Komfortuntersuchungen an Gebäudeanlagen mittels gekoppelter Coarse-Grid-CFD- und Systemsimulation

Stefan Wischhusen[1], Timo Tumforde[1], Johannes Brunnemann[1]
[1]XRG Simulation GmbH, Harburger Schlossstraße 6-12, 21079 Hamburg

wischhusen@xrg-simulation.de

[redacted]

1 Einleitung

Für die optimale Auslegung der Klimaanlage eines Gebäudes ist das ganzheitliche Verständnis von Anlage und auszurüstenden Räumlichkeiten erforderlich. Entsprechend der Nutzungsart, Fensterfläche, Sonnenausrichtung und Tageszeit fallen unterschiedliche Heiz- und Kühlbedarfe in den Räumen an, denen die Klimaanlage mit einem geeignet konditioniertem Luftmassenstrom begegnen muss. Während auf der Anlagenseite die für einen Ganzjahresbetrieb erforderlichen Leistungen und die geeignete Regelung bestimmt werden müssen, soll von Seiten der Nutzer die Behaglichkeit innerhalb des Raumes jederzeit sichergestellt sein. Dazu sind räumlich aufgelöste Daten über Temperaturen und Luftgeschwindigkeiten erforderlich, wie der vorliegende Artikel belegt. Eine Abbildung der klimatisierten Zone mit nur einem Kontrollvolumen und folglich einer gemittelten Temperatur erweist sich hingegen in Situationen mit hohen inneren Kühllasten als nicht ausreichend.

In diesem Beitrag wird das dynamische Modell eines Gebäuderaumes und der dazugehörigen Klimaanlage vorgestellt. Dabei kommt ein von XRG Simulation neu entwickeltes Modelica-Raummodell zum Einsatz, das auf einer dreidimensionalen finiten Volumen-Diskretisierung der Navier-Stokes-Gleichungen basiert und thermische Strahlung, interne Wärmelasten sowie Festkörper im Raum berücksichtigt.

Dadurch wird es möglich, das gesamte System Raum-Luft-Klimaanlage-Wetter in einem einzigen Modelica Systemmodell abzubilden und dynamisch zu simulieren.

Anhand eines durch reale Wetterdaten vorgegebenen Beispielszenarios wird die dynamische Wechselwirkung der unterschiedlichen Systemteile veranschaulicht. Dabei sollen auch Vorteile und Limitationen des gewählten neuartigen Modellierungsansatzes diskutiert werden.

of values from other systems. To obtain higher accuracy, a higher extrapolation order may be used instead of constant extrapolation. However, this can lead to instabilities for loose coupling methods of Jacobi or Gauss-Seidl type, which can be coped with by stabilization as in [5] or combinative algorithms ([2]).

Apart from the fact that the restriction to cases with special requirements leaves a lot of unexploited methods for further investigations, the - reasonable - restriction to systems fulfilling certain requirements holds a few risks. There exist several software tools allowing the more or less easy coupling of certain simulators. Unfortunately, these are often used without further investigation on the consequences regarding numerical stability - like, for example, testing the system and used algorithms for the requirements necessary to guarantee stability.

4 Conclusion and Outlook

The present work has explained the basic principles of cooperative and multirate simulation with regard to methodology, consequences and risks. Although the research on simulator coupling goes back more than two decades, still a lot of questions remain unanswered and some of them even need to be posed.

One further aim of this work is the study of some of the open fields of research in this area such as coupling of highly contrastive simulation approaches, discrete and continuous partial models, further the processing of events, for example, the development of an algorithm which not only adapts the macro step size in consideration of previous steps but also allows steps back in time.

For systems with similar time constants and model properties (such as, for example, equal domains in physical models) but which might already modelled in different environments due to different people developing them, the integration of both systems in one tool should be considered for the sake of accuracy and speed instead of co-simulating them as a sometimes easier applicable, but more error-prone solution.

This implies that a means is intended to be established of determining applications where co-simulation - and which method exactly therein - is the most suited method but also scenarios where it may not be the best solution and other ways of satisfyingly simulating large systems have to be considered.

References

[1] M. Busch. *Zur effizienten Kopplung von Simulationsprogrammen*, Kassel University Press, Germany, 2012.

[2] S. Liang, H. Zhang and H. Wang. *Combinative Algorithms for the Multidisciplinary Collaborative Simulation of Complex Mechatronic Products Based on Major Step and Convergent Integration Step*. In: Chinese Journal of Mechanical Engineering, S. 355, 2011.

[3] R. Kübler and W. Schiehlen. *Two Methods of Simulator Coupling*. In: Mathematical and Computer Modelling of Dynamical Systems, S. 93-113, 2000.

[4] M. Arnold and M. Günther. *Preconditioned Dynamic Iteration for Coupled Differential-Algebraic Systems*. In: BIT Numerical Mathematics, S. 1-25, 2001.

[5] M. Arnold. *Stability of Sequential Modular Time Integration Methods for Coupled Multibody System Models*. In: Journal of Computational and Nonlinear Dynamics, S. 031003, 2010.

Acknowledgement

This work is part of the *MeCoAMuCo* project funded by the Austrian Research Promotion Agency (FFG).

tems. This procedure is repeated until a given tolerance is obtained. Although in the first place this method seems to be very promising, it also holds a few difficulties. On the one hand, due to the iterations in every step, the execution takes a lot of computational time. On the other hand, in order to allow these iterations, not only does the overall algorithm (*master algorithm*) need to step back but the partial algorithms need to allow steps back and recalculations. The transfer of those newly calculated values to the master involves additional challenges, especially if different, maybe already prespecified software tools are combined.

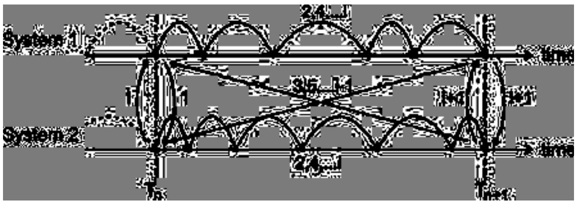

Figure 4: Waveform iteration method for two participating systems.

2.2.4 Other

Furthermore, there are mixed methods not fitting into the structure above, like the asynchronous algorithm proposed by [2]. In this approach, subsystems do not necessarily have any time steps in common but are simulated sequentially. The next system to be simulated is determined by the smallest current simulation time. In the scenario above, both systems start at simulation time T_0. Without loss of generality, system 1 is integrated first using interpolated values for variables from system 2. Now that system 1 has progressed one step and system 2 is still at T_0, therefore at a smaller simulation time, system 2 is required to make a step, using interpolated (or, if required, extrapolated) values for variables from system 1. If, as in the case of the depicted scene, the simulation time in system 2 is still smaller than in system 1, the next step is calculated for system 2. As soon as the simulation time of system 2 has exceeded the simulation time of system 1, the next step is calculated for system 1 with its own chosen time step and so on. This method takes more time since there are no parallel executions and more data exchanges than in Jacobi or Gauß-Seidl type methods, but the results remain stable and more accurate.

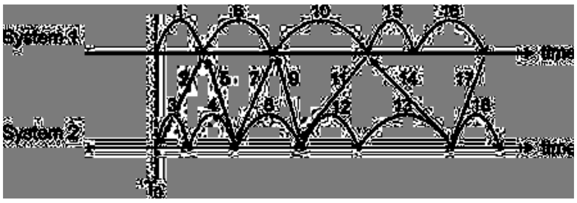

Figure 5: Illustration of an asynchronous coupling of two partial systems.

3 Developments, Challenges and Risks

In addition to the definition of these coupling methods, they have to be investigated for the numerical effects they have on separately nicely working integration algorithms. Since the 1990s there has been a lot of research on the stability of multirate simulation. As the field of numerics of differential equations and differential algebraic equations itself is already a seemingly infinite area of research, the combination of different methods out of this area is even harder to investigate from a general point of view. Hence many studies on stability of coupling methods are done on systems with certain limitations (see for example [5]) and introducing methods for stabilization ([3, 4]).
In [3] it is shown that zero-stability can not be guaranteed for loose-coupling co-simulation if there exist algebraic loops between the participating subsystems. In addition, two iterative schemes are described which always guarantee stability. On the other hand, a method introducing a filter causing the elimination of algebraic loops and hence again warranting zero-stability is shown. However, since the second method modifies the system itself, it actually cannot be seen as a plain coupling method.

[5] shows that for sequential algorithms, the order in which the subsystems are executed is important so a contractivity condition necessary for stability is fulfilled.

There are also studies on the order of extrapolation for the overall system are fulfilled. This also means that all participating systems have to set the same time steps throughout the whole simulation. These iterations allow high simulation accuracy but on the other hand lead to huge computational costs.

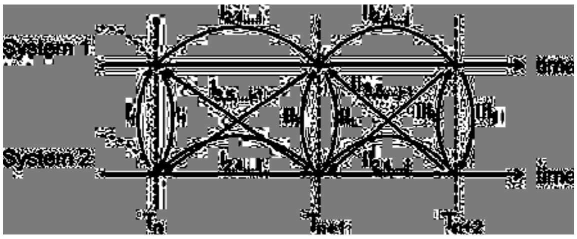

Figure 1: Illustration of the data exchange between two strongly coupled simulation algorithms.

2.2 Loose Coupling Methods

Loose coupling co-simulation (in literature also referred to as weak coupling co-simulation or only co-simulation, see also [1]) means that the partial systems are integrated separately with individual time steps and synchronization at given macro time steps, which implies that values from other systems have to be extrapolated (and sometimes interpolated) between two synchronization references. Loose coupling co-simulation enables a faster simulation of distributed systems in comparison to strong coupling co-simulation, but it only makes sense if the systems are only loosely coupled themselves, meaning minor dependencies on variables from other systems (see equation (2)) as in contrast to strong coupling methods, loose coupling algorithms are prone to error accumulation due to the required extrapolation.

$$\left\|\frac{\partial f_1}{\partial y_2}\right\| \ll \left\|\frac{\partial f_1}{\partial y_1}\right\| \text{ and } \left\|\frac{\partial f_2}{\partial y_1}\right\| \ll \left\|\frac{\partial f_2}{\partial y_2}\right\| \qquad (2)$$

Within algorithms for loose coupling, further distinctions can be made.

2.2.1 Gauß-Seidl Type Methods

Gauß-Seidl Type methods require sequential executions of partial systems. One system is integrated with its individual time step for one macro time step using extrapolated values for variables from other systems. The next system to be executed can then apply interpolation to gain the values from the first system and so on until all systems have been simulated for the current macro time step. The same procedure is followed for all subsequent macro time steps. The stability of Gauß-Seidl Type methods depends highly on the execution order of the participating subsystems.

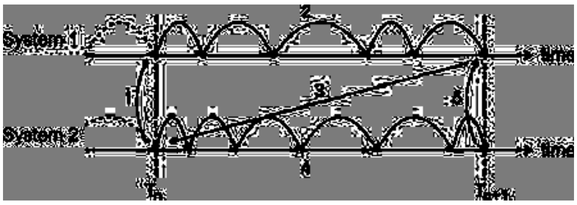

Figure 2: Sequential execution and data exchange for two Gauß-Seidl type coupled systems during one macro step.

2.2.2 Jacobi Type Methods

Applying algorithms of so-called Jacobi Type, all participating systems are simulated in parallel during each macro time step. At each synchronization reference, values needed from the respective other systems are exchanged and further extrapolated for usage in the next macro step. Between two synchronization references, every subsystem uses its own time step.

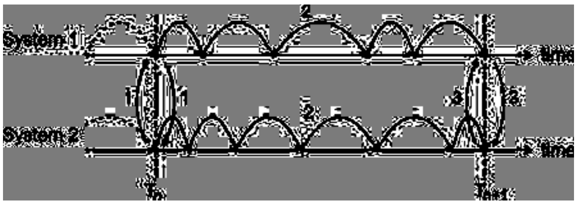

Figure 3: Communication between two systems loosely coupled according to the Jacobi type method.

2.2.3 Waveform Iteration Methods

So-called waveform iteration methods are an iterative enhancement of Gauß-Seidl Type methods. Again, all partial systems are simulated in parallel for one macro time step but before the next macro step follows, the partial systems are recalculated for the same step using values gained by interpolation from the other sys-

On the Methodolgy of Cooperative and Multirate Simulation

Irene Hafner[1], Niki Popper[1], Felix Breitenecker[2]
[1]dwh Gmbh, Neustiftgasse 57-59, 1070 Wien
[2]Technische Universität Wien, Wiedner Hauptstraße 8-10, 1040 Wien
irene.hafner@dwh.at

[redacted]

1 Introduction

In times of increasing complexity of systems aimed to be modelled and simulated, it has become more and more important to be able to simulate models with partial models of different complexity and differing requirements regarding solver algorithms step sizes and other model-specific properties. To meet these requirements, models of such complexity are approached via cooperative or multirate simulation algorithms.

2 Co-Simulation - State of the Art

Cooperative simulation (abbreviated co-simulation), in literature also called modular simulation, distributed time integration or simulator coupling, stands for the coupled simulation of at least two partial systems. This can be realized via different simulators, solver algorithms or at least individual solver step sizes. Every co-simulation with more than one rate of step sizes, i.e. at least two participating systems use different, individual step sizes for the calculation of internal values, is called multirate simulation. The crucial part of multirate co-simulation is the approximation of accuracy loss resulting from the extrapolation in between synchronization references.
In general, co-simulation can be divided into loose coupling and strong coupling methods Depending on the interdependencies between the partial systems. For further inspections, let us consider a system divided into two partial systems described by (1):

$$\begin{aligned} \dot{y}_1 &= f_L(y_1, y_2, t), \quad y_1(t_0) = y_{1,0} \\ \dot{y}_2 &= f_A(y_1, y_2, t), \quad y_2(t_0) = y_{2,0} \end{aligned} \tag{1}$$

2.1 Strong Coupling Methods

If high accuracy is required but co-simulation is necessary not due to highly differing time constants but different modelling approaches and requirements regarding the implementation, strong coupling will be considered. Using strongly coupled algorithms, solutions are iterated in every step until given error tolerances