SNE SIMULATION NOTES EUROPE

ARGESIM

ASIM — ASIM — ASIM

ASIM - Buchreihen / ASIM Book Series

Fortschritte in der Simulationstechnik / Frontiers in Simulation
Monographs - Conference Proceedings

Simulation und Optimierung in Produktion und Logistik.
L. März, W. Krug, O.Rose, G. Weigert , G. (eds.) , ISBN 978-3-642-14535-3, Springer, 2011

Integrationsaspekte der Simulation: Technik, Organisation und Personal.
Zülch, G., Stock, P. (eds.), ISBN 978-3-86644-558-1, KIT Scientific Publishing, Karlsruhe, 2010

Verifikation und Validierung für die Simulation in Produktion und Logistik, Vorgehensmodelle und Techniken.
M. Rabe, S. Spieckermann, S. Wenzel (eds.); ISBN: 978-3-540-35281-5, Springer, Berlin, 2008

Advances in Simulation for Production and Logistics Applications.
M. Rabe (ed.), , ISBN 978-3-8167-7798-4, Fraunhofer IRB-Verlag, Stuttgart, 2008

Modellierung, Regelung und Simulation in Automotive und Prozessautomation - Proceedings 5. ASIM-Workshop Wismar 2011. –
C. Deatcu, P. Dünow, T. Pawletta, S. Pawletta (eds.), ISBN 978-3-901608-36-0, ASIM/ARGESIM, Wien, 2011.

Reihe Fortschrittsberichte Simulation / Series Advances in Simulation

Ch. Steinbrecher: **Ein Beitrag zur prädiktiven Regelung verbrennungsmotorischer Prozesse**
FBS 18, ASIM/ARGESIM Vienna, 2010; ISBN 978-3-901608-68-1, ARGESIM Report 68

O. Hagendorf: **Simulation-based Parameter and Structure Optimisation od Discrete Event Systems**
FBS 17, ASIM/ARGESIM Vienna, 2010; ISBN 978-3-901608-67-4, ARGESIM Report 67

D. Leitner: **Simulation of Arterial Blood Flow with the Lattice Boltzmann Method**
FBS16, ASIM/ARGESIM Vienna, 2009; ISBN 978-3-901608-66-7, ARGESIM Report 16

Th. Löscher: **Optimisation of Scheduling Problems Based on Timed Petri Nets.**
FBS 15, ASIM/ARGESIM Vienna, 2009; ISBN 978-3-901608-65-0, ARGESIM Report 15

J. Wöckl: **Hybrider Modellbildungszugang für biologische Abwasserreinigungsprozesse.**
FBS 14, ASIM/ARGESIM Vienna, ISBN 3-901608-64-8, 2006, , ARGESIM Report 14,

M. Gyimesi: **Simulation Service Providing als Webservice zur Simulation Diskreter Prozesse.**
FBS 13, ASIM/ARGESIM Vienna, ISBN 3-901-608-63-X, 2006, ARGESIM Report 13

R. Fink: **Untersuchungen zur Parallelverarbeitung mit wissenschaftlich-technischen Berechnungsumgebungen.**
FBS 12, ASIM/ARGESIM Vienna, 2008; ISBN 978-3-901608-62-9, ARGESIM Report 12

H. Ecker: **Suppression of Self-excited Vibrations in Mechanical Systems by Parametric Stiffness Excitation.**
FBS 11, ASIM/ARGESIM Vienna, ISBN 3-901-608-61-3, 2006, ARGESIM Report 11

ARGESIM REPORTS

ARGESIM REPORTS

Editorial

Dear Readers – *This first issue of SNE Volume 23 comes along with an extended submission strategy introduced - individual submissions of scientific papers, and submissions of selected contributions from conferences of EUROSIM societies for post-conference publication (suggested by conference organizer and authors). This issue publishes post-conference publications from SIMS Conference 2012 (SIMS – Scandinavian Simulation Society), from MATHMOD 2012 (Vienna Conference on Mathematical Modelling, Vienna, Austria), and from ASIM SST 2011 (ASIM Symposium Simulation Technique, Zürich – Winterthur, Switzerland).*
Individually submitted contributions complement the very broad variety of modelling and simulation. Discrete approaches deal with agent-based simulation for hospital planning, and with schedule optimization based on Petri nets. In the Modelica framework fluid flow modelling, a Python package for variable-structure model, and loose coupling co-simulation are discussed; additionally, a globally-implicit framework for Physics-based s simulation of coupled thermo-hydro-mechanical problems is presented. Applications conclude this issue: power transfer by non-radiative electromagnetic fields, and mixed friction systems in the micro-scale.
We are glad that for SNE Volume 23 Vlatko Ceric, past president of CROSSIM, is providing his algorithmic art as design for SNE cover page – as for SNE Volume 21. The technique used for the picture series for the covers of SNE Volume 23 is alienation of 'classic' pictures by certain algorithms.

I would like to thank all authors for their contributions, and the organizers of the EUROSIM conferences for co-operation in post-conference publication, and the ARGESIM SNE staff for helping to manage the SNE administration and the improved SNE layout and extended templates for submissions (now also tex), and especially Vlatko Ceric for providing his graphics for SNE.

Felix Breitenecker, SNE Editor-in-Chief, eic@sne-journal.org; felix.breitenecker@tuwien.ac.at

Contents SNE 23(1)

SNE doi: 10.11128/sne.23.1.1016

Reader's Info

Simulation Notes Europe publishes peer reviewed *Technical Notes*, *Short Notes* and *Overview Notes* on developments and trends in modelling and simulation in various areas and in application and theory, with main topics being simulation aspects and interdisciplinarity.

Individual submissions of scientific papers are welcome, as well as post-conference publications of contributions from conferences of EUROSIM societies.
Furthermore SNE documents the ARGESIM Benchmarks on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions (*Benchmark Notes*). Special *Educational Notes* present the use of modelling and simulation in and for education and for e-learning.

SNE is the official membership journal of EUROSIM, the Federation of European Simulation Societies. A News Section in SNE provides information for EUROSIM Simulation Societies and Simulation Groups.

SNE is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With Online SNE the publisher ARGESIM follows the Open Access strategy, allowing download of published contributions for free. Since 2012 Online SNE contributions are identified by a DOI (Digital Object Identifier) assigned to the publisher ARGESIM (DOI prefix 10.11128). Print SNE, high-resolution Online SNE, full SNE Archive, and source codes of the *Benchmark Notes* are available for members of EUROSIM societies.

Author's Info

Authors are invited to submit contributions which have not been published and have not being considered for publication elsewhere to the SNE Editorial Office. Furthermore, SNE invites organizers of EUROSIM conferences to provide post-conference publication for the authors of their conference (with peer review).

SNE distinguishes different types of contributions (*Notes*):

- *Overview Note* – State-of-the-Art report in a specific area, up to 14 pages, only upon invitation
- *Technical Note* – scientific publication on specific topic in modelling and simulation, 6 – 8 (10) pages
- *Education Note* – modelling and simulation in / for education and e-learning; max. 6 pages
- *Short Note* – recent development on specific topic, max. 4 p.
- *Software Note* – specific implementation with scientific analysis, max 4 pages
- *Benchmark Note* – Solution to an ARGEIM Benchmark; basic solution 2 pages, extended and commented solution 4 pages, comparative solutions 4-8 pages

Further info and templates (doc, tex) at SNE's website.

SNE Contact & Info

In-Process Agent Simulation for Early Stages of Hospital Planning

Gabriel Wurzer

Institute of Architectural Sciences, Vienna University of Technology, Treitlstraße 3,
1040 Vienna, Austria; *wurzer@iemar.tuwien.ac.at*

Abstract. In the early stages of hospital planning, work processes are typically modelled in a static manner, using flow-charts or business process modelling notation as means. Diagrams of this kind are easily simulated, however, employed process engines lack possibilities for dealing with dynamic aspects of the process which depend on the building layout (e.g. elevators, behaviour of automatic delivery carts). If one could give planners the opportunity to employ dynamic entities without having to change their usual workflow, one of the benefits would be that they are not being forced to resort to naive assumptions (e.g. 15 seconds per floor) that are still commonplace in today's planning practice.

Introduction

Hospitals, like airports and some types of industrial facilities (e.g. oil platforms), are process-driven buildings: Their design depends foremost on the planned work processes that enable them to operate day and night, 365 days a year. Therefore, the process model of such a building constrains the architectural design, which must evolve in close cooperation between process planners and architects.

Because processes are modelled in highly-formalized manner (e.g. as flow-charts), one might think that the application of simulation lies at hand from the very start of a building project.

However, such *static* process descriptions lack the ability to also include aspects that *depend on the building layout*, such as the transition of persons and material from one space to the other, possibly using *dynamic* entities such as lifts as they move along. Resorting to naive assumptions (e.g. fixed passage times) might be inadequate (again taking the lift as example) and, furthermore, cumbersome to elaborate: In early planning, there are usually *several variants* of the spatial concept rather than only one for later phases.

Our work therefore focuses on overcoming the mentioned problems, by embedding dynamic entities into an otherwise static process model. Broken down into further detail, our contribution consists of:

- A thorough look at simulation needs in the early stages of process-driven building design (see 'Background', Section 2). Such a survey is (surprisingly) novel, as the community has previously targeted hospital simulation problems but not their context within the planning process.
- An extension of *static* process simulation such that *dynamic* entities (acting in a *spatial context*) can be represented. Technically, this is achieved by invoking an agent-based simulation on behalf of the process simulation (see An Early Stage Hybrid Simulation, Section 3).

As side-note and constraint, we want to augment the now-common working style of planners in a non-intrusive manner, i.e. extending rather than reinventing design tools available. The choice of an agent-based simulation *on top of* a process simulation fits exactly this line of reasoning.

1 Related work

Business Process Simulation (BPS) is based on linking a graph-based model to a *Discrete Event Simulation* (DES) that simulates its behaviour over time. There exists a variety of software packages implementing BPS (e.g. Rücker 2008, Dodds 2005), plus some DES packages that provide a 'flow-chart' like approach by means of a server/client based model (Sadowski and Bapat 2001, Nordgren 2001, Nordgren 2003, Concannon et al. 2003).

When it comes to simulation that requires an understanding of the spatial concept (as in the previously mentioned examples of elevators and automated delivery carts), we see that most of the DES seem to focus on late design phases, i.e. phases in which the spatial concept is already fixed and subject to optimization (e.g. via Laguna and Martí 2003). Especially in hospital planning, this might be a problem, since spatial design is subordinate to the process model, and might thus not be evolved as far as the latter when a simulation is performed. We have, therefore, previously presented a coupled pedestrian/process simulation targeted at early design stages, in which the early concept (also called schema) is taken into account (Wurzer 2010). Our efforts for this paper are approaching the problem from a different side: Our goal is to enable planners using a static modelling approach (i.e. flow-charts) to include dynamic entities into their process descriptions, based on a BPS being linked to a Agent-Based Simulation (ABS). We are aware of many approaching being occupied with this specific hybrid approach (e.g. Remondino 2003, Borshchev and Filippov 2003). However, none is focused on the planning context of early-stage hospital design, which is essential when producing an approach that is adapted to working routines now in place.

2 Background

The simulation needs for early-phase hospital design are closely connected to the design process. In the following subsections, we will describe the typical planning tracks and deliverables in early design, before coming to the actual problem areas in which simulation can provide valuable input when being used as a design tool. Because of space constraints, we have omitted a discussion on the influence of different design methods used, and may forward readers interested in this topics to (Jones 1992, Lawson 2005).

2.1 The Early Design Process

There are two design tracks that are important to early-phase hospital design: *Building Organization* and *Functional Planning*.

Building Organization. (see left part in Figure 1) is occupied with the planning of the organization from the side of business administration, i.e. definition of the organizational structures (departments, sub-departments), work processes and responsibilities within these. In essence, the planning work proceeds top-down: Starting with a very coarse outline of business activities required for operation (Figure 1a), a basic formulation of processes can be derived by introducing temporal and causal order (Figure 1b). The notation of these processes depends on preferences of the project team, two usual options are flow-charts or process graphs according to the recently standardized Business Process Modelling Notation (BPMN).

As the project progresses, some activities might need to be further detailed in order to be fully defined. This can be done by using sub-processes, which establishes a hierarchy of activities within activities (Figure 1c). Furthermore, when detailing a process, responsibilities for each activity are also assigned to different collaborating departments within the organization (see Figure 1d). The finished product and goal of the building organization is thus a description of the whole operation of the building from a business side (also called 'process model' of the organization). The process model acts as input and constraint for the functional planning track.

Functional Planning. (see right part in Figure 1) starts with a definition of building functions (i.e. capabilities of a building, see Figure 1e), based on the intended vision (laid out e.g. in the tender document, project description etc.) and process model of the organization. These functions are then correlated in a adjacency matrix (White 1986) by the degree of collaboration, ranging from 'adjacent' for closely collaborating areas to 'dislocated' for areas that do not cooperate or must be separated e.g. because of hygienic considerations (Figure 1f). Adjacent functions are then grouped into spaces (refer to Figure 1g): In the example given, 'operation theatre' and 'recovery' are put into one common space (signified by a dashed border), while 'trauma' stays isolated and gets its own space. The so-found spaces are then arranged in a preliminary floor plan called 'schema' (see Figure 1h), with each space being represented by a rectangle.
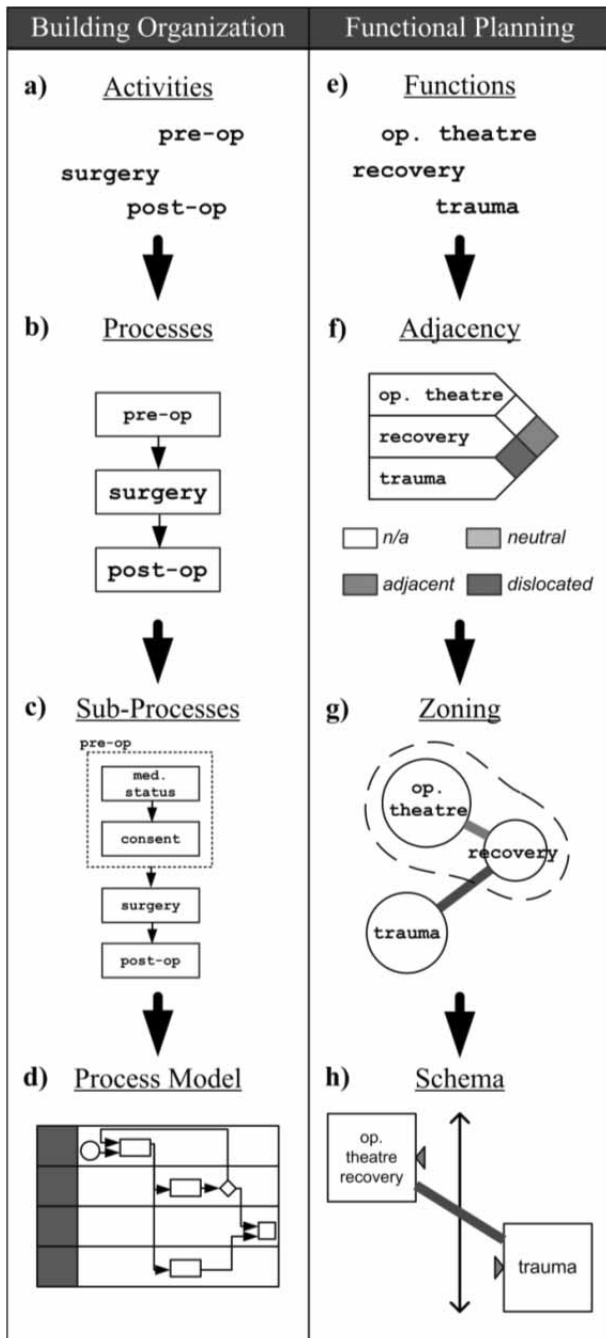
Figure 1: Early-stage planning tracks. (left) Building Organization: (a) Activities formed into (b) processes and (c) sub-processes. Furthermore, assignment of process responsibilities to different departments leads to (d) process model, which acts as input and constraint for (right) Functional Planning: (e) Functions are (f) related via an adjacency matrix, (g) grouped to form (h) spaces within the architectural schema. Circulation is additionally inscribed using arrows.

In this context, the rectangular form of every space is not to be taken literally, since it merely gives the proportion, approximate size and location in relation to other spaces. The concrete form for each space is beyond the work done in early-stage design - it occurs later, in a phase called 'Form Finding'. Apart from the spaces, the schema also contains arrows that give the preliminary circulation system (e.g. corridors) of the building. Aside from the graphical notation, the schema is typically also given as spreadsheet form ('Space Allocation Plan'), which listing spaces (often grouped by function), cardinality (e.g. $2x$) and usable area per space (e.g. $25m^2$), commonly regulated by guidelines and planning handbooks such as (Neufert 2000). Depending on project structure, the Space Allocation Plan may be produced during Functional Planning or be given before the actual planning work starts, as an input (e.g. when extending a building).

It is noteworthy that the activities of *Building Organization* and *Functional Planning* are not sequential but inherently parallel: As process model and the spatial concept are detailed and evolve side-by-side, the planning team has to ensure consistency of both models (Wurzer, 2010). Furthermore, the spatial concept might fork of a variety of alternative designs, which must then later be reduced or merged by design decisions (i.e. documented argumentation within the planning team leading to a set of choices, see Kunz and Rittel 1970, Rittel and Webber 1984). For process-based buildings in the early planning stages, these design decisions are typically based on:

Urban Context. The relation between the planned building as a whole (i.e. arrangement of spaces and circulation) with its surrounding environment and existing infrastructure (White, 1983). For example, traffic patterns resulting from local public transport and motorized individual traffic have to be taken as a constraint. Visibility of landmarks has to be preserved by (and likely used in) the proposed design.

Adjacency. Short paths between collaborating units (defined by the adjacency matrix), considering the adjacency matrix (White 1986), process model and expected volume of building users requiring service. Vice versa, a separation of spaces for reasons of privacy (e.g. secure areas vs. public spaces, inpatient vs. outpatient areas) and for sustaining building operation (typically by service corridors, allowing for repairs and delivery 'behind the scenes').

Separation of traffic. Different routing according to type of traffic (Lohfert 2005), e.g. separation of staff from visitors and patients, low-priority from high-priority traffic (e.g. emergencies), building users with appointment from the ones without, soiled from clean material, etc.

Location, size and proportion. Placement of spaces is not isolated from considerations of the building as a whole; for example, certain areas of work favour natural lighting (e.g. patient rooms, energy considerations for the whole building), while others can do without. Proportion and size of individual spaces determines the opportunity of future adaptations (e.g. change in equipment), while at the same time being subject to optimization (minimal area needed per function, compactness).

Orientation and wayfinding. Depending on intended user spectrum, orientation can play a vital role for the whole building project. The transition of building users from one space to the other must be considered both in terms of the process as well as existing previous knowledge about the building layout. Spaces serving processes used by temporary building users must be easy to reach (i.e. no signage required) and memorize (e.g. using a main corridor connecting all departments) . A clear readability of space can also help in fire safety and evacuation planning, conducted in later phases (Illera et al. 2008).

Extensibility and adaptability. Both extensibility and accessibility of a building is given by the configuration of spaces and circulation (Neufert 2000) - the first one deals with openness to the outer environment, the second with (usually multi-functional) hub spaces that serve as distribution points for pedestrian traffic, often located at prominent positions within the building. The ability to adapt the spatial concept to future requirements of the process also requires an evaluation from a multi-functional view (e.g. interdisciplinary use of a space, shared workspaces, etc.).

Adequacy of planned concept. The adequacy of both spatial concept and process model is an overall judgement of the building's design under consideration of the planning task. Argumentation focuses on whether the design satisfies the vision and financial context stated by the client. In the planning team, the discussion is centred around the types of functions present and sizes of their respective spaces as well as structure of the processes and needed resources.

2.2 Early-stage simulation needs for hospitals

Given the mentioned design decisions in early planning phases of process-based buildings, simulation can contribute tools for assessing a variety of parameters which can then be weighted according to the planning objectives (i.e. multi-objective analysis). Statements produced in this manner are necessarily qualitative, since spatial concept and process model are in a preliminary stage:

Visibility, accessibility and wayfinding. The analysis of these parameters may be done statically (for the whole building, its arrangement of spaces and circulation network) or dynamically (by simulating individual processes). In the first case, reachability analysis of the circulation network can be conducted for both interior and exterior space by using the methods provided by Space Syntax (Hillier and Hanson 1984, Hillier 1996), which can also compute the visibility from each point in the building (e.g. for hiding areas for supply and disposal). Viewshed computations, usually found in Geographical Information Systems (GIS), can be used for the same purpose. Wayfinding, on the other hand, requires a dynamic simulation of individuals following their processes (e.g. using ABM). In this connection, algorithms from pedestrian dynamics may be used to simulate the physical movement under the influence of congestion (e.g. using Helbing and Molnár 1995).

Space placement and dimensioning. Previously defined adjacency relations can be verified by simulating the planned processes by means of BPS, ABS or system dynamics (SD). The volumes of traffic between the spaces, distances travelled over the circulation and simulated times taken must correlate with the relationships given in the adjacency matrix. Furthermore, the dimensioning of spaces can be checked by considering the volume of persons present in each time step: In the simplest case, the occupation is related directly to the presence of persons in a space (e.g. in entrance areas). Moreover, presence in a space may relate to waiting for a shared function (e.g. examination), which can be modelled as server with a specified number of resources (e.g. 2 doctors) and one or more queues. By correlating the observed size of queues with the space requirements for waiting areas (distinguished e.g. for sitting and lying patients), it is possible to attain a hint at minimal areas required.

Norms and regulations further contribute to these space requirements, which could be checked using approaches from automated building code checking (Eastman 2009), albeit in a simplified form. A further opportunity for comparing the placement of spaces is that of building physics simulation: Some workplaces might require daylight; others must be protected from it. Preliminary environmental simulation (light, shadowing, wind), can also hint at energy demands which are elaborated in later phases.

Movement, circulation and traffic. Different options of route choice can be simulated by either assigning waypoints between subsequent activities of the process explicitly (e.g. delivery of goods in zigzag shape, one floor at a time) or by interpreting the circulative network as graph on which shortest paths are computed. As a matter of fact, spatial arrangements can be judged by the time it takes to move across the circulation (which is also depending on the processes in place). A separation of traffic can further be achieved by attributing the circulation with allowed types of traffic (e.g. patients, visitors, staff), and taking these into account during either automatic or manual route planning. A further attribution of the circulation arrows has to be performed for distinguishing between horizontal traffic (taking place in the same level) and vertical circulation (lifts and stairs), among which movement models and speeds might differ.

Usage. Functions give the purpose (or intent) of spaces, processes model their planned usage over time. By coupling activities of the process to the underlying functions, a static check for unused function or activities that have no reference to a function (i.e. the underlying spatial concept) can be made (Wurzer, 2010). Furthermore, temporal usage of functions obtained via process simulation can be used to compare the prominence of the spaces containing them and help think of possibilities for multi-functional use: Areas that are used only part-time (e.g. lunch room) may be conveniently used for other functions (e.g. meeting room) during the rest of the day.

3 An Early-stage Hybrid Simulation

The problems and design decisions presented under Section 2.1 demand, above all, a solution with reference to the spatial concept (Section 2.2), in which dynamic aspects such as wayfinding, movement etc. play a role.

This is a dilemma, since the process modelling approach commonly used is inherently abstract, i.e. non-spatial. In order to make dynamic spatial simulation within a static BPS possible, we have extended a commonly used process modelling platform (Microsoft Visio) for which a multitude of process simulation add-ons exists (e.g. ProModel Process Simulator, Simul8 and Arena Integration).

Our implementation follows the seemingly usual approach for adding simulation capabilities to Visio, by writing (1.) a script in Visual Basic for Applications (VBA) that (2.) exports the flowchart in a custom format and (3.) calls upon an external program to do the actual process simulation. In order to avoid licensing issues when making our work available, we have opted to use the open-source BPS implementation provided by (Baeyens 2005) instead of a commercial engine. Furthermore, (4.) we have employed NetLogo (Wilensky 1999) as ABS platform, which is also available under open-source license terms.

3.1 Calling an agent simulation from inside a process

Our central point of intervention lies in the introduction of a new type of activity in the process diagram that was coined as *agent node* (see Figure 2a). This new node type acts as an injection point for dynamic behaviour inside the static process.

An agent node is a proxy for an agent simulation that is executed on behalf of the BPS. Upon entering an agent node, the process execution is passivated and control is passed to an ABS, which performs a spatial simulation as required by the planning process; in the simplest case (which we have implemented) the route of an agent from one space to another is computed. In order to be able to call the ABS, an agent node holds parameters specific to its model (e.g. 'from' and 'to', Figure 2b). Generally, there will be several agent-based models running in parallel to the BPS, each one covering a different aspect of dynamic behaviour needed (refer to Section 1.2). As a matter of fact, every model will have its own type of agent node holding required parameters. Under the hood, the different agent models are implemented as servers listening on distinct ports (see Figure 2c showing the mapping of agent models to ports), communicating bidirectional with the BPS.
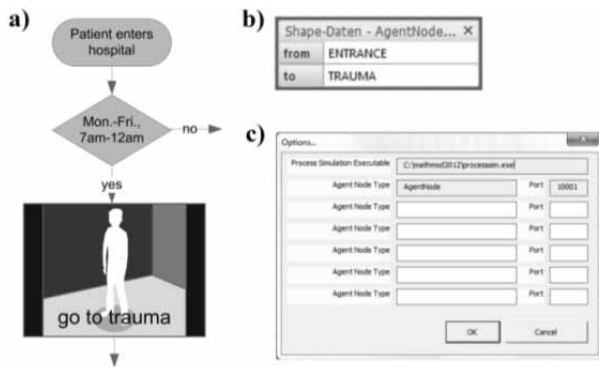
Figure 2: (a) An agent node being embedded into a static process. (b) Properties of an agent node. (c) Options dialog showing the connection settings, which let Visio communicate with process simulations and agent models.

3.2 Synchronization between process and agent simulation

As previously mentioned, a process execution that reaches an agent node is passivated for as long as the called agent-based model runs. However, because BPS and ABS typically differ in their time bases (future event list versus simulation in seconds), some synchronization construct is required.

In simplest approach (which we have undertaken), BPS and ABS both progress in seconds. For each time step, (1.) the BPS executes processes scheduled at that instance. Then, it (2.) issues the command to begin simulating that time-step to all ABS, which (3.) perform computation and (4.) report the list of agents that have completed their task in this time-step back to the BPS. (5.) The BPS waits for all ABS to finish computation, then (6.) reactivates process executions that have waited for the completion of an agent model.

A more elegant but also more complex way for synchronization would be to only simulate in time steps when needed (i.e. when ABS are active), and employ discrete scheduling based on the future event list in all other cases. We clearly see this as a future work item and optimization task for our approach.

3.3 Agent-based simulation

ABS simulations are occupied with simulating the dynamic aspects of the process depending on the building layout. The implementation we used computes the passage from one named space into another, taking into account the circulation and existence of lifts, which have their own behaviour (see Figure 3). On startup of NetLogo, the program opens its receiving communication port. Upon (1.) receiving the request to simulate a time-step, which may also contain the command to insert a new agent into the simulation, the simulation (2.) advances all agents and (3.) compiles a list of those that have finished their task, which it (4.) returns to process simulation. The wayfinding algorithm used is based on previous work, as given in (Wurzer, 2010). Furthermore, the agent-based models contains multiple variants of the spatial concept, which can be used to compare different spatial options. In our simulation, the spatial concept is chosen when starting the process simulation. The underlying data is hardcoded for the time being, in a full-blown implementation, this could be loaded from a Computer Aided Design (CAD) file or any other form of database containing the schema. Another way for obtaining the schema, which we find is preferable, is to use the agent model directly as a design tool (i.e. develop the schema directly inside the model) - thus embedding simulation fully into the design process.
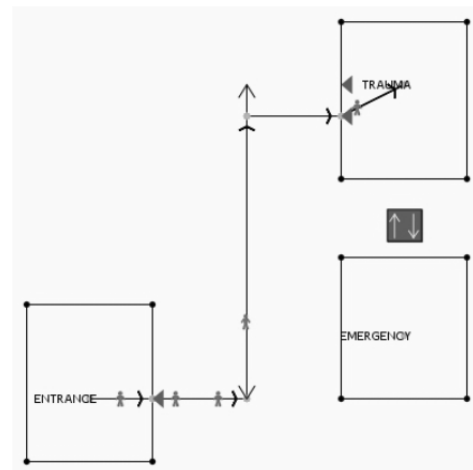
Figure 3: Agent-based NetLogo simulation for computing the passage between named spaces, taking the circulation (middle line with arrows) into account. Also simulates an elevator to the ward, which has its own behaviour.

3.4 Connectivity between implementations

An overview of the connectivity between the different applications written is shown in Figure 4: Visio uses the embedded VBA scripting engine to first output a file containing the process definition, then opens the process simulation app as well as all agent simulations. From then on, the communication proceeds solely between the process simulation and the agent simulations, bidirectional via sockets.
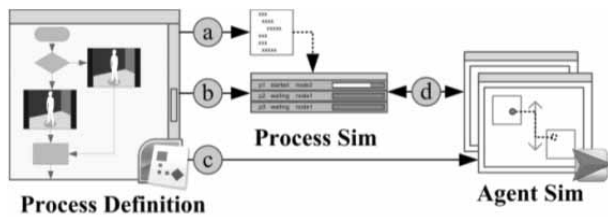
Figure 4: Implementation. (left) Processes drawn in Visio are (a) written to a file, which is fed into a (b) process simulation that can execute them. Visio furthermore (c) opens all agent simulations for the process model, which are (d) then invoked by the process simulation.

We have presented a simulation approach by which dynamic entities can be embedded into an otherwise static process. Our efforts are targeted at early stages of hospital planning, where simulation has the potential to become a design tool for qualitative decisions among a multitude of design variants. Our implementation (which is being made available under the website www.iemar.tuwien.ac.at/processviz/early-stage-sim), consists of an extension to Visio flowcharts in the form of a custom 'agent node' that invokes a process simulation communicating with an agent-based simulation. As outcome, we are able compare different spatial configurations with regard to how long it takes to cross different areas of a hospital being planned. However, our approach offers many more opportunities for being used in today's planning practice, as outlined under Section 1.2. Therefore, we see the technical part of our contribution only as minor outcome of the paper, the real impact lies in the possibility to shift simulation from late stages of design (where possibilities of change are limited), to early stages, where design decisions are of fundamental significance.

References

[1] Baeyens T. JBPM: Graph Oriented Programming [Internet]. [cited 2005] Available from: docs.jboss.org/jbpm/v3/userguide.

[2] Borshchev A, Filippov A. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. In: *Proceedings of the 22nd International Conference of the System Dynamics Society;* 2003.

[3] Concannon KH, Hunter KI, Tremble JM. (2003). Simul8-Planner simulation-based planning and scheduling. In: *Proceedings of the 2003 Winter Simulation Conference*; 2003. P. 1488-1493

[4] De Vries B, Jessurun AJ, and Dijkstra J. Conformance Checking by Capturing and Simulating Human Behaviour in the Built Environment. In: *Proceedings of Sixth Design and Decision Support Systems in Architecture and Urban Planning*; 2002. P. 378-391

[5] Dodds S, Designing improved healthcare processes using discrete event simulation, *British Journal of Healthcare Computing and Information Management*. 2005; 22(5): 14-16.

[6] Eastman C, Lee J, Jeong Y, Lee J. Automatic rule-based checking of building designs. In: *Automation in Construction*. 2009; 18: 1011-1033.

[7] Helbing D, Molnár P. Social force model for pedestrian dynamics. *Physical Review E 51*. 1995; 5: 4282–4286.

[8] Hillier B, Hanson J. *The Social Logic of Space*. Cambridge University Press; 1984.

[9] Hillier B. *Space is the Machine: A Configurational Theory of Architecture*. Cambridge University Press; 1996.

[10] Illera C, Fink M, Kath K, Waldau N, Rosic A, Wurzer G. NO_PANIC. Escape and Panic in Buildings. In: Proceedings of the fourth international conference on Pedestrian and Evacuation Dynamics; 2008. P. 733-742

[11] Jones JC, *Design Methods*. Wiley & Sons, London; 1992.

[12] Kunz W, Rittel H. Issues as elements of information systems, In: *Working Paper 131, Center for Urban and Regional Development,* University of California, Berkley; 1970.

[13] Laguna M, Martí R. The OptQuest Callable Library. *Operations Research/Computer Science Interfaces Series.* 2003; 18: 193-218

[14] Lawson B. *How Designers Think, The Design Process Demystified.* Architectural Press, Oxford; 2005.

[15] Lohfert P. *Methodik der Krankenhausplanung.* Lohfert & Lohfert AS, Copenhagen; 2005.

[16] Neufert E. *Architect's data.* Blackwell Science, Malden; 2000.

[17] Nordgren WB. Taylor Enterprise Dynamics. In: *Proceedings of the 2001 Winter Simulation Conference;* 2001. P. 269-271

[18] Nordgren WB. Flexible simulation (Flexsim) software: Flexsim simulation environment. In: *Proceedings of the 2003 Winter Simulation Conference;* 2003. P. 197-200

[19] Remondino M. Agent Based Process Simulation and Metaphors Based Approach. In: *Proceedings of the 4th Workshop on Agent-Based Simulation*; 2003. P. 93-97

[20] Rittel H, Webber M. Dilemmas in a General Theory of Planning. *Policy Sciences.* 1984; 4: 155–169

[21] Rücker B. *Building an open source Business Process Simulation tool with JBoss jBPM* [Master Thesis]. Stuttgart, University of Applied Science; 2008.

[22] Sadowski D, Bapat V. The Arena Product Family: Enterprise modeling solutions. In: *Proceedings of the 2001 Winter Simulation Conference*; 2001. P. 159-166

[23] White E. *Site Analysis: Diagramming information for architectural design.* Architectural Media, Tucson; 1983.

[24] White E. *Space adjacency analysis: Diagramming information for architectural design.* Architectural Media, Tucson; 1986.

[25] Wilensky U. *Netlogo* [Internet].c1999. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, Illinois. Available from: http://ccl.northwestern.edu/netlogo

[26] Wurzer G. Schematic Systems - Constraining Functions Through Processes (and Vice Versa). *International Journal of Architectural Computing.* 2010; 8(2): 201-217

Schedule Optimization based on Coloured Petri Nets and Local Search

Gašper Mušič

University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, 1000 Ljubljana, Slovenia;
gasper.music@fe.uni-lj.si

Abstract. The contribution deals with simulation-optimization of schedules that are modelled by simple Coloured Petri Nets (CPNs). CPN modelling of standard classes of scheduling problems is addressed and compact CPN representations of scheduling problems are proposed. It is shown how a combination of CPN representations with predefined transition sequence conflict resolution strategy can be used to optimize schedules by standard local search optimization algorithms. Possible neighbourhood construction procedures for various problem classes are proposed with the emphasis on solutions feasibility.

Introduction

Petri nets compose a general modelling formalism suitable for description of systems with highly parallel and cooperating activities. Among others, they are increasingly used for modelling and analysis in the field of manufacturing systems. An important advantage of Petri nets is that production systems' specific properties, such as conflicts, deadlocks, limited buffer sizes, and finite resource constraints can be easily represented within a single formal model (Tuncel and Bayhan, 2007).

The simplicity of model building, the possibility of realistic problem formulation as well as the ability of capturing functional, temporal and resource constraints within a single formalism motivated the investigation of Petri net based optimization of manufacturing planning and scheduling problems. In our previous work a simulation based optimization approach applying Petri nets was intensively studied, as well as other, more classical approaches, such as dispatching rules and reachability tree based heuristic search (Gradišar and Mušič, 2007; Löscher et al., 2007; Mušič, 2008).

In particular, the investigations focused on combinations of Petri net modelling approach and local search methods (Löscher et al., 2007; Mušič, 2009).

This paper focuses on Coloured Petri Net (CPN) models of various classes of scheduling problems. General modelling guidelines for standard problems, such as open shop, flow shop and job shop problems are proposed. The paper also explores the possibilities of use of the CPN models in conjunction with state-of-the-art local search algorithms provided a special type of parameterized conflict resolution strategy and neighbouring solution generation procedure are adopted. Constrained permutations on transition firing vectors are used to generate neighbouring schedule solutions that are always feasible, which improves the effectiveness of CPN based exploration of solutions compared to previous works.

1 CPN Representations of Scheduling Problems

Literature on deterministic scheduling classifies the manufacturing scheduling problems according to the machine environment structure, processing characteristics and constraints, and objectives (Pinedo, 2008). Standard machine environment structures lead to standard scheduling problems, e.g., open shop, flow shop and job shop problems, which are commonly studied. All three problem classes address a problem of sequencing n jobs (tasks) through a set of m machines (resources) where every job has to be processed once on every machine and every such job operation requires a specified processing time. The problems differ in restrictions on the job routings.
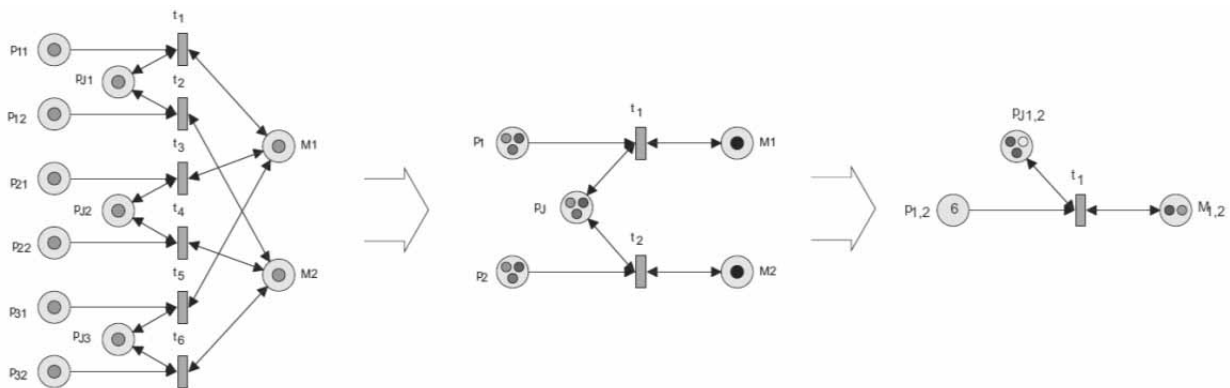
Figure 1: PN and CPN models of an open shop scheduling problem.
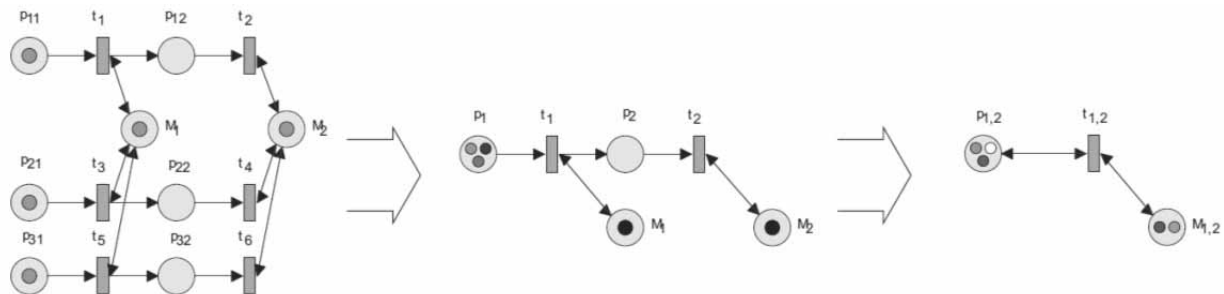
Figure 2: PN and CPN models of a flow shop scheduling problem.

Petri nets can be used to effectively model all three standard problem classes. In particular, Coloured Petri nets enable to develop compact models with a chosen level of abstraction, which are functionally equivalent to basic Place/Transition model. To illustrate this, Figure 1 shows a simple open-shop problem of three jobs and two machines.

The open shop problem structure states that the processing order of job operations is arbitrary, i.e., there are no restrictions with regard to the routing of each job through machine environment (Pinedo, 2008). Therefore the operations are represented as sequentially independent PN transitions that share a set of machines on one hand, and a set of jobs on the other. The shared resources are linked to transitions by self loops, which are for simplicity of drawing represented by double arrowed arcs.

Timed Petri nets are used where a token is declared unavailable for a specified time period after every transition firing. This way only a single operation of a job can be processed at a time and also only a single operation can be performed on a machine at a time. A single operation occurrence in a sequence is enforced by additional places which restrict the transition firing.

The model is shown in three levels of abstraction, with common Place/Transition Petri net (PN) structure used in the leftmost model and CPNs used in the other two models.

In the Place/Transition PN model the machines are denoted M_i and jobs by p_{J_i}. Start of every operation is modelled by transition firing. Since the transitions related to the same machine are in conflict, only one of the corresponding transitions can fire at a time. Conflict resolution strategy determines a schedule of machine assignments. Additional places p_{ij} correspond to j-th operation of job i and restrict the operations to be triggered only once in a sequence.

In the first CPN model (central model in Figure 1) all jobs are represented by the same set of places and transitions. The distinction among different jobs is achieved by introduction of token colours and transition occurrence colours. Every token colour corresponds to a job, whereas every occurrence colour corresponds to triggering of operation within the job. This way various operation durations can be represented by a single transition. In the next level of abstraction also the machines are folded into one place.

Consequently also the transitions representing operation triggering and places enforcing single operation triggering are folded together into only one place-transition pair. All the interrelations among different place tokens and transition firings are now encoded into complex colour based enablement conditions of different transition occurrence colours. E.g., for the model of Figure 1, marking of the final CPN is determined by 6 token colours in $p_{1,2}$, 3 token colours in $p_{J1,2}$ and 2 token colours in $M_{1,2}$, while t_1 has 6 occurrence colours. Every occurrence of t_1 defines a set of colour dependent weights of the incoming and outgoing arcs. The colour dependent weights related to a single arc can be represented by a vector, and the weights of this arc related to all transition occurrences form a matrix. In the CPN models the problem structure is therefore folded within arc weights that become matrices.

In general, an open shop problem with n jobs and m machines requires n token colours in job place, $n \cdot m$ token colours in operation place and m token colours in machine place. The number of transition occurrence colours equals the number of distinct operations, i.e. $n \cdot m$. The self loop arcs between job place and the transition are weighted by $n \times nm$ matrices, the arc between operation place and the transition by $nm \times nm$ matrix and self loop arcs between machine place and the transition are weighted by $m \times nm$ matrices.

Note that while the shown model is complete, it may be advantageous to add additional n colours to job place in order to represent the resulting schedule more easily. The final marking of the model in Figure 1 is obtained when all the operation tokens are consumed while all job and machine tokens are back in place and available. The schedule has to be reconstructed by observing unavailable tokens in operation and machine places. This is easier if tokens are deposited in operation place also during the last operation, although these are not employed to enable subsequent transition firing.

Similarly, any flow shop problem can be represented by a PN structure similar to Figure 2. Again the model is shown in three forms with increasing level of abstraction. Compared to open shop problems, the flow shop problem states that all jobs have the same routing, i.e. every job is processed on all machines in exactly the same machine order.

In Place/Transition PN model this is achieved by enforcing a directed token flow through a set of places representing operations of a single job. Places are linked in a chain with operation triggering transitions in between a pair of places. Transitions representing the same machine operation within all jobs are linked to a common place that represents a machine resource. No additional places are needed to ensure proper operation triggering as this is now achieved by sequential ordering of job operation places and transitions.

As before, the CPN model can be obtained by merging together the job operation chains, which are folded into a single sequence of places and transitions as shown in the center of Figure 2. The final CPN model is obtained by folding together the machine places, which consequently folds together also operation places and transitions. The required token flow is achieved by introduction of token colours, transition occurrence colours and arc weights in matrix form as described above. For a flow shop problem with n jobs and m machines, $n \cdot m$ token colours are required in job operation place and m token colours in machine place. The number of transition occurrence colours equals the number of distinct operations, i.e. $n \cdot m$. The arcs between job place and the transition are weighted by $nm \times nm$ matrices and arcs between machine place and the transition are weighted by $m \times nm$ matrices. The final marking of the model is obtained when all the job operation tokens are consumed while all machine tokens are back in place and available. Similarly as above, additional n colours can be added to job place to reconstruct the resulting schedule more easily.

In general, the model of Figure 2 represents a problem, where jobs are not processed in the same order on every machine. E.g., a job can pass another job while waiting for processing on a particular machine. Often the same job processing order is required on all machines, which forms a *permutation* flow shop problem. Within the optimization approach described in the next section, this can be easily achieved if the machine sequences are mutually dependent.

A job shop scheduling problem is modelled very much in the same way as the flow shop. The difference is only in machine assignments, which are not the same for every job. Instead, each jobs follows its own route through the set of machines, while it still visits every machine only once.

The different machine assignments reflect in diverse connections of operation transitions to machine places in P/T PN model. The final CPN model has the same graphical representation as in Figure 2; the machine assignments are hidden within arc weights in matrix form.

1.1 Simple Coloured Petri nets

Building on the two above examples a formal definition of Coloured Petri nets is given as follows. Note that the definition is different from Jensen (1997) in the sense that it does not alow for transition guards. Instead it closely follows one of the representations used in Basile et al. (2007) with an important difference: a different interpretation of transition delays is used, which is closer to that of Jensen (1997).

A $CPN = (\mathcal{N}, M_0)$ is a Coloured Petri net system, where:

$\mathcal{N} = (P, T, Pre, Post, Cl, Co)$ is a Coloured Petri net structure:

- $P = \{p_1, p_2, \ldots, p_k\}, k > 0$ is a finite set of places.
  $T = \{t_1, t_2, \ldots, t_l\}, l > 0$ is a finite set of transitions (with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$).

- Cl is a set of colours.

- $Co : P \cup T \to Cl$ is a colour function defining place marking colours and transition occurrence colours. $\forall p \in P, Co(p) = \{a_{p,1}, a_{p,2}, \ldots, a_{p,u_p}\} \subseteq Cl$ is the set of up possible colours of tokens in p, and $\forall t \in T, Co(t) = \{b_{t,1}, b_{t,2}, \ldots, b_{t,v_t}\} \subseteq Cl$ is the set of v_t possible occurrence colours of t.

- $Pre(p, t) : Co(t) \to Co(p)_{MS}$ is an element of the pre-incidence function and is a mapping from the set of occurrence colours of t to a multiset over the set of colours of $p, \forall p \in P, \forall t \in T$. It can be represented by a matrix whose generic element $Pre(p, t)(i, j)$ is equal to the weight of the arc from p w.r.t colour $a_{p,i}$ to t w.r.t colour $b_{t,j}$. When there is no arc with respect to the given pair of nodes and colours, the element is 0.

- $Post(p, t) : Co(t) \to Co(p)_{MS}$ is an element of the post-incidence function, which defines weights of arcs from transitions to places with respect to colours.

$M(p) : Co(p) \to \mathbb{N}$ is the marking of place $p \in P$ and defines the number of tokens of a specified colour in the place for each possible token colour in p. Place marking can be represented as a multiset $M(p) \in Co(p)_{MS}$ and the net marking M can be represented as a $k \times 1$ vector of multisets $M(p)$. M_0 is the initial marking of a Coloured Petri net.

1.2 Timed models

As described in Bowden (2000), there are three basic ways of representing time in Petri nets: firing durations (FD), holding durations (HD) and enabling durations (ED). When using FD principle the transition firing has a duration. In contrast, when using HD principle, a firing has no duration but a created token is considered unavailable for the time assigned to transition that created the token, which has the same effect. With ED principle, the firing of the transitions has no duration while the time delays are represented by forcing transitions that are enabled to stay so for a specified period of time before they can fire. The ED concept is more general than HD. Furthermore, in Lakos and Petrucci (2007) an even more general concept is used, which assigns delays to individual arcs, either inputs or outputs of a transition.

When modelling several performance optimization problems, e.g. scheduling problems, such a general framework is not needed. It is natural to use HD when modelling most scheduling processes as operations are considered non-preemptive. HD principle is also used in the timed version of CPNs defined by Jensen (1997), where the unavailability of the tokens is defined implicitly through the corresponding time stamps. While CPNs allow the assignment of delays both to transition and to output arcs, we further simplify this by allowing time delay inscriptions to transitions only. This is sufficient for the type of examples investigated here, and can be generalized if necessary.

To include time attributes of the marking tokens, every coloured token is accompanied with a time stamp where time stamps are elements of a time set TS, which is defined as a set of numeric values. In many software implementations the time values are integer, i.e. $TS = \mathbb{N}$, but will be here admitted to take any positive real value including 0, i.e. $TS = \mathbb{R}_0^+$. Timed markings are represented as collections of time stamps and are multisets over $TS : TS_{MS}$. By using HD principle the formal representation of a Coloured Timed Petri net is defined as follows.

$CTPN = (\mathcal{N}, M_0)$ is a Coloured Timed Petri net system, where:

- $\mathcal{N} = (P, T, Pre, Post, Cl, Co, f)$ is a Coloured Time Petri net structure with $(P, T, Pre, Post, Cl, Co)$ as defined above.
- $f : Co(t) \rightarrow TS$ is the time function that assigns a non-negative deterministic time delay to every occurrence colour of transition $t \in T$.
- $M(p) : Co(p) \rightarrow TS_{MS}$ is the timed marking, M_0 is the initial marking of a timed Petri net.

As mentioned before, in the CPN models the problem structure is folded in arc weights represented by matrices. This is convenient for automatic generation of the models as matrices can be easily constructed algorithmically and the graphical representation of CPNs is rather simple compared to graphical layout of P/T Petri nets with high number of places, transitions and arcs. CPNs can be therefore effectively applied as a modelling framework in conjunction with automatic model generation and optimization based on production management data.

2 Neighbourhood Solution Generation Strategy

In our previous work (Löscher et al., 2007; Mušič et al., 2008) different ways of solution space exploration were studied. Extensive testing of the reachability tree search based approaches has been performed and also several experiments with local search techniques have been made.

2.1 Prescribed transition firing sequences

In Löscher et al. (2007) the approach is presented, which extends the Petri net representation by sequences and priorities. When using sequences, disjoint groups of transitions are selected and mapped to sequence vectors. A firing list is defined by ordering transitions within the group. During the model evolution a set of sequence counters is maintained and all transitions belonging to sequences are disabled except of transitions corresponding to the current state of the sequence counters. After firing such a transition the corresponding sequence counter is incremented. This way the transition firing sequence can be parameterized.

If the model represents a scheduling problem, the sequence obtained by a sequence-supervised simulation run of the Petri net model from the prescribed initial to the prescribed final state is a possible solution to the problem, i.e. it represents a feasible schedule.

To illustrate this, consider a simple job shop example of four jobs and four machines. Operation durations are shown in Table 1 and resource requirements in Table 2.

The problem is modelled by a CPN similar to Figure 2. The model can then be simulated by applying SPT (Shortest Processing Time first) rule (Haupt, 1989) as a default conflict resolution mechanism. The resulting sequence represents a possible schedule, shown in Figure 3.

Operation/Job	J_1	J_2	J_3	J_4
o_1	54	9	38	95
o_2	34	15	19	34
o_3	61	80	28	7
o_4	2	79	87	29

Table 1: Operation durations for a simple job shop problem.

Operation/Job	J_1	J_2	J_3	J_4
o_1	3	4	1	1
o_2	1	1	2	3
o_3	4	2	3	2
o_4	2	3	4	4

Table 2: Machine requirements for a simple job shop problem.
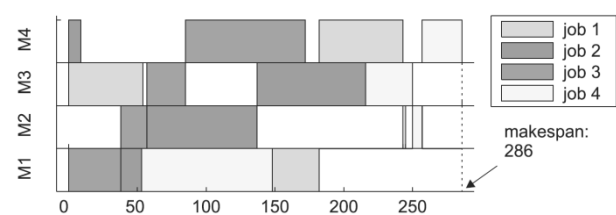

Figure 3: A possible solution of the given job-shop problem.

The same schedule can be obtained by fixing the sequential order of transitions in conflicts related to shared resources in the system. E.g., in the above example the shared resources are machines $M1$ to $M4$ modelled by four token colours in place M_{1-4}. Related sets of transition occurrence colours are:

$$
\begin{aligned}
S_{M1} &= \{t_{1,c3}, t_{1,c4}, t_{2,c1}, t_{2,c2}\} \\
S_{M2} &= \{t_{2,c3}, t_{3,c2}, t_{3,c4}, t_{4,c1}\} \\
S_{M3} &= \{t_{1,c1}, t_{2,c4}, t_{3,c3}, t_{4,c2}\} \\
S_{M4} &= \{t_{1,c2}, t_{3,c1}, t_{4,c3}, t_{4,c4}\}
\end{aligned} \tag{1}
$$

where $t_{i,cj}$ denotes the occurrence colour of t_{1-4}; c_j corresponds to job J_j and i corresponds to the operation number within the job.

If these sets are mapped to four independent sequences, and a set of index vectors $V = \{V_1, V_2, V_3, V_4\}$ is adjoined, where V_i is a corresponding permutation of integer values $i, 1 \leq i \leq 4$:

$$
\begin{aligned}
V_1 &= \{1, 4, 2, 3\} \\
V_2 &= \{1, 2, 4, 3\} \\
V_3 &= \{1, 3, 4, 2\} \\
V_4 &= \{1, 3, 2, 4\}
\end{aligned} \tag{2}
$$

a supervised simulation run, which forces the prescribed sequential order of conflicting transitions, results in the same schedule as above.

The sequence-supervised simulation is implemented by a simple modification of the regular CTPN simulation algorithm. After the enabled transitions are determined in each simulation step, the compliance of the set of enabled transitions to the state of the sequence counters is checked. Transitions that take part in defined sequences but are not pointed to by a counter are disabled.

The exploration of the solution space and the related search for the optimal schedule can then be driven by modifications of sequence index vectors. The problem of this approach is that by perturbing sequence index vectors the resulting transition firing sequence may easily become infeasible, which results in a deadlock during simulation (a sequence imposed deadlock). Such an infeasible solution can be ignored and a new perturbation can be tried instead. While this works for many problems, in some cases the number of feasible sequences is rather low and such an algorithm can easily be trapped in an almost isolated point in the solution space.

2.2 Generation of neighbourhood solutions

In the Operation Research (OR) literature several neighborhood generation operators have been proposed (Blazewicz et al., 1996; Jain et al., 2000).

The question is how to link these operators and related effective schedule optimization algorithms with Coloured Petri net representation of scheduling problems. As mentioned above the Petri net scheduling methods have advantages in unified representation of different aspect of underlying manufacturing process in a well defined framework. Unfortunately, the related optimization methods are not as effective as some methods developed in the OR field. The link of two research areas could be helpful in bridging the gap between highly effective algorithms developed for solving academic scheduling benchmarks and complex real-life examples where even the development of a formal model can be difficult (Gradišar and Mušič, 2007).

A possible way of such a link is the establishment of a correspondence of a critical path and the sequence index vectors. In a given schedule the critical path CP is the path between the starting and finishing time composed of consequent operations with no time gaps:

$$
CP = \{O_i : \rho_i = \rho_{i-1} + \tau_{i-1}, i = 2 \dots n\} \tag{3}
$$

where O_i are operations composing the path, ρ_i is the release (starting) time of operation O_i, and τ_i is the duration of O_i.

The operations O_i on the path are critical operations. Critical operations do not have to belong to the same machine (resource) but they are linked by starting/ending times.

Critical path can be decomposed in a number of blocks. A block is the longest sequence of adjacent critical operations that occupy the same resource.

The length of the path equals the sum of durations of critical operations and defines the makespan C_{max}:

$$
C_{max} = \sum_{O_i \in CP} \tau_i \tag{4}
$$

Figure 4 shows a redrawn gantt chart from Figure 3 with indication of the critical path (grey) and the sequence of critical operations. The shown critical path consists of 5 blocks.

Critical operations in Figure 4 are denoted by transition labels that trigger the start of a critical operation when fired. A transition that triggers a critical operation will be called a critical transition.

The scheduling literature describes several neighborhoods based on manipulations (moves) of critical operations (Blazewicz et al., 1996). One of the classical neighborhoods is obtained by moves that reverse the processing order of an adjacent pair of critical operations belonging to the same block (van Laarhoven et al., 1992). Other neighbourhoods further restrict the number of possible moves on the critical path, e.g. (Nowicki and Smutnicki, 1996).

Clearly every critical transition participates in one of the conflicts related to shared resources. If these transitions are linked to predefined firing sequences parameterized by index vectors, a move operator corresponds to a permutation of an index vector.
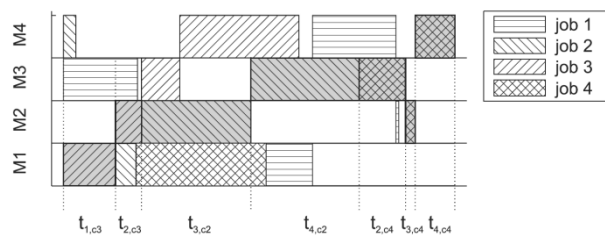
Figure 4: A critical path within a schedule and critical transitions.
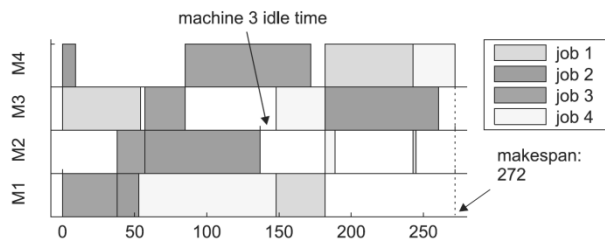
Figure 5: An optimized solution of the given job-shop problem.

For example, in the schedule shown in Figure 4 a move can be chosen, which swaps the two operations in the third block on the critical path. This corresponds to the swap of transitions $t_{4,c2}$ and $t_{2,c4}$ in the sequence S_{M3}, which is implemented by the exchange of the third and the fourth element within V_3 index vector:

$$move(V_3): \{1, 3, 4, 2\} \mapsto \{1, 3, 2, 4\} \qquad (4)$$

A new schedule obtained by simulation with modified V_3 results in a shorter makespan. Similar swap of the last two operations in the sequence S_{M2} leads to the optimal schedule (with the shortest makespan) shown in Figure 5.

When the move is limited to the swap of a pair of the adjacent operations in a block on the critical path this narrows down the set of allowed permutations. The most important feature of such a narrowed set of permutation on the index vector is that every permutation from this set will result in a feasible firing sequence, i.e. a feasible schedule. Therefore no deadlock solutions can be generated, which are often encountered when unrestricted permutations on the index vectors are used.

The described approach is well suited to optimization of job shop problems. With minor modifications it can however also be used for optimization of other scheduling problem classes. The problem class specific constraints can be enforced as additional permutation constraints. E.g., permutation flow shop restrictions can be enforced by keeping all the sequence vectors in synchronization. In contrast, open shop problems have no restrictions on job routings. Fixing transition firing sequences for each individual machine therefore does not resolve all the conflicts in the model. The remaining conflicts can be solved on the fly during simulation in a random manner in order to cover as much as possible wide set of problem solutions.

It is also important to note that such a neighbourhood function is comparable to exploring the reachability tree in an event driven manner. It is possible that certain feasible firing sequence imposes one or more intervals of idle time between transitions, i.e., some transitions are enabled but can not fire due to sequence restrictions. This is different from the exploration in a time driven manner when a transition has to be fired whenever at least one transition is enabled. The difference is important in cases when the optimal solution can be missed unless some idle time is included in the schedule as shown in Piera and Mušič (2011). In other words, the schedules generated in the proposed way belong to the class of semi-active schedules (Pinedo, 2008).

E.g., in the example of Figure 5 a small fraction of idle time is introduced before processing job 4 on machine 3. Job 2 has already been ready for processing on the same machine, which resulted in non-optimal schedule obtained by SPT rule (Figure 3). The chosen change in the firing sequence enforced the firing of $t_{2,c4}$ before $t_{4,c2}$ although $t_{4,c2}$ has been marking-enabled first. This way a schedule with shorter makespan was obtained.

3 Conclusion

The described neighbourhood generation procedure was coded in Matlab and used in combination with a simple Simulated annealing (SA) search algorithm. Comparison of the minimum makespan for some standard open shop, flow shop and job shop problems calculated by the proposed algorithm and some other standard algorithms has been performed. The proposed algorithm is able to improve the initial SPT solutions with a moderate effort, although it is not able to reach optimum for complex benchmarks.

Therefore a prototype implementation of tabu search algorithm has also been tested. The obtained results are comparable to the SA based search. It is expected that the tests with other neighbourhood operators would further improve the obtained results, which is one of the tasks for the future work.

Acknowledgments

References

[1] Basile F, Carbone C, Chiacchio P. Simulation and analysis of discrete-event control systems based on Petri nets using PNetLab. *Control Engineering Practice*. 2007; 15: 241–259.

[2] Blazewicz J, Domschke W, Pesch E. The job shop scheduling problem: Conventional and new solution techniques. *European Journal of Operational Research*. 1996; 93: 1–33.

[3] Bowden, FDJ. A brief survey and synthesis of the roles of time in petri nets. *Mathematical & Computer Modelling*. 2000; 31: 55–68.

[4] Gradišar D, Mušič G. Production-process modelling based on production-management data: a Petri-net approach. *International Journal of Computer Integrated Manufacturing*. 2007; 20(8): 794–810.

[5] Haupt, R. (1989). A survey of priority rule-based scheduling. *OR Spectrum*. 1989; 11(1): 3–16.

[6] Jain A, Rangaswamy B, Meeran S. New and 'stronger' job-shop neighborhoods: A focus on the method of nowicki and smutnicki(1996). *Journal of Heuristics*. 2000; 6(4): 457–480.

[7] Jensen K. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*. 2 edition. Berlin: Springer-Verlag; 1997. 236 p.

[8] Lakos C, Petrucci L. Modular state space exploration for timed Petri nets. *International Journal on Software Tools for Technology Transfer*. 2007; 9: 393–411.

[9] Löscher T, Mušič G, Breitenecker F. (2007). Optimisation of scheduling problems based on timed petri nets. In *Proceedings EUROSIM 2007*; 2007; volume II. Ljubljana.

[10] Mušič, G. (2008). Timed Petri net simulation and related scheduling methods: a brief comparison. In *The 20th European Modeling & Simulation Symposium*; 2008. Campora S. Giovanni (Amantea, CS). P. 380–385

[11] Mušič, G. (2009). Petri net base scheduling approach combining dispatching rules and local search. In *21th European Modeling & Simulation Symposium*, volume 2. Puerto de La Cruz, Tenerife. P. 27–32

[12] Mušič G, Löscher T, Breitenecker F. Simulation based scheduling applying Petri nets with sequences and priorities. In *UKSIM 10th International Conference on Computer Modelling and Simulation*; 2008. Cambridge. P. 455-460

[13] Nowicki E, Smutnicki C. A fast taboo search algorithm for the job shop problem. *Management Science*. 1996; 42(6): 797–813.

[14] Piera MA, Mušič G. Coloured Petri net scheduling models: Timed state space exploration shortages. *Math.Comput.Simul.*. 2001; 82: 428–441.

[15] Pinedo. ML. *Scheduling: Theory, Algorithms, and Systems*. 3rd edition. Springer Publishing Company, Incorporated; 2008.

[16] Tuncel G, Bayhan GM. Applications of Petri nets in production scheduling: a review. *International Journal of Advanced Manufacturing Technology*. 2007; 34: 762–773.

[17] van Laarhoven P, Aarts E, Lenstra J. (1992). Job shop scheduling by simulated annealing. *Operations Research*. 1992; 40: 113–125.

A Python Package for Simulating Variable-Structure Models with Dymola

Alexandra Mehlhase

Department of Software Engineering and Theoretical Computer Science, TU Berlin, Ernst-Reuter-Platz 7, 10587 Berlin, Germany; *a.mehlhase@tu-berlin.de*

Abstract. It becomes increasingly important to create more accurate models that can be simulated fast. To accomplish this we need models which can change their set of equations during runtime. These models are called variable-structure models. These models enable a user to specify a model with more than one mode and change between these modes during runtime. This can make a simulation faster and in some cases even more accurate. In this paper we present a Python package that enables the user to specify such models in an easy and intuitive manner. The introduced package provides means to use existing Dymola models as modes and simulate the variable-structure model with the Dymola simulation engine. Different examples are presented which were simulated with the new package and the advantages of variable-structure modeling with the Python package is discussed. Furthermore, requirements a model needs to fulfill to be used in a variable-structure model are explained.

Introduction

To study the behavior of a technical system early in the design phase (simulation-) models are often used. Such a model consists of variables and equations which specify the behavior of the model over time. The models are usually described through differential-algebraic equations (DAE). The models are then simulated with a numerical solver. The results of such a simulation can be used to analyze the behavior of a technical system without having to build the real system. Through the ever growing complexity of the real technical systems, the complexity of the models also needs to rise.

This leads to the problem that the simulation of a model might become too slow or that some systems cannot be modeled at all. We regard variable-structure models as a solution for this problem. These models consist of different modes between which they can switch. This means that a model can run through different 'modes', where each mode effectively is a model with its own set of equations which describe the physical behavior of this particular mode. When switching from one mode to the next one, the new mode needs to be initialized through the end values of the old mode.

For instance variable-structure models enable the user to model systems that change their behavior. An example for such a model is an airplane which is first a rolling vehicle then a cross between a rolling vehicle and a flying object and then becomes a flying object. In this paper we call models that change their equations in order to model different behavior 'variable-behavior models'.

Another example for variable-structure models is a model which changes its level of detail. With such a model the simulation can be as detailed as necessary and as easy as possible throughout the simulation. For instance if a model can reach critical regions and in these a complex model is needed but otherwise an easier model is sufficient, it would be feasible to use the less detailed model and only switch to the more complex model if needed. We call such models 'variable-detail models' and will show in the evaluation section that such models can save simulation time without significant accuracy losses. Of course there are models that are hard to place in only one category but usually the goals for these two are different. For variable-detail models the goal is to save simulation time or enhance the accuracy through the mode switch. For the variable-behavior models the simulation time is not the main goal but that a system can be simulated at all through the variable-structure approach.

A variable-structure model has always exactly one current mode and switches from mode to mode through defined transition. These transitions hold the information on how the simulation data of the old mode should be used to initialize the new model.

Common simulation tools like Simulink from [1] and Dymola from [2] do not support the change of the variables and the set of equations during simulation. There do exist simulation environments which enable a user to simulate variable-structure models. A brief overview of these environments and approaches are given here.

MOSILAB is a simulation environment based on Modelica which can be used to model and simulate variable-structure models. This tool enhances the Modelica language and uses a Statecharts view to specify the mode switches, see [3]. There is no index reduction implemented in the tool and thus only index-0 models are allowed.

The language SOL was developed by [4] and is an experimental Modelica like language. This language supports modeling variable-structure models, interpreting them and simulating them. An advantage of SOL is that when a mode switch occurs and the causalisation of the model changes only the necessary causalisations are done. This makes the mode switch quite elegant. SOL is an experimental language and thus far not available for large models, hopefully the results will sometime lead to a modeling language which supports variable-structure models. For now a user cannot work with SOL and reuse models from other tools all models would have to be specified in the new language. Another possibility for variable-structure models is Hydra which is described in [5]. Hydra is a language under development which supports variable-structure modeling. It is based on functional programming languages and is therefore not as easy to learn for modelers.

All these possibilities have great ideas and do different things better than our approach but to model and simulation variable-structure models the user has to learn a new language and remodel existing models. With the approach we present a common modeling tool can be used and existing models can be reused. Python as a free language is used for the package, so the package is accessible to anyone interested.

[6] describe an algorithm which transforms a variable-structure model to a normal model by reformulating the modes into one mode. This approach does work but makes the resulting model quite large and

will therefore extend the simulation time. This does not seem to be feasible for variable-detail models because simulation time will most likely not be saved. In Simulink enable blocks can be used to model different modes, but the definition of the transitions becomes rather complicated and the blocks that are disabled still take up simulation time, see [7] for more information. In Dymola a mode switch is possible, as long as the variables do not change and the causality of the equations does not change. If either needs to change, both equation systems need to be implemented in the model and through if-statements the switching needs to be defined. [8] describe a possibility for variable-structure models with if-statements in Modelica. Here the equations are reformulated, so depending on the mode the model is in, a multiplication with zero or one takes place. The equations therefore change during simulation. This approach does work but for large models with many mode switches it will become complicated. The equation system is also rather large and might slow down the simulation compared to a real mode switch. In this paper we will present an easy to use approach to define variable-structure models in Python and use the simulation tool Dymola for the simulation. This package enables the user to reuse existing Dymola models and still work with variable-structure models .

Section 2 introduces the Python package with its design and usability. In Section 3 different examples of variable-structure models which were modeled with the new Python package are presented. The last section provides the conclusion and future work.

1 A New Python Package

This section first gives an overview of how a variable-structure model can be modeled with its different modes and transitions. The design of the package and how the package can be used is afterwards explained.

1.1 A variable-structure model

As was already said in the introduction a variable-structure model is a model which consists of an arbitrary number of modes between which the model can switch. A model can switch from one mode to another mode via one transition. To model such a behavior an object-oriented approach seems feasible. Figure 1 shows how a variable-structure model can be modeled through objects. The ModelObject consists of different modes and each mode can have transitions.
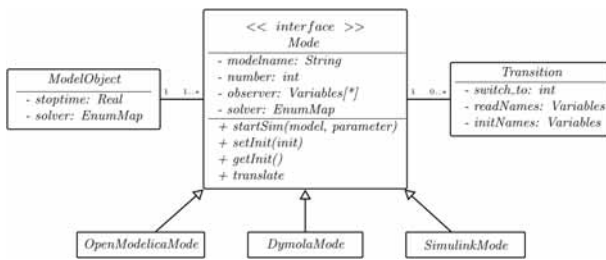
Figure 1: Class structure of a variable-structure model for different modeling environments.

This ModelObject holds all the necessary information about the variable-structure model. It defines the global stop time of the model, a default solver and the modes of the variable-structure model . To be able to integrate different simulation environments into the package an interface which is tool independent was defined. The most important attributes of a mode are:

- a unique mode number to identify the mode
- a model name with path to the original model
- observer variables which will be stored in a data matrix
- a specific solver which overwrites the default solver

Necessary methods in a mode class:

- start simulation (startSim) which starts the simulation in a specific simulation environment
- set initial values (setInit) which sets the initial values in the tool specific init file
- read end values (readEnd) which reads the necessary end values of the tool specific result data file
- translate model (translate) which compiles the model, if necessary

When a specific simulation environment needs to be added new class which implements the mode-interfaces has to be created. The other parts of the model do not have to be changes, as long as the given interface is not changed. For now Dymola is implemented and integrating OpenModelica and Simulink is planned. Each mode can have an arbitrary number of transitions which lead to the next modes. A transition is another class which defines the mode switch and is independent of the used simulation environment. In each transition an attribute exists which holds the identification number of the next mode. Furthermore the information on how the data of the old mode is used to initialize the new mode is stored in the transition.

1.2 Design of the package

In the previous section the object-oriented design of a variable-structure model was explained. This design is used in the Python package which makes it possible to integrate different simulation environments. For a modeler who is used to modeling in simulation environments it might be difficult to define this modeling structure in Python. Therefore the package provides a template to specify the variable-structure model. The user does not need any programming knowledge to be able to use this template.

The package uses the user given information to generate the necessary ModelObject. The basic idea of our approach is to use common simulation environment to simulate a variable-structure model and therefore use their capabilities. It is not the idea to create a new language as was done by [4] with SOL or by [5] with Hydra or with the tool [9, 10]. Our idea is to create a new modeling layer which can manage different modeling environments and which handles the switch from one mode to another during a simulation run. To accomplish this each mode of a variable-structure model needs to be an independent model which consists of variables and equations. Each of these modes has a stop condition which stops the simulation of this particular mode and defines the next mode. Figure 2 shows a schematic view of a breaking pendulum variable-structure model with two modes. One mode which is the normal pendulum and one mode which is a falling mass.
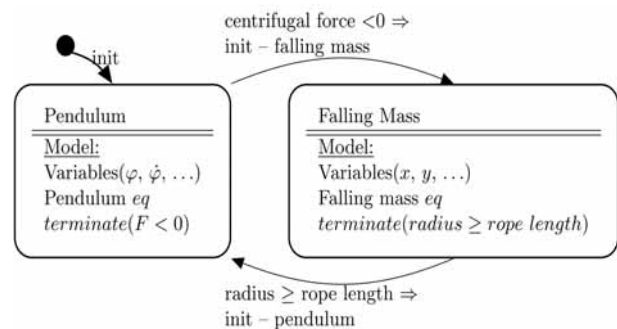

Figure 2: Schematic view of a breaking pendulum variable-behavior model.

To get from one mode to the next a Python script is used, this approach was already presented in [7] where it was tested with different scripting languages. The workflow of the script which handles the simulation of

the variable-structure model is shown in Figure 3. In our package the modeled ModelObject is used as input for the variable-structure simulation method.
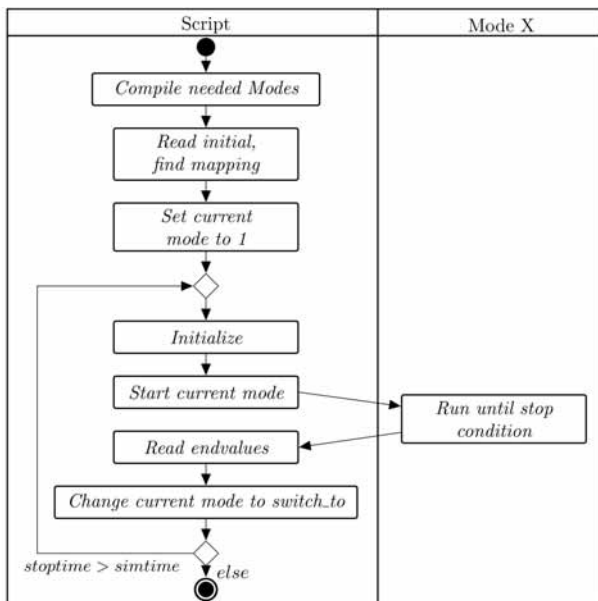

Figure 3: Program flow of the `switch.py` method.

At the beginning of this method the modes are compiled, which results in having an executable called 'dymosim.exe' and an initialization file called 'dsin.txt'. The dymosim.exe can be used to start the simulation of the model. Each created init file is loaded and results in having an initialisation matrix. An identical matrix can be loaded after a simulation of a Dymola model whereas this matrix then holds the end data of the simulation. To make the mode switches faster, the mapping for setting initial values in the init matrix through the end values of the end matrix, is saved in each transition. For now only a one to one mapping is allowed (`oldMode.x = newMode.z` is allowed, `oldMode.z = f(oldMode.x,oldMode.y,...)` is planned) which makes the mapping simple. This of course means that all values necessary to fully initialize the new mode need to be available in the old mode. In case a value is not available the modeler can set values himself in the package. The initialization routine of the specific simulation environment of the mode is then used to initialize the whole model. The modeler is therefore responsible to specify the initialization within the package to get a stable and continuous solution. The method then enters a while-loop which only stops when the user defined stop time is reached. The loop

starts with the user defined start mode with given initial values. When the simulation stops because of a specific stop condition the transition to the next mode is known through the ModelObject. The end values of the simulation (dsfinal.txt) are then loaded which gives the end value matrix. The simulation data of the variables to observe are saved in a result matrix. The new mode is then used as current mode and the while-loop is entered again. The mapping which was saved in the transition is then used to set the initial values of the current mode.

If the stop time of the simulation is reached the while-loop is not entered again. After the simulation is done the observed values are saved in a data-file which can later on be used to post-process the simulation data.

1.3 Creating variable-structure models

As an example on how the package can be used, we look again at the pendulum model. First lets consider the Modelica models needed for the variable-structure model. The package approach was chosen because we wanted to be able to use existing models and to reuse the models afterwards again. The package allows us to use our old models on its own because they are valid models, but each model needs a stop condition to be used as a mode in a variable-structure model so the model needs to be altered. Modelica with its object-oriented approach Modelica2010 helps us to keep our old models as they are and extend our needed modes from the old models. Figure 4 shows the pendulum model.
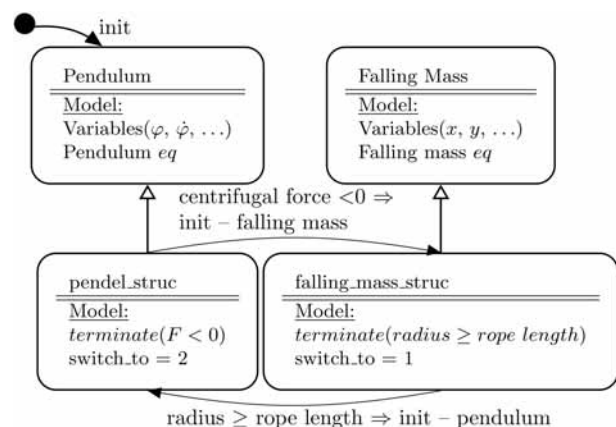

Figure 4: Using inheritance for variable-structure modeling.

The two models 'Pendulum' and 'Falling mass' represent the original models. The other models are extended models of the two and have the stop condition

(here called terminate) added and a variable 'switch_to' which specifies the mode that needs to be entered next. Here it can be seen, that the original models have not changed and can be used as before only the extended models now represent the modes in the variable-structure model .

After the models for the modes are defined the template provided in the package is used to define the mode switches for the variable-structure model .

```
stop = 10 # stoptime

model = ['Pendelum.mo'] # filename

mode1='pendel_struc' #first mode
mode2='falling_mass_struc' #second mode
modes=[mode1, mode2] #list of modes
sol=EULER # global solver

# SWITCH MODE 1 – > MODE 2
out1=['x','y','der(x)','der(y)']
in2=['x','y','vx','vy']
transition.append([1,2,out1,in2])

# SWITCH MODE 2 – > MODE 1
out2=['x','der(phi)']
in1=['x','dphi']
transition.append([2,1,out2,in1])

obs(['x','y'],['x','y'])

switch(stop,sol,model,modes,transition,obs)
```

For each model the name of the modelfile (or files) and the name of the modes have to be specified. These modes will later be compiled with Dymola. Afterwards the transitions have to be defined. A user defines a transition with the mode numbers of the two modes between which the transition is. Furthermore, the variables to read from the old mode (out1 and out2) and the variables that will be set in the new mode (in2 and in1) have to be specified. The variable 'observer' defines which variables should be saved in a data matrix at the end of the simulation. The data of each mode is mapped and saved in one data matrix (for instance: mode1.x and mode2.x will be written in one column of the data matrix). The data matrix is per default saved as MAT-File but the user can specify other output filetypes as well. All the information is then given to the switch method which creates the ModelObject and all the mode objects with their transitions. This template can be used to specify an arbitrary number of modes and switches between these modes.

2 Evaluation

In this section we present different variable-structure models. With these models it is shows how useful variable-structure models are. We then use an easy variable-behavior model to analyze the scalability of the Python package. At the end of this section requirements a model needs to fulfill to be used as a mode in a variable-structure model are discussed.

2.1 Variable-detail models

To show that variable-detail models can save simulation time we look at a diesel combustion engine model, see Figure 5. In this model the environment pressure changes every five seconds and thus the pressure of the manifold changes. When the pressure of the manifold and environment are almost the same the throttle and manifold are not necessary anymore and can be taken out. When the pressure changes again the model has to become more detailed again to simulate the dynamic pressure change in the manifold.
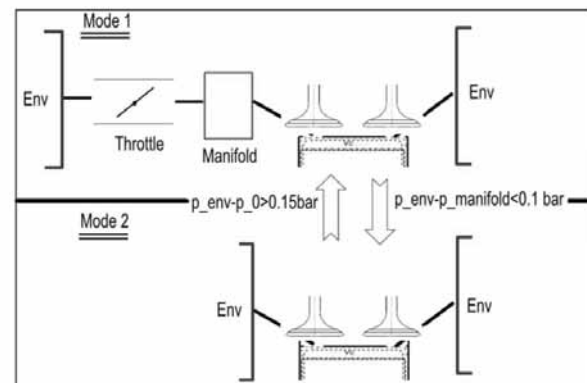
Figure 5: Schematic view of a diesel combustion engine variable-detail model.

The measured simulation times can be seen in table 1. Here the simulation times of the model with only one level of detail and with two levels of detail are presented. The model is always simulated for 20 seconds and has 7 mode switches. The *Compilationtime* is the time needed to compile the Dymola models and the *Residualtime* is the time the script needs for starting the simulation, setting initial values and so on. The variable-detail model takes less time than the one level of detail model even though two compilations are necessary and the residual time is larger. This leads to the

conclusion that variable-detail models can make a simulation faster. We also compared the results of the cylinder pressure and temperature and the difference was less than half a percent which shows that we were able to make the simulation faster without significant loss of accuracy.

Stop time 20sec /7 switches	One detail Dymola	Variable detail Dymola/Python
Simulation time	17	7.3
Compilation time	1*1.2 = 1.2	2*1.2 = 2.4
Residual time	1	2.3
Total time	**19.2**	**12**

Table 1: Simulation time for the diesel combustion engine in seconds.

2.2 Variable-behavior models

As an easy variable-behavior model we present a bouncing ball model. Here the bouncing ball does not just change its velocity when it hits the floor but becomes a spring and damper system which means the ball is elastic and bounces differently depending on the damping constant. Figure 6 shows the bouncing ball results with different damping constants. Interesting is that the ball can never fall below the surface as happens if only a 'when' statement n which the velocity is negated and multiplied by a factor (without extra precautions) is used in Modelica.

Furthermore, the deformation of the ball can now be modeled dependent on the current velocity of the ball, which makes the model more realistic. As a model with more than two transitions we present a breaking pendulum model where the rope can get stuck on a nail. The model is simplified to make it easier to understand:

- The pendulums suspension point is (0,0)
- The nail position is $x \leq 0$ and $y < 0$ ('nailPoint')
- The falling mass model is valid left of the nail

The simplified model is shown in Figure 7 (only a few important variables are shown in this view).
We still have two modes but when the rope passes the angle where the nail is located the rope length changes and therefore the suspension point of the pendulum. We make a mode switch into the same model but change the model parameters. If the centrifugal force goes below
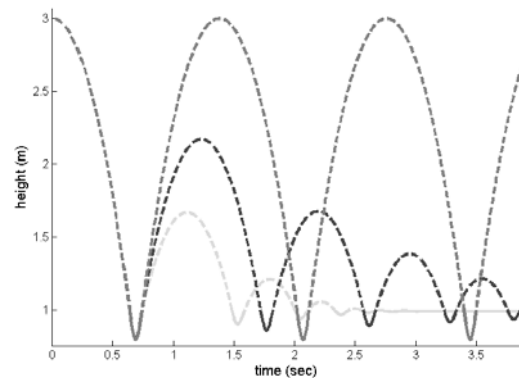
Figure 6: Simulation result of the center point of the bouncing ball variable-structure model .
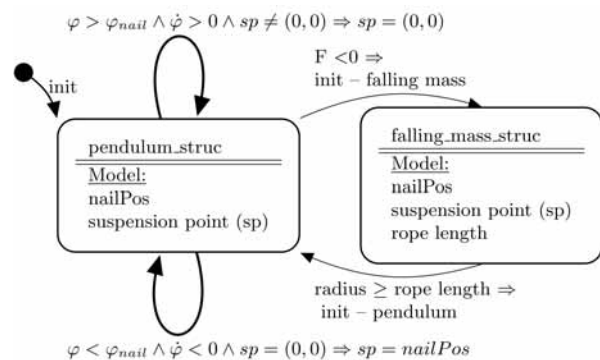
Figure 7: Constrained and breaking pendulum variable-structure model .

zero the pendulum becomes a falling mass otherwise it either turns around the nail or becomes the normal pendulum again. Figure 8 shows different movements of the pendulum for different start values of the angular velocity (dphi (rad/sec)) and the damping constant (D (N sec/m)). The normal suspension point and the nail are markedy as dots.

2.3 Scalability of the simulations

The scalability is always an issue with programs as presented here especially if one goal is to save simulation time. To test if the scripting approach with Python scales with the number of switches and the number of variables for initialization two different tests were made. For both test a bouncing ball model is used which has a transition to itself as soon as the ball touches the ground. The velocity is then negated and we have a ball that never stops bouncing.
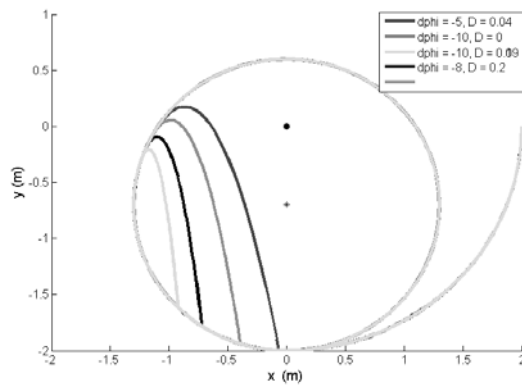
Figure 8: Simulation results of the breaking pendulum
model with nail.

This model has only 2 statevariables (height,velocity)
and these have to be initialized for each mode switch.
As first test this model is simulated with 10, 100, 1000,
10000 mode switches, see table 2. It can be seen that
the simulation time scales with the number of switches
and it can also be seen where most of the time is lost.

Switch	overall	mapp	while-loop			
	time	time	CPU	dymosim	read	init
10	0,64	0,046	0,02	0,33	0,22	0,02
100	5,21	0,039	0,10	2,90	1,98	0,17
1000	55,34	0,049	0,68	30,38	22,35	1,79
10000	544	0,039	6,20	298,00	220,22	18,33

Table 2: Simulation time for many switches in seconds.

The start of the dymosim.exe takes up a long time, but
we do not have any means to change anything on the dy-
mosim.exe routine (each time the executable is started
the license is checked, which also takes up time). The
other part that takes a long time is the reading of the
end values of the old mode. In the implementation the
dsinfinal.txt (which holds the end values of the simua-
tion) is changed into a Matlab file, because it is easier to
load. This process does take up a long time and we are
currently trying to find a better solution. The other mea-
sured times are the CPU time which is the time the sim-
ulation runs, the init time is the time it takes to set the
initial values in the initial file. This of course is a rather
drastic example because the idea of variable-structure
models is to use larger models with a long CPU time
and a few switches and not a model with almost no CPU
time and many switches.

As second example we use the same bouncing ball
but this time we create an array with many of these
balls. We always simulate for 10 switches but with
10, 100, 1000, and 10000 bouncing balls. This leads to
many statevariables which have to be initialized. Now
we see in table 3 that the CPU time takes up most of the
time, which was to be expected from larger models. We
see that finding the mapping at the beginning of the sim-
ulation takes up a long time. There it can be seen that it
is feasible to search the mapping once at the beginning
and not for each switch because the needed time would
be even greater. All other measured times are rather in-
significant compared to the large CPU time.
We see here that there is still some improvement nec-
essary for the index search but otherwise the approach
seems good for large models.

Balls	overall	mapp	while- loop			
	time	time	CPU	dymosim	read	init
10	5,00	0,06	0,4	4,3	0,17	0,02
100	6,75	0,07	0,23	5,5	0,24	0,45
1000	14,93	2,4	7,4	4,03	0,21	0,61
10000	322,27	62,15	252,2	5,47	0,77	0,85

Table 3: Simulation time for large models in seconds.

2.4 Model requirements

After introducing the Python package, its design, and
presenting examples of variable-structure models we
are now discussing the most important requirements for
variable-structure models with our Python package.

Looking at the scalability test it is clear that it is not
feasible to create variable-structure models with lots of
mode switches especially if the models them self are re-
ally small. Many switches lead to an overhead in simu-
lation time through the scripting. For variable-behavior
models this might still be reasonable because one might
otherwise not be able to simulate the system at all.

Another problem with many mode switches is the
initialization of the new mode. Each time a switch oc-
curs an initialization problem has to be solved. If the
values are chosen incorrectly or cannot be calculated
from the old mode the numerical solution might be-
come wrong or even instable. The initialization is there-
fore a great issue for variable-structure models and it is
only possible to switch from one mode to the next if the
new modes statevariables can be calculated through the

variables of the old mode. This means not all models are fit to be used as modes in variable-structure models.

For variable-detail models it is important that the models have a CPU time which is greater than the time the script consumes and also that the less detailed model at least compensates the scripting time otherwise no simulation time can be saved.

3 Conclusion and Future Work

Our approach is not able to re-causalize only the needed equations or to have a just-in-time compiler as some other language and tools for variable-structure models have but we are able to use a common simulation environment for our simulation and therefore use the strength of this tool.

We can use a tool like Dymola and give modelers the opportunity to test if variable-structure models are feasible for them.

Our approach helps to easily create variable-structure models from existing models and use easy means to describe the models. We therefore hope that with our Python package knowledge of variable-structure models can be gained.

In the future the package will be enhanced to a framework which will support different simulation tools and a graphical user interface. The framework should enable the user to use models from different tools as modes.

With the planned framework researches are planned on what a tool needs to be usable for variable-structure modeling and when variable-structure models should be used to be feasible.

References

[1] MATLAB/Simulink Release 2010b, The MathWorks, Inc., Natick, Massachusetts, United States.

[2] dynasim: Dymola [Internet]. Dassault Systèmes c2002-2014 [cited 2014 Dec]. Available from: www.dynasim.se

[3] Nytsch-Geusen C., Ernst T., Nordwig A. et al. Mosilab: Development of a modelica based generic simulation tool supporting model structural dynamics. In G. Schmitz, editor. *Proceedings of the 4th International Modelica Conference*; 2005 Mar, TU Hamburg, 527–535.

[4] Zimmer, D. *Equation-Based Modeling of Variable-Structure Systems* [dissertation]. [Swiss Federal Institute of Technology (CH)]. ETH Zürich; 2010.

[5] Nilsson H, Giorgidze G. Exploiting structural dynamism in Functional Hybrid Modelling for simulation of ideal diodes. *Proceedings of the 7th EUROSIM Congress on Modelling and Simulation*; 2010; Prague: Czech Technical University Publishing House.

[6] Urquia A., Dormido, S. Object-oriented description of hybrid dynamic systems of variable structure. *Simulation*. 2003; 79(9): 485–493.

[7] Mehlhase, A. Varying the level of detail during simulation. *ASIM 2011, Symposium Simulationstechnik*; 2011.

[8] Elmqvist H, Cellier F.E., Otter, M. Object-oriented modeling of hybrid systems. *Proc. 1993 European Simulation Symposium*; 1993; Delft.

[9] MOSILAB [Internet]. 2011 [cited 2014 Dec]. Available from: http://mosim.swt.tu-berlin.de/wiki/doku.php?id=projects:mosilab:home

[10] Nordwig, A. *Integration von Sichten für die objektorientierte Modellierung hybrider Systeme* [dissertation]. [Institut für Softwaretechnik und Theoretische Informatik (DE)]. Technische Universität Berlin; 2003.

Fluid Flow Modelling with Modelica

Marco Bonvini[1*], Mirza Popovac[2]

[1]Dipartimento di Elettronica e Informazione, Politecnico di Milano, Via Ponzio 34/5, 20133 Milano, Italia;
 *bonvini@elet.polimi.it

[2]Austrian Institute of Technology, Sustainable Building Technology, Gieffinggasse 2, 1210, Vienna, Austria

Abstract. This work shows how the problem of modelling fluids motion can be addressed in Modelica. This innovative approach makes possible to face such a problem in a multi-physic modelling language as Modelica is. In this way it is possible to simulate together the fluid and the system that interacts with it, without any additional effort and taking advantage of the Modelica libraries representing buildings, power plants, water treatment systems, HVAC and so forth.

Introduction

Modelling fluid flows is extremely important in simulating many engineering processes. When the fluid is constrained to move in ducts or pipes strong assumptions/simplifications can be taken into account without affecting the description of the fluid properties (e.g. temperatures, pressures, densities,...) and their distributions. The mentioned simplification for such cases where a spatial coordinate prevails the others leads to zero or one dimensional models where the spatial dependence is respectively disregarded or limited to just one coordinate (e.g. pipes). However there are elements like tanks (in the context of hydraulic systems) or rooms (in the context of HVAC systems) where zero or one dimensional models are not appropriate. The standard practice when simulating such more complicated scenery, is to employ CFD codes. Despite this approach is capable of representing in a very detailed way the fluid thermal dynamics, it has some drawback. The main one is in its modularity. CFD cannot be easily integrated with other models in order to represent the entire system, the only way for doing such a task is to employ the so called co-simulation techniques, that introduce a communication overhead and some non trivial convergence problem as shown by Trčka et al. [8].

The aim of this work is to provide a general methodology for modelling 2D or 3D fluid flows with Modelica. Modelica is a multi-physic Object-Oriented modelling language [**?**]. In Modelica several modelling libraries, representing a variety of systems are already available [6], and new ones can be developed. Thanks to the modularity of the language and the deffinition of standard interfaces, models belonging to different physical domains can be coupled together. Providing a way for modelling fluid flows in such an environment is a step ahead in the direction of a real integrated multi-domain simulation tool, thus avoiding co-simulation and its drawbacks [8], [10]. The proposed modelling approach aims at representing simple scenery in cases where the powerful capabilities of CFD software are not needed. More precisely, complex geometries and high velocities are not taken into account , however a wide range of application like rooms, portion of buildings, storage tanks can be modelled. As consequence, despite the apparent simplicity of the proposed approach a widespread set of relevant cases can be investigated.

The structure of the paper is the following: an introductory section where the governing equations are shown is followed by a section in which the discretisation approach is presented. Then, the implementation in the Modelica language of the discretised equation is discussed. The last section concerns the validation of the models, and more in detail, a comparison between experimental data coming from a natural convection case is reported. Some conclusions as well future works complete the paper.

1 The Governing Equations

The motion of fluid is described by the equations of mass, energy and momentum balance, and this set of equations is often referred to as the Navier Stokes equations (NS). In the case of the Newtonian fluid they can be written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad \text{(mass)}$$

(1a)

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho \mathbf{v} h) = \nabla \cdot (k \nabla T) \quad \text{(energy)}$$

(1b)

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}^T) + \nabla p = \nabla \cdot (\mu \nabla \mathbf{v}) + \mathbf{f} \quad \text{(momentum)}$$

(1c)

where the scalars p, T, e, h, ρ, k and μ are respectively the fluid pressure, temperature, specific energy, specific enthalpy, density, thermal conductivity and dynamic viscosity; the vectors $\mathbf{v}$ and f are the fluid velocity and the external forces only, such as gravity, acting on the fluid.

With the scalar projection brought in, and for simplicity analysing only 2D case, the momentum equation (1c) is decomposed into two scalar equations:

$$\frac{\partial \rho v_x}{\partial t} + \frac{\partial \rho v_x v_x}{\partial x} + \frac{\partial \rho v_x v_y}{\partial y} =$$
$$f_x - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left(\mu \frac{\partial v_x}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial v_x}{\partial y}\right)$$

(2a)

$$\frac{\partial \rho v_y}{\partial t} + \frac{\partial \rho v_y v_x}{\partial x} + \frac{\partial \rho v_y v_y}{\partial y} =$$
$$f_y - \frac{\partial p}{\partial y} + \frac{\partial}{\partial x}\left(\mu \frac{\partial v_y}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial v_y}{\partial y}\right)$$

(2b)

where the subscripts x, y denote the components of the 2D Cartesian coordinate system. In the case of natural convection having as vertical axis the y one, clearly $f_x = 0$ and $f_y = -\rho g$, with g being the gravity acceleration. In order to solve numerically the momentum equation (as well the continuity and the energy ones), each term appearing has to be properly represented.

2 The Discretised Model

The conservation equation for mass (refeqn:mass), energy (refeqn:energy) and momentum (1c) can be represented in the standard Convection-Diffusion (CD) form, which (again for simplicity) in the 2D case reads:

$$\underbrace{\frac{\partial \rho \Phi}{\partial t}}_{\text{local}} + \underbrace{\frac{\partial \rho v_x \Phi}{\partial x} + \frac{\partial \rho v_y \Phi}{\partial y}}_{\text{convective}} =$$
$$\underbrace{\frac{\partial}{\partial x}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial y}\right)}_{\text{diffusive}} + \underbrace{S_\Phi}_{\text{source}}$$

(3)

where the generic quantity Φ is the scalar quantity transported by the fluid moving with velocity $\mathbf{v} = (v_x, v_y)$, and Γ_Φ is the diffusivity coefficient. The time dependent variable Φ can be either one velocity component, the internal energy or the mass fraction of a chemical species. The source term S_Φ is the generation rate of the scalar quantity Φ per unit volume.

The generic CD equations (3) states that the (unsteady) local change of the scalar quantity Φ is equal to the sum of the convective change, the diffusive change, and the generation from a source. For example, replacing Φ with v_y, Γ_Φ with the viscosity μ and collecting in the source term S_Φ both the gravity $-\rho g$ and the pressure gradient dp/dy, the y-momentum equation (2b) is obtained. Following the numerical procedure for solving CD, described in [7], the general CD equation (3) is integrated over a grid of Control Volumes (CV) as shown in Figure 1.
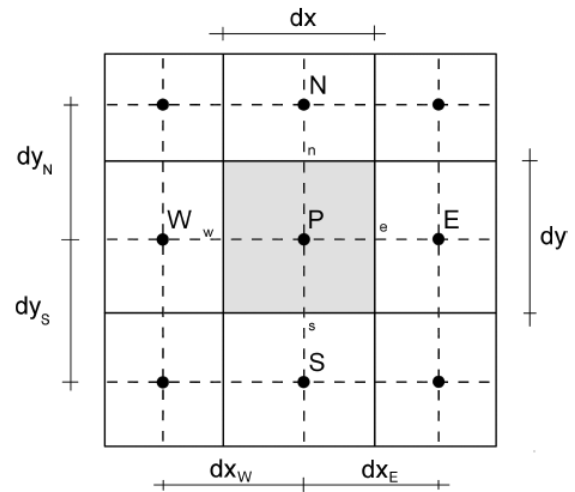
Figure 1: Grid employed for the spatial discretisation of the CD equation.

Applying the Gauss' theorem, the volume integrals are replaced with surface ones:

$$\int_V \frac{\partial \rho \Phi}{\partial t} dV + \int_S (\partial \rho v_x \Phi) \mathbf{i} \cdot \mathbf{n} dA + \int_S (\partial \rho v_y \Phi) \mathbf{j} \cdot \mathbf{n} dA =$$
$$\int_S \left(\Gamma_\Phi \frac{\partial \Phi}{\partial x}\right) \mathbf{i} \cdot \mathbf{n} dA + \int_S \left(\Gamma_\Phi \frac{\partial \Phi}{\partial y}\right) \mathbf{j} \cdot \mathbf{n} dA + \int_V S_\Phi dV$$

(4)

where $\mathbf{i}$ and $\mathbf{j}$ are respectively the x and y components of the unit vector, while $\mathbf{n}$ is the outgoing unit normal vector from the surface element dA. The surface integrals that appear in (4) can be approximated via sums over the faces of the considered control volume.

The unsteady term and the sources that represent respectively the variation of the scalar quantity Φ over the time, and the rate of generation into the CV, are replaced as follows:

$$\int_V \frac{\partial(\rho\Phi)}{\partial t}dV \simeq V\frac{d(\rho\Phi)}{dt} \qquad (5a)$$

$$\int_V S_\Phi dV \simeq V S_\Phi \qquad (5b)$$

where V is the volume of the CV. In particular, implementing the Final solution in Modelica, there is no need for explicitly implement a time discretisation method. Indeed, using the capabilities of the Modelica solvers (e.g. [**?**]) several adaptive time step solvers can be employed without any additional effort required.

When the CD equation aim at representing the NS equations, the pressure gradients appearing in (2) are included into the source terms. Once integrated over the CV and converted into surface integrals they read

$$\int_S P\mathbf{i}\cdot\mathbf{n}dS \simeq A_x(P_e - P_w) \qquad (6a)$$

$$\int_S P\mathbf{j}\cdot\mathbf{n}dS \simeq A_x(P_n - P_s) \qquad (6b)$$

where $A_{x,y}$ are the surfaces of the CV normal to the x and y direction respectively, and $P_{e,w,n,s}$ are the pressures on the boundaries of the CV. The diffusive term of the equation (4) can be approximated as:

$$\int_S \left(\Gamma_\Phi \frac{\partial\Phi}{\partial x}\right)\mathbf{i}\cdot\mathbf{n}dA \simeq \\ D_{\Phi_e}(\Phi_E - \Phi_P) - D_{\Phi_w}(\Phi_P - \Phi_W) \qquad (7a)$$

$$\int_S \left(\Gamma_\Phi \frac{\partial\Phi}{\partial y}\right)\mathbf{j}\cdot\mathbf{n}dA \simeq \\ D_{\Phi_n}(\Phi_N - \Phi_P) - D_{\Phi_s}(\Phi_P - \Phi_S) \qquad (7b)$$

where $D_{\Phi_{e,w,n,s}}$ are the diffusivity coefficients evaluated at the CV faces. For the east face (omitting the others for brevity) it is computed as:

$$D_{\Phi_e} = \Gamma_{\Phi_e} \frac{A_x}{dx_E} \qquad (8)$$

where dx_E is the distance between the center of the CV and the neighbour close to the E face. The diffusivity of the fluid is a property that may vary between adjacent CVs (e.g. the fluid viscosity or the thermal conductivity vary in time and space). For such a reason the diffusivity Γ_{Φ_e} is computed on the boundaries of the CV as a weighted mean of the fluid properties in the cell P and the cell E.

The most influential in a fluid flow is the convective term. Assuming that the velocities are normal to the surfaces of the CV, (v_x normal to faces E and W, while v_y normal to faces N and S), they can be approximated as:

$$\int_S (\rho v_x \Phi)\mathbf{i}\cdot\mathbf{n}dA \simeq F_e\Phi_e - F_w\Phi_w \qquad (9a)$$

$$\int_S (\rho v_y \Phi)\mathbf{j}\cdot\mathbf{n}dA \simeq F_n\Phi_n - F_s\Phi_s \qquad (9b)$$

where $F_{e,w,s,n}$ are the mass fluxes over the faces of the control volume. Again for brevity, for the e face the mass flux is computed as

$$F_e = (\rho v_x)_e A_x \qquad (10)$$

where the subscript indicates that the value is computed on the e boundary of the CV, while A_x is the surface of the CV normal to x directions. The values $\Phi_{e,w,n,s}$ introduced in (9) are the values of the scalar variable Φ on the boundaries of the CV. The way these values are computed has a strong impact on the numerical solution. The standard practice is to employ the first order accurate UPWIND scheme (11), where the scalar value on the boundary is taken from the one computed in the cell from where the fluid is flowing (hence the name upwind). Other methods, extensively described in [7], have been implemented.

$$\Phi_e = \begin{cases} \Phi_E & \text{if } F_e > 0 \\ \Phi_P & \text{if } F_e < 0 \end{cases} \qquad (11)$$

3 Implementation in Modelica

Having all the terms appearing in the CD equation (3) discretised over a given CV, the last step is to transform such an equation in a compact form that can be written in Modelica. The standard form, employed in all CFD tools and carefully explained in [7] has been adapted in order to be straightforwardly implemented in Modelica. The equation reads

$$V\rho\frac{d\Phi_P}{dt} + a_P\Phi_P = a_E\Phi_E + a_W\Phi_W + a_N\Phi_N + a_S\Phi_S + S \qquad (12)$$

where the coefficients $a_{E,W,N,S,P}$ are a compact representation of both the diffusive and convective terms, while S are the possible sources (e.g. gravity, heat sources, external forces ... depending on the nature of Φ) or in the case of the momentum equation the

pressure gradients. Coefficients $a_{E,W,N,S}$ are defined as:

$$a_E = D_e A \left(\left| \frac{F_e}{D_e} \right| \right) + \| -F_e, 0 \| \qquad (13a)$$

$$a_W = D_w A \left(\left| \frac{F_w}{D_w} \right| \right) + \| F_w, 0 \| \qquad (13b)$$

$$a_N = D_n A \left(\left| \frac{F_n}{D_n} \right| \right) + \| -F_n, 0 \| \qquad (13c)$$

$$a_S = D_s A \left(\left| \frac{F_s}{D_s} \right| \right) + \| F_s, 0 \| \qquad (13d)$$

with $a_P = a_E + a_W + a_N + a_S$. In (13), $\|a,b\|$ is the maximum between a and b while $A(\cdot)$ is a function that represents the convective scheme employed (e.g. UPWIND or Central Difference) as described in [7].

Such a generalised version of the CD equation can be used for discretising the mass, the energy and the momentum balance equations. These equations have been spatially discretised over a staggered grid as suggested by Versteeg and Malalasekera ([9]), an example of such a grid is shown in Figure 2. The basic idea behind the staggered grid is to integrate the balance equations over CVs that differs, in order to avoid numerical problems as evidenced in [7], [9].
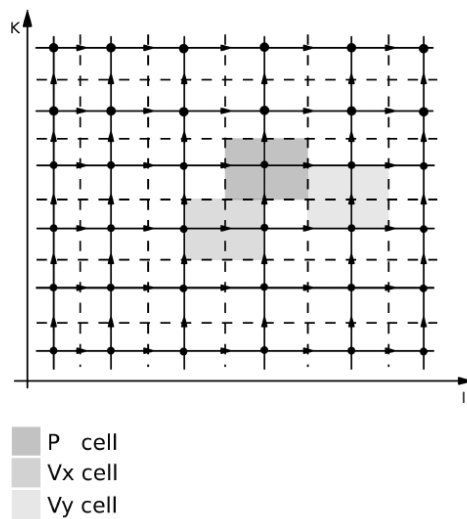

Figure 2: Staggered grid – Mass and Energy balance equations are discretised over P cells, while x-Momentum and y-Momentum equations are discretised over Vx and Vy cells.

For the boundary conditions, either the value (e.g. temperatures, velocities) or the gradients (e.g. heat fluxes) can be described. The grid of CVs is implemented as a matrix of nodes. The value of each node represents one of the scalar variable for which the CD equation is solved (e.g. the temperature, or velocity components). Boundary conditions are a given subset of values of these matrices. Boundary conditions can be extended by employing connectors, in such a way the values of a particular quantity (e.g. the temperature) instead of being defined a priori can be assigned by an other model. It is important to underline that Connectors are standard interfaces between the model and its neighbours. If the behaviour of the model has been properly described, just by knowing the information provided by its connectors, the model can be linked together with any model that implements the same connectors. This is a crucial point in the context of OO modelling, and this is the key that allows a real and powerful multi-physic simulation. The model is completed with a description of the fluid (the fluid state equation) that introduce a relationship between the temperature, pressure and density of the fluid. Such a relationship has been kept intentionally simplified, in order to reduce the complexity of the model. In particular a linearised version of the ideal gas relationship has been introduced.
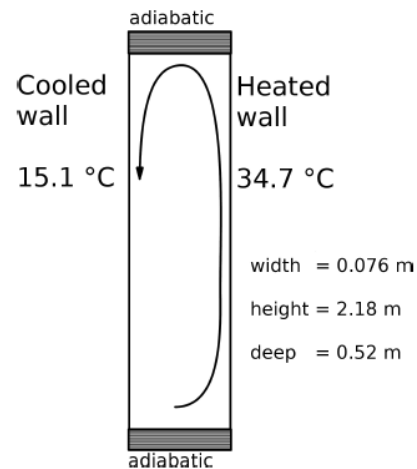
Figure 3: Scheme of the tall cavity.

A turbulence model and also a wall function representing the interaction between the fluid and the domain boundaries have been introduced. The complexity of the above mentioned models has been kept as low as possible, in order to reduce the computational effort. Therefore, following the idea of Prandtl ([?]), a zero-equation turbulence model is used, with the wall function (Launder and Spalding, [4]) used for imposing the wall boundary conditions for momentum equation.

(a) *Temperature at* $y = 0.1Y$.

(b) *Temperature at* $y = 0.4Y$.

(c) *Temperature at* $y = 0.6Y$.

(d) *Temperature at* $y = 0.9Y$.

(e) *Air vertical velocity at* $y = 0.1Y$.

(f) *Air vertical velocity at* $y = 0.4Y$.

(g) *Air vertical velocity at* $y = 0.6Y$.

(h) *Air vertical velocity at* $y = 0.9Y$.

Figure 4: Temperature (a - d) and air vertical velocity (e - h) distributions at different heights.

4 Validation

The validation is performed investigating the case of natural convection in a tall cavity (see Figure 3) where the right wall is heated while the left one is cooled. The sizes of the cavity are $0.076 \times 2.18 \times 0.52 [m]$.

Experimental results for such a cavity are taken from [2]. The shape of the cavity, as well its symmetry, allows to describe the fluid with a 2D grid, without reducing the accuracy in the description of the temperature distribution and the air flow field. For such a reason a non-uniform grid of 11×21 volumes has been used. The comparisons between simulation data and experimental results are listed in Figures 4. In particular, both temperature and vertical velocity profiles at different heights ($y = \{0.1, 0.4, 0.6, 0.9\}Y$, where Y is the height of the cavity) are shown. More in detail, Figures 4 (a-d) are the temperature profiles, while 4 (e-h) are the vertical velocity ones. In each plot experimental data are compared against 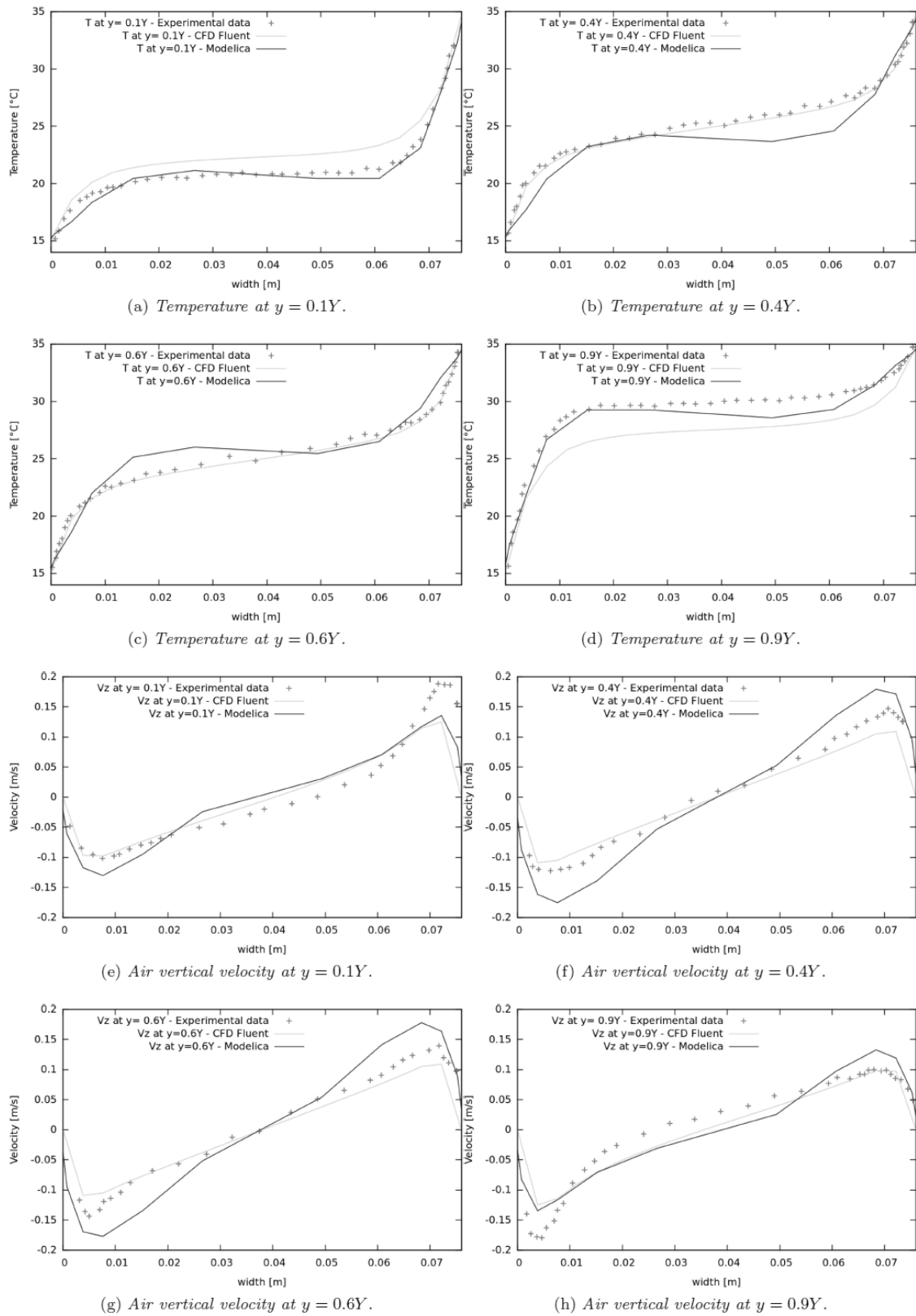simulation data provided by a standard CFD code ([1]) and simulation data obtained with Modelica models. The agreement between results provided by Modelica models and both CFD as well as experimental data is very good as can be seen in the various Figures.

To stress that the aim of Modelica models is not to give more accurate results with respect to CFD ones, but to give comparable ones by using a modelling paradigm that offers the possibility to integrate not only the fluid motion but also the interaction with other systems (e.g. the walls that surround the ambient, the environmental conditions as well as a suitable representation of the heat sources acting on the system).

5 Conclusion

A model capable of simulating fluid flows with an approach which is not the standard CFD has been proposed. Such a model makes possible to face the fluid-flows problem in a multi-domain modelling language, such as Modelica.

Despite the simplicity of the numerical scheme employed as well the geometry description taken into account, the big advantage is that now it is possible to simulate together the fluid and the system that interacts with it, without any additional effort and taking advantage of the Modelica libraries and avoiding co-simulation.

References

[1] Ansys fluent home page. URL http://www.ansys.com.

[2] Betts P L, Bokhari I H. Experiments on turbulent natural convection in an enclosed tall cavity. *International Journal of Heat and Fluid Flow*. 2000; 21(6): 675 - 683. doi:10.1016/S0142-727X(00)00033-3

[3] Dymola 7.4. URL http://www.3ds.com/products/catia/portfolio/dymola.

[4] Launder B, Spalding D. The numerical computation of turbulent flows. *Comput. Meth. Appl. Mech. Engrg.*. 1974; 2(3): 269 - 289. doi:10.1016/0045-7825(74)90029-2

[5] Mattsson S, Elmqvist H, Otter M. Physical system modeling with Modelica. *Control Engineering Practice*. 1998; 6(4), 501 - 510. doi:10.1016/S0967-0661(98)00047-1

[6] Modelica home page. URL http://www.modelica.org/.

[7] Patankar S. *Numerical heat transfer and fluid flow*. Edition 1. Washington, DC: Hemisphere Publishing Corp.; 1980. 197 p.

[8] Trčka M, Hensen J L, and Wetter M. Co-simulation for performance prediction of integrated building and hvac systems - an analysis of solution characteristics using a two-body system. *Simulation Modelling Practice and Theory*. 2010; 18(7): 957 - 970. doi:10.1016/j.simpat.2010.02.011

[9] Versteeg H, Malalasekera W. *An introduction to computational fluid dynamics: the finite volume method*. Edition 2. Upper Saddle River, NJ, USA: Pearson Prentice Hall; 2007. 503 p.

[10] Zhai Z, Chen Q Y. Numerical determination and treatment of convective heat transfer coefficient in the coupled building energy and cfd simulation. *Building and Environment*. 2004; 39(8): 1001 - 1009. doi:10.1016/j.buildenv.2004.01.023

Power Transfer by Non Radiative Electromagnetic Fields between High-Q Resonant Coupled Circuits

José Alberty

Department of Science and Technology Industries, University of Applied Sciences of Western Switzerland, Rue de la Prairie 4, CH-1202 Genève, Switzerland; *jose.alberty@hesge.ch*

Abstract. Between two high quality-factor (Q) resonant magnetically coupled circuits, non-radiative power transfer is modelled and observed in agreement with predictions found in recent works from MIT. The physical behaviour of the receptor as well as the geometry of the power flux lines (Poynting) are explained in terms of general behaviour of the power flux near completely absorbing targets. Practical consequences are extracted and generalisations of the source-receptors' geometries are proposed.

Introduction

It was recently shown [7], [6] that strongly coupled oscillating electrical circuits at resonance having high Q factors, are able of exchanging electromagnetic power in ways which are markedly different from more common coupled circuits, such as primary and secondary circuits in a transformer or even close emitter-receptor antennae pairs. What distinguishes power exchanges in traditional devices from those in the new ones relates to the way in which the power flux organizes itself geometrically between the source/emitter and the target/receiver regions. This particular organisation is a result of the interplay between the wave character of the field and the conditions of near-perfect wave absorption which are met at the receptor level at resonance. In this sense the same type of qualitative phenomena are found in all domains where strong wave absorption occurs (optical waves around totally absorbing bodies, particle wave function in the neighbourhood of resonant nuclei, etc.).

A complete analytical model is developed under the assumption of large electromagnetic wavelengths λ, compared to the set-up characteristic dimensions.

On the modelling and simulation side, we extensively used *Mathematica* to develop the entire analytical model and to visualise the fields and the fluxes in 3D, to compute - symbolically and numerically - the iterative solutions to the mixed induction-law equations and to interactively explore the space of parameters available in the this type of physics problem. This latter facility offered by this software was very helpful in understanding the physical subtleties involved in this type of systems. Finally the possibility offered by *Mathematica* of defining functions with generic arguments which remain unevaluated until called for by numerical or functional operations, allowed for changes in the definition of these arguments at later stages in the modelling chain, changes which are immediately and automatically impacted on all intermediate stages of the chain, without the need to redefine the original function for each new type of argument (reusable functions).

1 Setup Parameters and Geometry

In this work we consider a system made of one or more copper solenoids, represented by loops of radius $b = 5.5$ [cm] and diameter $a = 1.5$ [mm], even if each solenoid may contain more than one loop. Actually we use 5.5 turns per coil. At least one of the solenoids - the reference solenoid - is always present, located on the OXY -plane and centred at the origin O of the axis (see Figure 1).

We will use polar and Cartesian coordinates, depending on the type of calculation needed. For a general point P in space, these coordinates are shown in Figure 1. To begin with and for the sake of comparison with later results, we model the electro- magnetic field generated by an alternating current in the reference loop at frequencies $f1 = 5$ [MHz] and $f2 = 35$ [MHz], neglecting its inductive, capacitive and ohmic characteristics.
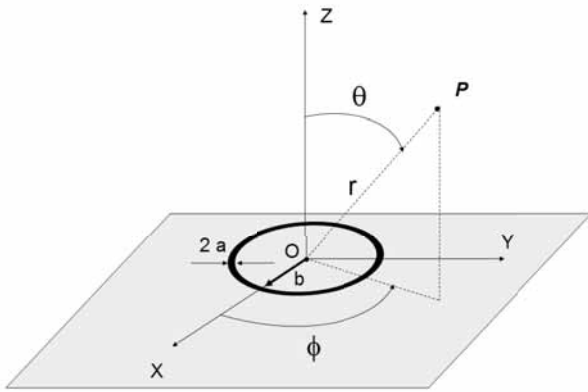
Figure 1: Cartesian (X,Y,Z) and polar (r,ϕ,θ) coordinates of a point P. The horizontal circle of radius b is a single conduction loop of diameter $2 \cdot a$.

Later we will model the interaction of an external field with the reference solenoid. The full impedance elements of the solenoid's equivalent circuit must then be taken into account. For a solenoid with the characteristics indicated above, the inductance will be $L = 7.7\ [\mu H]$. We model the capacity C as a tunable capacity fixed at $C = 3.3\ [nF]$ in series with L and with the total resistance which is quite small ($R = 0.01\ [\Omega]$) and mainly due to the wire' s ohmic surface resistance per turn(We will neglect the presence of distributed capacities and radiative resistance.). The RLC oscillating circuit formed by the solenoid, the capacity and the resistance will thus resonate at frequency $f_{res} = \frac{1}{2\pi\sqrt{LC}} = 998'649\ [Hz]$.

2 Fields and Power Fluxes

The physics behind the phenomena which interest us here are simply Biot-Savart's and Faraday-Lenz's induction laws, iterated several times. An important assumption we make is that the magnetic fields' wavelength must be far larger than the typical circuits dimensions. This amounts to saying that the current distributions inside each circuit may safely be considered as being uniform along the circuits' length. If this condition is not met, our results will not be generally correct. The present modelling approach is similar to [3]. However our main interest relates to the dynamics of the electromagnetic field in the entire space and its relation to the effect of power flux concentration at resonances.

2.1 Single solenoid source

At an arbitrary point P in free space, with coordinates (r,ϕ,θ), the electromagnetic field created by an oscillating uniform current of frequency f and peak intensity I, circulating inside an horizontal (XY plane) conducting loop of radius b, made of wire of radius a and centered at the origin, is given by [1]

$$
\begin{aligned}
H_r &= H_r(f,I,\theta,r) \\
&= \frac{\iota k b^2 I \cos\theta}{2 r^2} e^{-\iota k r}\left(1 - \frac{\iota}{k r}\right) \quad (1) \\
H_\theta &= H_\theta(f,I,\theta,r) \\
&= -\frac{(k b)^2 I \sin\theta}{4 r} e^{-\iota k r}\left(1 - \frac{\iota}{k r} - \frac{1}{(k r)^2}\right)(2) \\
E_\phi &= E_\phi(f,I,\theta,r) \\
&= \eta\, \frac{(k b)^2 I \sin\theta}{4 r} e^{-\iota k r}\left(1 - \frac{\iota}{k r}\right) \quad (3)
\end{aligned}
$$

where H_r and H_θ denote the radial and zenith components of the magnetic field and E_ϕ is the azimuthal component of the electric field at P. Here $k = \frac{2\pi}{\lambda}$ is the field's wave-number and $\eta = \sqrt{\frac{\mu_0}{\eta_0}}$ the vacuum impedance.
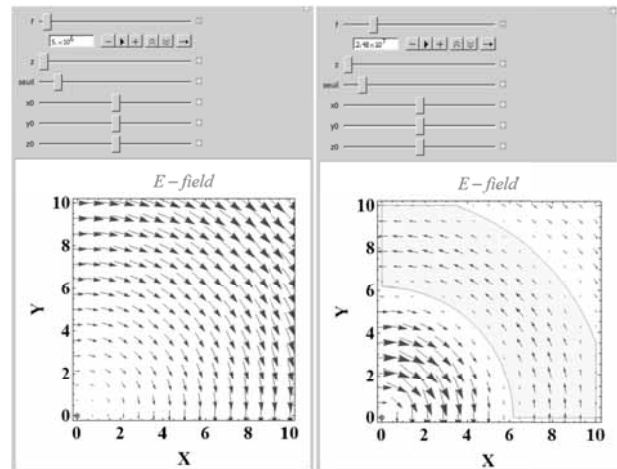

Figure 2: E-field lines at frequencies f_1 and f_2 in the XY-plane. Coloured sector (yellow) corresponds to $E_\phi > 0$. The solenoid is indicated by a fat red dot at the origin representing its center.

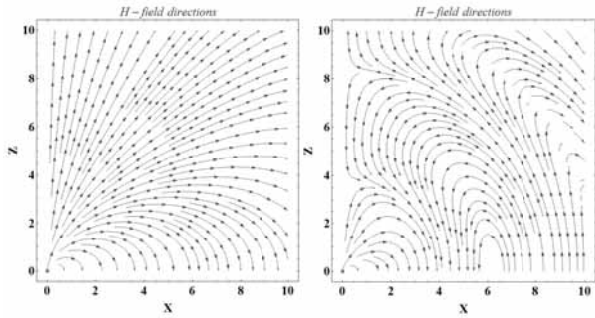The E and H-field lines are shown in Figure 2 and Figure 3.

Figure 3: H-field lines at frequencies f_1 and f_2 in the XZ-plane.
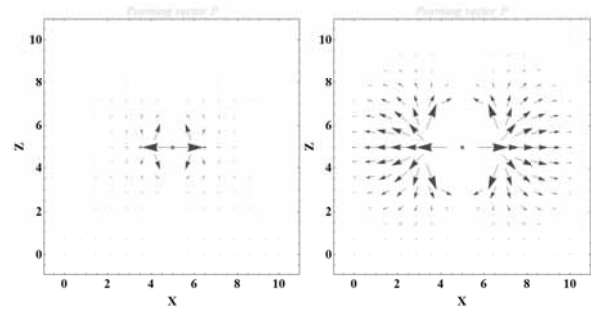

Figure 4: Poynting vector field (power flux) in the XZ-plane under the same conditions as in Figures 2 and 3. Arrow lengths are proportional to the power flux intensities (Poynting norm).

The power fluxes are determined by using the *Poynting vector*, defined as

$$\vec{P} = \Re\vec{E} \wedge \Re\vec{H} \qquad (4)$$

where the Cartesian expressions of the E and H-fields in the XY and XZ-planes are obtained from the field components in equation (1) after rotating by the angles ϕ and θ respectively:

$$\vec{E} = R_z(\phi)\,\{0, E_\phi, 0\} \qquad (5)$$
$$\vec{H} = R_y(\theta)\,\{H_r, 0, H_\theta\} \qquad (6)$$

where $\quad R_z(\phi) = \begin{pmatrix} cos\phi & -sin\phi & 0 \\ sin\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad$ and

$$R_y(\theta) = \begin{pmatrix} sin\theta & 0 & cos\theta \\ 0 & 1 & 0 \\ cos\theta & 0 & -sin\theta \end{pmatrix}.$$

For future reference, the power flux lines (field of Poynting vectors) at frequencies f_1 and f_2 are shown in Figure 4. The maximal power fluxes occur on the loop plane and the minimal (zero) flux is perpendicular to this plane near the loop, in agreement with the typical shape of power distribution polar diagrams for small loop antennas.

3 Single Coil in an External Magnetic Field

3.1 Uniform magnetic field

Now we consider a single solenoid inside an homogeneous magnetic field of amplitude H_{ext}, oscillating at frequency f and parallel to the solenoid's axis. The solenoid is connected in series to a variable capacity C and an ohmic resistance R, forming an *RCL* circuit. The oscillating current $I_{ind} = I_{ind}(f, H, C)$ induced in the solenoid creates a scattered electromagnetic field with components

$$
\begin{aligned}
H_r^{scatt-PW}(f,H,\theta,r,C) &= H_r(f, I_{ind}(f,H,C), \theta, r), \\
H_\theta^{scatt-PW}(f,H,\theta,r,C) &= H_\theta(f, I_{ind}(f,H,C), \theta, r) \\
E_\phi^{scatt-PW}(f,H,\theta,r,C) &= E_\phi(f, I_{ind}(f,H,C), \theta, r).
\end{aligned}
$$

The extension PW stands for plane-wave.

In Figure 5 we show the Poynting-field lines calculated from these scattered fields, with L, C and the resonant frequency f_{res} of the equivalent *RCL* circuit given in section 1. The frequencies shown are at $f = 0.90 f_{res}$, $f = f_{res}$ and $f = 1.10 f_{res}$. Notice the clear qualitative change of the Poynting flux lines at fres with respect to the cases with $f = f_{res}$. This is due to the fact that, at resonance, it is the real part of the oscillating induced current that dominates (similarly to the case when the fields stem from an imposed real current in the solenoid, as in section 3), whereas the imaginary part of Iind dominates away from fres. The fact that the external field is homogeneous and oriented along OZ implies that, at all frequencies, the scattered magnetic and Poynting flux diagrams in the XZ plane exhibit a $X \to -X$ symmetry and there is no net power flux along X.

The full power fluxes $\vec{P}$ are the sum of the scattered Poynting fields $\vec{P}_{scatt}$ shown above and the power fluxes $\vec{P}_{ext}$ associated to the externally imposed magnetic field $\vec{H}_{ext}$:

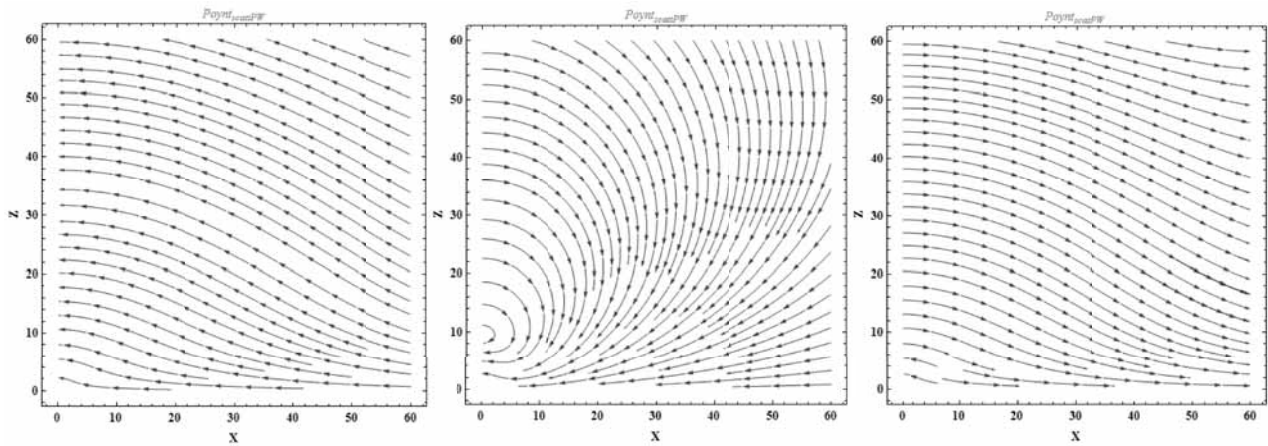**Figure 5:** Scattered Poynting-vector fluxes at $f < f_{res}$ (upper left), $f = f_{res}$ (upper right) and $f > f_{res}$ (bottom).

$$\vec{P} = \vec{P}_{ext} + \vec{P}_{scatt}$$

Clearly, an harmonic external pure magnetic field $\vec{H}$ which is spatially homogeneous will carry no power flux, not only for reasons of symmetry (in which direction would the flux be oriented ?) but also because an electric field is needed to create a Poynting vector. Last, but not least, such a pure magnetic field does not exist, according to Faraday's law

$$rot\,\vec{E}_{ext}(\vec{r},t) = -\frac{\partial \vec{B}_{ext}(\vec{r},t)}{\partial t} \qquad (7)$$

where $\vec{E}_{ext}$ is the induced external electric field and $\vec{B}_{ext} = \mu_0 \vec{H}$ is the external *magnetic induction* field.

3.2 Near plane-wave external magnetic field

We assume that $\vec{B}_{ext}$ varies harmonically in time that is, $\vec{B}_{ext}(\vec{r},t) = \vec{B}_{ext}(\vec{r})\,e^{\iota\,\omega\,t}$. Then, by linearity, $\vec{E}_{ext}$ and $\vec{B}_{ext}$ will have the same dependency on t and equation (7) reduces to

$$rot\,\vec{E}_{ext}(\vec{r}) = -\iota\,\omega\,\vec{B}_{ext}(\vec{r}) \qquad (8)$$

We make the further assumptions that $\vec{B}_{ext}$ is everywhere oriented along OZ and that its eventual variations are along OXs that is, $\vec{B}_{ext}(\vec{r}) = \{0,0,B_0(x)\}$. Then, if we restrict ourselves to the OXZ-plane, where $\vec{E}_{ext}$ has only a y-component ($E_x = E_z = 0$), then equation 8 becomes

$$\frac{d\vec{E}_{ext}(x)}{dx} = -\iota\,\omega\,\vec{B}_0(x) \qquad (9)$$

Were $B_0(x)$, and thus $\vec{B}_{ext}$, constant, the resulting electric field would have $E_y(x) = -\iota\,\omega\,B_0\,x + E_y^{(o)}$. Therefore it would become arbitrarily large at sufficiently large x, which is physically unacceptable. To avoid this, we introduce a x-dependency in $\vec{B}_{ext}$ in such a way that $\vec{B}_{ext}$ is almost uniform in the region surrounding the solenoid.

Furthermore, to have a globally net power flux, this magnetic field will be a planewave magnetic for which we define a propagation direction, sense and wavenumber $k = \frac{2\pi}{\lambda}$, by means of a wave-vector $k = k\vec{u}$, with $\vec{u}$ the unitary vector giving the magnetic wave direction and sense. Consider thus the following external magnetic-induction field $\vec{B}_{ext}(\vec{r}) = \{0,0,B_0(x)\}$ with

$$B_0(x) = b_0\,e^{-\frac{(x-x_0)^2}{2\sigma^2} - \iota\,k\,x} \qquad (10)$$

and the induced field is $\vec{E}_{ext}(\vec{r}) = \{0, E_y(x), 0\}$, with

$$E_y(x) = -b_0\sqrt{\frac{\pi}{2}}\,e^{-\frac{1}{2}k^2\sigma^2}\,\sigma\,\omega\,Erfi\left(\frac{\iota\,x - k\,\sigma^2}{\sigma\sqrt{2}}\right) \qquad (11)$$

and $Erfi$ is the imaginary error function, $Erfi(z) = -\iota\,erf(\iota\,z)$.

For an external magnetic induction of amplitude $b_0 = 1\,[T]$, with a spatial spread of $\sigma = 70\,[m]$ oscillating at the resonating frequency $f_{res} = 998'649\,[Hz]$, the Poynting vector $\vec{P}_{ext} = Re[E_y(x,\sigma,k,b_0)] \cdot Re[B_0(x,\sigma,k,b_0)/\mu_0]$ is aligned in the positive X-direction and varies slowly in x with a scalar value close to $P_x^{(o)} = 1.35 \times 10^{-4}\,[W/m2]$.
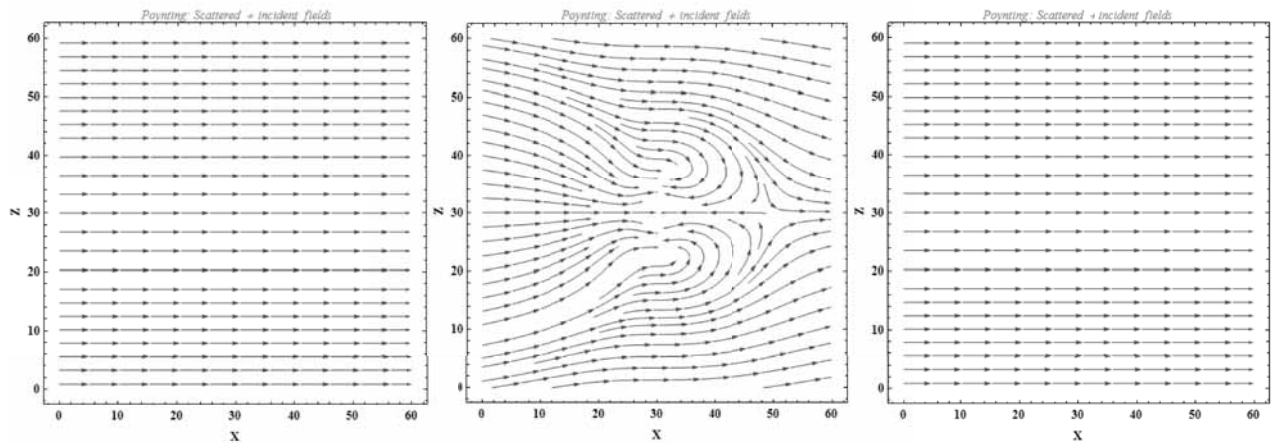
Figure 6: Power flux lines in the XZ-plane at frequencies $0.90\,f_{res}$ (left), f_{res} (center) and $1.10\,f_{res}$ (right). The field arrows are not drawn to scale, they only show the local field directions. Notice that the solenoid's centre is now at $x_0 = z_0 = 30\,[m]$.

The superposition of this external Poynting field and the scattered one is shown in Figure 6 at frequencies $0.90\,f_{res}$, f_{res} and $1.10\,f_{res}$. The concentration of flux lines is particularly strong at resonance. Away from the resonance, this effect is either absent (low-f) or less pronounced (high-f).

4 Poynting Concentration, Resonance and Total Absorption

The clear convergence of the Poynting flux lines around the solenoid at resonance with an external magnetic field is related to the sharp increase of power transfer from the external field to the RCL circuit of the solenoid at resonance. A similar phenomenon occurs in optics when a monochromatic light beam incident upon a completely absorbing full disk produces a bright and tiny spot (Poisson-Arago spot) behind the disk, at the centre of the circular shadow region [4]. Another well-known effect is the sudden opacity developed by a cloud of micron-sized metallic particles when the frequency of the incident light coincides with the frequency of resonances due to plasmon oscillations on the particles' surfaces.

We also simulated the concentration of power flux around a small absorbing sphere placed in an electromagnetic field at resonance with the sphere's plasmon oscillations, as shown in Figure 7. The numerical calculation was done with finite elements using Comsol.

At the resonant frequencies, the target (solenoid, dark disk, particle) behaves like a larger object from the point of view of the total power flux geometric distribution [2], [8]. The wave nature of the surrounding fields is essential for explaining these power concentration effects since the wave-fronts are deformed to satisfy the constraints imposed by the condition of total absorption at the target. Put differently, the target is actually the source of a scattered wave field which interferes with the external field and produces convergent Poynting around the target.
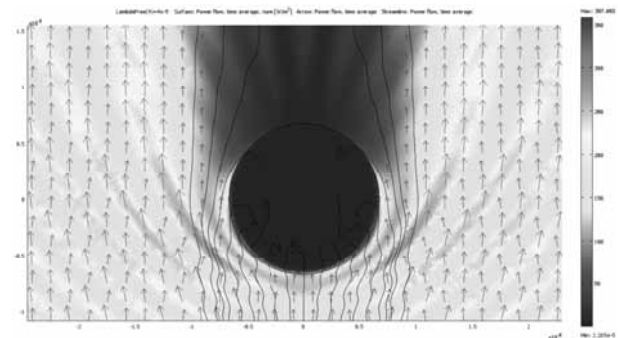

Figure 7: Strong absorption of an EM wave of wavelength $\lambda = 4\,[nm]$ by a spherical particle with a diameter of $15\,[nm]$: Black full lines represent Poynting flux, with norm indicated by the background colours. Arrows represent the Poynting vectors.
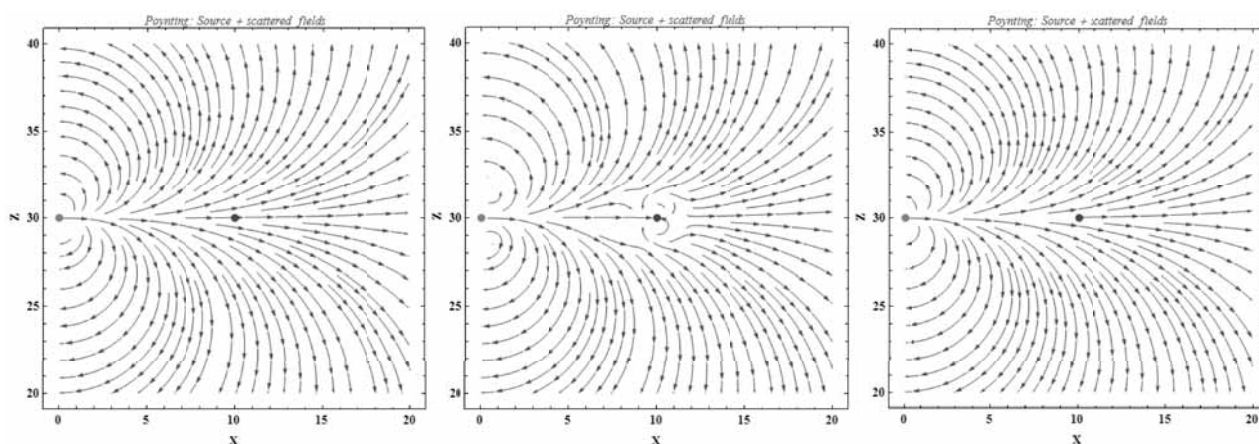
Figure 8: Total power flux lines for the sum of the source (left coil in each diagram) and target (right coil) Poynting vectors fields. The frequencies are $0.90\, f_{res}$ (left), f_{res} (centre) and $1.10\, f_{res}$ (right). We assume that the left coil (source) contains only the external current (i. e., no self- or mutually-induced currents). The power flow is then one-way and the resonance frequency of the pair of coils stays equal to fres, otherwise there would be resonance-splitting. The left (right) fat dot represents the source (target) coil.

5 Mutually Coupled Coils

If the resonating coil of section 3.2 is now placed inside a magnetic field created by a single coil with current oscillating at frequency f_{res} (as in section 2.1), it is reasonable to expect that the same type of concentration of power flux lines will occur around the resonating coil, as seen in section 3.2. This is confirmed by the present model as can be observed from the shape of the Poynting flux lines at $f = f_{res}$, in Figure 8, which form a sort of tube between the source at left and the target coil at right (middle of the central diagram).

In this calculation we used an unreasonably large distance ($10\,[m]$) between the source and the target coils to emphasize the presence of this power transfer mechanism even when that distance is large compared to the coils' dimensions. It should be stressed that the source-target distance is nevertheless smaller than the typical size of the *near-field* zone, $\frac{\lambda}{2\,\pi} \approx 50\,[m]$. In this zone, the fields are normally *evanescent* that is, their amplitudes decrease very fast with the distance from the source and they do not radiate.

If there is a region where the flux lines are tubular however, as happens at $f = f_{res}$, the power flux density decreases less rapidly in that region than what is expected to happen with evanescent, non-radiative fields. It is as if a limited region of radiative far-field was brought inside the near-field zone. These model results are in qualitative agreement with the literature on wireless power transfer already cited.

The role of the high quality factor Q of the oscillating circuits involved should now be clear, given the extreme sensitivity of the power-flux concentration effect with respect to the frequency f : Frequency shifts of 10% around the resonance f_{res} - and actually much less than that - completely wipe-out all traces of the Poynting field convergence effect. Resonant circuits with small values of Q won't have enough frequency stability to allow for the observation of this effect. The quality factor of our model resonating target coil is very high: $Q \approx \omega_{res}\frac{L}{R} \approx 3'620$.

The same concentration of flux lines occurs when the source and target coils are not placed in the same plane, as is shown in Figure 9, where the upper row diagrams correspond to axial coils and the lower row to a more general geometry, with the coils' axis remaining however parallel.

6 Applications

The idea of transferring electromagnetic power over long distances is an old one, going back to Tesla in the late nineteenth century. His ideas actually inspired the MIT team [7], [6]. Since the publication of these later works, a large number of publications appeared in parallel with various proposals for industrial applications. The literature on this subject is by now quite extensive. Some reviews can be found at [10] and on the Web [11], [5] , [9] (with many references and patent listing).
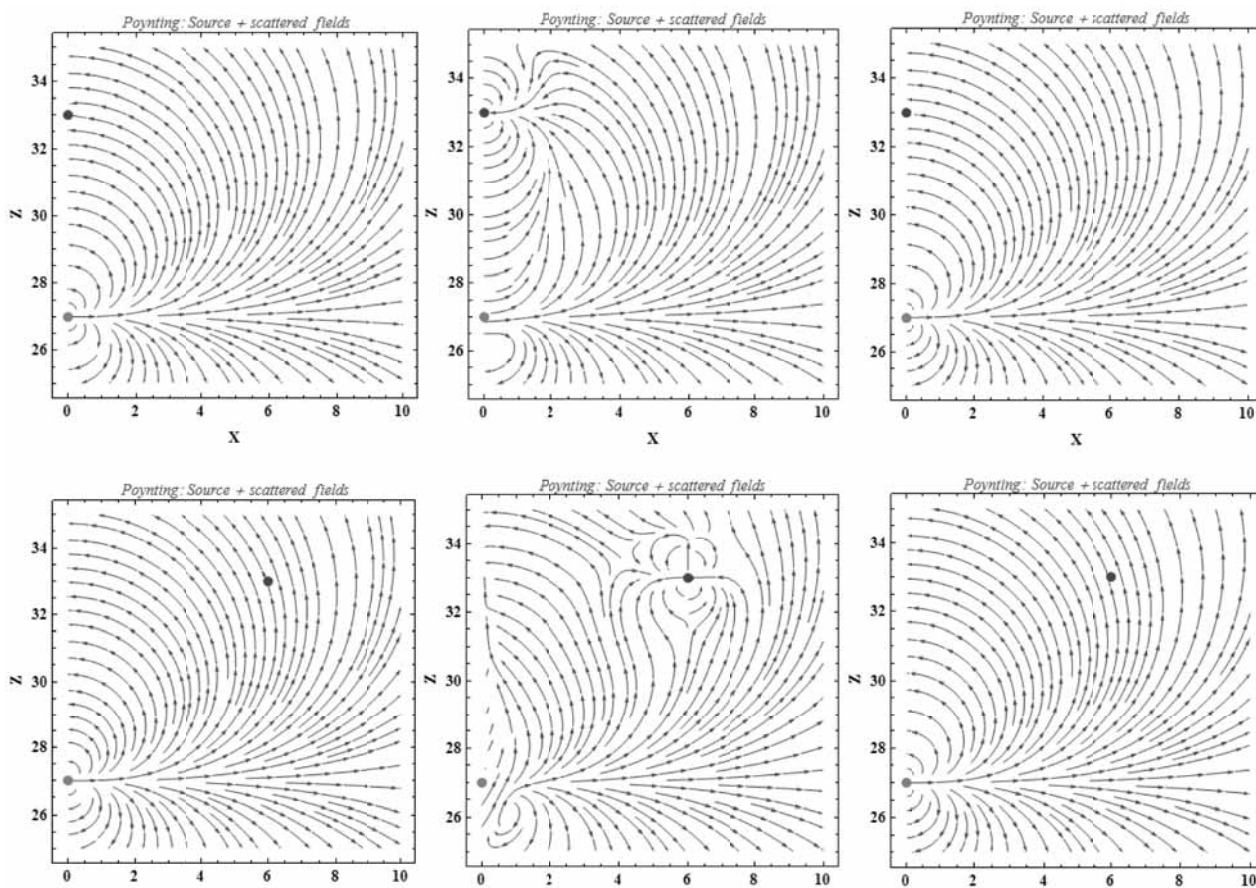
Figure 9: Generation of power-flux tubes under more general source / target geometries.

The same physical mechanism that underlies power transfer and which was modelled here could also be used in principle, for the transfer of information. In fact the RFID devices use the same principle of mutual induction between magnetically coupled oscillating circuits. There is however an important difference between RFID's functioning and the present schemes: RFIDs contain relatively low Q (quality-factor) oscillators, which makes them quite insensitive to small frequency changes between source and tar- get. For the phenomena discussed here, on the contrary, strong resonance (akin to total absorption) is critical to producing clear power flux concentration.

7 Conclusions

1. A clear transition was modelled and occurs in the spatial pattern of electromagnetic power flux when a solenoid is embedded in an harmonic magnetic field oscillation exactly at the frequency of resonance of the solenoid circuit. At this frequency,

the power flux lines crossing a region surrounding the solenoid tend to converge very strongly past the solenoid, in sharp contrast to what is observed away from resonance, where the flow patterns are only mildly affected by the solenoid oscillating circuit.

2. This phenomenon was studied with the help of an analytical model using the program *Mathematica*. In this model, the interactions between the different subsystems (coupling between fields and currents) was interpreted under the form of hierarchies of functions whose arguments were other functions.

3. Power concentration related to strong absorption was related to other well known phenomena found in different areas of Physics. A numerical simulation by finite elements, made using the program *Comsol*, numerically confirmed the power concentration effect. Finally, the basis of the effects was argued to be the wave nature of the fields involved.

References

[1] Balanis, CA. *Antenna Theory - Analysis and Design*. Edition 1. New York: Harper and Row; 1982. 816 p.

[2] Bohren CF. How can a particle absorb more than the light incident on it?. *Am J Phys*. 1983; 51(4):323–327. doi:10.1119/1.13262.

[3] Feng Y, Braaten B, Nelson R. Analytical expressions for small loop antennas - with application to emc and rfid systems. In IEEE. EMC 2006 Proceedings Volume 1. *2006 IEEE International Symposium on Electromagnetic Compatibility*; 2016 Aug; Portland, Oregon USA. 445 Hoes Lane, Piscataway, NJ 08855-1331 USA : IEEE. 63-68. doi:10.1109/ISEMC.2006.1706264.

[4] Gondran M, Gondran A. Energy flow lines and the spot of Poisson-Arago. *Am J Phys*. 2010; 78(6):598–602. doi:10.1119/1.3291215

[5] *Intel's Wireless Power Technology Demonstrated*. http://thefutureofthings.com/news/5763/intel-s-wireless-power-technology-demonstrated.html 2009.

[6] Karalis A, Joannopoulos JD, Soljacic M. Efficient wireless non-radiative mid-range energy transfer. *Annals of Physics*. 2008; 323:34–48. doi:10.1016/j.aop.2007.04.017

[7] Kurs A, Karalis A, Moffatt R, Joannopoulos JD, Fisher P, Soljacic M. Wireless power transfer via strongly coupled magnetic resonances. *Science*. 2007; 317:83. doi:10.1126/science.1143254

[8] Paul H, Fischer R. Comment on "how can a particle absorb more than the light incident on it?". *Am J Phys*. 1983; 51(4):327. doi:10.1119/1.13471

[9] PowerPedia. Wireless transmission of electricity. http://peswiki.com/index.php/Powerpedia:Wireless_transmission_of_electricity.

[10] van Bussel R, Franken j, Golchin S, Leijenaar R. Wireless Power Supply. MDP1, Final report (V. 2) 14-12-2007, 2007.

[11] Wireless Electricity. http://emergingtechnology.wordpress.com/2007/10/02/wireless-electricity. 2007.

A Numerical Approach to Investigate Mixed Friction Systems in the Micro-scale by means of the Coupled Eulerian Lagrangian Method

Albert Albers[*], Benoit Lorentz

Karlsruhe Institute of Technology, IPEK – Institute of Product Engineering, Campus Süd, Kaiserstr. 10, 76131 Karlsruhe, Deutschland; * *albert.albers@kit.edu*

Abstract. An approach for numerical investigation of mixed lubricated systems is presented in this article. By means of the Finite Element Method, a two dimensional model is built and composed of one fluid lubricating two sliding rough surfaces. The challenge of such a model resides in the complexity of interactions between the fluid and the solid structure. The used meshing method called Coupled-Eulerian-Lagrangian is utilized for high contact topology changes, a phenomenon occurring in case of large translation of rough surfaces in contact with another viscous body.
A model based on a two dimensional axial bush bearing is developed in order to evaluate the abilities of such an approach in calculating a contact pressure and the friction coefficient between both lubricated solids. The main friction coefficient is separated into solid-solid and fluid-solid friction part. The present approach gives the opportunity to identify the influence parameters on the tribological behaviour of mixed friction systems.

Introduction

In a current context of global warming, improvements in energy saving are highly demanded. Friction effects occurring in mechanical systems, responsible for up to 10% loss of the overall worldwide produced energy [1], need to be reduced. That would be possible if the knowledge of the different friction phenomena is increased. To improve the understanding of such phenomena, a numerical approach principally based on mathematical models is developed.

An advantage of such an approach in comparison with experimentations remains in a better modularity when systems become complex. Indeed, only the CAD data of the studied system are needed whereas experimentations require the building of a prototype which is more expensive. In the contrary to real experimentations; the numerical approach are better adapted for investigations at the microscopic scale necessary conditions for presented mixed lubricated systems analyses.

This paper aims at evaluating the potential of the most recent implemented coupling method of the Finite Elements (FE) code ABAQUS in modelling complex fluid structure interactions like in mixed lubricated systems. Analyses of are then provided to determine the influence of the different parameters on the friction behaviour. Results are compared to the literature with a view to validate the model. Finally the outgoing results are synthesized and an outlook of upcoming experimentations is given.

1 Numerical Model

1.1 Phenomenon

A friction system is submitted to mixed lubrication (see Figure 1) when the lubricating fluid is broken and some solid-solid contacts occur in addition to fluid-solid contacts. The occurring mixed lubrication configuration described as in [2] where R. Stribeck presented the relationship of the friction coefficient μ and the fluid film thickness $log\ h$ in function of the ratio $\eta V/F_N$ where F_N the normal applied load, V the sliding speed and η the lubricant viscosity.

Considering dynamic effects occurring in such system configurations, a transient analysis is required for numerical investigations. The movement of the asperities induces transient fluid flow changes affecting implying dynamic loads.
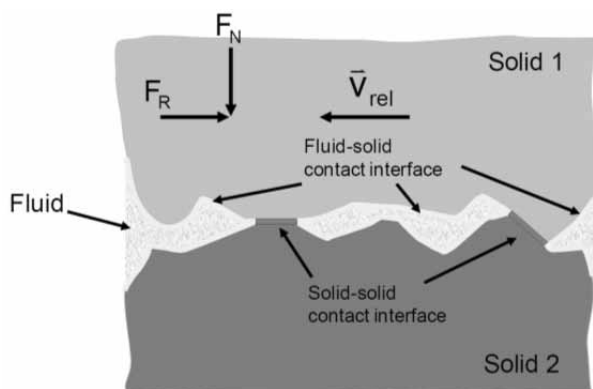
Figure 1: Mixed friction system configuration.

1.2 Coupling method and contact algorithm

The main challenge in this work is to handle with the large fluid mesh distortion provoked by the translation of the upper rough body. To cope with this problem, a remeshing technique, the Arbitrary-Lagrangian-Eulerian (ALE) technique of C.W. Hirt [3] is particularly interesting for such cases. Using this approach, L. Nowicki has modelled in his work [4, 5] Fluid-Structure-Interaction (FSI) with following limitations. When the fluid mesh becomes too small, it is necessary to actualise it manually. To overcome this encountered ALE limitations, the recent approach called Coupled-Eulerian-Lagrangian (CEL) method [6] and based on the volume of fluid technique (VOF) [7] is used for the present investigations.

In comparison with conventional FSI analyses, the fluid nodes are not attached onto the solid nodes. This concept allows for the first time the overlapping of a lagrangian and an eulerian mesh. In fact the material domain of the eulerian mesh is determined by using the free surface method. Based on the material ratio criterion the contact forces are only active if the surrounding material is of 50%. When this ratio is lower than this ratio, the fluid penetrates into the structure as shown in the edge on Figure 2.

To avoid any fluid intrusion it is necessary to use round edges and adapt the meshing refinement but this phenomenon cannot always be avoided.

1.3 Material properties and geometry

Using the commercial FE code ABAQUS 6.9-1 [8] a model is made in two dimensions for a question of computing time. The usual lagrangian meshing method is used for the solids whereas the eulerian method is applied to the fluid domain. The Mie-Grüneisen equation of state [9] based on the Hugoniot linear relationship between the particle velocity U_p and the shock velocity U_s is employed to model the fluid.
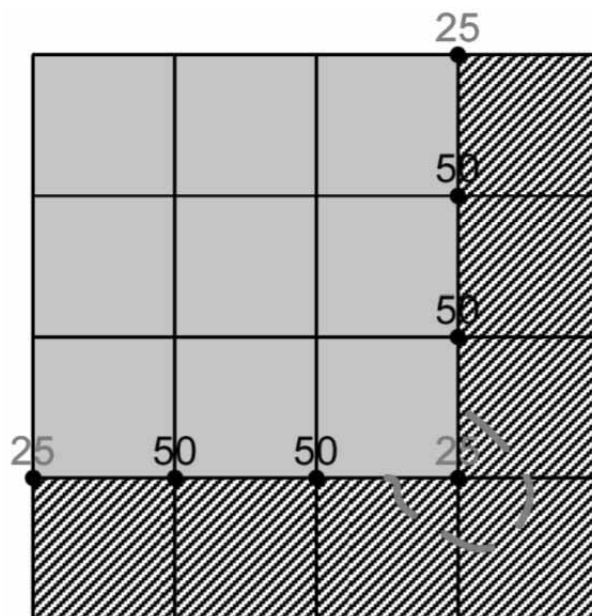
Figure 2: Contact conditions between solid and fluid.

The approximation of constant viscosity in function of the pressure is made because of low hydrodynamic pressure [10]. According to the hypothesis of incompressible fluid the parameters s and Γ are set to 0 which implies that the shock wave velocity is independent from pressure (see Table 1).

For the solid structure, a conventional structural steel is used where plasticization is also taken into account. The developed model is based on two rough simple cast surfaces having a roughness R_a calculated using definition [11].
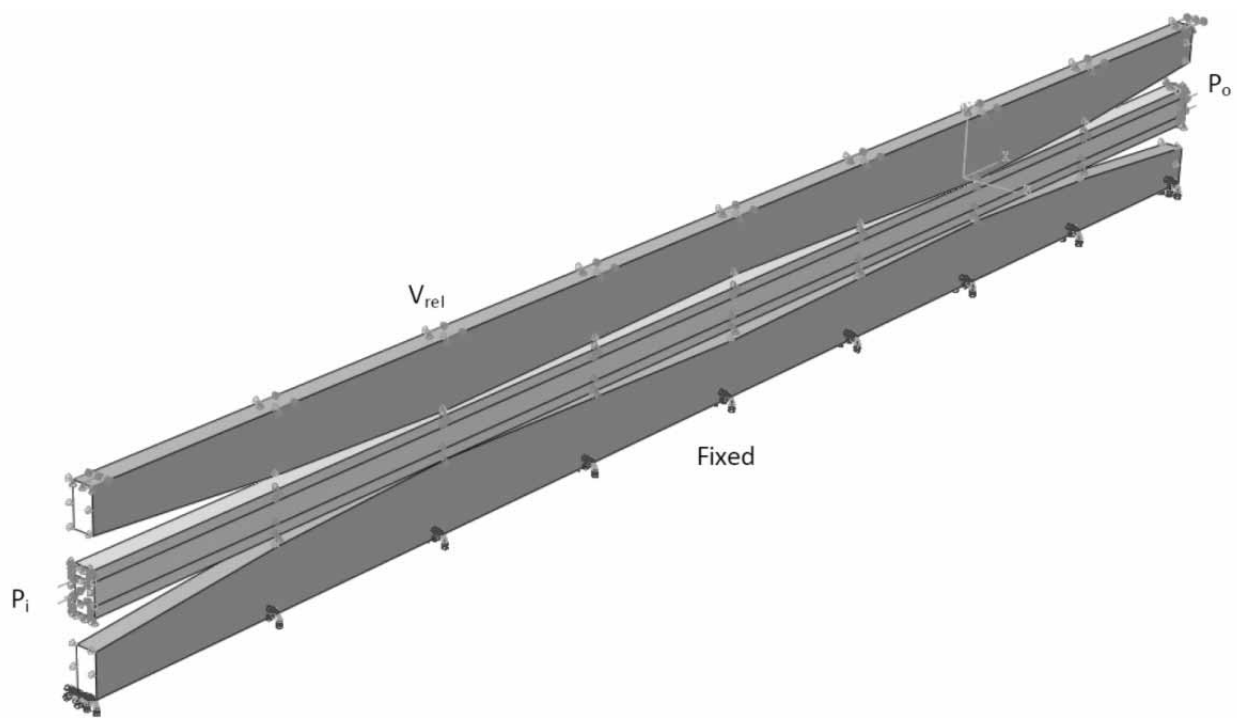
Figure 3: Model with boundary conditions.

Symbol fluid	Quantity	Value	Symbol solid	Quantity	Value
ρ	density	880 kg/m^3	ρ	density	7800 kg/m^3
η	dynamic viscosity	0.088 Pa.s	E	Young coefficient	210.10^9 Pa
c_0	sound velocity	2135 m/s	v	Poission coefficient	0.33
Γ	Grüneisen ratio	0	R_e	Yield stress	234.10^6 Pa
s	slope of the $U_s - U_p$ curve	0	ε_e	Yield strain	0.18

Table 1: Fluid structure parameters (on the left) and solid structure parameters (on the right)

1.4 Model configuration and simulation process

The two dimensional model has one rank of elements in the Z direction (see Figure 3). The used boundary conditions are applied in order to fit to the reality. Regarding the objectives, a two dimensional model will not be able to simulate a complete mixed lubricated problems but only a combination of fluid-solid and solid-solid interaction. The aim is to analyze the plausibility of the resulting output in comparison with the real effects.

As the model is a section of a three dimensional one, X and Y rotations, as well as the displacements in the normal direction to the section, represented by the Z axis in Figure 3, are forbidden. The lower body is fixed on the ground at its lower surface. Concerning the fluid, there is the need to avoid fluid loss in the X direction what is also the case in the Y direction whether some fluid parts penetrate into the solid body.

After setting preceding boundary conditions, it is necessary to follow a global procedure in order to make a parametrical study. Investigations are only relevant when the model is working in quasi-static conditions that means when a constant friction velocity is reached. To do that, a rigorous procedure needs to been chosen. In initial phase an inlet pressure P_i of $110\ MPa$ at the inlet and an outlet pressure P_0 of $109.99\ MPa$ are applied. In order to vary the fluid film thickness a second step is needed for the setting vertical position. Displacements are imposed to the whole bodies because of present high accelerations. After positioning phase, a new step has to initiate the wanted sliding velocity on the upper body. Once the fluid contact is stabilized again, a final step is set to make the relevant quasi-static investigations where the sliding velocity remains constant. When mixed friction is initiated, the solid-solid contact coefficient is of 0.15 is admitted according to the literature.

Due to the high nonlinearity coming from the fluid modelling, the explicit solver is needed. However its computational costs highly depend on the element size according to the stable increment criterion [12] which is depending principally of the characteristic length associated with the elements. This limitation is essential for the simulation duration because the model is made at the micro scale where the global element size is in the order of $0.05\ mm$.

2 Results

Many different investigations where realized in order to analyze the frictional behaviour ofthe system. Three parameters where varied in order to know each of their influence.

2.1 Variation of the film thickness in hydrodynamic lubrication

The main analyses conditions are the same as for the previous one. The friction coefficient is decreasing when the lubricant film becomes thicker. This is explained by the local pressure which decreases when the flow section becomes bigger.

This trend is followed with the three different roughness profiles. A rule could not been defined in order to get the real influence of the roughness.

Figure 4: Influence of the film thickness.

Figure 6: Contact pressure (MPa) in the mixed lubricated system

2.2 Noise effects appearing in using the CEL method

Some noise effects such as vibrations that are a combination of elastic behaviour and fluid vibrations coming from the pressure application (see Figure 5). The three signals are analyzed with the Fast Fourier Transformation (FFT) showing that all signals have the same main frequency of 0.2 peaks pro frame which means a peak each 5 frames. That shows that the frequency is independent of the fluid shock wave because the solid velocity is not influencing this noise. This effect was known by the developers but can only be avoided by using shorter simulation time than the time needed by the wave to reflect against the fluid boundaries.

On Figure 5 oscillations are observed, these ones correspond to the reflecting oscillations of the fluid against both sides where a pressure load is applied.
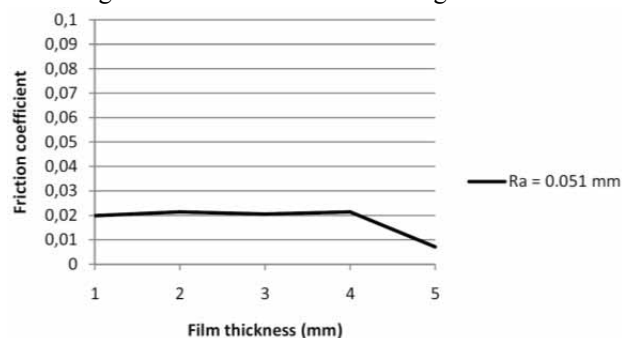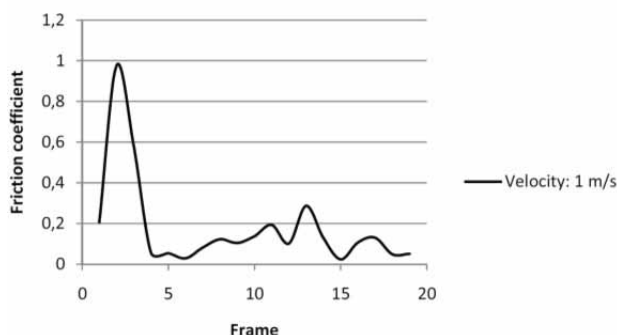

Figure 5: Application of the translation velocity.

2.3 Mixed friction model

The previous presented investigations were made in order to have a basis to investigate more complex system behaviour. In this work the case of mixed friction systems are investigated. Figure 6 shows how the contact pressure is distributed in the contact. The contact pressure goes until $1.1\ GPa$ what is usual for solid-solid contact occurring in mixed friction contacts.

In such systems the solid friction can be separated from the fluid friction into two coefficient μ_s for the solid friction and μ_f the fluid friction. The global coefficient is calculated by the FE solver and its value is of 0.2. The main friction coefficient can be separated into two terms [13].

The adhesion effects issuing from the interaction between the fluid and the structure are calculated with a post processing treatment. The hydrodynamic friction, which is coming from the shear stress of the fluid, is directly taken into account by the software. After post-processing treatment the value of 0.039 as hydrodynamic friction is established which is in accordance with the literature.

3 Conclusion and Outlook

Finally, the new CEL approach offers many possibilities to simulate interactions between fluid and solids. This procedure avoids the difficulty of remeshing fluid structures when contact topologies get too large changes, what is impossible with other methods but with lower precision. Nevertheless, it has to be noticed that the method needs many CPU resources.

The hypothesis to use this method is valid because the outgoing results are very similar to those found in the literature. This method enables investigations of micro structures in order to calculate the local friction coefficient and to understand the phenomena at this scale. The next step will be to develop a method which investigates the phenomena in real three dimensions and enables a design of experiment for determining most influencing parameters.

Then, to complete this method, adhesion models have to be implemented to take into account the adhesion effects between both solids. One last aspect to consider in the modelling is to eliminate and overcome the noise coming from the numerical method.

References

[1] ASME. *Strategy for energy conservation through tribology*. 1977.

[2] Stribeck R. Die wesentlichen Eigenschaften der Gleit- und Rollenlager (The basic properties of sliding and rollingbearings). *Zeitschrift des Vereins Deutscher Ingenieure*. 2002; 36(46) 1341-1348, 1432-1438, 1463-1470.

[3] Hirt CW. *An arbitrary Lagrangian-Eulerian computing technique*. Los Alamos Scientific Laboratory, University of California.

[4] Albers A, Nowicki L, Enkler H-G. Development of a method for the analysis of mixed friction Problems. *International Journal of Applied Mechanics and Engineering*. 2006; 11(3), 479-490.

[5] Albers A, Nowicki L, Enkler H-G. Methode zur Berechnung von geschmierten Friktionsproblemen in Mischreibungsgebieten. *GFT*. 2006.

[6] Van Loon R, Anderson PD, Van de Voss FN, Sherwin SJ. Comparison of various fluid–structure interaction methods for deformable bodies. *Computers and Structures*. 2007; 85, 833-843.

[7] Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*. 1981; 39.

[8] ABAQUS. *Analysis User's Manual, Materials 21.2.1 Equation of state*. Version 6.9; 2009.

[9] Singh RN, George AK, Arafin S. Specific heat ratio, Grüneisen parameter and Debye temperature of crude oil. *Journal of Physics D: Applied Physics*. 2006; 39, 1220-1225.

[10] Gras R. *Tribologie: Principes et solutions industrielles*. Dunod; 2008. 321 p.

[11] NF EN ISO 4287-4288

[12] ABAQUS. *Analysis User's Manual, Analysis Procedure – 6.3.3 Explicit Dynamic analysis*. Version 6.9; 2009.

[13] Nowicki L. *Raue Oberflächen in geschmierten Tribokontakten* [dissertation]. [Institute of Product Development, (G)]. University of Karlsruhe; 2008.

An Investigation on
Loose Coupling Co-Simulation with the BCVTB

Irene Hafner[1*], Bernhard Heinzl[1,2], Matthias Rössler[1]

[1]Simulation Services, dwh GmbH, Neustiftgasse 57-59, 1070 Vienna, Austria; *irene.hafner@dwh.at

[2]Institute for Analysis and Scientific Computing, Vienna University of Technology, Wiedner Haupstraße 8-10, 1040 Vienna, Austria

Abstract. This paper introduces several methods of co-operative simulation. Apart from the general classification and method descriptions, the numerical stability and consistency of one loose coupling approach is discussed. It is shown that consistency is maintained, although possibly of lower order, and zero stability persists as long as no algebraic dependencies between partial systems occur. The methodology of Jacobi-Type loose coupling is applied for a case study using the co-simulation tool BCVTB (see [1]). The study shows that this tool is well suited for the fast co-simulation of many instances of certain simulators, but allows no synchronisation step size control and only equidistant synchronization step sizes.

Introduction

In times of increasing environmental awareness, the prediction of energy and resource consumption is becoming more and more important. A very important auxiliary means for this issue is computer-aided mathematical simulation. Since the simulation of whole production halls including building geometry, machinery, control systems and building services needs detailed modelling of all parts, where every system requires an individual modelling approach, the method of cooperative simulation needs to be considered. Co-simulation allows the overall simulation of complex systems consisting of partial systems requiring different modelling approaches, solver step sizes or even solver algorithms.

1 Co-Simulation Types

In general, co-simulation methods are divided into two types, depending on whether data exchange takes place iteratively in every time step or only at specified synchronisation time steps.

1.1 Loose coupling

Simulations coupled via loose coupling exchange data only at certain points in time. These synchronisation references do not have to be predefined or equidistant, but hereafter only co-simulation methods at fixed, equidistant times are considered for reasons given in section 4.2. In the following a system of two partial systems depending on each other is given.

$$\dot{x_1} = f_1(x_1, y_2) \tag{1}$$
$$y_1 = g_1(x_1, y_2) \tag{2}$$
$$\dot{x_2} = f_2(x_2, y_1) \tag{3}$$
$$y_2 = g_2(x_2, y_1) \tag{4}$$

Equations (1) and (2) describe System 1 and equations (3) and (4) describe System 2. $y_i, i \in \{1, 2\}$ are required in the respective other system to calculate the internal state variables. These values are synchronised at given points in time. Depending on the synchronisation order, two methods of loose coupling are distinguished.

Gauß-Seidl type. At the start time of the simulation, the initial values for all variables are exchanged. For each following synchronisation reference, the data exchange follows the procedure shown in Figure 1. Between two synchronization references, without loss of generality in System 1 the values of y_2 are extrapolated and the states of the internal variables are calculated at the individual time steps defined by the solver for System 1.
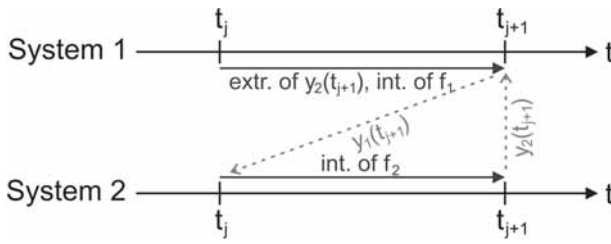
Figure 1: Overview of loose coupling co-simulation with the Gauß-Seidl type between two synchronisation references

As soon as t_{j+1} is reached, the values of y_1 until t_{j+1} are transferred to System 2, so the values needed at the time steps demanded by the solver for System 2 between t_j and t_{j+1} can be interpolated instead of extrapolated. At t_{j+1}, the values for y_2 are again transferred to System 1.

Jacobi type. This method allows all systems to calculate in parallel between two synchronisation references, which also means that each partial system has to extrapolate the values needed from the other systems. For the given example, values of of y_1 and y_2 are exchanged simultaneously at each synchronisation reference and extrapolated until the next data exchange takes place. A sketch of the method is illustrated in Figure 2.
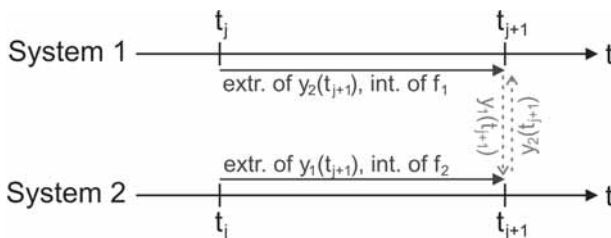

Figure 2: Overview of loose coupling co-simulation with the Jacobi type between two synchronisation references

1.2 Strong coupling

Co-simulation methods using strong coupling iterate the values needed from other partial systems in every time step until a specified accuracy is achieved. This approach obviously leads to far more accurate results but also boosts computing times.

2 Numerical Background

Coupling the simulation of two or more equation systems of course influences the behaviour of the underlying solver algorithms. Hence it is very important to investigate these effects to be able to determine consequences regarding numerical stability.

2.1 Consistency

For the analysis of consistency with co-simulation, let a multi-step method for solving ordinary differential equations be given as in (5):

$$\sum_{j=0}^{k} \alpha_{k-j} y_{i+1-j} = \Delta t \cdot \Phi_f(t_{i+1-j}, y_{i+1-j}, \Delta t), \quad (5)$$

where Φ stands for the increment function of the method and Δt for the (equidistant) step size between t_i and t_{i+1}, $i = 1, \ldots n$.

Consistency defines the method's error per step. The consistency error is defined by

$$\tau^{i+k}(\Delta t) := \sum_{j=0}^{k} \alpha_{k-j} y(t_{i+1-j}) - \Delta t \cdot \Phi_f(t_{i+1-j}, y(t_{i+1-j}), \Delta t).$$

$$(6)$$

A method is called consistent if

$$\lim_{\Delta t \to 0} \left(\frac{\tau^{i+k}(\Delta t)}{\Delta t} \right) = 0 \quad (7)$$

is fulfilled for arbitrary initial values. A method is called consistent of order p if there exists a constant $C > 0$ so that

$$\left\| \tau^{i+k}(\Delta t) \right\| \leq C \cdot (\Delta t)^{p+1} \quad (8)$$

To discuss consistency of a loose coupling co-simulation method, a linear one-step method is considered, so (5) becomes

$$\alpha_0 y_i + \alpha_1 y_{i+1} = \Delta t \left(\beta_0 f(t_i, y_i) + \beta_1 f(t_{i+1}, y_{i+1}) \right) \quad (9)$$

The consistency error of this method is calculated by

$$\tau^{i+1}(\Delta t) = \alpha_0 y(t_i) + \alpha_1 y(t_{i+1}) - \Delta t \left(\beta_0 f(t_i, y(t_i)) + \beta_1 f(t_{i+1}, y(t_{i+1})) \right). \quad (10)$$

In the case of a co-simulation, the values for $y(t_{i+1})$ are needed from another partial system and hence are not known but extrapolated during one time step.

Let $y_c(t_{i+1})$ be the extrapolated value. Therefore follows the consistency error

$$
\begin{aligned}
\tau_c^{i+1}(\Delta t) =& \alpha_0 y(t_i) + \alpha_1 y(t_{i+1}) \\
& - \Delta t \left(\beta_0 f(t_i, y(t_i)) + \beta_1 f(t_{i+1}, y_c(t_{i+1})) \right) \\
=& \alpha_0 y(t_i) + \alpha_1 y(t_{i+1}) \\
& - \Delta t (\beta_0 f(t_i, y(t_i)) + \beta_1 f(t_{i+1}, y(t_{i+1})) \\
& - \beta_1 f(t_{i+1}, y(t_{i+1})) + \beta_1 f(t_{i+1}, y_c(t_{i+1}))) \\
=& \tau^{i+1}(\Delta t) \\
& + \Delta t \cdot \beta_1 \cdot (f(t_{i+1}, y(t_{i+1})) - f(t_{i+1}, y_c(t_{i+1})))
\end{aligned}
$$

in the co-simulation. To determine consistency, we consider

$$
\begin{aligned}
\left\| \frac{\tau_c^{i+k}(\Delta t)}{\Delta t} \right\| =& \\
& \left\| \frac{\tau^{i+k}(\Delta t)}{\Delta t} + \beta_1 \cdot (f(t_{i+1}, y(t_{i+1})) - f(t_{i+1}, y_c(t_{i+1}))) \right\| \\
\leq& \left\| \frac{\tau^{i+k}(\Delta t)}{\Delta t} \right\| \\
& + |\beta_1| \cdot \| f(t_{i+1}, y(t_{i+1})) - f(t_{i+1}, y_c(t_{i+1})) \| \\
\leq& \left\| \frac{\tau^{i+k}(\Delta t)}{\Delta t} \right\| + L \cdot |\beta_1| \cdot \| y(t_{i+1}) - y_c(t_{i+1}) \|
\end{aligned}
$$

where the Lipschitz continuity of f with Lipschitz constant L conditions the last inequality. If the most simple extrapolation, i.e. taking the last known value, $y(t_i)$, for $y_c(t_{i+1})$, is applied and the Taylor approximations for $y(t_{i+1})$ and $y(t_i)$ around $t_i + \alpha \Delta t$ for an arbitrary $\alpha \in (0,1)$ are considered, the constant terms cancel each other out in the subtraction. Thus follows

$$
\left\| \frac{\tau_c^{i+k}(\Delta t)}{\Delta t} \right\| \leq \left\| \frac{\tau^{i+k}(\Delta t)}{\Delta t} \right\| + L|\beta_1| \cdot O(\Delta t). \quad (11)
$$

Hence for a method of consistency order 1 the consistency order is maintained in a co-simulation, for methods of higher order consistency is maintained but of lower order.

2.2 Zero stability

A zero stabile numerical method yields a bounded solution of $y(t) = 0$ for arbitrary initial conditions. To determine zero stability for a method, the first characteristic polynomial is needed. For a given multi-step method (see (5)), the first characteristic polynomial is given in (12):

$$
\rho(\zeta) := \sum_{j=0}^{k} \alpha_j \zeta^j \quad (12)
$$

A method is called zero stabile if every zero λ of the first characteristic polynomial fulfils $|\alpha| < 1$ and every zero with $|\alpha| = 1$ is a single zero.

In the following zero stability of a linear one-step method in a loosely coupled co-simulation is considered. Regarding (9), we see that the characteristic polynomial depends solely on the equation's left side. Co-simulation inflicts changes only on the calculation of f when using $y(t_{i+1})$, so only the right side is being affected. This means that co-simulation does not influence zero stability as long as only ordinary differential equation systems are considered. For differential-algebraic equation systems, [2] states that zero stability is maintained as long as no algebraic interdependencies between the partial systems occur.

3 Co-Simulation with the BCVTB

The Building Controls Virtual Test Bed (BCVTB) is a co-simulation tool which has been developed at the University of California, Berkeley and allows co-simulation of the building simulation software Energy-Plus [6], simulators of the object-oriented standard Modelica [4], MATLAB [3] and its toolboxes Simulink and Simscape, Radiance [7] and Functional Mockup Interfaces [5].

BCVTB is based on Ptolemy and provides on the one hand so-called *simulator actors* which are part of the BCVTB environment and define the simulators and corresponding source files to be co-simulated. On the other hand, functions or function blocks for the communication with BCVTB are provided for each participating simulator. The communication itself takes place via so-called BSD sockets, which have also been developed at the University of California for inter-process communication. BCVTB allows only loose coupling co-simulation of Jacobi type with predefined, equidistant synchronisation references.

4 Case Studies

4.1 Production hall in Energy Plus

In the following case study, the model of a production hall is co-simulated with the BCVTB. The machines located in the different halls emit heat which has to be transferred to the respective rooms. Figure 3 shows the sketch of the building model to be simulated.
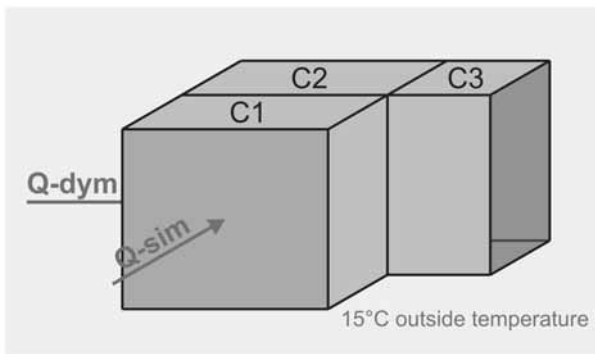

Figure 3: Overview of the building to be simulated.

The building itself is modelled in EnergyPlus, the machines are implemented in Simscape and Dymola respectively.   A sketch of the communication via BCVTB is shown in Figure 4. The solvers used for the simulation of the individual models can be found in the corresponding brackets.
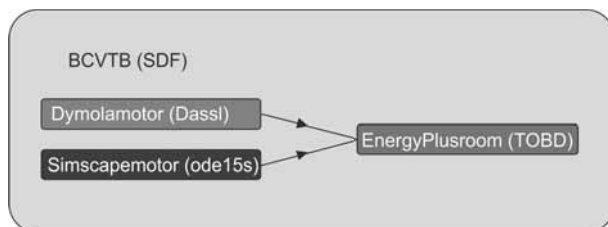
Figure 4: Overview of the intended communication between the individual simulators via BCVTB.

Both machines are switched on at 8 a.m.   in the morning and switched off at 4 p.m. in the afternoon. The temperature of the environment of the hall is defined in a weather data sheet which is needed as an input to the EnergyPlus model. Figure 5 shows the heat emission of both machines and Figure 6 shows the room temperature in all rooms over one week responding to the heat insertion.
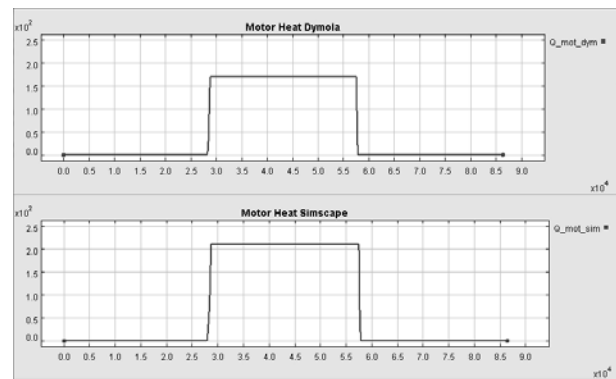
Figure 5: Heat loss of the machine models.
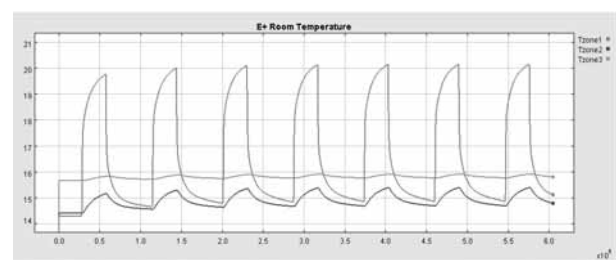
Figure 6: Room temperature over one week.

Regarding Figure 6 it becomes clear that the room temperature reacts clearly to the heat loss of the machines but still manages to cool down without further ado (like HVAC systems controlling the temperature). It is also evident that the room containing the machines (C1) reacts to the emission with much faster temperature increase than the rooms without a machine. Furthermore, since room C3 contains a window, its temperature is also able to cool down rather rapidly. Between two synchronisation references, every simulation uses the time steps given by its own solver and at each reference the data needed from other partial systems is exchanged. This procedure is shown in Figure 7.
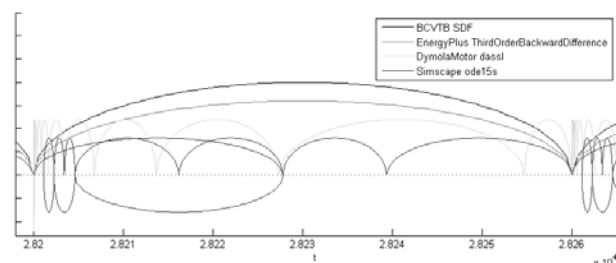
Figure 7: Steps taken by the individual solvers between two synchronisation references.

It can be observed that the solvers for the machine models need to discard steps after the data exchange since they are intended too large in the first place. In Energy-Plus the overall step size is also defined for the internal calculation as no major changes take place internally and furthermore EnergyPlus fires data exchange at every internal step, so the co-simulation step size has to be the very same as the step size for EnergyPlus.

In Simscape and Dymola it is possible to calculate internal states at additional points in time after minor modifications of the predefined communication devices. To keep the room temperature in an interval convenient for human beings even in high summer or in a well insulated hall, a control (e.g. in Simulink) can be added to the model but since this requires a feedback (see Figure 8), a loop has to be broken by the insertion of a time lag of one step into the control model.
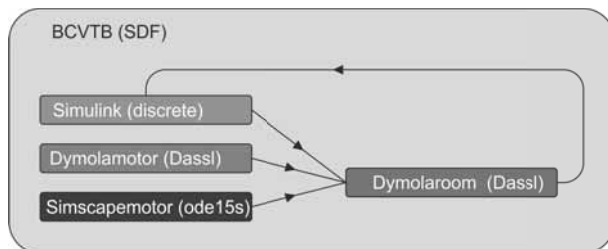

Figure 8: Overview of the communication via BCVTB with the inclusion of temperature control.

4.2 Experiment with step size control

Since BCVTB in general offers also a *continuous director* meaning an ODE solver with step size control for the overall simulation, this possibility has also been tried out. The basic model for this experiment was the model of a motor implemented in Simscape which transfers its waste heat to BCVTB where it is simply plotted. The BCVTB model for this scenario is given in Figure 9.
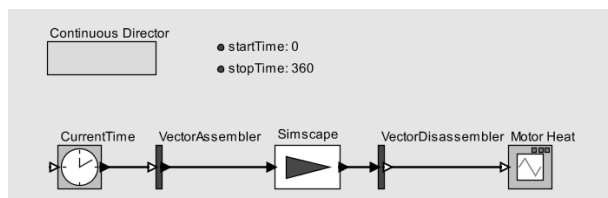
Figure 9: BCVTB model with only one partial model, a machine implemented in Simscape.
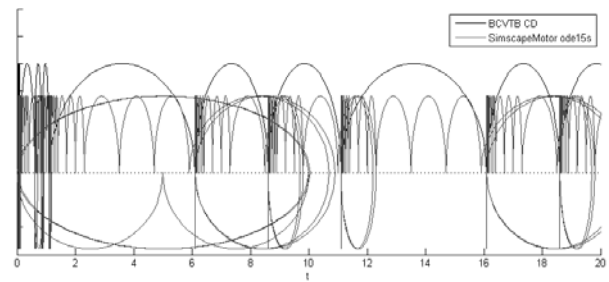
Figure 10: Steps taken by the *continuous director* of BCVTB and ode 15s in Simscape at the beginning of the simulation.

So far, everything works fine. However, the use of a continuous solver does not make much sense for this system, since nothing actually takes place in the BCVTB model. This can also be seen by inspecting the steps taken by the *continuous director*, which show periodic behaviour after a few smaller steps at the beginning, which are shown in Figure 10. Even after the beginning of the heat emission of the machine at $t = 200s$ the *continuous director* maintains its periodic behaviour of taking one bigger and two smaller steps consequently (see Figure 11).
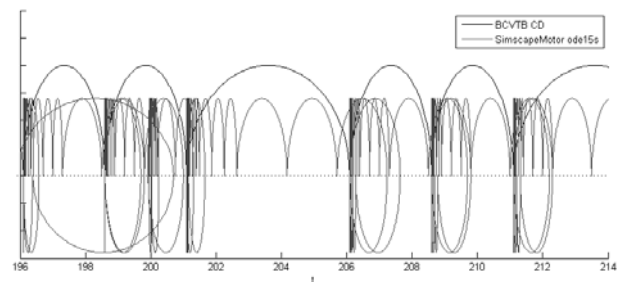
Figure 11: Steps taken by the *continuous director* of BCVTB and ode15s in Simscape at the beginning of the motor heat emission.

Furthermore the Simscape solver has to make many redundant steps since the time for synchronization has to be iterated.

If an integrator is inserted to calculate the energy consumption of the machine (see Figure 12), which justifies the usage of step size control, a somehow predictable performance occurs: up to switching the motor on, the same behaviour as in the model without integrator can be observed. As soon as the motor starts to emit heat, the BCVTB solver realizes that the output of the Simscape *simulator actor* has changed gravely and so it
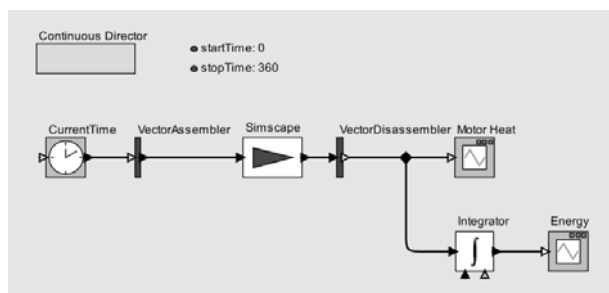
Figure 12: BCVTB model with an integrator determining the energy discharged by a machine.

wants to step back to iterate the time of change more precisely. Simscape in the meantime does not know BCVTB has stepped back and hence does not step back itself but waits for BCVTB to fire the next synchronization event – which can never occur in the future since it already happens in the past. This, however, means that BCVTB waits for Simscape to respond to its call and Simscape never does since it does not look back, so both simulators would wait forever for each other. This quite simple experiment proves that the use of a *continuous director* for the overall simulation is not suited for co-simulation with BCVTB as either the model has to be simple enough that the solver does never need to step back, which means a fixed step solver would be just as well or even better suited, or a deadlock would occur if the overall solver is obliged to discard steps.

5 Conclusion

This article shows that in a loose coupling co-simulation consistency is maintained but possibly of lower order and zero stability can be held with restrictions to algebraic dependencies. Apart from the numerical analysis, the possibilities and limits of co-simulation with the BCVTB have been investigated with respect to the number of participating simulators, diversity of models and methods of numerical solution approaches. Related work has shown that BCVTB allows the co-simulation of many instances of several simulators in a quick and rather easy way, but as described in this paper BCVTB covers only loose coupling co-simulation at equidistant points in time since step size control leads to a deadlock, see section 4.2. Further research will aim the investigation of additional co-simulation and multi-rate simulation methods, consideration of numerical issues and testing of other given co-simulation tools with regard to generality and provided coupling methods.

References

[1] Wetter M. *Co-Simulation of Building Energy and Control Systems with the Building Controls Virtual Test Bed*. Journal of Building Performance Simulation. 2011; 4(3):185-203. doi: 10.1080/19401493.2010.518631.

[2] Kübler R, Schielen W. Two methods of simulator coupling. *Mathematical and Computer Modelling of Dynamical Systems*. 2000; 6(2): 93-113. doi: 10.1076/1387-3954(200006)6:2;1-M;FT093.

[3] mathworks: Products and Services [Internet]. The MathWorks, Inc.: c1994-2014 [cited 2014 Nov 13]. Available from: `http://uk.mathworks.com/products/index.html?s_tid=gn_loc_drop`

[4] modelica: Modelica and the Modelica Association [Internet]. Modelica Association c2000-2014 [cited 2014 Nov 13]. Available from: `https://www.modelica.org/`

[5] fmi-standard: Functional Mock-up Interface [Internet]. Modelica Association c2000-2014 [cited 2014 Nov 13]. Available from: `https://www.fmi-standard.org/`

[6] eere.energy: EnergyPlus Energy Simulation Software [Internet]. U.S. Department of Energy [cited 2014 Nov 13]. Available from: `http://apps1.eere.energy.gov/buildings/energyplus/energyplus_documentation.cfm`

[7] radsite: Radiance [Internet]. Building Technology and Urban Systems Department c2014 [cited 2014 Nov 13]. Available from: `http://radsite.lbl.gov/radiance/`

[8] Trčka M. *Co-simulation for Performance Prediction of Innovative Integrated Mechanical Energy Systems in Buildings* [dissertation]. [Faculty of Architecture, Building and Planning, (NL)]. Technische Universiteit Eindhoven; 2008.

[9] Hafner I. *Möglichkeiten der Co-Simulation mit dem Building Controls Virtual Test Bed für den Bereich der objektorientierten Modellbildung physikalischer Systeme* [diploma thesis]. [Institute of Analysis and Scientific Computing, (AT)]. Vienna University of Technology; 2013.

A Globally-Implicit Computational Framework for Physics-based Simulation of Coupled Thermo-Hydro-Mechanical Problems: Application to Sustainability of Geothermal Reservoirs

Robert Podgorney[1], Hai Huang, Mitch Plummer, Derek Gaston

[1] Idaho National Laboratory, 2525 Fremont Ave, Idaho Falls, ID 83415

Abstract. This paper highlights the development of a fully-coupled and fully-implicit modeling tool for predicting the dynamics of fluid flow, heat transport, and rock deformation using a GIA named FALCON (Fracturing And Liquid CONvection). The code is developed on a parallel Multiphysics Object Oriented Simulation Environment (MOOSE) computational framework developed at Idaho National Laboratory (INL) for providing finite element solutions of coupled system of nonlinear partial differential equations. In this paper, a brief overview of the governing equations numerical approach are discussed, and an example simulation of strongly coupled geothermal reservoir behavior is presented.

Introduction

Numerical modeling has played an important role in understanding the behavior of geothermal systems since as early as the 1970s. While capabilities of geothermal reservoir simulators have grown since then, the prospect of simulating more challenging classes of geothermal problems–such as reservoir creation and operation of engineered geothermal systems (EGS), high enthalpy supercritical magmatic systems, etc–pose additional, and very significant, computational challenges that the current generation of continuum or dual-continuum hydrothermal models are ill-equipped to describe.

Interest in multiphysics simulation techniques is growing rapidly with a focus on more realistic and higher fidelity analysis of geothermal and engineering systems. This in-crease in activity is typically attributed to increasing computer power and more robust computational schemes [9, 11, 12], but in truth, advanced numerical methods are playing an equal role. The phrase 'multiphysics simulation' is used to describe analyses which include disparate physical phenomena–such as coupled multiphase, multicomponent fluid flow, enthalpy transport, and geomechanics and their feedbacks – are examined in a simultaneous manner. Examples of multiphysics problems in subsurface energy applications are numerous and include pressure and temperature driven permeability creation and evolution in geothermal reservoirs, temperature driven phase evolution of in-situ kerogen processing of oil shale reservoirs, kinetically controlled reactive transport in the flow of contaminants, etc. In addition to multiphysics coupling, most of these problems also have multiscale issues to resolve.

Examining coupled physics for fluid flow, energy transport, and geomechanical deformation is a relatively new area for the geothermal community; however, simulating coupled problems has been an important topic of study in the reactive transport community for decades. Yeh and Tripathi [21] and Steefel and MacQuarrie [16] cite three major approaches that differ in the way coupling transport and reaction have been considered for reactive transport modeling: (1) GIA (fully-coupled) approach that solves all governing nonlinear equations simultaneously at each time step using various forms of Newton's method, (2) sequential iteration approach

(SIA) that subdivide the reactive transport problem into transport and reaction subproblems, solves them sequentially, and then iterates, and (3) sequential non-iteration approach (SNIA) that solves the transport and reaction problems sequentially without iteration, which is often referred as operator-splitting. The operator-splitting approach is perhaps the simplest to implement and requires the least computational resources in terms of the memory and CPU time; thus, it became the method of choice for subsurface reactive transport modeling during the past three decades.

However, the drawback of the operator-splitting approach is the splitting error when the physics (either reactions-transport or flow-mechanics) are tightly coupled; the solution becomes inaccurate and requires very small time steps [17]. For most potential EGS reservoirs fluid flow, heat transport, and rock deformation will be strongly nonlinearly coupled. The changes in flow and energy transport properties due to fracturing and/or dissolution add further complexity and nonlinearity to the problem. For such situations, the global implicit approach (GIA) solves all solution variables simultaneously during each time step by seeking the solution of a large system of nonlinear equations via some form of Newton's method and is a more robust solution than the other two approaches [3, 8, 14].

One potential limitation of the GIA approach is the need to compute, store and invert the Jacobian matrix. This could become problematic for large systems which would be expected for reservoir-scale geothermal problems. As the number of solution variables grows, the matrix holding the Jacobian entries also grows. The increased size of the Jacobian matrix results in greater memory usage and more CPU time to solve the resulting system of linear equations within the Newton iterations. For highly nonlinear processes involving strong fluid-reservoir interactions and significant changes of flow and transport properties due to fracturing, the true Jacobian is often difficult to describe in analytical formulas. For reasons such as these, during the past three decades, despite its numerical merits of greater robustness and the ability to take larger time steps, the fully-coupled GIA method was considered to be too CPU-time and memory-intensive [21] or to be computationally inefficient [16]. It has been used primarily only as a research tool for small one- or two-dimensional problems with a few thousands of unknowns. Since the first attempts of implementing the GIA approach in the early

1980s [17, 13], only a handful of examples based on this approach have been reported in the literature, compared with numerous examples of applications based on an operator splitting approach [19, 20, 15].

1 Architecture and Design

FALCON has been designed for the simulation of geothermal reservoirs, both conventional hydrothermal and EGS. The architecture of FALCON has a plug-and-play modular design structure based on representing each piece of the residual term in a weak form of the governing PDEs as a 'Kernel'. Kernels may be coupled together to achieve different application goals. All kernels are required to supply a residual, which usually involves summing products of finite element shape functions. The basic architecture of the code allows convenient coupling of different processes and incorporation of new physics.

Figure 1 shows the basic architecture of FALCON, with the Kernels at the uppermost level, directly underlain by the numerical framework and solver libraries used to couple the Kernels and perform reservoir simulations. Currently primary Kernels (primary variables) have been written to describe the following physics:

- Single-phase flow of water
- Two-phase flow of water and steam
- Conservative heat transport
- Enthalpy transport
- Fluid and Energy Sources/Sinks
- Displacement (all mechanics are solved in terms of displacement)
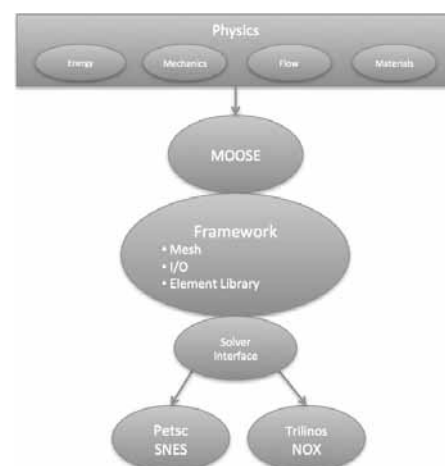

Figure 1: Kernel and Object Oriented Architecture used to develop the FALCON simulator.

For any given simulation, any combination of the primary kernels can be applied to make the problem as simple or complex as necessary, with some exceptions. Single phase flow of water problems must be cast in terms of temperature, while steam-water flow problems must use enthalpy for energy transport. The option of single phase temperature formulation was the basis for early versions of FALCON and have been retained because of their computational and memory efficiency. As an example of the modular framework, one can simply choose only a single phase pressure kernel, and solve a simple LaPlace equation based on imposed boundary conditions, totally ignoring energy and mechanics kernels.

An auxiliary variable system has been built into FALCON to handle solving most all of the derived quantities and variables that are dependent on the primary kernels mentioned above. The number of auxiliary kernels needed for a given simulation depends on the choice of primary variables and whether they are formulated in terms of pressure-temperature or pressure-enthalpy. In general, a simulation run with the pressure-enthalpy formulation, considering geomechanical displacement and damaging, requires the most auxiliary kernels and has the highest computational burden. The auxiliary kernels consist of

- Equation of state calculations
  - Steam and water density
  - Steam and water viscosity
  - Derivatives of steam and water density to pressure, temperature, or enthalpy as required
- Stress and Strain
- Fluid Velocities
- Damage Menchanics (or fracturing)

In addition to the primary and auxiliary physics kernels, other kernels are required for the mesh, material properties (and some additional supporting calculations), boundary conditions, code execution/solver parameters, and data output.

1.1 Code uses and limitations

As stated above, the FALCON code has been developed to support simulation of both conventional hydrothermal and EGS reservoirs, with a primary design focus on EGS resources. While we are using the IAPWS-97 formulation [18], which has an quite an effective operating range of pressure ($\leq 100M\ Pa$) and temperature

($\leq 800C$). Code development to date has focused on subcritical conditions.

Maximum mesh sizes are related to the number of kernels, and hence the total system wide Degrees of Freedom (DoFs), used in a simulation. In practice, the true limitations are based on computational power and available memory. The parallel scaling and performance example testing used 1 million grid blocks and more than 20 million DoFs, and showed remarkable scalability. Code tests have used computational meshes with greater than 30 million elements and also showed excellent scaling performance [7]. For any parallel simulation runs, a minimum of 20,000 DoFs per processor is recommended for good scalability.

2 Numerical Methodology

FALCON has been developed using INL's MOOSE framework [6]. This framework provides a strong numerical foundation for rapid development of multidimensional, parallel, fully implicit, fully-coupled, nonlinear simulation capabilities. MOOSE is based on a finite element discretization strategy and utilizes state-of-the-art preconditioned Jacobian-Free Newton-Krylov (JFNK) nonlinear solution method that requires only residual evaluations of the discrete system. Strategic use of this feature results in a modular, pluggable architecture that greatly simplifies adding new physics and coupling them together. The MOOSE framework incorporates multiple parallel solution capabilities including both Message Passing Interface (MPI) and threading utilizing the Intel Threading Building Blocks (TBB), which allows application codes developed upon MOOSE to run efficiently on multicore workstations, laptops and supercomputers. All parallel activities are completely hidden from application developers, enabling scientists and engineers to focus on the physics of problem they wish to solve instead of parallel programing practices.

In addition, applications developed upon MOOSE also inherit many advanced computing capabilities such as dimension-independence, massive paral- lelism, high-order finite elements and adaptive mesh refinement/coarsening with both structured and unstructured meshes.
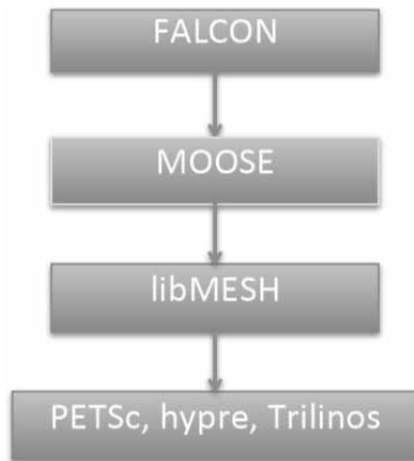
Figure 2: Hierarchical framework used to build the FAL-CON simulator, based up the INL developed MOOSE library [6]. The libMesh finite element framework developed by the CFDLab at the University of Texas at Austin [34] provides a core set of parallel finite-element libraries and couples with interfaces to linear and nonlinear solvers from both Petsc [2] and Trilinos [9] along with other packages such as Hypre [4].

The MOOSE framework has a layered structure, as shown in Figure 2. The lower layer interfaces with several open-source libraries from multiple universities and national laboratories. In particular, the libMesh finite element framework developed by the CFDLab at the University of Texas at Austin [10] provides a core set of parallel finite-element libraries. Coupled with interfaces to linear and nonlinear solvers from both PETSc [2] and Trilinos [9] along with other packages such as Hypre [4], MOOSE and application codes developed upon it provide considerable flexibility including the abilities to swap out solver libraries and to utilize diverse large scale parallel computing resources.

The middle layer of MOOSE provides a set of core functionalities necessary for residual and Jacobian (more precisely, the preconditioner) evaluations required by the preconditioned JFNK approach, such as fetching the designated test and shape functions, numerical integration using Gaussian quadrature, and coupling physics. The top layer of MOOSE, referred as the kernel is the interface with physics where the FALCON application is built (see Figure 2). It is convenient to think of a kernel as a piece of the residual term in the weak forms of PDEs, for example, the diffusion term, advection term, time accumulation term in the weak form of

general enthalpy transport equations. Kernels may be coupled together to achieve different application goals. All kernels are required to supply a residual, which usually involves summing products of finite element shape functions.

Kernels may also provide diagonal and off-diagonal blocks of the (approximate) Jacobian matrix for the purpose of building certain preconditioners.

In order to further clarify the Kernel concept, we provide a simple example (single phase water pressure diffusion) kernel here. The diffusion of pressure written as $\nabla \cdot \left(\frac{k \rho_w}{\mu_w} \cdot \nabla p_w \right)$ equation, which contributes to the overall residual in the system is provided as an example. Figure 3 shows the actual codes of the pressure diffusion kernel. In this figure, `_test` is the test function evaluated at the quadrature point `_qp` and `_grad_phi` is the gradient of shape function evaluated at the quadrature point `_qp` (both provided by MOOSE), `_u` and `_grad_u` are the current solution variable and the gradient of the current solution variable this kernel operates on evaluated at the quadrature point `_qp`. `_permeability` is the intrinsic material permeability defined in material kernel that the physics kernels can access, `_dens_water` and `_visc_water` are the water phase fluid density and viscosity, respectively, as returned from the equation of state auxiliary Kernel. Every other term based upon the MOOSE framework, the FALCON code has developed a set of *'physics'* kernels handling the time derivatives, single- and two-phase flow equations, heat and energy transport, source-sink terms, geomechanics, as well as a set of *'Auxiliary'* and *'Material'* kernels for equations of state (EOS) and flow-transport-mechanical properties required for geothermal reservoir simulations. As shown and discussed in the sections that follow, these kernels all have modular, pluggable structure, and can be coupled in arbitrary ways depending on the type of problems of interest. It is also worth noting that the MOOSE framework provides a material kernel. Flow and transport properties such as porosity, permeability, and relative permeability can all be defined within this material kernel and can be accessed by the physics kernels during each residual evaluation. Furthermore, the material kernel has access to state variables if needed. This feature is particularly useful for hydrofracturing applications where fracturing significantly modifies the porosity and permeability of porous media.

```
Real WaterMassFluxPressure_PT::computeQpResidual()
{
_tau_water[qp] = _permeability[qp] * _dens_water / _visc_water;
return _tau_water[_qp]*_grad_u[_qp] * _grad_test[_i][_qp];
}
```

```
Real WaterMassFluxPressure_PT::computeQpJacobian()
{
_tau_water[qp] = _permeability[qp] * _dens_water / _visc_water;
return _tau_water[_qp]*_grad_phi[_j][_qp] * _grad_test[_i][_qp];
}
```

Figure 3: Residual (left) and Jacobian (preconditioner, right) evaluations inside the pressure diffusion kernel for single phase flow of water.

3 Example Applications

3.1 Comparison with analytical solution

Our first example problem is to solve a simple one-dimensional heat conduction-convection problem using FALCON and compare the numerical solution with the analytical solution. In this particular example, only two equations, fluid flow and heat transport, are solved.

The analytical solution compared in this example is derived from the solution by Faust and Mercer [5], by omitting the heat exchange between confined aquifer and surrounding rock matrix. In order to obtain the analytical solution, the thermodynamic and transport properties, such as water density and viscosity are assigned as constants. Then the mass conservation equation reduced to a Laplacian equation of pressure (Equation 1), which gives a uniform velocity v_w along x-direction.

$$\tau \nabla^2 P + q'_w = 0 \qquad (1)$$

And the energy equation reads as:

$$K_m \frac{\partial^2 u}{\partial x^2} - v_w \rho_w c_w \frac{\partial u}{\partial x} = \rho_m c_m \frac{\partial u}{\partial t} \qquad (2)$$

where $\rho_m c_m = \varphi \rho_w c_w + (1-\varphi)\rho_r c_r$. The c is specific heat capacity of water (subscripted with w) or rock (subscripted with r). And u is normalized temeperature $u = \frac{T-T_0}{T_i-t_0}$, T_i and T_0 are the injection and initial temperature, respectively. K_m is the heat conductivity of wet rock.

The analytical solution for Equivation (1) and (2) is given by Avdonin [1]:

$$u(\chi,\tau) = \frac{\chi}{(\pi\tau)^{1/2}} \int_0^1 exp\left[-(s\gamma(\tau)^{\frac{1}{2}}\right.$$
$$\left. - \frac{\chi}{2s(\tau)^{1/2}})^2\right]\frac{ds}{s^2} \qquad (3)$$

where $\chi = \frac{2x}{b}$, $\tau = \frac{4K_m t}{c_m \rho_m b^2}$, $\gamma = \frac{Qc_w \rho_w}{4K_m}$, Q is the injection rate, and b is the reservoir thickness (1-m in this example).

Parameter	Value	Units
Porosity	0.20	-
Permeability	1×10^{-15}	m^2
Rock Density	2.5×10^3	kg/m^3
Rock Specific Heat	0.92×10^3	$J/kg°C$
Thermal Conductivity	1.5	$W/m°C$
Water Density	1×10^3	kg/m^3
Water Specific Heat	4.186×10^3	$J/kg°C$

Table 1: Parameters used for the 1-dimensional convection-conduction problem numerical-analytical comparison.

In FALCON simulations, the geometry used for this example is a 100 meter long rectangle, 1 meter in width, with a 1 meter grid resolution. The mesh consisted of 100 elements and 102 nodes. Table 1 summarizes the parameters used for this example. Initial conditions are set as pressure $P = 10MPa$, temperature $T = 200°C$, uniformly. The BC are set as: injection pressure $P_i = 10.5MPa$, temperature $Ti = 150°C$ at left side, and constant pressure $10MPa$, temperature $200°C$ are assigned at right side.
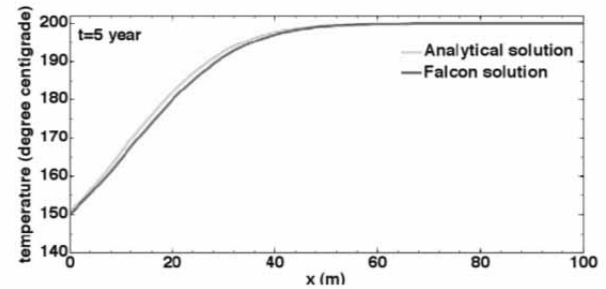
Figure 4: Comparison of the numerical and analytical solutions for 1-dimensional heatconduction-convection problem. Temperature profile calculated by FALCON and analytical solution at 5 years. The small discrepancy is caused by the pressure and temperature dependent density and viscosity of water used in the FALCON simulations. The analytical solution assumes a constant uid density and viscosity, which essentially decouplesthe the flow and transport problem.

Figure 4 shows the comparison between numerical and analytical solutions after 5 years of simulated transport. It is clear that the numerical solution agrees well with the analytical solution. The small discrepancy is caused by the pressure and temperature dependent density and viscosity of water used in the FALCON simulations. The analytical solution assumes a constant fluid density and viscosity, which essentially decouples the flow and transport problem.

3.2   Thermal stimulation of a geothermal reservoir

Management of fluid reinjection is of critical importance for maintaining geothermal reservoir performance. Reinjection has posed a problem for portions of the Hellisheidi Geothermal Field, southwest Iceland, where a number of wells are drilled into active faults. The Hellisheidi Geothermal Field is located in the southern part of the Hengill Area, an active volcanic system consisting of Mt. Hengill and fracture/fault zones to the north- and south-west (Smundsson, 1967; Franzson et al., 2005). Injection tests have resulted in swarms of small earthquakes and with the injectivity of the wells exhibiting a high dependence on temperature of the reinjected water. Strongly coupled thermo- hydro-mechanical effects on fractures in the fracture-governed reservoir likely explain the temperature dependent injectivity.

A number of injectivity tests have been conducted to support the development of the Hellisheidi Power Plant. For several of the wells, injection experiments were conducted using three types of water; 120℃ untreated brine directly from the low-pressure boiler, a 90℃ mixture of brine and condense water (7:3) from the turbines, and 15℃ cold groundwater. These experiments were done in the three most promising wells in the Hsmli Reinjection Zone, HN-09, HN-12, and HN-16. The injectivity vs. T is plotted for all the wells in Figure 5. It should be mentioned here that the values for cold water in wells HN-12 and HN-16 are inaccurate. The wells are so permeable that the pressure changes in the pumping tests were not very clear.

The injection tests for estimating the injectivity were conducted as described below. Maximum flow of water at preferred temperature was injected into the well for several days. The wellhead pressure was monitored in order to estimate when the well had reached equilibrium.

A pressure and temperature sensor was placed in the well at the depth of its main feed zone. The flow was lowered in three steps, each lasting for approximately 3 hours to allow the pressure to equilibrate to the new injection rate. Figure 6 is an example of pumping test results for the hot water injection into well HN-09. The pressure and temperature are plotted over the duration of the injection steps.

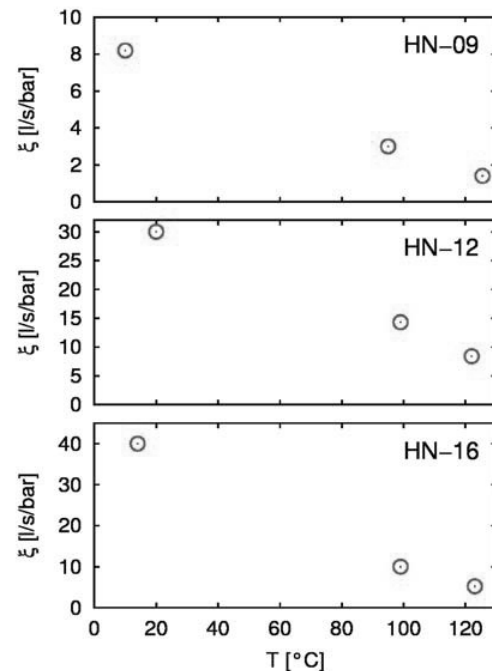**Figure 5:** Injectivity at different values of temperature (T) in three wells in the Hsmli Reinjection Zone. The injectivity values for the lowest temperatures in wells HN-12 and HN-16 are not very accurate.
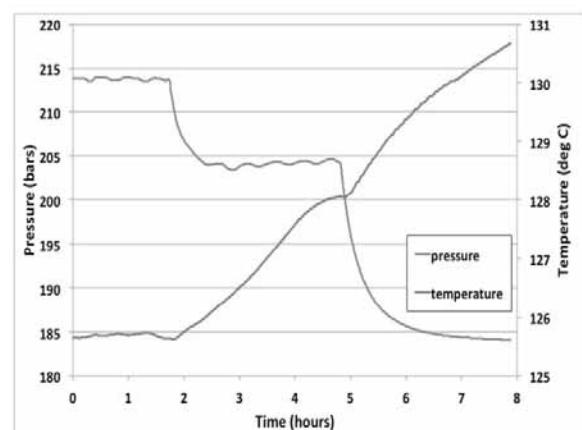

Figure 6: Injection test in well HN-09 using 120 C water. The pressure sensor is placed 30m above the bottom of the well showing the pressure and temperature over the dura- tion of the test.

Model Setup

A radial structured mesh, with a radius of 250 meters, is being used to simulate the injection into well HN-09. The simulation domain is 100 meters thick, with the production zone begin represented as a 5-meter thick zone of fractured rock embedded in lower permeability (unfractured) reservoir rock. Figure 7 shows the simulation domain and the computational mesh.
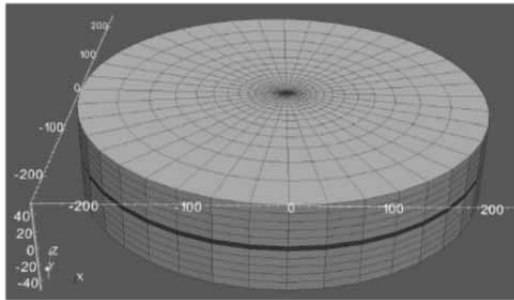

Figure 7: Computational domain used for the numerical simulations of injection into well HN-09. The red zone in the middle of the domain represents the fracture zone/fault system comprising the feed zone of the well.

Initial conditions being used in the simulations mimic those that exist in the reservoirs. The initial temperature distribution used in the simulation is shown in Figure 8, with the resulting water density, as calculated from the IAPWS-97 steam tables, shown on Figure 9. The temperature in the feed zone is approximately 264℃, with the temper- ature ranging from 263℃ to 265℃. The water density in the reservoir ranged from approximately 772 to 776 kg/m^3 in the initial conditions.

The initial pressure in the reservoir was specified to a uniform 185 bars, and as the majority of the flow was expected to be primarily horizontal and limited to exist only within the thin fracture zone, the effects of gravity of the fluid flow and heat transport were neglected. This approach greatly simplified the specification of the boundary conditions needed for the simulations.
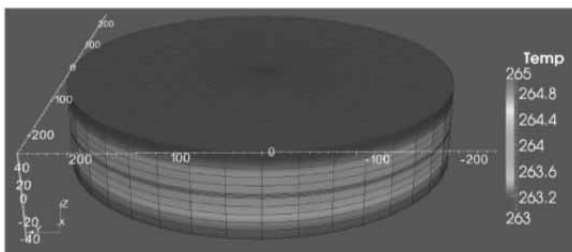
Figure 8: Initial temperature used for the simulations.
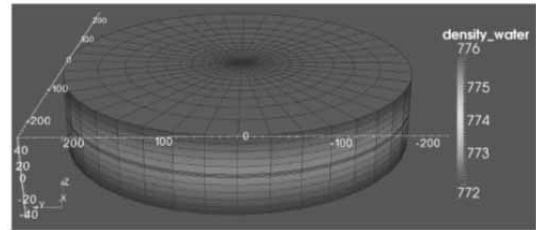
Figure 9: Initial water density distributions used for the simulations.
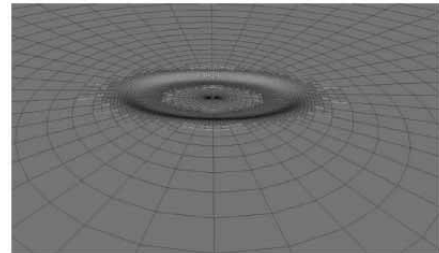
Figure 10: Predicted thermal contraction of the reservoir matrix in the feed zone in the vicinity of well HN-09. Note that the deformation is greatly exaggerated for illustration purposes.

Feedback between the geomechanics and fluid flow are being implemented by revising the permeability of the feed zone by an effective fracture aperture, as calculated by the thermo-mechanical deformation of the mesh resulting from the injection of cold fluid into the initially hot reservoir. As the host reservoir rock thermally contracts, the effective permeability from the fractured feed zone increases using a cubic law approxi- mation.

An initial modeling scenario, consisting of injecting approximately 30 l/s of 20℃ pure water into the approximately 260℃ reservoir, is currently underway. Preliminary simulation results are encouraging, where permeability increases on the order of 10X to 100X are predicted in the vicinity of well HN-09's feed zone. The reservoir matrix contraction in the feed zone near the injection well is predicted to approach 10^{-4} meters.

Figure 10 shows the results of the geomechanical deformation in the vicinity of the injection well in the feed zone, along with the mesh that is adaptively refined by FALCON to capture the strong gradient in the temperature and resulting thermal deformation in the reservoir.

Acknowledgment

This contribution is a post-conference publication from SIMS 2012 Conference (53rd SIMS Conference, Reykjavík, Iceland, October 4 - 6, 2012). The contribution is a (partly) modified publication from the paper published in the Proceedings of SIMS 2012, published by Orkustofnun, National Energy Authority Iceland, ISBN: 978-9979-68-318-6, electronically available at http://www.scansims.org/sims2012/SIMS2012.pdf.

References

[1] Avdonin, NA. Some formulas for calculating the temperature of a stratum subject to thermal injection. *Izvestiya Vysshikh Uchebnykh Zavedenii*. 1964; 7(4):37–41.

[2] Balay S, Buschelman K, Eijkhout V, Gropp WD, Kaushik D, Knepley MG, McInnes LC, Smith BF, Zhang H. *PETSc users manual*. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[3] Carrayrou J, Knabner P, Hoffmann J, Krautle S, de Dieuleveult C, Erhel J, van der Lee J, Lagneau V, Mayer KU, MacQuarrie KTB. Comparison of numerical methods for simulating strongly nonlinear and heterogeneous reac- tive transport problems-the momas benchmark case. *Computational Geosciences*. 2010; 14(3):483–502.

[4] Falgout RD, Yang UM. hypre: A library of high performance preconditioners. In *Computational Science-Iccs* 2002, Berlin, 2002. Springer-Verlag.

[5] Faust CR, Mercer JW. Geothermal reservoir simulation .1. Mathematcial- models for liquid-dominated and vapot-dominated hydrothermal systems. *Water Resources Research*. 1979; 15(1):23–30.

[6] Gaston D, Newman C, Hansen G, Lebrun-Grandié D. MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*. 2009; 239: 1768–1778.

[7] Guo L, Huang H, Gaston D, Permann C, Andrs D, Redden G, Lu C, Fox D, Fujita Y. A parallel fully-coupled fully-implicit solution to reactive transport in porous media using preconditioned jacobian-free newton-krylov method. *Journal of Computational Physics, in review*.

[8] Hammond GE, Valocchi AJ, Lichtner PC. Application of jacobian-free newton-krylov with physics-based preconditioning to biogeochemical transport. *Advances in Water Resources*. 2005; 28(4):359–376.

[9] Heroux M et al. Trilinos: an object-oriented software framework for the solution of large-scale, complex multi-physics engineering and scientific problems. http://trilinos.sandia.gov, 2008.

[10] Kirk BS, Peterson JW, Stogner RH, Carey GF. libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. *Eng Comput- Germany*. 2006; 22(3-4): 237–254.

[11] Knoll D, Park R, Smith K. Application of the Jacobian-free Newton-Krylov method in computational reactor physics. In American Nuclear Society *2009 International Conference on Advances in Mathematics, Computational Methods, and Reactor Physics*, May 2009; Saratoga Springs.

[12] McHugh PR, Knoll DA. Inexact Newton's method solutions to the incompressible Navier-Stokes and energy equations using standard and matrix-free implementations. *AIAA J*. 1994; 32: 2394.

[13] Miller CW, Benson LV. Simulation of solute transport in a chemically reactive heterogeneous system - model development and application. *Water Resources Research*. 1983; 19(2): 381–391.

[14] Molins S, Carrera J, Ayora C, Saaltink MW. A formulation for decoupling components in reactive transport problems. *Water Resources Research*. 2004; 40(10).

[15] Rutqvist J, Wu YS, Tsang CF, Bodvarsson G. A modeling approach for analysis of coupled multiphase fluid flow, heat transfer, and deformation in fractured porous rock. *Int. Journal of Rock Mechanics and Mining Sci*. 2002; 39(4): 429–442.

[16] Steefel CI, MacQuarrie KTB. Approaches to modeling of reactive transport in porous media. *Reviews in Mineralogy and Geochemistry*, 1996; 34: 83–129.

[17] Valocchi AJ, Street RL, Roberts PV. Transport of ion-exchanging solutes in groundwater - chromatographic theory and field simulation. *Water Resources Research*. 1981; 17(5):1517–1527.

[18] Wagner W, Cooper JR, Dittman A, Kijima J, Kretzschmar HJ, Kruse A, Mares R, Oguchi K, Sato H, Stocker I, Sifner O, Takaishi Y, Tanishita I, Trubenbach J, Willkommen Th. The IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam. *Trans. ASME*. 1997; 150(122):150–182.

[19] Xu T, Sonnenthal E, Spycher N, Pruess K. TOUGHREACT - A simulation program for non-isothermal multiphase reactive geochemical transport in variably saturated geologic media: Applications to geothermal injectivity and CO2 geological sequestration. *Computers & Geosciences*. 2006; 32(2):145–165.

[20] Yeh GT, Sun J, Jardine PM, Burgos WD, Fang Y, Li MH, Siegel MD. HYDROGEOCHEM 5.0: A Coupled Model of Fluid Flow, Thermal Transport, and HYDROGEOCHEMical Transport through Saturated-Unsaturated Media: Version 5.0. Technical Report ORNL/TM-2004/107, Oak Ridge National Laboratory, Oak Ridge, TN, 2004.

[21] Yeh GT, Tripathi VS. A critical-evaluation of recent developments in hydrogeochemical transport models of reactive multichemical components. *Water Resources Research*. 1989; 25(1): 93–108.

SNE Simulation News

EUROSIM Data and Quick Info

EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013

www.eurosim2013.info

Contents

Simulation Notes Europe SNE is the official membership journal of EUROSIM and distributed / available to members of the EUROSIM Societies as part of the membership benefits. **SNE** is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With **Online SNE** the publisher **ARGESIM** follows the **Open Access** strategy for basic **SNE** contributions. Since 2011 **Online SNE** contributions are identified by DOI 10.11128/sne.xx.nnnnn. for better web availability and indexing.

Print SNE, high-resolution **Online SNE**, and additional SNE contributions are available for subscription via membership in a **EUROSIM** society.

This *EUROSIM Data & Quick Info* compiles data from EUROSIM societies and groups: addresses, weblinks, officers of societies with function and email, to be published regularly in SNE issues.

If you have any information, announcement, etc. you want to see published, please contact a member of the editorial board in your country or the editorial office.

EUROSIM
Federation of European Simulation Societies

General Information. EUROSIM, the Federation of European Simulation Societies, was set up in 1989. The purpose of EuroSim is to provide a European forum for regional and national simulation societies to promote the advancement of modelling and simulation in industry, research, and development.
→ www.eurosim.info

Member Societies. EUROSIM members may be national simulation societies and regional or international societies and groups dealing with modelling and simulation. At present EUROSIM has thirteen full members and three observer member:

ASIM	Arbeitsgemeinschaft Simulation *Austria, Germany, Switzerland*
CEA-SMSG	Spanish Modelling and Simulation Group *Spain*
CROSSIM	Croatian Society for Simulation Modeling *Croatia*
CSSS	Czech and Slovak Simulation Society *Czech Republic, Slovak Republic*
DBSS	Dutch Benelux Simulation Society *Belgium, Netherlands*
FRANCOSIM	Société Francophone de Simulation *Belgium, France*
HSS	Hungarian Simulation Society *Hungary*
ISCS	Italian Society for Computer Simulation *Italy*
LSS	Latvian Simulation Society *Latvia*
PSCS	Polish Society for Computer Simulation *Poland*
SIMS	Simulation Society of Scandinavia *Denmark, Finland, Norway, Sweden*
SLOSIM	Slovenian Simulation Society *Slovenia*
UKSIM	United Kingdom Simulation Society *UK, Ireland*
ROMSIM	Romanian Society for Modelling and Simulation, *Romania, Observer Member*
RNSS	Russian National Simulation Society *Russian Federation, Observer Member*
LIOPHANT	LIOPHANT Simulation Club *Italy & International, Observer Member*

EUROSIM Board / Officers. EUROSIM is governed by a board consisting of one representative of each member society, president and past president, and representatives for SNE Simulation notes Europe. The President is nominated by the society organising the next EUROSIM Congress. Secretary and Treasurer are elected out of members of the Board.

President	Khalid Al.Begain *kbegain@glam.ac.uk*
Past President	Mikuláš Alexík (CSSS), *alexik@frtk.fri.utc.sk*
Secretary	Borut Zupančič (SLOSIM) *borut.zupancic@fe.uni-lj.si*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
SNE Repres.	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

SNE – Simulation Notes Europe. SNE is a scientific journal with reviewed contributions as well as a membership newsletter for EUROSIM with information from the societies in the *News Section*. EUROSIM societies are offered to distribute to their members the journal SNE as official membership journal. SNE Publishers are EUROSIM, ARGESIM and ASIM.

Editor-in-chief	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

→ www.sne-journal.org,
✉ office@sne-journal.org

EUROSIM Congress. EUROSIM is running the triennial conference series EUROSIM Congress. The congress is organised by one of the EUROSIM societies.

EUROSIM 2013 will be organised by UKSIM in Cardiff, Wales, UK, September 10-13, 2013.

Chairs / Team EUROSIM 2013
Khalid Al.Begain, *kbegain@glam.ac.uk*
Richard Zobel, *r.zobel@ntlworld.com*
David Al-Dabass, *david.al-dabass@ntu.ac.uk*
Alessandra Orsoni, *a.orsoni@kingston.ac.uk*
Richard Cant, *richard.cant@ntu.ac.uk*

→ www.eurosim2013.info

ASIM
German Simulation Society
Arbeitsgemeinschaft Simulation

ASIM (Arbeitsgemeinschaft Simulation) is the association for simulation in the German speaking area, servicing mainly Germany, Switzerland and Austria. ASIM was founded in 1981 and has now about 700 individual members, and 30 institutional or industrial members. Furthermore, ASIM counts about 300 affiliated members.

→ *www.asim-gi.org* with members' area

✉ *info@asim-gi.org, admin@asim-gi.org*

✉ ASIM – Inst. f. Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

ASIM Officers	
President	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*
Vice presidents	Sigrid Wenzel, *s.wenzel@uni-kassel.de*
	T. Pawletta, *pawel@mb.hs-wismar.de*
Secretary	Anna Mathe, *anna.mathe@tuwien.ac.at*
Treasurer	I. Bausch-Gall, *Ingrid@Bausch-Gall.de*
Membership Affairs	S. Wenzel, *s.wenzel@uni-kassel.de*
	W. Maurer, *werner.maurer@zhwin.ch*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
Universities / Research Inst.	S. Wenzel, *s.wenzel@uni-kassel.de*
	W. Wiechert, *W.Wiechert@fz-juelich.de*
	J. Haase, *Joachim.Haase@eas.iis.fraunhofer.de*
	Katharina Nöh, *k.noeh@fz-juelich.de*
Industry	S. Wenzel, *s.wenzel@uni-kassel.de*
	K. Panreck, *Klaus.Panreck@hella.com*
Conferences	Klaus Panreck *Klaus.Panreck@hella.com*
	A. Gnauck, *albrecht.gnauck@tu-cottbus.de*
Publications	Th. Pawletta, *pawel@mb.hs-wismar.de*
	Christina Deatcu, *christina.deatcu@hs-wismar.de*
	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
Repr. EUROSIM	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
	N. Popper, *niki.popper@drahtwarenhandlung.at*
Education / Teaching	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	N. Popper, *niki.popper@drahtwarenhandlung.at*
	Katharina Nöh, *k.noeh@fz-juelich.de*
International Affairs	H. Szczerbicka, *hsz@sim.uni-hannover.de*
	O. Rose, *Oliver.Rose@tu-dresden.de*
Editorial Board SNE	T. Pawletta, *pawel@mb.hs-wismar.de*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
Web EUROSIM	Anna Mathe, *anna.mathe@tuwien.ac.at*

Last data update December 2012

ASIM Working Committee. ASIM, part of GI - Gesellschaft für Informatik, is organised in Working Committees, dealing with applications and comprehensive subjects in modelling and simulation:

ASIM Working Committee	
GMMS	Methods in Modelling and Simulation Th. Pawletta, *pawel@mb.hs-wismar.de*
SUG	Simulation in Environmental Systems Wittmann, *wittmann@informatik.uni-hamburg.de*
STS	Simulation of Technical Systems H.T.Mammen, *Heinz-Theo.Mammen@hella.com*
SPL	Simulation in Production and Logistics Sigrid Wenzel, *s.wenzel@uni-kassel.de*
EDU	Simulation in Education/Education in Simulation N. Popper, *niki.popper@drahtwarenhandlung.at*
	Working Groups for Simulation in Business Administration, in Traffic Systems, for Standardisation, for Validation, etc.

CROSSIM – Croatian Society for Simulation Modelling

CROSSIM-*Croatian Society for Simulation Modelling* was founded in 1992 as a non-profit society with the goal to promote knowledge and use of simulation methods and techniques and development of education. CROSSIM is a full member of EUROSIM since 1997.

→ *www.eurosim.info*

✉ *vdusak@foi.hr*

✉ CROSSIM / Vesna Dušak
Faculty of Organization and
Informatics Varaždin, University of Zagreb
Pavlinska 2, HR-42000 Varaždin, Croatia

CROSSIM Officers	
President	Vesna Dušak, *vdusak@foi.hr*
Vice president	Jadranka Božikov, *jbozikov@snz.hr*
Secretary	Vesna Bosilj-Vukšić, *vbosilj@efzg.hr*
Executive board members	Vlatko Čerić, *vceric@efzg.hr*
	Tarzan Legović, *legovic@irb.hr*
Repr. EUROSIM	Jadranka Božikov, *jbozikov@snz.hr*
Edit. Board SNE	Vesna Dušak, *vdusak@foi.hr*
Web EUROSIM	Jadranka Božikov, *jbozikov@snz.hr*

Last data update December 2012

CSSS – Czech and Slovak Simulation Society

CSSS -The *Czech and Slovak Simulation Society* has about 150 members working in Czech and Slovak national scientific and technical societies (*Czech Society for Applied Cybernetics and Informatics*, *Slovak Society for Applied Cybernetics and Informatics*). The main objectives of the society are: development of education and training in the field of modelling and simulation, organising professional workshops and conferences, disseminating information about modelling and simulation activities in Europe. Since 1992, CSSS is full member of EUROSIM.

→ *www.fit.vutbr.cz/CSSS*
✉ snorek@fel.cvut.cz
✉ CSSS / Miroslav Šnorek, CTU Prague
FEE, Dept. Computer Science and Engineering,
Karlovo nam. 13, 121 35 Praha 2, Czech Republic

CSSS Officers	
President	Miroslav Šnorek, *snorek@fel.cvut.cz*
Vice president	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Treasurer	Evžen Kindler, *ekindler@centrum.cz*
Scientific Secr.	A. Kavička, *Antonin.Kavicka@upce.cz*
Repr. EUROSIM	Miroslav Šnorek, *snorek@fel.cvut.cz*
Deputy	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Edit. Board SNE	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Web EUROSIM	Petr Peringer, peringer@fit.vutbr.cz

Last data update December 2012

FRANCOSIM – Société Francophone de Simulation

FRANCOSIM was founded in 1991 and aims to the promotion of simulation and research, in industry and academic fields. Francosim operates two poles.

- Pole Modelling and simulation of discrete event systems. Pole Contact: *Henri Pierreval, pierreva@imfa.fr*
- Pole Modelling and simulation of continuous systems. Pole Contact: *Yskandar Hamam, y.hamam@esiee.fr*

→ *www.eurosim.info*
✉ *y.hamam@esiee.fr*
✉ FRANCOSIM / Yskandar Hamam
Groupe ESIEE, Cité Descartes,
BP 99, 2 Bd. Blaise Pascal,
93162 Noisy le Grand CEDEX, France

FRANCOSIM Officers	
President	Karim Djouani, *djouani@u-pec.fr*
Treasurer	François Rocaries, *f.rocaries@esiee.fr*
Repr. EUROSIM	Karim Djouani, *djouani@u-pec.fr*
Edit. Board SNE	Karim Djouani, *djouani@u-pec.fr*

Last data update December 2012

DBSS – Dutch Benelux Simulation Society

The Dutch Benelux Simulation Society (DBSS) was founded in July 1986 in order to create an organisation of simulation professionals within the Dutch language area. DBSS has actively promoted creation of similar organisations in other language areas. DBSS is a member of EUROSIM and works in close cooperation with its members and with affiliated societies.

→ www.eurosim.info
✉ *a.w.heemink@its.tudelft.nl*
✉ DBSS / A. W. Heemink
Delft University of Technology, ITS - twi,
Mekelweg 4, 2628 CD Delft, The Netherlands

DBSS Officers	
President	A. Heemink, *a.w.heemink@its.tudelft.nl*
Vice president	W. Smit, *smitnet@wxs.nl*
Treasurer	W. Smit, *smitnet@wxs.nl*
Secretary	W. Smit, *smitnet@wxs.nl*
Repr. EUROSIM	A. Heemink, *a.w.heemink@its.tudelft.nl*
Deputy	W. Smit, *smitnet@wxs.nl*
Edit. Board SNE	A. Heemink, *a.w.heemink@its.tudelft.nl*

Last data update April 2006

HSS – Hungarian Simulation Society

The Hungarian Member Society of EUROSIM was established in 1981 as an association promoting the exchange of information within the community of people involved in research, development, application and education of simulation in Hungary and also contributing to the enhancement of exchanging information between the Hungarian simulation community and the simulation communities abroad. HSS deals with the organization of lectures, exhibitions, demonstrations, and conferences.

→ *www.eurosim.info*
✉ *javor@eik.bme.hu*
✉ HSS / András Jávor,
Budapest Univ. of Technology and Economics,
Sztoczek u. 4, 1111 Budapest, Hungary

HSS Officers	
President	András Jávor, *javor@eik.bme.hu*
Vice president	Gábor Szűcs, *szucs@itm.bme.hu*
Secretary	Ágnes Vigh, *vigh@itm.bme.hu*
Repr. EuroSim	András Jávor, *javor@eik.bme.hu*
Deputy	Gábor Szűcs, *szucs@itm.bme.hu*
Edit. Board SNE	András Jávor, *javor@eik.bme.hu*
Web EuroSim	Gábor Szűcs, *szucs@itm.bme.hu*

Last data update March 2008

PSCS – Polish Society for Computer Simulation

PSCS was founded in 1993 in Warsaw. PSCS is a scientific, non-profit association of members from universities, research institutes and industry in Poland with common interests in variety of methods of computer simulations and its applications. At present PSCS counts 257 members.

→ *www.ptsk.man.bialystok.pl*

✉ *leon@ibib.waw.pl*

✉ PSCS / Leon Bobrowski, c/o IBIB PAN,
   ul. Trojdena 4 (p.416), 02-109 Warszawa, Poland

PSCS Officers	
President	Leon Bobrowski, *leon@ibib.waw.pl*
Vice president	Andrzej Grzyb, Tadeusz Nowicki
Treasurer	Z. Sosnowski, *zenon@ii.pb.bialystok.pl*
Secretary	Zdzislaw Galkowski, *Zdzislaw.Galkowski@simr.pw.edu.pl*
Repr. EuroSim	Leon Bobrowski, *leon@ibib.waw.pl*
Deputy	Tadeusz Nowicki, *tadeusz.nowicki@wat.edu.pl*
Edit. Board SNE	Zenon Sosnowski, *z.sosnowski@pb.ed.pl*
Web EuroSim	Magdalena Topczewska *m.topczewska@pb.edu.pl*

Last data update December 2012

ISCS – Italian Society for Computer Simulation

The Italian Society for Computer Simulation (ISCS) is a scientific non-profit association of members from industry, university, education and several public and research institutions with common interest in all fields of computer simulation.

→ *www.eurosim.info*

✉ *Mario.savastano@uniina.at*

✉ ISCS / Mario Savastano,
   c/o CNR - IRSIP,
   Via Claudio 21, 80125 Napoli, Italy

ISCS Officers	
President	M. Savastano, *mario.savastano@unina.it*
Vice president	F. Maceri, *Franco.Maceri@uniroma2.it*
Repr. EuroSim	F. Maceri, *Franco.Maceri@uniroma2.it*
Secretary	Paola Provenzano, *paola.provenzano@uniroma2.it*
Edit. Board SNE	M. Savastano, *mario.savastano@unina.it*

Last data update December 2012

SIMS – Scandinavian Simulation Society

SIMS is the *Scandinavian Simulation Society* with members from the four Nordic countries Denmark, Finland, Norway and Sweden. The SIMS history goes back to 1959. SIMS practical matters are taken care of by the SIMS board consisting of two representatives from each Nordic country (Iceland one board member).

SIMS Structure. SIMS is organised as federation of regional societDjouaniies. There are FinSim (Finnish Simulation Forum), DKSIM (Dansk Simuleringsforening) and NFA (Norsk Forening for Automatisering).

→ *www.scansims.org*

✉ *esko.juuso@oulu.fi*

✉ SIMS / Esko Juuso, Department of Process and Environmental Engineering, 90014 Univ.Oulu, Finland

SIMS Officers	
President	Esko Juuso, *esko.juuso@oulu.fi*
Vice president	Erik Dahlquist, *erik.dahlquist@mdh.se*
Treasurer	Vadim Engelson, *vadim.engelson@mathcore.com*
Repr. EuroSim	Esko Juuso, *esko.juuso@oulu.fi*
Edit. Board SNE	Esko Juuso, *esko.juuso@oulu.fi*
Web EuroSim	Vadim Engelson

Last data update December 2012

SLOSIM – Slovenian Society for Simulation and Modelling

SLOSIM - Slovenian Society for Simulation and Modelling was established in 1994 and became the full member of EUROSIM in 1996. Currently it has 69 members from both slovenian universities, institutes, and industry. It promotes modelling and simulation approaches to problem solving in industrial as well as in academic environments by establishing communication and cooperation among corresponding teams.

→ www.slosim.si

✉ *slosim@fe.uni-lj.si*

✉ SLOSIM / Rihard Karba, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia

SLOSIM Officers	
President	B. Zupančič, *borut.zupancic@fe.uni-lj.si*
Vice president	Leon Žlajpah, *leon.zlajpah@ijs.si*
Secretary	Vito Logar, *vito.logar@fe.uni-lj.si*
Treasurer	Milan Simčič, *milan.simcic@fe.uni-lj.si*
Repr. EuroSim	B. Zupančič, *borut.zupancic@fe.uni-lj.si*
Deputy	Rihard Karba, *rihard.karba@fe.uni-lj.si*
Edit. Board SNE	Rihard Karba, *rihard.karba@fe.uni-lj.si*
Web EuroSim	Vito Logar, *vito.logar@fe.uni-lj.si*

Last data update December 2012

UKSIM - United Kingdom Simulation Society

UKSIM has more than 100 members throughout the UK from universities and industry. It is active in all areas of simulation and it holds a biennial conference as well as regular meetings and workshops.

→ *www.uksim.org.uk*

✉ *david.al-dabass@ntu.ac.uk*

✉ UKSIM / Prof. David Al-Dabass
    Computing & Informatics,
    Nottingham Trent University
    Clifton lane, Nottingham, NG11 8NS
    United Kingdom

UKSIM Officers	
President	David Al-Dabass, *david.al-dabass@ntu.ac.uk*
Vice president	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Secretary	Richard Cant, *richard.cant@ntu.ac.uk*
Treasurer	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Membership chair	K. Al-Begain, *kbegain@glam.ac.uk*
Univ. liaison chair	R. Cheng, *rchc@maths.soton.ac.uk*
Repr. EuroSim	Richard Zobel, *r.zobel@ntlworld.com*
Deputy	K. Al-Begain, *kbegain@glam.ac.uk*
Edit. Board SNE	Richard Zobel, *r.zobel@ntlworld.com*

Last data update December 2012

CEA-SMSG – Spanish Modelling and Simulation Group

CEA is the Spanish Society on Automation and Control In order to improve the efficiency and to deep into the different fields of automation, the association is divided into thematic groups, one of them is named 'Modelling and Simulation', constituting the group.

→ *www.cea-ifac.es/wwwgrupos/simulacion*

→ *simulacion@cea-ifac.es*

✉ CEA-SMSG / María Jesús de la Fuente,
    System Engineering and AutomaticControl department,
    University of Valladolid,
    Real de Burgos s/n., 47011 Valladolid, SPAIN

CAE - SMSG Officers	
President	M. À. Piera Eroles, *MiquelAngel.Piera@uab.es*
Vice president	Emilio Jiminez, *emilio.jiminez@unirioja.es*
Repr. EuroSim	Emilio Jiminez, *emilio.jiminez@unirioja.es*
Edit. Board SNE	Emilio Jiminez, *emilio.jiminez@unirioja.es*
Web EuroSim	Mercedes Peres, *mercedes.perez@unirioja.es*

Last data update December 2012

LSS – Latvian Simulation Society

The Latvian Simulation Society (LSS) has been founded in 1990 as the first professional simulation organisation in the field of Modelling and simulation in the post-Soviet area. Its members represent the main simulation centres in Latvia, including both academic and industrial sectors.

→ *briedis.itl.rtu.lv/imb/*

✉ *merkur@itl.rtu.lv*

✉ LSS / Yuri Merkuryev, Dept. of Modelling
    and Simulation Riga Technical University
    Kalku street 1, Riga, LV-1658, LATVIA

LSS Officers	
President	Yuri Merkuryev, *merkur@itl.rtu.lv*
Secretary	Artis Teilans, *Artis.Teilans@exigenservices.com*
Repr. EuroSim	Yuri Merkuryev, *merkur@itl.rtu.lv*
Deputy	Artis Teilans, *Artis.Teilans@exigenservices.com*
Edit. Board SNE	Yuri Merkuryev, *merkur@itl.rtu.lv*
Web EuroSim	Oksana Sosho, *oksana@itl.rtu.lv*

Last data update December 2012

ROMSIM – Romanian Modelling and Simulation Society

ROMSIM has been founded in 1990 as a non-profit society, devoted to theoretical and applied aspects of modelling and simulation of systems. ROMSIM currently has about 100 members from Romania and Moldavia.

→ *www.ici.ro/romsim/*

✉ *sflorin@ici.ro*

✉ ROMSIM / Florin Stanciulescu,
    National Institute for Research in Informatics, Averescu
    Av. 8 – 10, 71316 Bucharest, Romania

ROMSIM Officers	
President	Florin Stanciulescu, *sflorin@ici.ro*
Vice president	Florin Hartescu, *flory@ici.ro* Marius Radulescu, *mradulescu@ici.ro*
Repr. EuroSim	Florin Stanciulescu, *sflorin@ici.ro*
Deputy	Marius Radulescu, *mradulescu@ici.ro*
Edit. Board SNE	Florin Stanciulescu, *sflorin@ici.ro*
Web EuroSim	Zoe Radulescu, *radulescu@ici.ro*

Last data update December 2012

RNSS – Russian Simulation Society

NSS - The Russian National Simulation Society (Национальное Общество Имитационного Модели-рования – НОИМ) was officially registered in Russian Federation on February 11, 2011. In February 2012 NSS has been accepted as an observer member of EUROSIM.

→ *www.simulation.su*

✉ *yusupov@iias.spb.su*

✉ RNSS / R. M. Yusupov,
St. Petersburg Institute of Informatics and Automation RAS, 199178, St. Petersburg, 14th lin. V.O, 39

RNSS Officers	
President	R. M. Yusupov, *yusupov@iias.spb.su*
Chair Man. Board	A. Plotnikov, *plotnikov@sstc.spb.ru*
Secretary	M. Dolmatov, *dolmatov@simulation.su*
Repr. EuroSim	R. M. Yusupov, *yusupov@iias.spb.su*
Deputy	B. Sokolov, *sokol@iias.spb.su*
Edit. Board SNE	Y. Senichenkov, *sneyb@dcn.infos.ru*

Last data update February 2012

LIOPHANT Simulation

Liophant Simulation is a non-profit association born in order to be a trait-d'union among simulation developers and users; Liophant is devoted to promote and diffuse the simulation techniques and methodologies; the Association promotes exchange of students, sabbatical years, organization of International Conferences, organization of courses and stages in companies to apply the simulation to real problems.

→ *www.liophant.org*

✉ *info@liophant.org*

✉ LIOPHANT Simulation, c/o Agostino G. Bruzzone, DIME, University of Genoa, Polo Savonese, via Molinero 1, 17100 Savona (SV), Italy

LIOPHANT Officers	
President	A.G. Bruzzone, *agostino@itim.unige.it*
Director	E. Bocca, *enrico.bocca@liophant.org*
Secretary	A. Devoti, *devoti.a@iveco.com*
Treasurer	Marina Masseimassei@itim.unige.it
Repr. EuroSim	A.G. Bruzzone, *agostino@itim.unige.it*
Deputy	F. Longo, *f.longo@unical.it*
Edit. Board SNE	F. Longo, *f.longo@unical.it*
Web EuroSim	F. Longo, *f.longo@unical.it*

Last data update December2012

SNE – Simulation Notes Europe

Simulation Notes Europe publishes peer reviewed *Technical Notes*, *Short Notes* and *Overview Notes* on developments and trends in modelling and simulation in various areas and in application and theory. Furthermore SNE documents the ARGESIM Benchmarks on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions (*Benchmark Notes*). Special *Educational Notes* present the use of modelling and simulation in and for education and for e-learning.

SNE is the official membership journal of EUROSIM, the Federation of European Simulation Societies. A *News Section* in SNE provides information for EUROSIM Simulation Societies and Simulation Groups. SNE also offers possibilities for post-conference publication of contributions to conferences of the EUROSIM member societies.

SNE is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With Online SNE the publisher ARGESIM follows the Open Access strategy, allowing download of published contributions for free. Since 2011 Online SNE contributions are identified by an DOI (Digital Object Identifier) assigned to the publisher ARGESIM (DOI prefix 10.11128). Print SNE, high-resolution Online SNE, source codes of the *Benchmarks* and other additional sources are available for subscription via membership in a EUROSIM society.

Authors Information. Authors are invited to submit contributions which have not been published and have not being considered for publication elsewhere to the SNE Editorial Office. SNE distinguishes different types of contributions (*Notes*):

- *Overview Note* – State-of-the-Art report in a specific area, up to 14 pages, only upon invitation
- *Technical Note* – scientific publication on specific topic in modelling and simulation, 6 – 8 (10) pages
- *Education Note* – modelling and simulation in / for education and e-learning; max. 6 pages
- *Short Note* – recent developments, max. 4 pages
- *Software Note* – development in simulators, max 4 pages
- *Benchmark Note* – Solution to an ARGESIM Benchmark; basic solution 2 pages, extended and commented solution 4 pages, comparative solutions on invitation

Interested authors may find further information at SNE's website → www.sne-journal.org layout templates for *Notes*, requirements for benchmark solutions, etc.).

EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013

www.eurosim2013.info

EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013

www.eurosim2013.info

EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013

www.eurosim2013.info

EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013

EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM2013, we are soliciting original submissions describing novel research and developments in the following (and related) areas of interest: Continuous, discrete (event) and hybrid modelling, simulation, identification and optimization approaches. Two basic contribution motivations are expected: M&S Methods and Technologies and M&S Applications. Contributions from both technical and non-technical areas are welcome.

Congress Topics

The EUROSIM 2013 Congress will include invited talks, parallel, special and the poster sessions. The Congress topics of interest include, but are not limited to:

Intelligent Systems and Applications
Hybrid and Soft Computing
Communication Systems and Networks
Case Studies, Emergent Technologies
Workflow Modelling and Simulation
Web-based Simulation
Security Modelling and Simulation
Computer Games and Simulation
Neural Networks, Fuzzy Systems & Evolutionary Computation
Autonomous Mental Development
Bioinformatics and Bioengineering
Circuits, Sensors and Devices

e-Science and e-Systems
Image, Speech & Signal Processing
Human Factors and Social Issues
Industry, Business, Management
Virtual Reality, Visualization and Computer Games
Internet Modelling, Semantic Web and Ontologies
Computational Finance & Economics
Systems Intelligence and Intelligence Systems
Adaptive Dynamic Programming and Reinforcement Learning

Methodologies, Tools and Operations Research
Discrete Event /RT Systems
Mobile/Ad hoc wireless networks, mobicast, sensor placement, target tracking
Control of Intelligent Systems and Control Intelligence
Robotics, Cybernetics, Control Engineering, & Manufacturing
Energy, Power, Transport, Logistics, Harbour, Shipping and Marine Simulation
Semantic & Data Mining

Congress Venue / Social Events

The Congress will be held in the historic and magnificent City Hall in the heart of Cardiff, the capital city of Wales. The Gala Dinner will be held in the main hall of the National Museum of Wales. Social activities include visits to Cardiff Castle and Caerphilly Castle.

Congress Team: K. Al-Begain, A. Orsoni, R. Zobel, R. Cant, D. Al-Dabass; *kbegain@glam.ac.uk*

Info: www.eurosim2013.info