_2S, which is the most important component to be analyzed in this process section, but also to calculate the consumption of NaOH and NaOCl. A complete illustrative example of results is presented before illustrating all the exiting molar fractions of H_2S from the second and third trap.

The outlet of the first trap previously presented is considered as input of the illustrative example, representing the second trap; the results are shown in the following:

COMPONENT	MOLAR FLOWRATE OUT [mol/s]	MOLAR FRACTION OUT
H_2S	1.7875 e-6	0.005
CO_2	1.4830 e-5	0.042
H_2O	1.4585 e-5	0.041
CO	2.0460 e-6	0.007
H_2	5.8754 e-6	0.017
NaOH	2.8499 e-6	/
NaOCl	7.1399 e-5	/

Table 5 – example of trap 2 results

The NaOH and NaOCl results are based on a fictitious charge of 1.0 e-5 mol/s of NaOH and 1.0 e-4 mol/s of NaOCl, so actually the fictitious outlet molar flowrate represent the consumption of the compounds, in particular 7.15 e-6 mol/s of NaOH and 2.86 e-5 mol/s of NaOCl. These data correspond of about 1 g/h and 7.6 g/h of NaOH and NaOCl respectively in the second trap for this example, but the order of magnitude of these solutions can be extended for all the laboratory cases with engineering interest, thus excluding the scenario with a H_2S conversion less than 70%. In the following tables only the H_2S molar flowrate and molar fractions outlet will be presented including all the scenarios illustrated in the previous paragraphs for the second and third trap. The third trap outlet represents also the process outlet which goes directly on the top of the fume hood, thus the following tables conclude the unit operations evaluation in MOSAIC.

RESULTS OF LAST TRAPS FOR 10 l/h

Temperature [K]	out trap 2 [mol/s]	out trap 2 y H_2S	out trap 3 [mol/s]	out trap 3 y H_2S
1073	9.30E-07	1.10E-04	0.008455938	1.86064E-07
1173	7.79E-07	1.10E-04	0.007056459	1.55779E-07
1273	4.84E-07	1.08E-04	0.004470014	9.68E-08
1323	3.64E-07	1.10E-04	0.003311483	7.27327E-08
1373	2.85E-07	1.10E-04	0.002593855	5.69333E-08
1423	2.35E-07	1.10E-04	0.002137612	4.69314E-08
1473	1.94E-07	1.10E-04	0.001763768	3.87324E-08
1523	1.60E-07	1.10E-04	0.001454899	3.19539E-08
1573	1.32E-07	1.10E-04	0.001202008	2.6401E-08

Table 6 – complete results for trap 2 and trap 3 at 10 l/h

RESULTS OF LAST TRAPS FOR 20 l/h

Temperature [K]	out trap 2 [mol/s]	out trap 2 y H_2S	out trap 3 [mol/s]	out trap 3 y H_2S
1073	1.86E-06	2.20E-04	0.008465512	3.72566E-07
1173	1.59E-06	0.000221	0.007178232	3.17143E-07
1273	1.14E-06	0.000221	0.005146771	2.27423E-07
1323	8.39E-07	0.00022	0.003806702	1.67716E-07
1373	6.00E-07	0.00022	0.002732714	1.20076E-07
1423	4.71E-07	0.00022	0.002145096	9.41971E-08
1473	3.88E-07	0.000235	0.001653856	7.75708E-08
1523	3.20E-07	0.00022	0.0014549	6.39078E-08
1573	2.64E-07	0.00022	0.001202008	5.2802E-08

Table 7 complete results for trap 2 and trap 3 at 20 l/h

RESULTS OF LAST TRAPS FOR 30 l/h

Temperature [K]	out trap 2 [mol/s]	out trap 2 y H_2S	out trap 3 [mol/s]	out trap 3 y H_2S
1073	2.79097E-06	0.00033	0.008468708	5.58194E-07
1173	2.38931E-06	0.000331	0.00721939	4.77862E-07
1273	1.78779E-06	0.000331	0.005393553	3.57558E-07
1323	1.37218E-06	0.000331	0.004148885	2.74435E-07
1373	9.72924E-07	0.00033	0.002951457	1.94585E-07
1423	7.1934E-07	0.000329	0.002186063	1.43868E-07
1473	5.80709E-07	0.000329	0.001764966	1.16142E-07
1523	4.78766E-07	0.000329	0.0014549	9.57532E-08
1573	3.95579E-07	0.000329	0.001202008	7.91158E-08

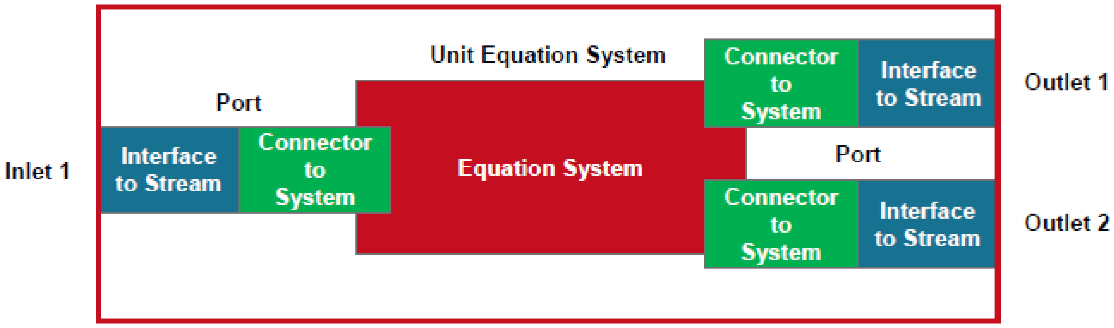
Table 8 - complete results for trap 2 and trap 3 at 30 l/h

In the last two paragraphs of MOSAIC section the complete process is built assembling all the unit operations already described.

6.4.8 Building the complete process in MOSAIC: creating ports

After explaining all the unit operations models in MOSAIC, the final paragraph is used to explain the process building in the environment. It’s almost evident that MOSAIC works with the equations, while the standard simulators, e.g. Dynsim, use a different approach, the flowsheeting. In particular, the simulator are based on the ‘box and arrows’, threating the process components as unit operations linked each other by streams. The importance is then focused not on the equations characterizing the system, but on the inputs and outputs of the different unit operations, which are streams of information between the equipment. MOSAIC enables the user to create a complete process linking the unit operations each other, using the 'Flowsheeting' tab; for this purpose, the 'ports and streams' functions will be better explained.

The principal goal of the flowsheeting is to enable the exchange of information between the unit operations which symbolize the equation systems previously discussed. Every large equation system is the mathematical model of a piece of equipment, which is in the real process connected with inlets and outlets to the other devices. In order to allow the 'speaking' between unit operations, some passages must be done. At first, the equation system needs to be equipped with 'ports', which are the 'doors' thanks to receive inputs from the external or send outputs. To do that, it's necessary the creation of the 'connectors' and of an 'interface'. The connectors are able to translate the unit operation inputs and outputs to the interface language; in this way the interface, which has its own language and notation, can 'speak' with the equation system which receive and send information with its language and notation. The interface between the connectors and the streams is used to define the stream type of the information passing through it. In this way a port is created able to 'speak' with a stream which can carry the information to other unit operations. A simplified scheme of the 'ports and streams' configuration is presented below:



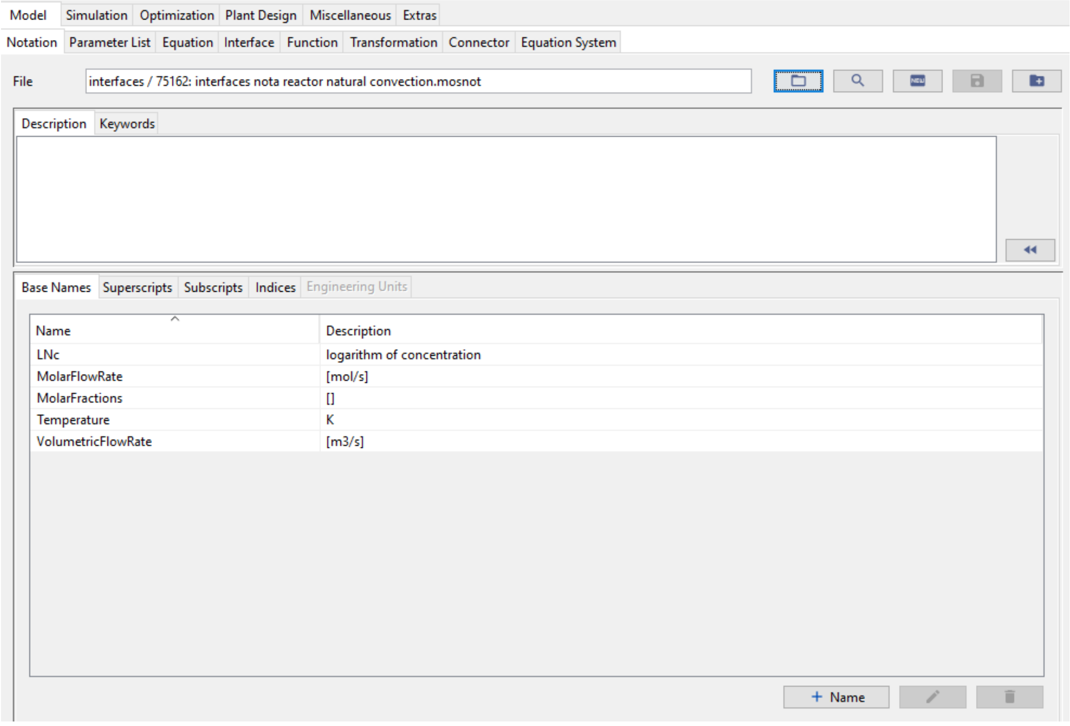
Connector connects variables from the System to the Interface

Interface defines the Stream Type (Heat flow, Mass flow, etc.)

Figure 124 – ports and streams configuration

To better specify the behavior of the ports and streams, it can be useful the example of an equation system sending an output information. The connector to system created for the outputs converts the equation system notation into the interface one, then the interface receive the connector data and defines which type of information is carried to the stream. Now the stream can receive the output of the equation system, knowing the type of information contained in it; in this way the stream can provide the data to the interface of the recipient unit operations, specifying the type of information carried. The recipient interface speaks with the respective connector which converts the data in the equation system language, and so the output of the first equation system is considered as input from the second unit operation.

Thus, as written before, to perform the linking between the equation systems, the following passages are necessary. At first, the notation of the interface has to be created:

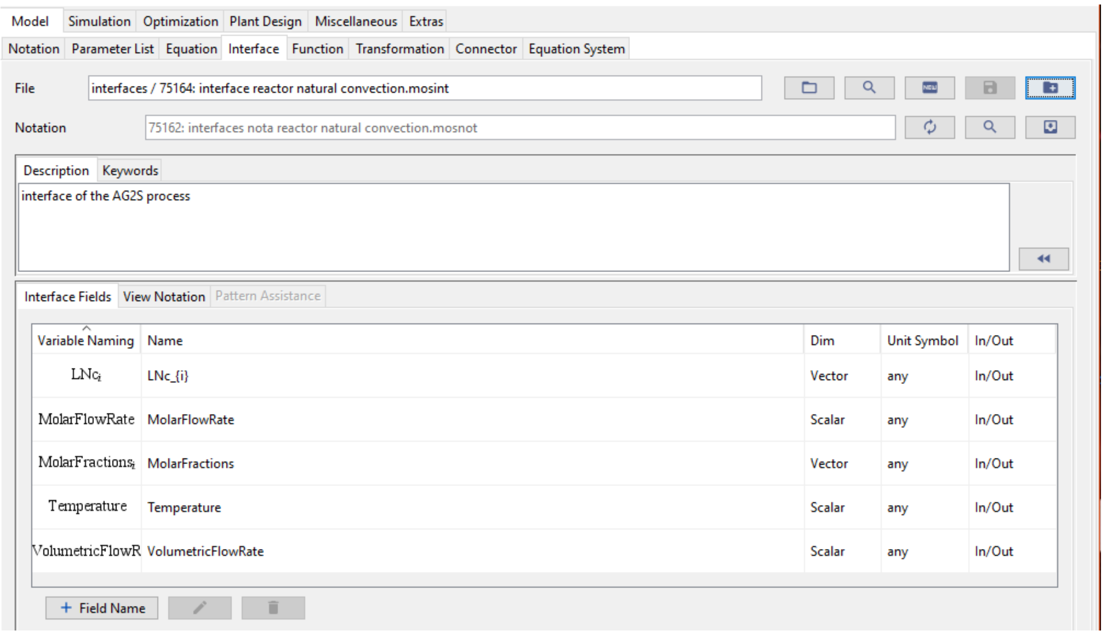


File: interfaces / 75162: interfaces nota reactor natural convection.mosnot

Name	Description
LNc	logarithm of concentration
MolarFlowRate	[mol/s]
MolarFractions	[]
Temperature	K
VolumetricFlowRate	[m3/s]

Figure 125 – notation of the interface

thus, the interface can be built using its notation. In the interface fields not only the name of variables, but also their variable namings and dimension must be specified. In fact notice that the molar fractions and the logarithm of concentrations are considered as vectors, because they have indices.



File: interfaces / 75164: interface reactor natural convection.mosint

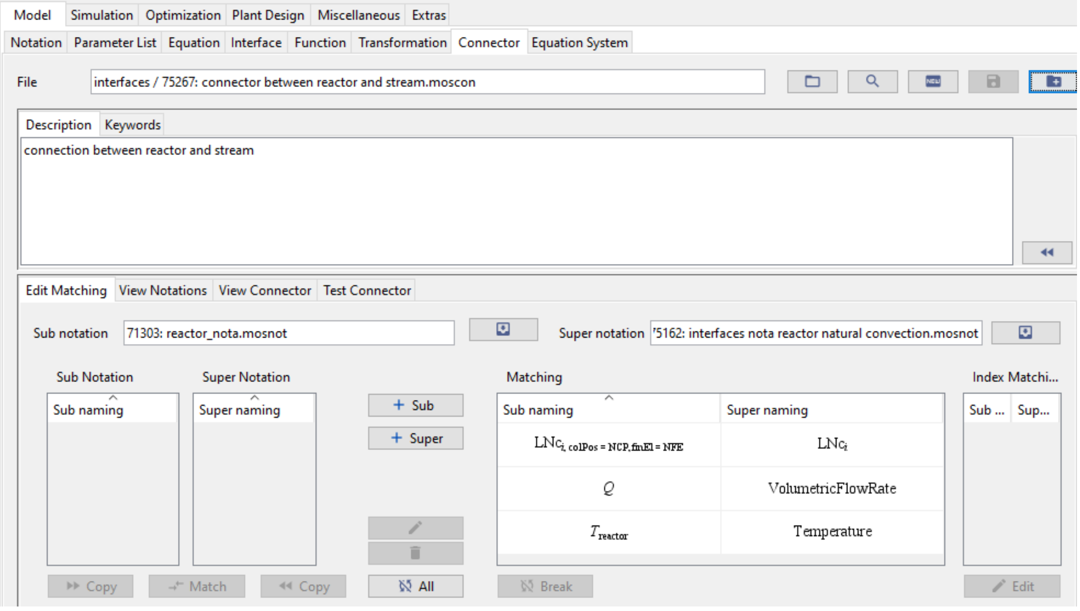
Notation: 75162: interfaces nota reactor natural convection.mosnot

interface of the AG2S process

Variable Naming	Name	Dim	Unit Symbol	In/Out
LNc_i	LNc_{i}	Vector	any	In/Out
MolarFlowRate	MolarFlowRate	Scalar	any	In/Out
$MolarFractions_i$	MolarFractions	Vector	any	In/Out
Temperature	Temperature	Scalar	any	In/Out
VolumetricFlowR	VolumetricFlowRate	Scalar	any	In/Out

Figure 126 – interface creation

The following work is to build the connectors. While the interface can be common to all the unit operations, because it contains the type of information carried by the streams (molar or volumetric flowrate, molar fractions or concentrations, temperature), the connectors have the task to link the general interface language to the specific unit operation language; for this reason the connectors are strictly associated to their equation systems. In the figure below some connectors examples are presented.



File: interfaces / 75267: connector between reactor and stream.moscon

connection between reactor and stream

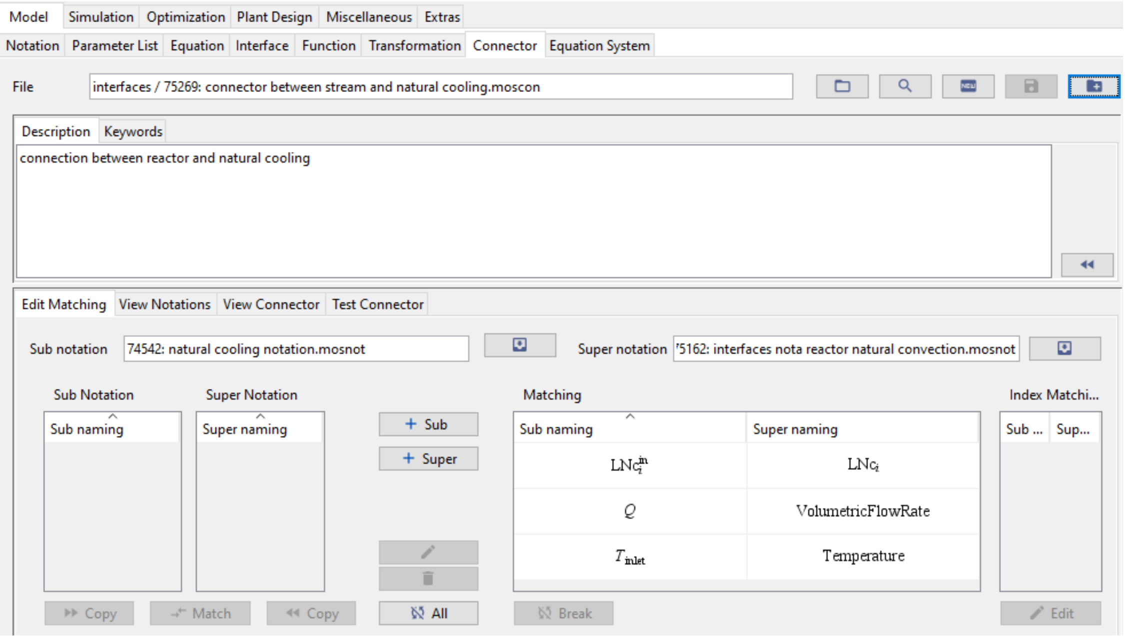
Sub notation: 71303: reactor_nota.mosnot

Super notation: '5162: interfaces nota reactor natural convection.mosnot

Matching

Sub naming	Super naming
$LNc_{i,\ colPos=NCP,finEl=NFE}$	LNc_i
Q	VolumetricFlowRate
$T_{reactor}$	Temperature

Figure 127 – connector between reactor and stream



File: interfaces / 75269: connector between stream and natural cooling.moscon

connection between reactor and natural cooling

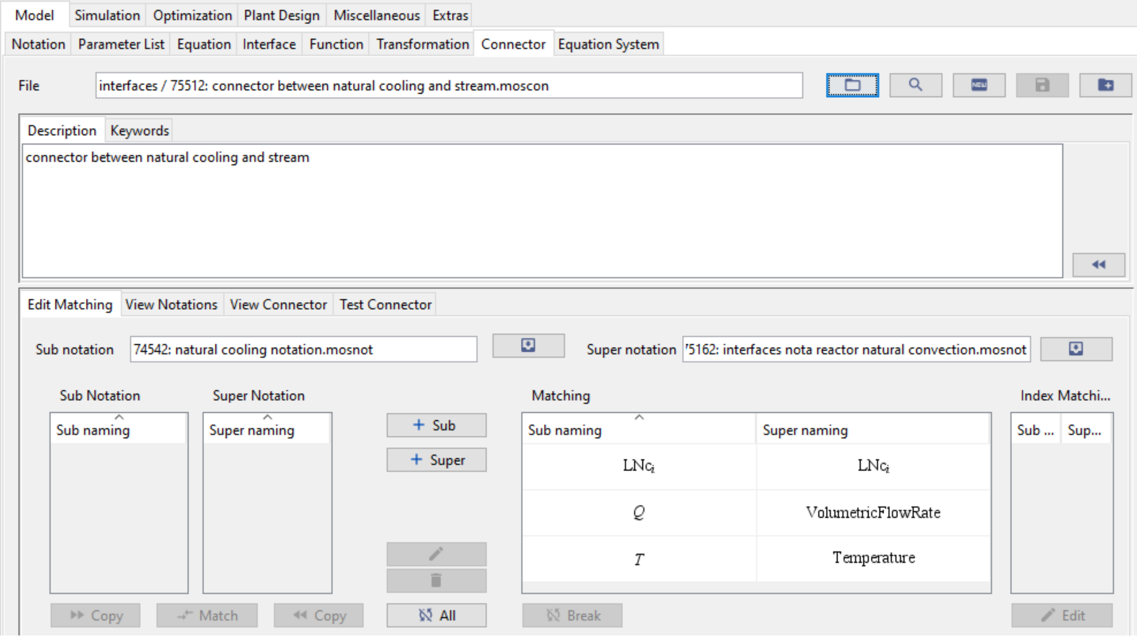
Sub notation: 74542: natural cooling notation.mosnot

Super notation: '5162: interfaces nota reactor natural convection.mosnot

Matching

Sub naming	Super naming
$\mathrm{LNc}_i^{\mathrm{in}}$	LNc_i
Q	VolumetricFlowRate
T_{inlet}	Temperature

Figure 128 – connector between stream and natural cooling



File: interfaces / 75512: connector between natural cooling and stream.moscon

connector between natural cooling and stream

Sub notation: 74542: natural cooling notation.mosnot

Super notation: '5162: interfaces nota reactor natural convection.mosnot

Matching

Sub naming	Super naming
LNc_i	LNc_i
Q	VolumetricFlowRate
T	Temperature

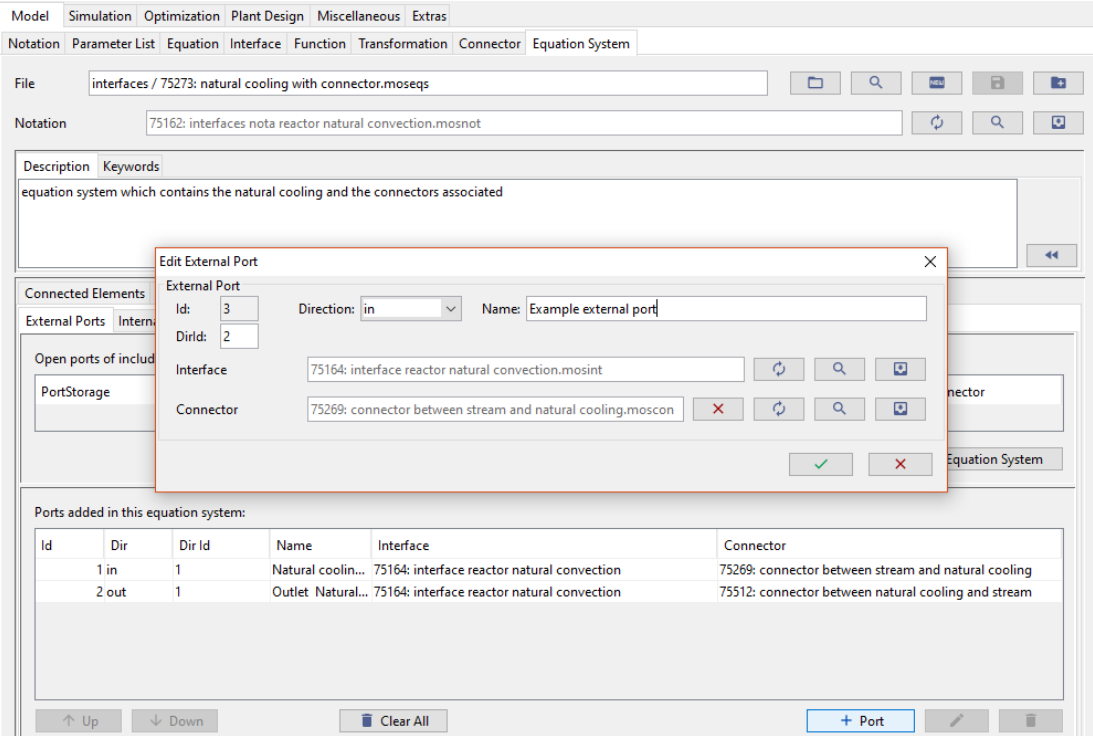
Figure 129 – connector between natural cooling and stream

The first thing to notice is that the equation systems can have different connectors for the inputs and outputs. In this case the ‘connector between stream and natural cooling’ refers to the input section of the equation system, in which the temperature is converted in T_{in} from the connector, while in the output section of the unit operation the connector operates differently; in fact in the second case the temperature to be transmitted is the outlet temperature of the natural cooling, which in the equation system notation is represented by T symbol. Notice even that while the temperature can assume different notations and meanings in the equation systems, as evident in the three examples chosen, the stream ‘Temperature’ is always the same, and represents only a generic temperature information. It’s the task of the connector to assign to the stream data its meaning in the considered equation system.

The reactor connector example is a good introduction to better explain the choice of discretizing the first unit operation of the process. The reactor is the crucial unit operation of the process, and it’s modelled as an ADE (algebraic differential equation) system. The calculation of the reactor model is so a differential calculation on the length z; applying a discretization on the reactor model, each part of the unit operation is treated as a separate algebraic equation system, so the solutions of the system are not continuous but exactly discretized. In this way the output of the reactor is better defined, because it’s the value of the desired variable at the final finite element and final collocation position of the discretized reactor. In this way the definition of output variables in the ADE reactor system is better readable for MOSAIC, without risk of imprecisions in taking them. This procedure can be applied even for the natural cooling section, but in this work thesis presentation the second unit operation is left in differential form because the solvability of the system is very much higher, and the only variable to be passed that changes in the natural cooling section is the temperature, while the other variables are left with the same inlet value, so the risk to get not the real final value is really restricted.

As last, to create the connectors it’s necessary to choose as sub-notation the considered unit operation notation, and as super-notation the interface notation, then the equation systems variables must be connected to their respective streams data.

At this point, there are all the instruments to create the equation systems ports. To do this is sufficient to go in ‘Equation System’, selecting the ‘Flowsheeting’ tab and adding the external ports using the ‘Add ports’ button as in the next figure.



File: interfaces / 75273: natural cooling with connector.moseqs

Notation: 75162: interfaces nota reactor natural convection.mosnot

equation system which contains the natural cooling and the connectors associated

Id: 3
Direction: in
Name: Example external port
DirId: 2
Interface: 75164: interface reactor natural convection.mosint
Connector: 75269: connector between stream and natural cooling.moscon

Ports added in this equation system:

Id	Dir	Dir Id	Name	Interface	Connector
1	in	1	Natural coolin...	75164: interface reactor natural convection	75269: connector between stream and natural cooling
2	out	1	Outlet Natural...	75164: interface reactor natural convection	75512: connector between natural cooling and stream

Figure 130 – creation of an external port

In the window that appears it’s necessary to choose the direction of the port (input or output), and the correspondent interface and connector to be used. In the figure above the natural cooling external ports are visible: the input and the outlet are listed in the ‘Ports added in this equation system’ window, with the references at the interface and connectors used. Clicking on ‘Print units’ it’s also possible to have a schematic summary of the equation system.

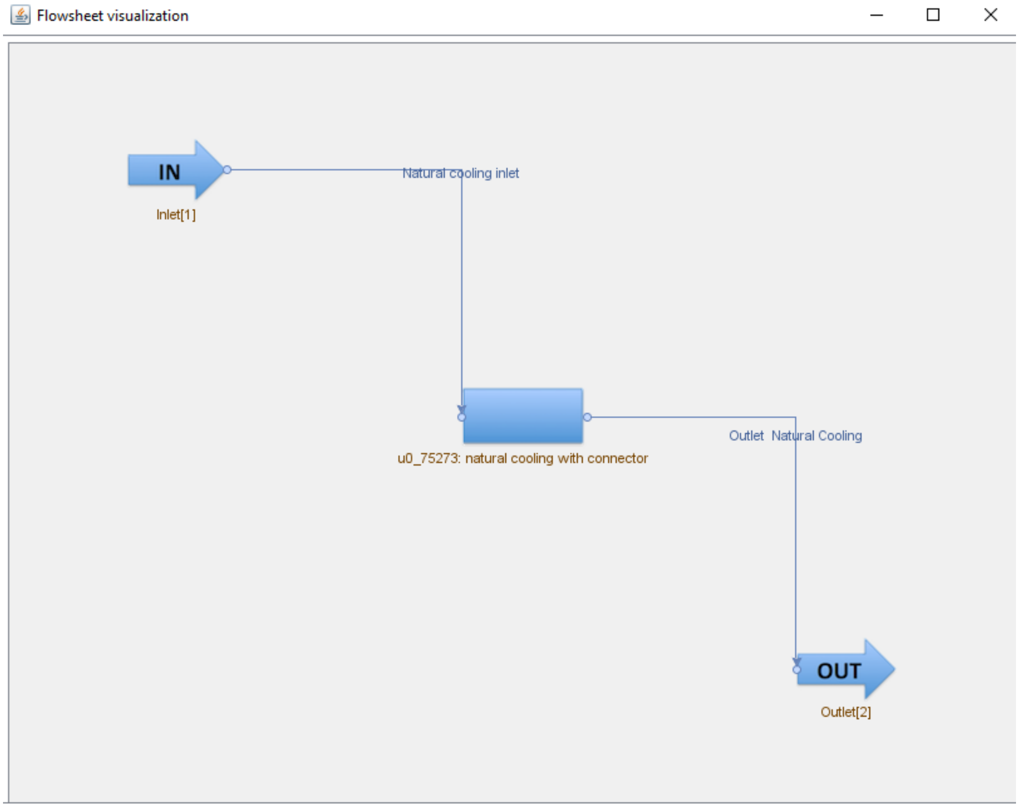
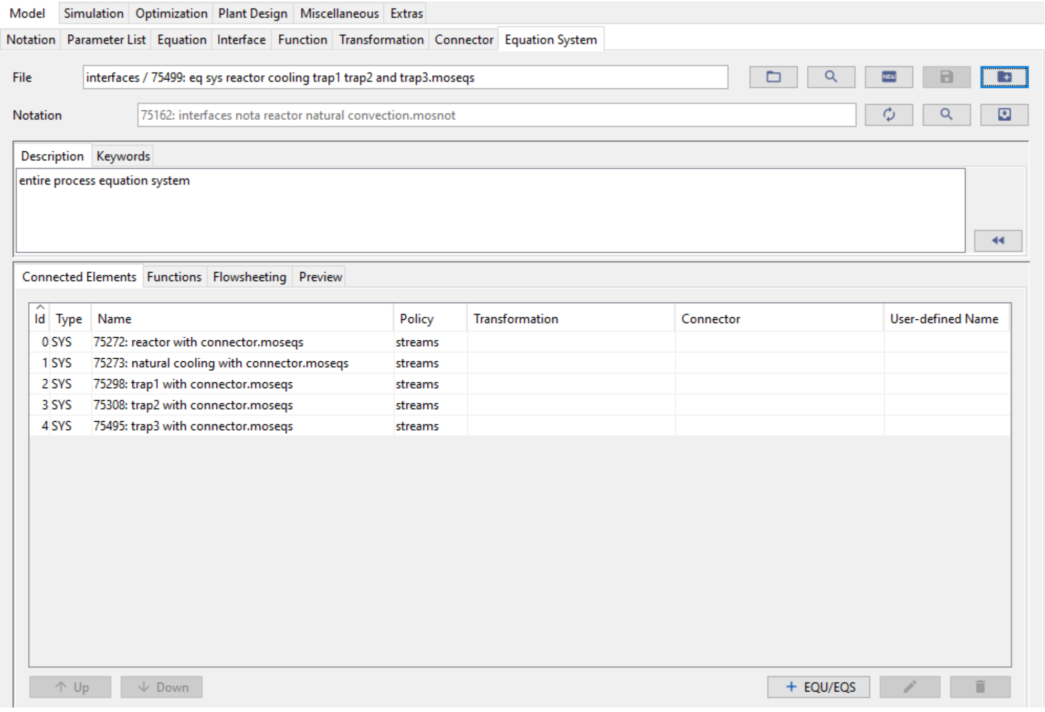


Figure 131 – printed natural cooling section

MOSAIC recognizes that the equation system expects input data and provides an output, but at this point no streams are generated. In order to build the complete process using ports and streams, another passage is required.

6.4.9 Building the complete process in MOSAIC: connecting ports with streams

To create the entire flowsheeting, it's necessary to connect all the ports of the different equation systems with the correct streams. For this purpose, after creating all the equation systems which represent the unit operations complete of their external ports, another passage is needed; this last point is the creation of the complete process equation system, which includes all the unit operations equation systems with their external ports.



File: interfaces / 75499: eq sys reactor cooling trap1 trap2 and trap3.moseqs

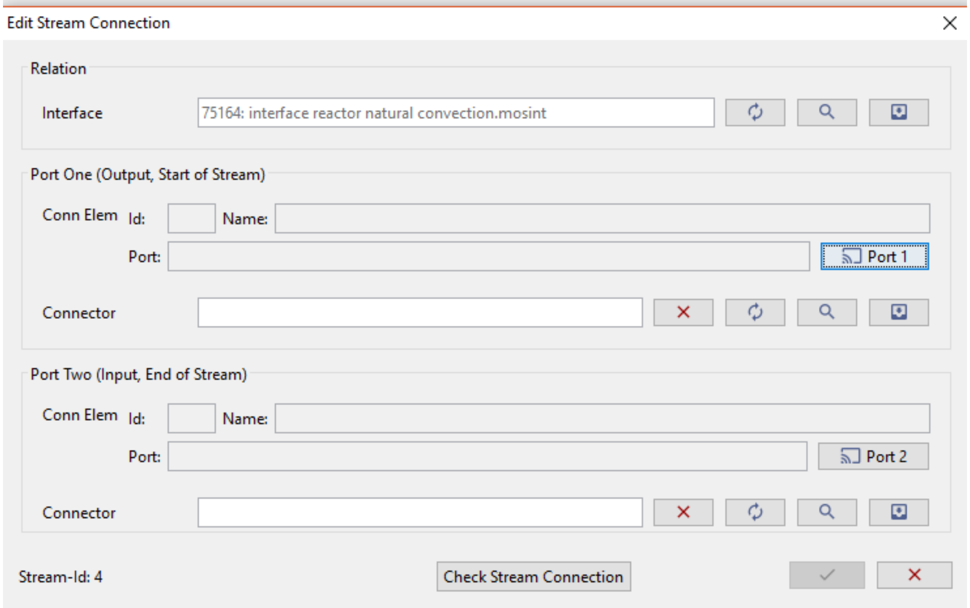
Notation: 75162: interfaces nota reactor natural convection.mosnot

Description: entire process equation system

Id	Type	Name	Policy	Transformation	Connector	User-defined Name
0	SYS	75272: reactor with connector.moseqs	streams			
1	SYS	75273: natural cooling with connector.moseqs	streams			
2	SYS	75298: trap1 with connector.moseqs	streams			
3	SYS	75308: trap2 with connector.moseqs	streams			
4	SYS	75495: trap3 with connector.moseqs	streams			

Figure 132 – entire process equation system

Notice that in that case the unit operations connected in this huge equation system are considered as ‘streams’ and not ‘integrated’. This is because the system recognizes that they contain the external ports useful to perform the connection between them. In order to create the effective linking between unit operations is sufficient to go on ‘Internal streams’ tab, and create the streams which connect the unit operations.

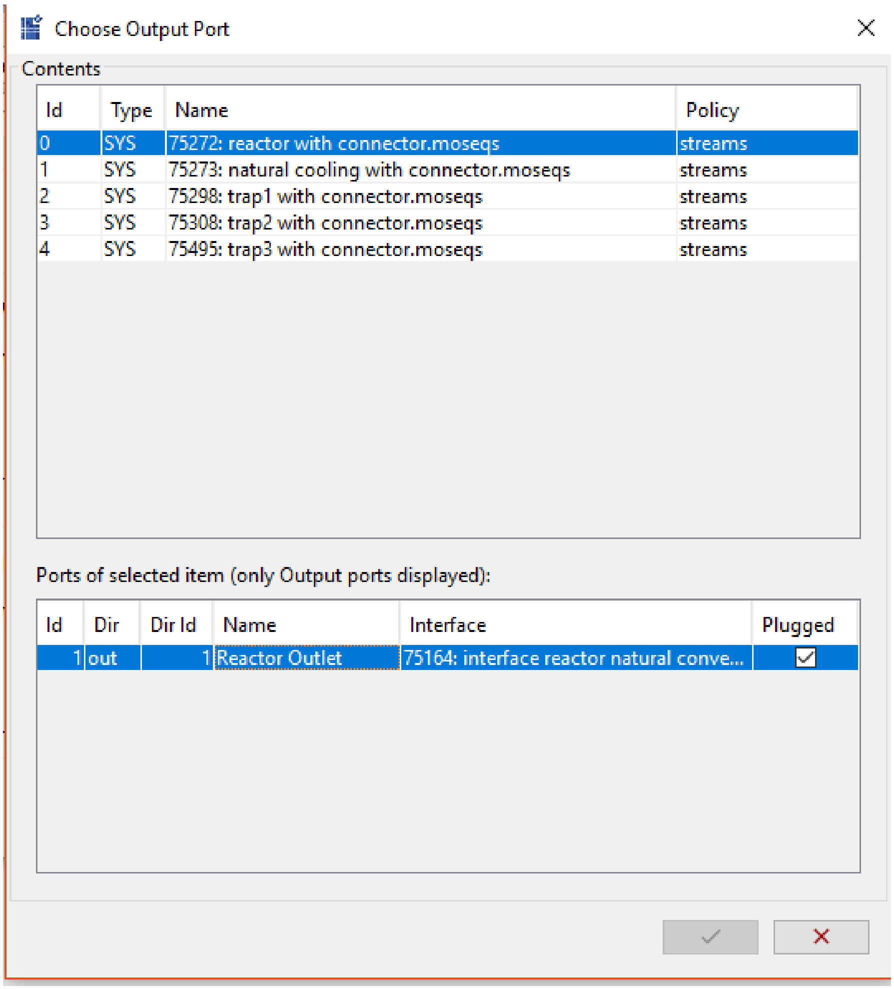


Interface: 75164: interface reactor natural convection.mosint

Stream-Id: 4

Figure 133 – edit internal streams

In ‘edit stream connection’ the interface and the two ports to be connected need to be specified. As evident from the figure, an internal stream can connect only two ports, precisely one output and one input port. If an unit operation need two or more inputs or outputs, it’s necessary to create different ports for each of them. The port1 is dedicated to the output ports. Clicking on the ‘Port 1’ button a screen with all the possible output ports appears:



Contents

Id	Type	Name	Policy
0	SYS	75272: reactor with connector.moseqs	streams
1	SYS	75273: natural cooling with connector.moseqs	streams
2	SYS	75298: trap1 with connector.moseqs	streams
3	SYS	75308: trap2 with connector.moseqs	streams
4	SYS	75495: trap3 with connector.moseqs	streams

Ports of selected item (only Output ports displayed):

Id	Dir	Dir Id	Name	Interface	Plugged
1	out	1	Reactor Outlet	75164: interface reactor natural conve...	☑

Figure 134 – internal streams creation: output port

Choosing the output and the input port following this simple method enables to create a real connection between the selected ports; practically, a stream of information is created between the two unit operations.

At the end of the procedure, the *AG2S™* process flowsheeting is completed.

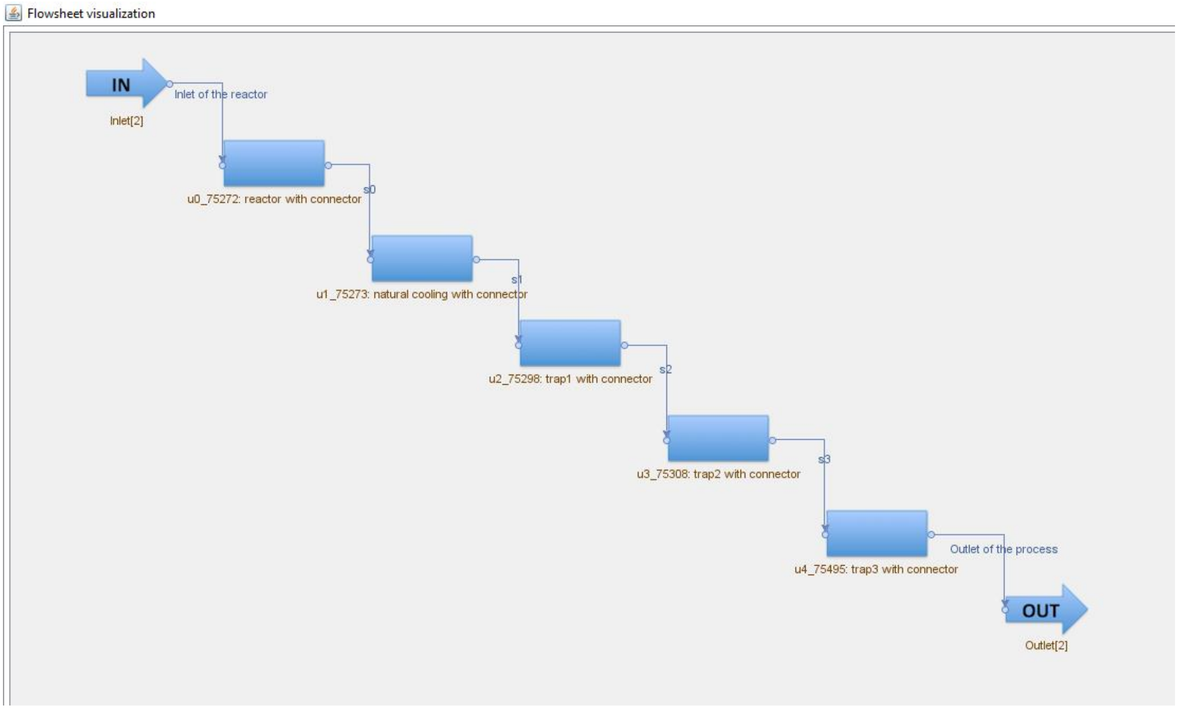
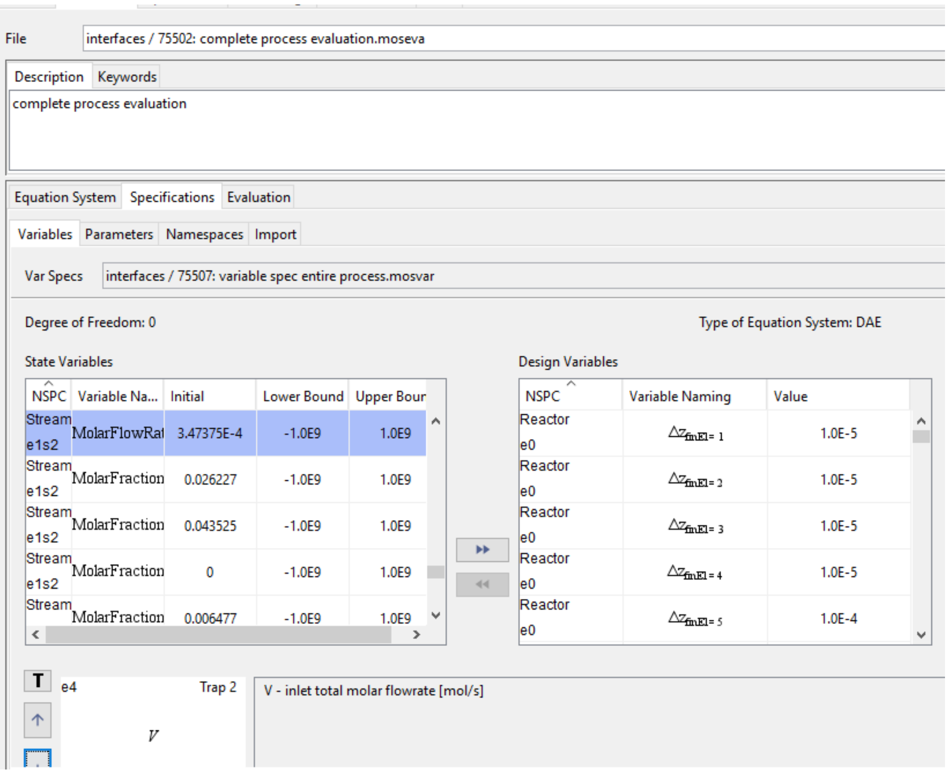


Figure 135 – flowsheeting of the entire process in MOSAIC

The blocks represent the unit operations with their names (e.g. ‘reactor with connector’), while the arrows represent the streams with their progressive names (e.g. ‘s0’, ‘s1’). In this thesis work is chosen to create an input stream before the reactor and an outlet stream for the last trap, in order to evidence the fact that all the designed variables of the reactor are user inputs, while the outputs of the last trap are the process outlet.

The great advantage to create the flowsheeting is that MOSAIC automatically recognize the output and input streams of the unit operations, so to make an example the temperature used in the isothermal reactor is automatically assigned as an input of the natural cooling sections, as the temperature exiting the natural cooling section is considered the temperature at the inlet of the first trap. This scenario could be possible assigning the correct notation to all the variables in the system, which would be totally inconvenient; in fact using the potential of MOSAIC it’s possible to have separate notations for all the unit operations but at the same time connect the variables assigning them the right meaning in every equation system. The re-usability of equations and equation systems is in this way total, and the size of the process studied isn’t a problem, because all the unit operations have a separate notation and equation system at which refer. The only limitation is that MOSAIC doesn’t recognize stream of information which are not present in one of the unit operation connected. This could appear trivial because a passing data is supposed to be used in the unit operation which receives it, but in some cases like in the *AG2S™* process this is not true. In particular, in the 'Natural cooling' equation system, only energy balance is needed to solve the unit operation, but in the following equipment, also the mass balances must be used because of the separation in vapor and liquid streams. In this case, trying to pass the concentrations of the compounds from the reactor to natural cooling and then to the first trap is not allowed by MOSAIC if there aren't equations regarding the LNc_i. This is why in natural cooling section the trivial balance equations $LNc_i^{IN} = LNc_i$; in this way MOSAIC recognizes the presence of the concentration data and can pass them to the first trap. Another possible solution is to pass the concentration information from the reactor directly to the first trap, but as said before MOSAIC uses streams connecting only one output port to one input port, and a single port cannot accept more than one stream. Thus, in this second solution other two ports are needed. The external output port for the concentration stream in the reactor, and the external input port for the concentration stream in the 'trap 1'. The first solution appears more simple and immediate, so in this thesis work the concentrations data are passed from the natural cooling section to the first trap using trivial equations to explicit the LNc_i in the 'Natural cooling' unit operations. In fact it's possible to see in the process flowsheeting that there is only one stream connecting each equation system, and not two for reactor and first trap as necessary for the second solutions.

The entire system solving procedure is the same of the single unit operations method. In this thesis work the solutions for the single units are found before connecting the entire process in order to understand the MOSAIC structure starting from simpler solutions and then increasing the complexity, in a learning procedure due to the complete new approach to the simulation environment, but the entire process could also be solved directly in its complete connected form. The variable specifications are taken from the previous simulations for each unit operation, and the results are the same of the solutions found with separate unit operations. The last thing which can be interesting to show about the complete reactor scheme is how MOSAIC treats the stream variables in 'Variable Specifications'.



File: interfaces / 75502: complete process evaluation.moseva

complete process evaluation

Var Specs: interfaces / 75507: variable spec entire process.mosvar

Degree of Freedom: 0

Type of Equation System: DAE

State Variables

NSPC	Variable Na...	Initial	Lower Bound	Upper Bour
Stream e1s2	MolarFlowRat	3.47375E-4	-1.0E9	1.0E9
Stream e1s2	MolarFraction	0.026227	-1.0E9	1.0E9
Stream e1s2	MolarFraction	0.043525	-1.0E9	1.0E9
Stream e1s2	MolarFraction	0	-1.0E9	1.0E9
Stream	MolarFraction	0.006477	-1.0E9	1.0E9

Design Variables

NSPC	Variable Naming	Value
Reactor e0	$\Delta z_{finEl=1}$	1.0E-5
Reactor e0	$\Delta z_{finEl=2}$	1.0E-5
Reactor e0	$\Delta z_{finEl=3}$	1.0E-5
Reactor e0	$\Delta z_{finEl=4}$	1.0E-5
Reactor e0	$\Delta z_{finEl=5}$	1.0E-4

e4 Trap 2

V

V - inlet total molar flowrate [mol/s]

Figure 136 – complete process variable specification

The stream variables are indicated with their interface notation, but clicking in the tab in the bottom left side, it’s possible to view all the variable namings associated with them. In this exemplifying figure the ‘MolarFlowRate’ of the stream 2 (from ‘Trap 1’ to ‘Trap 2’) is so indicated in the variable specification, but in the bottom side tab it’s possible to see how the ‘MolarFlowRate’ is considered in the equipment ‘Trap 2’, in the origin equipment ‘Trap 1’ and in all the possible notation in which this stream is included. In the figure is shown that in ‘Trap 2’ the ‘MolarFlowRate’ corresponds to the ‘V’ symbol. This is very useful to check the path of the variables within the entire process following their notation and description in the various unit operations.

This was the last interesting characteristic of the MOSAIC to be presented, and the last software description of this work thesis. In the final chapter there will be a resume of the work done, the obtained results and some consideration about the hereafter of the project.

7. RESUME AND FINAL CONSIDERATIONS

This thesis work is concluded; the objectives fixed at the starting point were the 3D model production, the dynamic simulation modeling and the depth analysis of the MOSAIC environment potential.

The 3D model of the process is realized, thanks to the support of Intergraph® Company and after the 2D realization of the laboratory in AutoCAD. The following steps of the 3D modeling are many; at first, to provide an immersive experience to the operator, the entire laboratory room must be modelled. For this purpose, with the support of laser scanning, it's possible to produce a point cloud and export it in Smart 3D.This technique is available for big environments and usually applied for rooms, but hasn't the same quality reported in smaller scales like the fume hood inside, which is in fact realized manually using the presented Smart 3D tasks without point cloud support. After the external laboratory room realizing, the 3D model must be render with adding the correct textures and lighting effects; the procedure must be realized inside the virtualizer, so before starting the rendering of the 3D model, the virtualizer software choice has to be made. This choice will be crucial because the virtualizer receives the inputs from the 3D modeling and the dynamic simulation software (with or without the MOSAIC interfacing) and performs the immersive 3D simulation, so it's connected with all the programs used and has the prior task to produce the final product of the work. The choice of the virtualizing software will be the most important step to do in the future, to ensure the 'speaking' between really different software and the possibility to perform an immersive simulation thanks to good texturing of the objects and real time response of the manipulated variables.

The dynamic simulation of the process is realized with Dynsim™ software. The possible next steps can be to improve the kinetic data of the process using the experimental results which will be produced in the *AG2S*™ laboratory located at 'Politecnico di Milano'. In particular the *AG2S* reactions can be better studied in their real behavior; in fact at now the kinetic data are taken from DSmoke program, which has an accurate theoretical descripting model, but the experimental results could show some unexpected involved reaction or different behavior of the compounds in the desired conditions. Besides, the H_2S removal reactions could be depth and a kinetic model can be built if required using the experimental data. At the end, the way to extract the desired data in real time from the dynamic simulation has to be found, in order to send the required information to directly the virtualizer, or with the intermediation of another ‘generalizing’ environment which could be MOSAIC.

The MOSAIC simulation environment is analyzed and proved to discover its potential in the immersive simulation project. The most important characteristics of MOSAIC are the possibility to write completely customized mathematical models of the process unit operations, the re-usability of the equations and the equations systems thanks to the ability of using different notations in the same model, and eventually the capacity of managing different languages which enables the import and export of data from and to the desired environments. Substantially, MOSAIC can be used as an hub in which importing the dynamic simulation data and exporting in the language specification of the virtualizer, or as the simulator itself, mostly in the case of laboratory process where the dynamic simulation is still not available or when the process requires high customize equations not provided by standard simulators. In both the cases, MOSAIC can provide the desired information to the virtualizing software in the required language specifications, after programming the interface between them. The real problem of the actual version of MOSAIC using in the immersive simulation project is the impossibility to perform a data exchange in real time, which is a fundamental requirement in this case.

In the continuation of the project lots of choices must be made. The possibility to use a hub like MOSAIC to standardize the output of dynamic simulation software is strictly connected to the possible development of the real time exchange of data in the environment. A work in this direction can be started in order to preserve the possibility to use whatever the dynamic simulation programs having the same input to the virtualizer. If the MOSAIC in real time project won’t be chosen or will not reach the expected results, the possible ways are to abandon the possibility to perform the immersive simulation with different dynamic simulator, which would be a big loss in term of flexibility of the package offered to the industrial companies, or to search another way to generalize the output of the dynamic simulators.

Besides, in the future steps the important choice of the virtualizing software must be addressed. This will be a crucial point because of the importance of the virtualizer, which is connected with all the inputs and outputs of the immersive simulations.

When all the problems will be faced, the immersive dynamic simulation can be performed. The final results will guarantee a new possibility for the operator formation, with completely safety working or accidents simulations and also the opportunity to assess the training performances using standard simulations and analyzing the training effectiveness. Besides, in the academic world, the immersive simulation can be a completely new way to form the student for their experimental activities. The idea is to perform immersive simulations for the students, training them for their specific task and acts in the laboratory, assessing their performances with a standard test; after passing the immersive simulation training session, the student could be allowed to work in the real laboratory. In this way the student formation would be made in completely safety conditions, and with the added values of an immersive environment and the assessment possibility on the specific task in laboratory compared to the actual computer presentations method and safety tests on paper. The double valence of the project, in academic and industrial world, is another important value which allows embracing two completely different worlds but with a common needing of safety and performance assessments.

REFERENCES

3D model with Intergraph® Smart 3D

[1] Intergraph® Company, *CIS2 Interface*, 2014
[2] Intergraph® Company, *CloudWorx for Smart3D User's Guide,* 2014
[3] Intergraph® Company, *Smart3D Civil Task*, 2014
[4] Intergraph® Company, *Smart3D Common Application Tutorial Master,* 2014
[5] Intergraph® Company, *Smart3D Equipment Tutorial,* 2014
[6] Intergraph® Company, *Smart3D Piping Tutorial Master,* 2014
[7] Intergraph® Company, *Smart3D Point Cloud*, 2014
[8] Intergraph® Company, *Smart3D Reference 3D,* 2014
[9] Intergraph® Company, *Structural Import and Export using CIS2*, 2014

Kinetic data regression and AG2S™ process description

[10] C. Frau, E. Loria, F. Poggi, F. Manenti, A. Bassani, E. Ranzi, *Sviluppo di un sistema innovativo per il trattamento dei gas di coda provenienti dalla desolforazione del syngas da carbone,* (255) 4-7
[11] Dipartimento di Chimica Industriale e Ing. Chimica 'Giulio Natta', *Dsmoke user manual,* 1998
[12] Guido Buzzi-Ferraris, Flavio Manenti, *BzzMath: Library Overview and Recent Advances in Numerical Methods, Computer-Aided Chemical Engineering,* 30(2), 1312-1316, 2012
[13] Guido Buzzi-Ferraris and Flavio Manenti, *Kinetic Models Analysis*, *CHEMICAL ENGINEERING SCIENCE*, 64(5), 1061-1074, 2009
[14] Guido Buzzi-Ferraris and Flavio Manenti, *Better Formulation of Kinetic Models, COMPUTERS & CHEMICAL ENGINERING,* 34(11), 1904-1906, 2010
[15] Luke Chen, James Huang and Chen-Lu Yang, *Absorption of H_2S in NaOCl Caustic Aqueous Solution*, Department of Water Resources and Environmental Engineering, Tamkang University, Tamsui, Taipei Hsien, Taiwan, Hazardous Substance Management Research Center, New Jersey Institute of Technology, University Heights, Newark, 3(20), 175-181, 2001

Dynsim™ dynamic simulation

[16] Invensys – SimSci-Esscor, *Dynamic Simulation Suite Base Equipment,* 2007
[17] Invensys – SimSci-Esscor, *Dynamic Simulation Suite Flash Tanks,* 2007
[18] Invensys – SimSci-Esscor, *Dynamic Simulation Suite User Guide,* 2007
[19] Invensys – SimSci-Esscor, *Dynamic Simulation Suite Process,* 2007
[20] Invensys – SimSci-Esscor, *Dynamic Simulation Suite Process Equipment,* 2007
[21] Invensys – SimSci-Esscor, *Getting Started With Dynsim*, 2007

MOSAIC description and AG2S™ process modeling in the environment

[22] dbta (Department of Process Dynamics and Operation), *Unit Operations in MOSAIC,* Berlin

[23] dbta (Department of Process Dynamics and Operation), *Flowsheeting with MOSAIC,* Berlin

[24] Erik Esche, David Muller, Gunter Wozny, Lorenz T. Biegler, Jens-Uwe Repke, *Structural Reformulation of Process Models for Improved Numerical Behavior in Process Simulation and Optimization*, $_1$Technische Universitat Berlin, Germany School of Process Science, Chair of Process Dynamics and Operations, $_2$Evonik Technology & Infrastructure GmbH, Process Tehnology & Engineering, CAPE & Automation, Marl, Germany, $_3$Carnegie Mellon University, Dpt. of Chemical Engineering, Pittsburgh, PA, United States of America, 2016

[25] Erik Esche, *Orthogonal Collocation Optimization in Process Science,* DBTA (Department of Process Dynamics and Operation), Berlin, 2016

[26] Stefan Kuntsche, *MOSAIC Learning Trail*, dbta (Department of Process Dynamics and Operation), Berlin, 2010

[27] Stefan Kuntsche, *MOSAIC User Guide*, dbta (Department of Process Dynamics and Operation), Berlin, 2011