Deficiencies of Modelica and its simulation environments for large fluid systems

Kilian Link, Haiko Steuer, Axel Butterlin
Siemens AG, Energy Sector, Fossil Power Generation, Energy Solutions, Erlangen, Germany
{kilian.link, haiko.steuer, axel.butterlin}@siemens.com

Abstract

Modeling of large fluid systems requires in-house (specialized) tools, since applicability of Modelica and existing environments is limited.

Nevertheless Modelica is a very powerful and descriptive modeling language, which is best suited for physical modeling in a heterogeneous environment. Its object oriented approach, the built-in documentation and the availability of commercial and free libraries justifies the decision for Modelica as the preferred modeling language within Siemens Energy.

For an appropriate analysis of transient power plant processes, there often are large fluid systems to be modeled, i.e. there can be several thousand states. For such plant models, we use our in-house tool Dynaplant (DP), which is specialized for large fluid systems. A comparison between DP and Dymola[1] reveals some deficiencies of the Modelica world concerning performance and plant model construction: Especially, successive initialization and sparse matrix solvers are important features in need.

Keywords: Fluid simulation; workflow; performance

1 Introduction

Providing clean and affordable electric power to all human beings is one of the most ambitious challenges in our world. Technological efforts and innovations lead to high effective and environmental gentle methods of power production. Here, transient simulation has become an inevitable tool. Especially matters of unit safety in respect to material stresses of components do require detailed modeling and computational intensive dynamic simulations.

A typical use case which requires a detailed transient simulation of a power plant is a dynamic stability analysis. In the remaining of the introduction the system under consideration and the use case are introduced.

1.1 Plant model

The system under consideration is a special kind of evaporator modeled via several tubes with a total length of some 100 meters. The tubes are filled with water/steam, mostly in the two-phase region. Most of the tubes are heated by a hot flue gas flow through the tube metal wall.

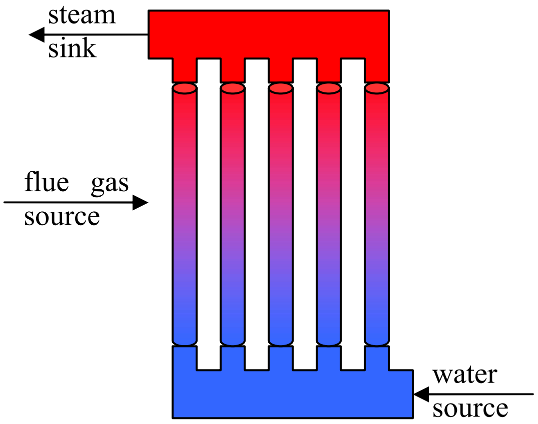


Figure 1 Evaporator part model composed of a splitter, five heated tubes and a mixer.

Usually the plant model is composed of two of the evaporator parts shown in Figure 1. For the purpose of this article, we will focus on one single evaporator part only.

A detailed one-dimensional hydrodynamic tube model is used. The resulting differential-algebraic equation system (DAE) is stiff and non-linear. In addition, there is a narrow spatial discretization along the tubes, such that up to several thousand states have to be considered.

1.2 Use case “dynamic stability analysis”

The purpose of a stability analysis is to avoid spontaneous mass flow fluctuations in the evaporator, which could lead to material fatigue [6]. Therefore, a large fluid system has to be built up using component models with detailed geometry parameters and many states.

At first a start point steady state has to be established. Secondly, a dynamic experiment has to be performed starting from this initial steady state with a temporary perturbation. The perturbation is put into the system via a temperature shift of the inlet flue gas. Here, if the system relaxes to the steady state again, it is stable. Otherwise, the evaporator design should be modified.

Our in-house tool DP is specialized for such kind of applications. Its component library is limited but suitable for a dynamic stability analysis. Dymola, on the other hand, is a multi-purpose simulation environment. It is at present the only tool based on Modelica, which supports fluid systems including Modelica.Media and Modelica_Fluid elements.

Both tools will be compared with respect to the reference work flow which covers a use case similar to a stability analysis with a reduced plant model.

2 Reference workflow (using DP)

In this section, we will introduce a typical DP workflow, which is very similar to a dynamic stability analysis. It covers the plant build-up, initialization and a dynamic experiment.

2.1 Plant build-up with successive initialization

The scope of the plant model is a single evaporator part as shown in Figure 1. It is composed of a water source, a splitter for water, several parallel heated tubes, a mixer for steam, a steam sink and a flue gas flow. The flue gas heats the tubes via their metal walls. In each of the parallel tubes, due to the splitter and mixer, the same pressure loss but different heating is applied, such that a certain mass flow distribution will arise.

We will cover two different system sizes, which distinguish only via their spatial discretizations (“small” resp “large” system). In DP, the total plant model results in a DAE system with 101 algebraic and 440 (resp 895) dynamic degrees of freedom for the small (resp large) system system.

The plant model can be edited in DP as follows: Adding of components per drag & drop, editing its parameters and setting start values at the connection points can be carried out via a graphical user interface. The build-up of the plant model takes place using successive initialization:

(A) Starting with just a few components and specifying start values at the connection points, a dynamic simulation with constant boundary conditions can be performed. The inner degrees of freedom of the initial state are computed using interpolations of the connection values. After an appropriate simulation time, the system will be relaxed into a steady state.

(B) The resulting plant model including steady state variables is loaded in DP.

(C) Here, the plant model can be modified. Further aggregates can be added. At new connections, initial state information has to be specified. For new aggregates, the inner state variables are unknown, such that they will be interpolated from these connection values.

(A) In a next simulation, the “old” components start with already computed state variables, and the “new” components start with interpolated values. After this run, there is a new steady state for this larger plant model, in which the “old” states may be modified.

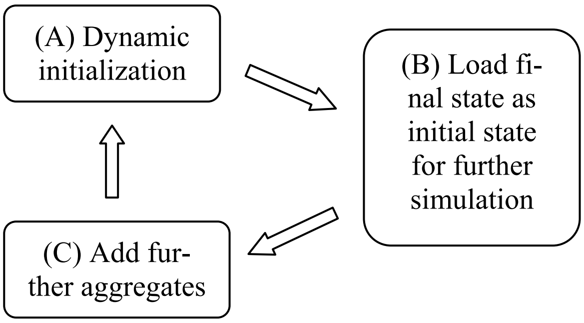


Figure 2 Successive initialization for plant build-up

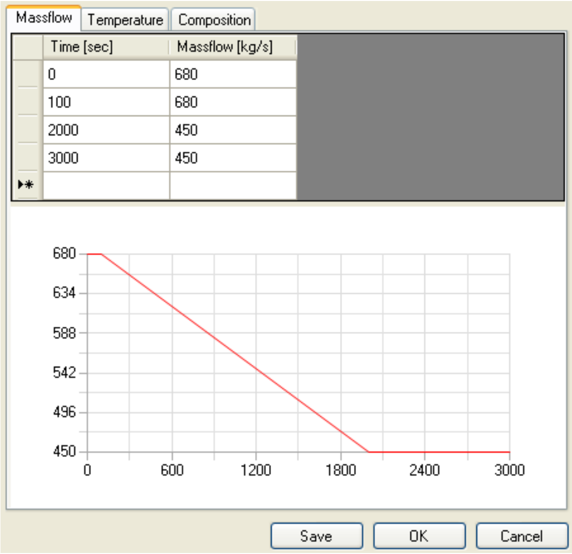
Such, step by step, the plant model can be enlarged in order to successively build up the steady state of the total plant model (see Figure 2).

2.2 Dynamic experiment

Finally, the constructed plant model with a steady state can be used as starting point of a dynamic experiment, which is specified via certain time-dependent boundary conditions.

Depending on the dynamics of the experiment, the DAE system may be more or less difficult to solve. DP uses a fast and stable DAE solver with sparse Jacobian. The partial derivatives are defined in special sub-model functions. The total Jacobian structure is built only once at the beginning of the simulation taking into account potential flow reversals. The water/steam property computations are performed using fast table based functions [5].

The boundary conditions are used to force the evaporator to switch from 100% load case to a part load case. Therefore, the water source changes its mass flow rate and specific enthalpy, the steam sink changes its pressure and the gas source changes its mass flow rate and temperature.



Time [sec]	Massflow [kg/s]
0	680
100	680
2000	450
3000	450

Figure 3 DP parameter dialog of the gas source for specifying dynamic boundary conditions.

In DP, dynamic boundary conditions are set in the boundary components using time table parameters (see Figure 3). The system simulation time is 3000sec.

3 Comparison: DP-Dymola

3.1 Application range

Unlike general purpose tools as Dymola the development of DP is in line with the clearly defined use case: Modeling of one-dimensional hydrodynamics of large fluid systems. For a larger application range two main features are missing,

- Handling of hybrid systems, i.e. discrete states, including advanced event handling.
- Model libraries and the ability for user defined components.

3.2 Modeling and Simulation

The explicit implementation of the Jacobian in DP is time-consuming and error-prone.

Coping with very large systems, as described above, heavy difficulties during computing the initial state may occur, where iterative assembling and initialization of sub models is missing. For the recent case of hard to find steady state solutions, the support for sub model initialization and setting the states to known values is a key feature to gain success. Currently setting fixed start values for parts of the model is a time consuming task in Dymola. This is strongly related to the Modelica specific requirement to define initial values inside of models, while DP allows the definition of initial values in the connection set. In addition identical initial values are not propagated or checked for consistency, hence it is difficult to find out which initial or guess values are in use.

3.3 Performance

In this chapter the performance of DP and Dymola [1], the by far best suited Modelica tool for fluid simulations is compared for the dynamic experiment of the reference work flow. In Dymola, evaluation of parameters as well as the “NoGuard” `userdefs.h` option is used. The Modelica model is slightly simplified, since some details of the DP tube model are not yet implemented in Modelica. It is based on the Modelica_Fluid interfaces [2].

	small system	**large system**
Dymola	691 sec	6780 sec
Dynaplant	46 sec	80 sec

Table 1 CPU times for reference dynamic experiment. The small/large system has about 400 resp. 800 continuous time states.

The main results of the performance comparison are that the Dymola CPU time is very large and critically depends on the system size. This is mainly due to

- Missing high performance water/steam property calculation [5].
- Missing sparse matrix solver.

4 Conclusions

The comparison with Dynaplant reveals features in need for large fluid systems in Modelica simulation environments. Below, they are sorted by priority. The important features in need are tool related. The other items cannot clearly be addressed to the tool vendors alone, since enhancing Modelica will also be necessary. Our intention is however, that the further development of Modelica *and* tools may consider the demands of large fluid system simulations.

4.1 Important features in need

- For generating a steady state successive initialization, i.e. "reload" of old simulation results with component specific states, should be possible.
- A sparse matrix DAE solver is necessary.

4.2 Further improvements

- A standardized solver interface would simplify the usage of external solvers An External Model Interface for Modelica: http://www.modelica.org/events/modelica2008/Proceedings/sessions/session5f.pdf[3], [4].
- Pre-compiled sub-models would reduce the compilation time of large models.
- High performance water/steam property calculation.

4.3 Nice to have features

Both nice-to-have features are related to the setting of guess values used in the initialization routine.

- Redundant specifying of guess values for the ports inside the sub-models may be replaced by specifying values at the connection points.
- Propagation of guess values would further simplify the setting-up of large plant models. This can be done using simple rules for flow variables or by using more sophisticated information from pre-compiled sub-models.

References

[1] Dymola7.1 http://www.dynasim.se/index.htm

[2] Modelica_Fluid: http://www.modelica.org/libraries/Modelica_Fluid

[3] An External Model Interface for Modelica: http://www.modelica.org/events/modelica2008/Proceedings/sessions/session5f.pdf

[4] Modelisar: http://www.itea2.org/public/project_leaflets/MODELISAR_profile_oct-08.pdf

[5] Butterlin, A.; Schiesser, D.; Steuer, H.: Usage of Water & Steam Properties in Computational Intensive Dynamic Simulation, ICPWS XV, September 2008.

[6] Franke, J., Brückner, J.: Dealing with tube cracking at Herdecke and Hamm-Uentrop in Modern Power Systems, October 2008.