

Application of Richardson Extrapolation to the Co-Simulation of FMUs from Building Simulation

Christoph Clauß, Kristin Majetta, Richard Meyer

Fraunhofer IIS EAS, Zeunerstraße 38, D-01069 Dresden, GERMANY

christoph@clauss-it.com, {kristin.majetta, richard.meyer}@eas.iis.fraunhofer.de

Abstract

The application of the FMI technology gains ground in building simulation. As far as specialized tools support the FMI simulator coupling becomes an important option to simulate complex building models. Co-simulation needs a master algorithm which controls the communication time steps as well as the signal exchange between FMUs. Often a constant communication step size is applied chosen by the user. The Richardson extrapolation approach allows variable master step sizes. An extension of this approach is presented, and the method is applied to both academic test examples as well as examples of building simulation which co-simulate FMUs from NANDRAD and SimulationX. Although variable step size control could improve the performance this cannot be observed at the building simulation examples presented. But Richardson extrapolation turns out to guarantee finding an appropriate step size at the prize of downgraded performance.

Keywords: Building Simulation, FMI, Co-Simulation, Richardson Extrapolation, Variable Time Step Size

1 Introduction

In order to reduce the primary energy production by both reduction the consumption in buildings and growing the portion of renewable energy a much higher knowledge of the dynamic energy and mass fluxes is essential. Especially the daily and hourly fluctuations of sun and wind based energy generation require detailed dynamic considerations by simulation. Since the first publication of the FMI standard well established simulation tools have been improved to support FMI both for model exchange and for co-simulation. This allows the combination of dedicated tools as well as their model libraries which contain results of a long period of investigations. Basing on tool as well as model combination by co-simulation a big step to generate detailed simulation results was managed.

Modern buildings typically are divided into the “proper” building (walls, roof, windows ..., thermal, hygric behavior), HVAC (heating, ventilation, air conditioning devices...), and often a central acting control software (building energy management systems, BEMS). Within

the German research project EnTool:CoSim the tools NANDRAD [Nicolai, 2012] for building simulation, and the Modelica simulation tool SimulationX [ESI ITI GmbH] for mainly HVAC and control simulation were prepared to export FMUs for co-simulation. Since the FMI standard does not offer dedicated master algorithms which control the coordinated simulation of different FMUs, master algorithms have been investigated and implemented [Bastian. 2011]. So far master algorithms with a constant step size were considered mostly. FMUs generated from SimulationX for HVAC models often have a higher performance than building FMUs generated by NANDRAD. Furthermore, different “activity ranges” can be recognized (less activity at night, weekend, less heating in summer ...). The required time intervals to be simulated can be very long (years). Often the user is overstrained to define a suitable master step size, especially if the co-simulation method shall leave the research area to be applied in building practice. These issues as well as the hope for improved performance suggested the investigation of variable step size master algorithms, and furthermore asynchronous algorithms. In this paper results of the investigation of synchronous variable step size algorithms are presented. After the introduction of Richardson extrapolation methods some small academic test examples are presented, followed by three building simulations of different complexity which apply Richardson extrapolation.

2 Algorithms

The task of co-simulation of m Simulators (FMUs) can be described according to (1) with S_i , ($i = 1 \dots m$) being the simulators. Q_i and P_i are matrices which project the output values of S_i into, and the input values of S_i out of the vector of coupling values $x(t) \in [0, T] \rightarrow R^n$. The argument $P_i x$ is missing if the simulator S_i lacks input values. In [Petridis, 2015] this description is derived, and basic solution methods for cycles are presented there.

$$\begin{cases} x = \sum_{\substack{i=1 \\ Q_i \text{ exists}}}^m Q_i S_i(P_i x) & Q_i \text{ exists} \\ S_i(P_i x) & Q_i \text{ does not exist} \end{cases} \quad (1)$$

The projector P_i takes the input values of the simulator S_i out of the coupling values x . The simulator S_i calculates its output values which are written into the coupling value vector x via Q_i . Since all coupling values x are output values of exactly one simulator, and since no output value is input of the same simulator the summarizing is possible. This is described in the first line of (1). The second line clarifies that simulators without output values also have to be called.

Due to the FMI-intention the interval $[0, T]$ is divided into communication intervals $[0, T] = \cup [tc_k, tc_{k+1}]$ with tc_k being the k^{th} communication point, and $hc_k = tc_{k+1} - tc_k$ the communication step size.

The communication step size can be chosen synchronously by calling each simulator with the same step size, or asynchronously by using individual step sizes for each simulator. Otherwise, the step size can be constant, or it can vary. Both properties are independent from each other. In this paper Richardson extrapolation is applied as a method of variable but synchronous step size control.

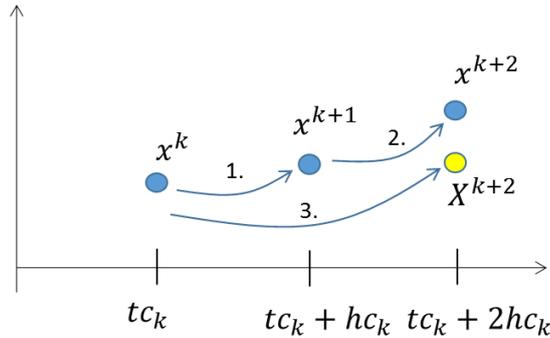


Figure 1. Richardson extrapolation

According to [Hairer, 1993], [Schierz, 2013] the **Richardson extrapolation** algorithm consists of the following steps (Figure 1):

1) Start at tc_k , and simulate two steps using the step same size hc_k which results in the coupling variables x^{k+2} . At the first step hc_0 is provided by the user. For following steps hc_k is calculated by previous steps.

2) Roll back to tc_k and simulate one step using the doubled step size $2hc_k$ which results in X^{k+2} .

3) Calculate an individual error estimation for each coupling variable ($j = 1 \dots n$) with q being the degree of the interpolation polynomial of input values:

$$errest_j = (x_j^{k+2} - X_j^{k+2}) / (1 - 2^{q+1}) \quad (2)$$

4) Calculate a total error estimation according to

$$ERR = \sqrt{\frac{1}{m} \sum_{j=1}^m \left(\frac{errest_j}{ATol_j + RTol_j |x_j^{k+2}|} \right)^2} \quad (3)$$

with $ATol$ and $RTol$ being absolute and relative error limits which can be chosen individually for each coupling variable.

5) Calculate the new step size h_{new} according to

$$h_{new} = hc_k \min \left\{ Q_{max}, \max \left\{ Q_{min}, \frac{Q_s}{\sqrt[p]{ERR}} \right\} \right\} \quad (4)$$

The heuristic values $Q_{max} \in [1.5, 5]$, $Q_{min} \in [0.2, 0.5]$, and $Q_s \in [0.8, 0.9]$ prevent too “strong” step size variations. p has to be $p = q + 1$, if there are no algebraic dependencies between inputs and outputs, otherwise $p = q + 2$ is necessary.

6) If $ERR \leq 1$: Both time steps are accepted, $tc_{k+2} = tc_k + 2hc_k$, $hc_{k+2} = h_{new}$, $k := k + 2$, go to 1)
If $ERR > 1$: Both time steps are rejected $hc_k = h_{new}$, go to 1)

Due to the steps 2) and 6) the FMUs must be able to be set back to a former communication time step. Otherwise Richardson extrapolation cannot be applied. This Richardson extrapolation algorithm assumes that the simulators S_i solve DAEs, and their output variables depend on input variables. If output variables do not depend on input variables, but on any purely time depending formula or algorithm, then intermediate time steps do not at all influence the results. Even if simulators without inputs solve DAEs their output values are in general not influenced by the master communication step size. For such components l of the coupling variables $errest_l = 0$ follows because of $x_l^{k+2} = X_l^{k+2}$. If the components l cover the vector of coupling values totally ERR becomes zero, and Richardson extrapolation cannot be applied reasonably. In such cases the step size of the output values has to be chosen such that simulators which take the output values as inputs can reconstruct the output values without losses. The step size must meet the sampling theorem [Kotelnikov, 1933]. Therefore, a maximum communication step size could be calculated if the fastest frequency component of the output values is known. Since this is not the case in general, the step size is controlled similar to classic predictor-corrector approaches [Hairer, 1993] by simply keeping the deviation from linear extrapolation small (Figure 2). The linear extrapolation of the first step is compared with the values of the second step to generate an error estimation. This algorithm called **Linear extrapolation** algorithm throughout this paper is identical to Richardson extrapolation except the error estimation formula (2) of step 3), which is replaced by

$$errest_j = x_j^{k+2} - 2x_j^{k+1} + x_j^k \quad (5)$$

The double sized step to calculate X^{k+2} is no more necessary there.

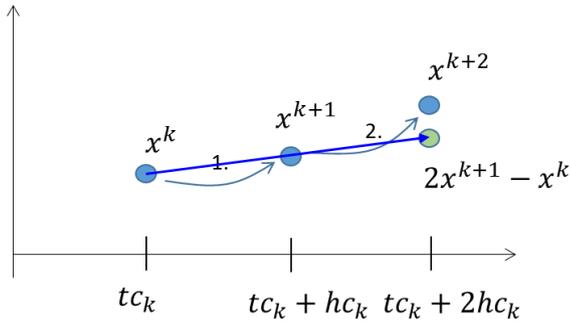


Figure 2. Linear extrapolation

This way the more communication points are inserted the more the coupling variable behavior is nonlinear. If the behavior is linear temporarily or generally then ERR also becomes zero. For such cases ERR gets a small positive minimum value. It is known that this extrapolation method on its own gives no reliable step size control if DAEs are solved. Therefore, it is combined with the Richardson extrapolation method to vanishing errors in Richardson extrapolation reasonably: At first $errest_j$ is calculated according to Richardson, formula (2) within step 3). If $errest_j$ vanishes ($|errest_j| < 1.e - 12$) then $errest_j$ is replaced by the linear extrapolation error estimation according to (5). This approach is called **Extended Richardson extrapolation** method in this paper.

In summary three methods with variable step size control are available:

- Richardson extrapolation comprising formula (2)
- Linear extrapolation algorithm comprising formula (5) instead of formula (2), without the double sized step. It does not guarantee reliable step size control in case of DAEs.
- Extended Richardson extrapolation as a combination of both of them

These algorithms are implemented in the "EAS master tool" [Petridis, 2015] which is a proprietary tool for testing master algorithms.

3 Simple Test Examples

The following simple academic examples were developed to test features of the co-simulation algorithm. They illustrate the implemented Richardson extrapolation algorithms.

3.1 Precision Test Example

The precision test example presented in [Petridis 2015] consists of three FMUs according to

Table 1. Each table line describes the equations of one FMU. The example is designed such that $y(t)$ is zero.

Table 1. Equations of the precision test example

Input	Equations	Output
x_2	$x_1 = -x_2$	x_1
x_1	$\frac{\partial x_2}{\partial t} = x_1, x_2(0) = 1$	x_2
x_2	$e^{-t} - x_2 = y$	y

The example contains one cycle which is treated by Newton's method. Since $x_2(t)$ is differentiated, and all coupling values depend on $x_2(t)$, the pure Richardson extrapolation algorithm offers correct results. Two cases of different tolerances are regarded (care for formula (3) in the algorithm):

- Case 1 (usually default values): $ATol = 1.e - 6$, and $RTol = 1.e - 4$
- Case 2 (higher precision): $ATol = 1.e - 8$, and $RTol = 1.e - 6$

The limitation of the Richardson step size variation is kept far (min 1.e-5 s, max 5 s) to not restrict the step size choice.

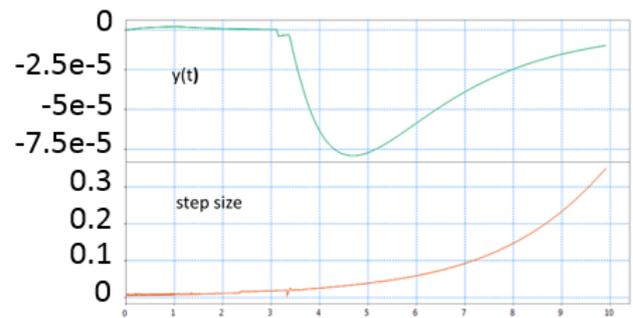


Figure 3. Precision test example results case 1

In case 1 (Figure 3) the accepted step size is growing which is expected since the solution converges to the steady state. $y(t)$ is numerically near zero.

If the tolerances demand a higher accuracy (case 2, Figure 4) $y(t)$ is closer to zero. The step size starts not significantly smaller than in case 1 but does not increase as fast as in case 1. That indicates that the smallest possible accuracy seems to be reached using the step size 0.1 s.

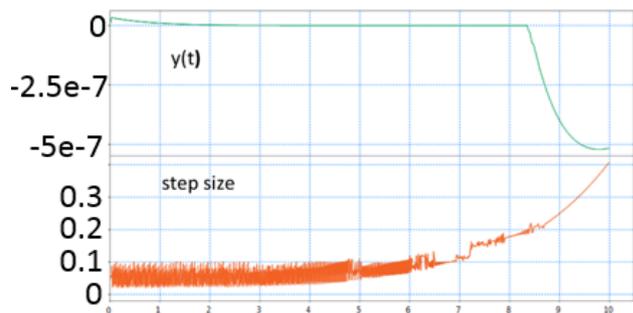


Figure 4. Precision test example results case 2

This Richardson extrapolation study advises to choose a constant step size of about 0.01 s if a constant step size algorithm should be used. Doing that Figure 5 shows the reasonable result.

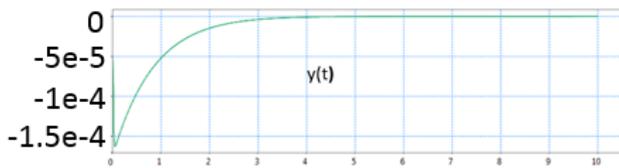


Figure 5. Precision test example using the constant step size of 0.01 s

When setting the CPU time necessary in the Richardson case 1 to be one, the normalized CPU times are listed in Table 2. Although the step size in Richardson extrapolation exceeds 0.01 s clearly in the second half of the time interval, the constant step size simulation is significantly faster than Richardson extrapolation. The reason is that Richardson extrapolation simulates the whole interval more than twice.

Table 2. Normalized CPU time comparison

Richardson case 1	Richardson case 2	Constant step size 0.01 s
1	1.14	0.42

3.2 Linear System of Equations

The linear system of equations with time dependent system matrix according to Table 3 was already presented in [Petridis 2015].

Table 3. Linear system of equations

In	Equations	Out
	$r_1 = 1, r_2 = t, r_3 = 1$	r_1, r_3, r_3
x_2, x_3, r_1	$3x_1 + (0.1 + t)x_2 + 0.2x_3 = r_1$	x_1
x_1, x_3, r_2	$0.1x_1 + 3x_2 + (0.1 + t)x_3 = r_2$	x_2
x_1, x_2, r_3	$(0.1 + t)x_1 + 0.2x_2 + 4x_3 = r_3$	x_3
x_1, x_2, x_3	$x_1 + x_2 + x_3 = y$	y

Applying Richardson extrapolation (Figure 6) the step size increases since each variable depends on the point in time only. The step size does not affect the Richardson error calculation. The values calculated at each time step are correct indeed, but there are too less time steps generated. This drawback is overcome applying the Linear extrapolation time step method instead of Richardson extrapolation (Figure 7). This example shows that the combination of both variable step size methods is necessary. The Extended Richardson extrapolation method shows the same results as the Linear extrapolation method.

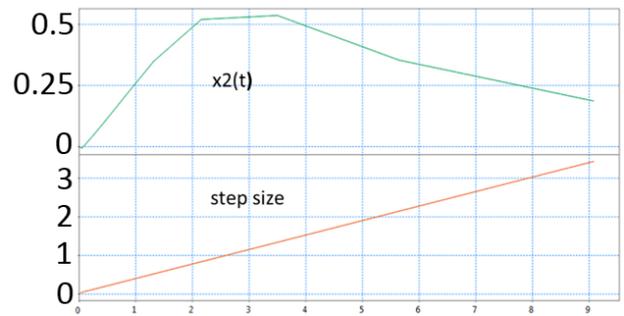


Figure 6. Linear system of equations example results applying Richardson extrapolation

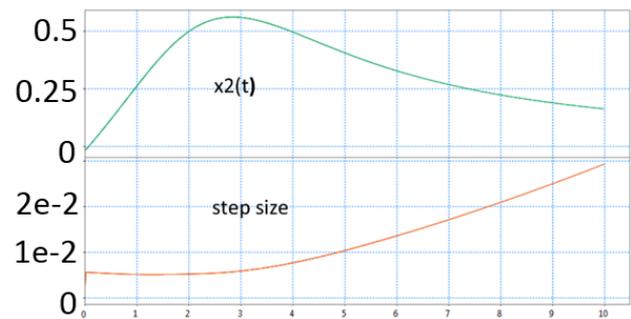


Figure 7. Linear system of equations example results applying the Linear extrapolation method

3.3 Touching Mass Example

Similar to a bouncing ball the touching mass example [Klein 2015] switches on a stiff spring as soon as the mass touches the base. This accelerates the mass into the opposite direction, and the stiff spring is switched off when the base is left.

Table 1 shows the equations of this example, separated into two parts (FMUs). The spring part contains switching as well as the calculation of the stiff spring force f .

Table 4. Equations of the touching mass example

Input	Equations	Output
f	$\frac{\partial s}{\partial t} = vs(0) = 10$ $\frac{\partial v}{\partial t} * 0.1 = f - 0.1f(0) = 0$	s
s	$f = \begin{cases} -1e6 * s & s < 0 \\ 0 & s \geq 0 \end{cases}$	f

All tests reported use Newton’s method to calculate cyclic equations. The constant step size approach needs a very small step size to handle the reversal process correctly. Figure 8 shows that the step size of 1.e-5 calculates a nearly correct result.

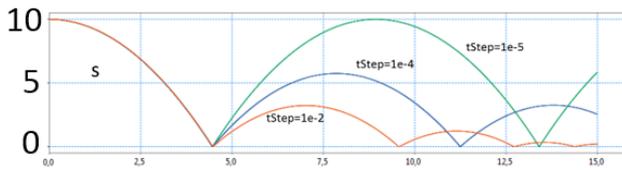


Figure 8. Course of $s(t)$ at different communication step sizes

Using Richardson extrapolation (max. step size 0.1 s, min. step size 1.e-15 s, start step size 1.e-3 s, absolute tolerance ATol 1.e-15, relative tolerance RTol 1.e-8) the maximum step size is used. Only to calculate the reversal regions (see Figure 9) the step size decreased down to about 1.e-7 s (Figure 10).

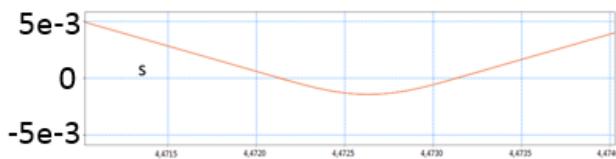


Figure 9. Reversal region calculated by Richardson extrapolation

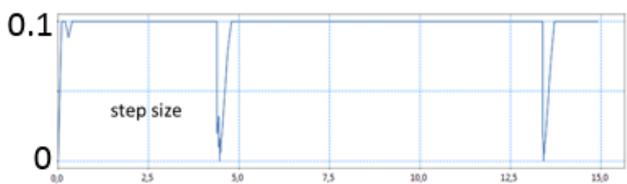


Figure 10. Varying step size using Richardson extrapolation

To calculate the reversal region correctly a small step size is necessary, otherwise the result becomes useless. The usage of the necessary small step size over the whole interval as a constant step size increases the CPU time abnormally (Table 5). The varying step size provided by Richardson extrapolation is the method of choice. The Linear extrapolation method does not succeed since the step size does not increase after decreasing. The reason is still to investigate.

Table 5. Normalized CPU time comparison for touching mass example

Step size 0.01 s	Step size 1e-4 s	Step size 1e-5 s	Richardson
1	56	572	1.3

If the spring constant is 1.e10 instead of 1.e6, the constant step size 1.e-5 s does no more show the correct result (Figure 11). Richardson extrapolation test calculates the expected result within a short CPU time of less than 1s. The step size decreases to 1.8e-9 s.

This example demonstrates the importance of the Richardson extrapolation method.

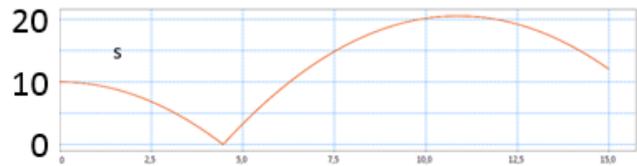


Figure 11. Course of $s(t)$ with spring constant 1.e10, constant step size 1.e-5 s

4 Application in Building Simulation

Three examples from building simulation are presented to study the obviousness of Richardson extrapolation at realistic use cases. The examples of different complexity consist of two FMUs each. One FMU describes both the heating facility and heating control modeled using Modelica and SimulationX [ESI ITI GmbH]. The other FMU of each example contains the building physics description as well as weather data using the NANDRAD tool [Nicolai, 2012] which solves PDEs. Table 6 shows roughly the structure of all examples.

Table 6. Macrostructure of the building examples

Input	Equations, Tool	Output
Room temperature, weather	DAEs (heating, heating control), SimulationX, Green Building, Modelica	Heat flow
Heat flow	PDEs (building physics), NANDRAD	Room temperature, weather

4.1 Single Room

The single room model is based on a small conference room (up to 20 people, Figure 12).



Figure 12. Meeting room on which the model is based

The room has a floor space of 52 m², and a height of 3,3 m, one outer wall (west oriented), at which ambient conditions are applied. The boundary temperature of the opposite wall and the ceiling is set to constantly 18 °C, for the other walls to 20 °C. The four walls consist of a

heavy construction from clinker bricks and plastering, both ceiling and floor from lightweight concrete. Furthermore, an intermediate ceiling is included made of papier mâché. The room is equipped with a radiator heating operating a supply temperature of 70 °C. A valve that can operate continuously between valve position 0 and 1 regulates its volume flow.

Figure 13 shows the single room model. It consists of a thermal zone and heating facilities. The green framed part of Figure 13 contains both the thermal zone and the weather source, both modeled using NANDRAD, and therefore placed within the NANDRAD-FMU “thermal zone” which solves PDEs. The remaining model part of Figure 13 contains the heating facilities including the controller both taken from the GreenBuilding library [EA Systems Dresden, Unger et al. 2012]. This model part is written in Modelica, and exported as FMU “facility” using SimulationX 3.7.4

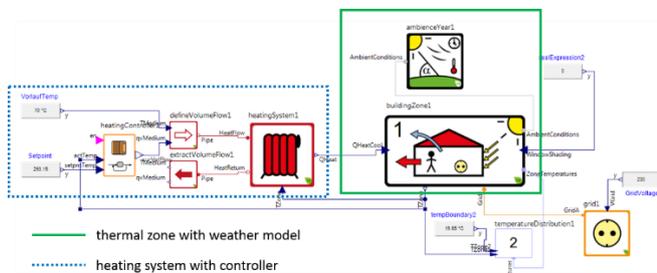


Figure 13. Single room model

This co-simulation task of both the FMUs “thermal zone” and “facility” has 17 coupling variables according to Table 7. Since one of the heating power components is zero, and the ambient variables do not depend on inputs, there are two “true” coupling values which form a cycle.

Table 7. Coupling variables of the single room example

	variables	th. zone	facility
2	Heating power	input	output
2	Zone temperature	output	input
10	Ambient values	output	input
2	Temperature set points	output	input
1	Electric power consumption	output	input

Using the Gauss-Jacobi method for solving the cyclic equations the three step size methods

- Constant step size 60 s
- Richardson extrapolation (tstepMax: 3600 s, tstepMin: 1 s, tstepStart: 60 s, default accuracy ATol: 1e-6, RTol: 1e-4)
- Linear extrapolation (tstepMax: 3600 s, tstepMin: 1 s, tstepStart: 60 s, default accuracy ATol: 1e-6, RTol: 1e-4)

calculate the same result (Figure 14) which does not differ from the reference solution obtained without coupling.

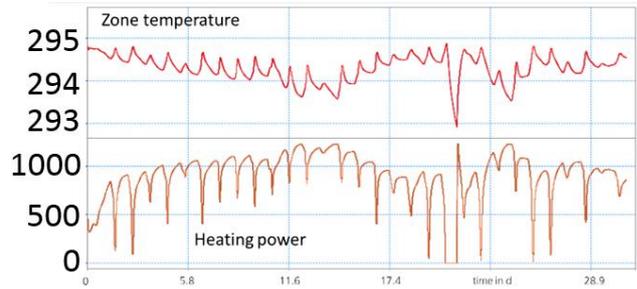


Figure 14. Room temperature and convective thermal heat load

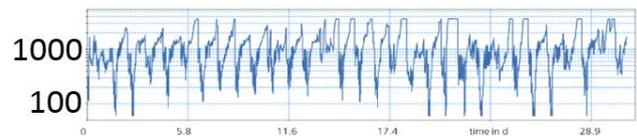


Figure 15. Step size variation in Richardson extrapolation

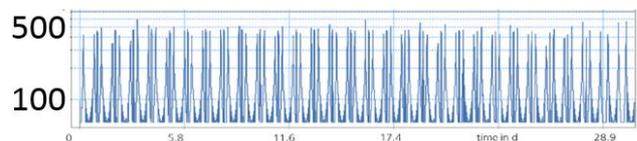


Figure 16. Step size variation in Linear extrapolation

According to Figure 15 the accepted step size in Richardson extrapolation varies considerably. The constant step size of 60 s is much smaller than most of the Richardson steps. Therefore, the Richardson method is even faster than constant step size simulation (Table 8). The Linear extrapolation time step method calculates smaller step sizes than Richardson extrapolation (Figure 16). In this example, Richardson extrapolation is the best choice since it is fast, and the user does not have to define a constant step size.

Table 8. Normalized CPU time comparison single room

Step size 60 s	Richardson	Lin. Extr.
1	0.7	1.6

4.2 Row House

The row house is a building according to Figure 17 with three floors. The heat to the ground floor and the first floor is provided by a volume flow controlled heating system (underfloor heating), the attic is not heated. A thermal storage buffer (50 m³) which is provided with warm water by a heat pump supplies the heating system.

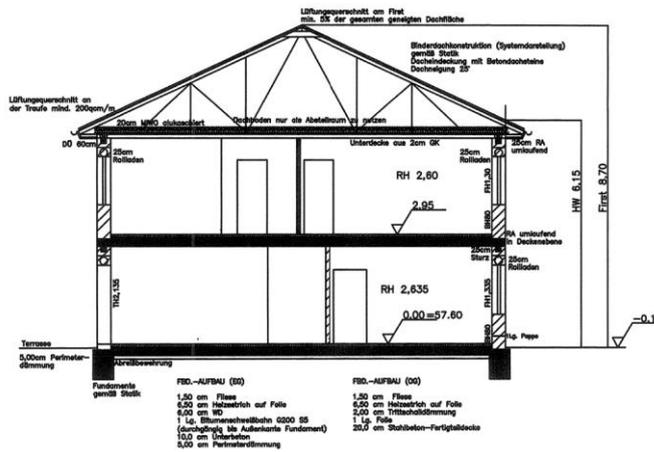


Figure 17. Row house sketch

Similar to the single room model the row house is modeled with different tools. Using NANDRAD the thermal zones including their interdependencies and additionally the weather were modeled, and exported as one FMU “thermal zones”. The facility model including heat pump, buffer, and the heating system are modeled using the GreenBuilding library. This model part is exported as “facility” FMU by SimulationX.

Figure 18 shows the graphical model representation of the row house. The green dashed frame shows the thermal zones, which form together with the weather model the “thermal zones” FMU. All other parts are within the “facility” FMU. Table 9 gives an overview on the 26 coupling variables.

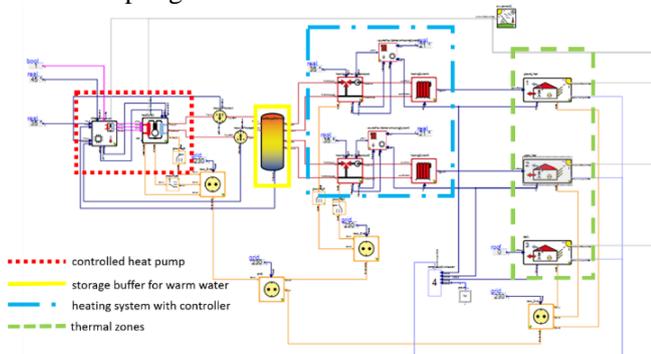


Figure 18. Row house model

Table 9. Coupling variables of the row house example

	<i>variables</i>	<i>th. zones</i>	<i>facility</i>
4	Heating power	input	output
2	Zone temperature	output	input
2	Zone mean radiant temp.	output	
9	Ambient values	output	input
1	Ambient values	output	
2	Heating setpoints	output	input
2	Cooling setpoints	output	
2	User load	output	
2	Electric power consumption	output	

The following results are based on Newton’s method for solving the cyclic equations. Figure 19 shows the zone temperatures, and Figure 20 shows the heat flow into the heated zones over a time interval of 31 days using constant step size of 60 s. This step size was chosen based on experience. The temperatures differ less than $5e-3$ K from reference results obtained by closed simulation via model exchange. The constant step size cannot be enlarged considerably, since already a constant step size of 300 s creates clear deviations, see Figure 21.

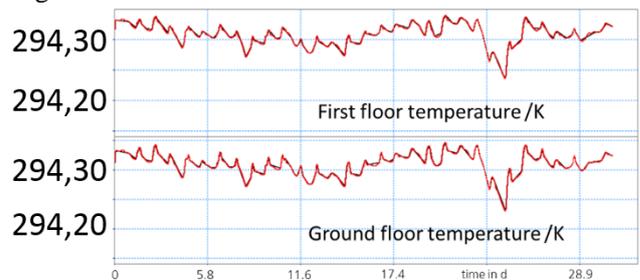


Figure 19. Row house room temperatures, constant step size 60 s

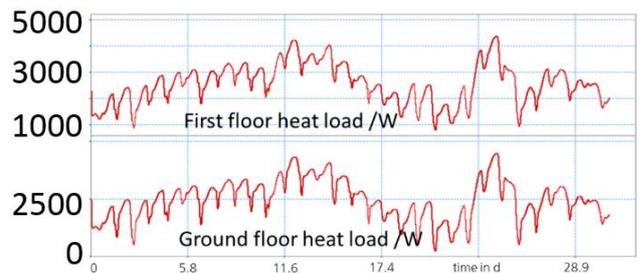


Figure 20. Row house convective thermal heat load, constant step size 60 s

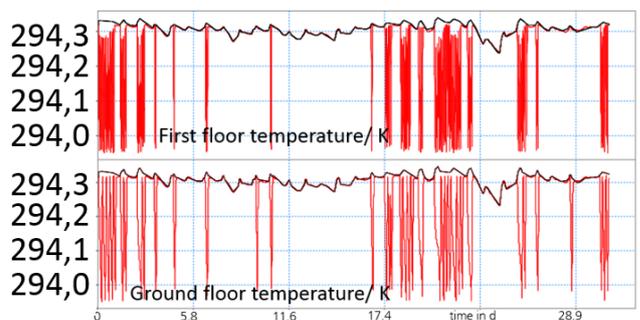


Figure 21. Row house room temperatures with deviations, constant step size 300 s

Richardson extrapolation calculates the same results as shown in Figure 19 and Figure 20, differences are negligible. Figure 22 shows the step size variation which was allowed to vary in a wide range from 0.01 s up to 3600 s, the lower limits were not reached.

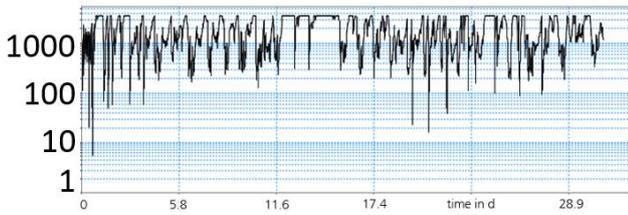


Figure 22. Row house Richardson accepted step size

According to Figure 22 the constant step size of 60 s is not a bad choice. This inspires to use a short time Richardson extrapolation for finding an appropriate constant step size. Table 10 compares different simulations. All variable step size methods calculate wrong results using the default tolerances. If a higher precision is applied correct results are achieved. Richardson extrapolation with higher precision is more twice as slow as well chosen constant step sizes. The reason is that Richardson extrapolation simulates the whole interval more than twice. Furthermore, it is to notice that some wrong simulations take much more CPU than correct ones at that example.

Table 10. Row house comparison of simulation runs

method	step size	deviation**	CPU*
Constant	60 s	0.004 K	21 min
Constant	300 s	0.4 K	20 min
Richardson	3600 s...60 s	0.4 K	1 day
Linear extr.	3600 s...60 s	0.4 K	24 min
Extended Richardson	3600 s...60 s	0,4 K	52 min
Richardson***	3600 s...60 s	0.004	1.1 h
Linear extr.***	3600 s...60 s	0.004	38 min
Extended Richardson***	3600 s...60 s	0.004	1.1 h

* Desktop-PC, SSD, Intel 2, 1 GHz, 8 GB RAM, Windows 7 (64 bit), ** max. deviation of the first floor room temperature from reference values, *** tighter tolerances (ATol=1e-8, RTol=1e-6)

4.3 Apartment Building

The apartment building has four floors, and three staircases. Per staircase and per floor there are three flats so that the building comprises 36 flats, see Figure 23.

The model consists of 168 thermal zones which are described using NANDRAD like at the row house and single room model. The thermal zones are exported altogether with one FMU2.0 “zones” for co-simulation. The heat supply of the building consists of a thermal storage buffer which is recharged by a both a block heat and power plant and a gas boiler. To keep the huge model smaller the heat supply model was simplified by prescribing the temperature of the medium that supplies the radiators. The 168 heating systems of the thermal zones comprise the model of a radiator, a controller model for the volume flow, and a controller model for the supply temperature each, see Figure 24. The heat

supply is modeled using the Green Building library (Modelica) and SimulationX, and exported as one FMU 2.0 “facility”.

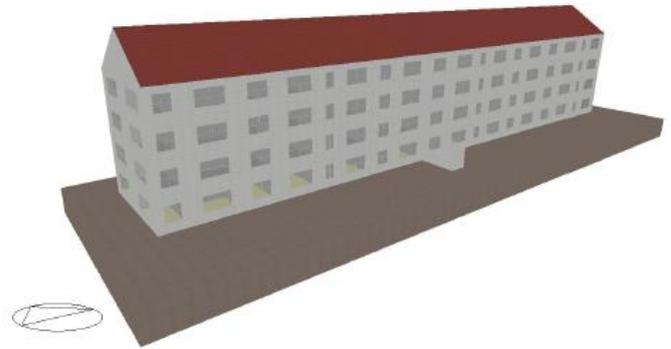


Figure 23. CAD model of the apartment building

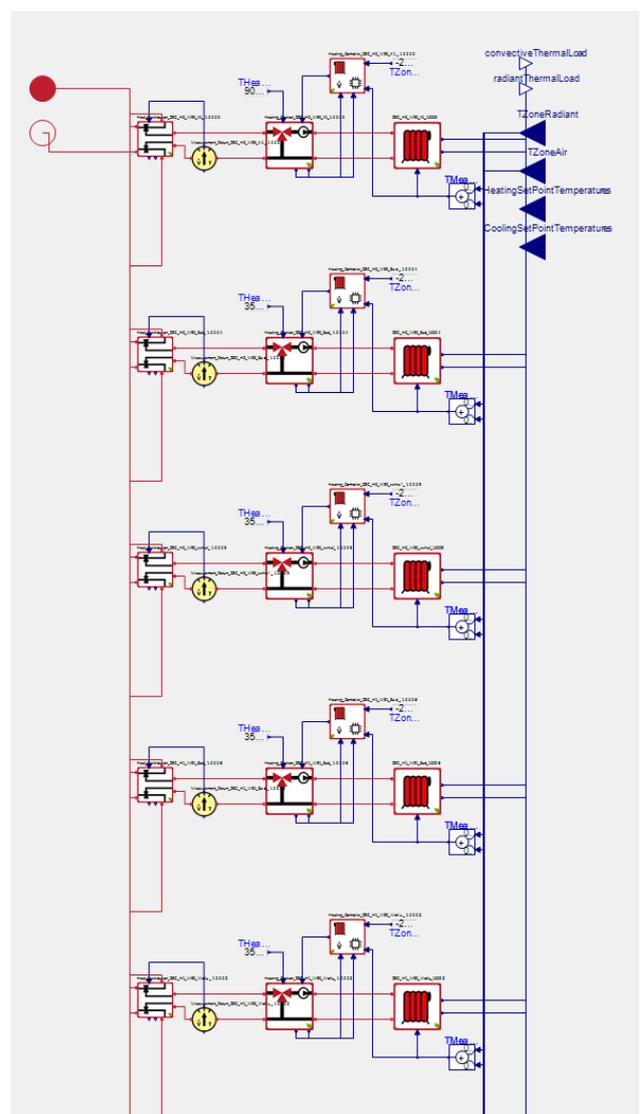


Figure 24. Heating system models of 5 thermal zones

The apartment house example has 1186 coupling variables which are roughly explained in Table 11.

Table 11. Coupling variables of the apartment building example

	<i>variables</i>	<i>th. zones</i>	<i>facility</i>
336	Heating power	input	output
168	Zone temperature	output	input
168	Zone mean radiant temp.	output	input
10	Ambient values	output	input
168	Heating setpoints	output	input
168	Cooling setpoints	output	input
168	Electric power consumption	output	input

Because of the bad performance the model is simulated over 7 days only. The following considerations are based on the Gauss-Seidel method with one iteration for cycle handling. Newton’s method is not applicable due to its extremely bad performance. Figure 25 shows the mean zone temperatures as well as the thermal load of some rooms which are result of Richardson extrapolation with step sizes varying between 0.01 s and 1 hour. This result coincidences with a reference solution obtained by a co-simulation using SimulationX 3.7.4 for the facility part with included NANDRAD FMU for the thermal zones. Therefore, the result is regarded to be correct.

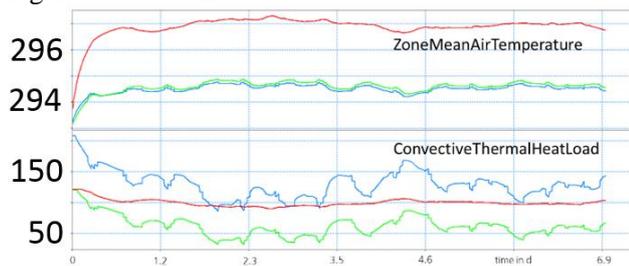


Figure 25. Apartment building: temperatures and heat load, Richardson extrapolation

Figure 26 shows the accepted step size variation of Richardson extrapolation. The step size varies between 2 seconds and an hour, however, only some peaks are below 100s. Therefore, a constant step size simulation was tested, which shows no visible deviation from the Richardson extrapolation result at some selected signals. A gradual increase of the constant step size up to 1 hour does not change the calculated signals clearly. At 1 hour step size the differences are about 0.05 K at some temperatures, and 0.5 W at thermal loads. A more detailed comparison is necessary. It is an advantage of Richardson extrapolation that no fixed step size needs to be defined.

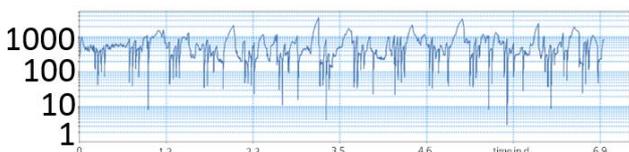


Figure 26. Apartment building: accepted step size variation during Richardson extrapolation, limited by 3600 s...0.01 s

Table 12 shows the performance of the different simulations. Reasonable constant step size simulations are about twice as fast as Richardson extrapolation. The Extended Richardson extrapolation as well as the Linear extrapolation approach also calculate correct results. But their performance is worse than Richardson extrapolation since it uses smaller step sizes (Figure 27).

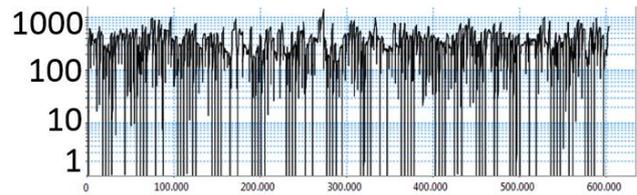


Figure 27. Apartment building: step size variation during Extended Richardson extrapolation

Table 12. Apartment building: performance comparison

<i>method</i>	<i>step size</i>	<i>CPU*</i>
Richardson	3600 s ... 0.01 s	11.4 min
Constant	100 s	7.3 min
Constant	200 s	6.4 min
Constant	400 s	6.0 min
Constant	1000 s	5.4 min
Constant	3600 s	5.4 min
Linear extrapolation	3600 s...0.01 s	14.3 min
Extended Richardson	3600 s...0.01 s	18.2 min

* Desktop-PC, SSD, Intel 2, 1 GHz, 8 GB RAM, Windows 7 (64 bit)

This example demonstrates that Richardson extrapolation seems to ensure finding the correct solution. Furthermore, it is useful for finding adequate step sizes for constant step size simulations. But it is not an approach to obtain a somewhat high performance.

5 Conclusion

Richardson extrapolation is recognized to be an important and useful approach for co-simulation. It was shown that enhancements are necessary for the cases of outputs that do not depend on inputs which control DAEs.

There are examples which need a variable step size approach in co-simulation. The touching mass example requires the Richardson extrapolation approach.

To apply Richardson extrapolation in building simulation three differently sized examples are presented. The results which are by far not yet representative to building simulation models at all, are:

- The performance of Richardson extrapolation is worse than the performance of constant step size method, although Richardson extrapolation partly uses higher step sizes.

- Richardson extrapolation with a wide step size limitation can be applied to find out a trustable constant step size. This helps the user to define the step size. This approach should be automated.
- Furthermore, the building simulation examples show that a high number of coupling variables is to be expected. This frustrated the application of Newton's method for cycles. Therefore, modifications of Newton's method should be investigated.

Acknowledgements

This paper is based on the results of the German research project „Entwicklung der Kopplungstechnologie von Komplexmodellen für Bauteil-, Raum- und Gebäudesimulation mit Modelica-basierten Anlagen-, Regelungs- und Nutzermodellen“ (EnTool:CoSim), funding reference 03ET1215C.

Gefördert durch:



aufgrund eines Beschlusses
des Deutschen Bundestages

The authors are much obliged to the contributors of the research project J. Bastian, T. Blochwitz (ESI ITI GmbH), A. Nicolai and Anne Paepke (IBK TU Dresden), Torsten Schwan, and Monika Wicke (EA Systems Dresden) as well as Kosmas Petridis and Andreas Klein for support and discussions.

References

- Andreas Klein. Private information, 2015.
- Andreas Nicolai, Anne Paepke. Die Gebäudesimulationsplattform NANDRAD – Physikalisches Modell, Umsetzungskonzept und Technologien im Überblick. BauSIM 2012, Berlin, 26.-28. September 2012.
- EA Systems Dresden: Portfolio. Die neue Generation intelligenter Energiekonzepte. Company internal document, via info@ea-energy.de, 2015.
- Ernst Hairer, Syvert Paul Norsett, Gerhard Wanner. Solving Ordinary Differential Equations. Berlin, Springer, 1993.
- ESI ITI GmbH: Simulation software SimulationX, website <https://www.simulationx.de/simulationssoftware.html>
- Jens Bastian, Christoph Clauß, Susann Wolf, Peter Schneider. Master for CoSimulation Using FMI. 8th International Modelica Conference, Dresden, March 20-22, 2011.
- Kosmas Petridis, Christoph Clauß. Test of basic co-simulation algorithms using FMI. 11th International Modelica Conference, Versailles, 2015.
- Vladimir Aleksandrovitch Kotelnikov. On the transmission capacity of the ether and of cables in electrical communications. Proc. of the first All-Union Conference on the technological reconstruction of the communications sector and low-current engineering, Moscow 1933.

Rene Unger, Torsten Schwan et alt.. "Green Building"-Modelling Renewable Building Energy Systems and Electric Mobility Concepts Using Modelica. 9th International Modelica Conference, Munich, Germany, 2012.

Tom Schierz, Martin Arnold, Christoph Clauß. Cosimulation with communication step size control in an FMI compatible master algorithm. 9th International Modelica Conference, Munich, 2012.

Tom Schierz. Modulare Zeitintegration gekoppelter Differentialgleichungssysteme in der technischen Simulation. Fortschritt-Berichte, VDI Reihe 20 Nr. 447. Düsseldorf, VDI Verlag 2013.