

Co-simulation for performance prediction of integrated building and HVAC systems - An analysis of solution characteristics using a two-body system

Marija Trčka^{a*}, Jan L.M. Hensen^a and Michael Wetter^b

^a*Eindhoven University of Technology, PO Box 513, 5600MB Eindhoven, The Netherlands;* ^b*Simulation Research Group, Building Technologies Department, Environmental Energy Technologies Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

**Corresponding author. Email: m.trcka@tue.nl*

Abstract

Integrated performance simulation of buildings and heating, ventilation and air-conditioning (HVAC) systems can help reducing energy consumption and increasing occupant comfort. However, no single building performance simulation (BPS) tool offers sufficient capabilities and flexibilities to analyze integrated building systems and to enable rapid prototyping of innovative building and system technologies. One way to alleviate this problem is to use co-simulation to integrate different BPS tools. Co-simulation approach represents a particular case of simulation scenario where at least two simulators solve coupled differential-algebraic systems of equations and exchange data that couples these equations during the time integration.

This article analyzes how co-simulation influences consistency, stability and accuracy of the numerical approximation to the solution. Consistency and zero-stability are studied for a general class of the problem, while a detailed consistency and absolute stability analysis is given for a simple two-body problem. Since the accuracy of the numerical approximation to the solution is reduced in co-simulation, the article concludes by discussing ways for how to improve accuracy.

Key words: co-simulation, building system performance simulation, integrated building simulation

1. Introduction

The building industry is one of the most important industrial and economical sectors influencing the quality of life and the environment. It is a known fact that in developed countries, buildings account for 30%-40% of the energy consumed. Depending on the building type, heating, ventilation and air-conditioning (HVAC) systems are responsible for 10%-60% of the total building energy consumption. The long life-cycle of buildings further compounds the importance of architectural and engineering design decisions.

Challenging goals are set by new initiatives and energy policies. For example, the European Union has defined ambitious goals for reducing emissions of CO₂ for the industrialized countries, which should be achieved by 2020. Also, the U.S. Department of Energy and ASHRAE have defined their vision for 2030 [2] in a form of net zero energy buildings. Thus, modern buildings are required to be energy efficient while adhering to the ever increasing demand for better indoor environmental quality.

Further, new buildings consist of numerous dynamically interacting components that are nonlinear, dynamic, and complex. This requires an integrated approach that treats buildings and the systems that service them as complete entities, not as separately designed subsystems.

To enable energy efficient designs in this complex setting, the concept of integrated building performance simulation (BPS) has been developed. BPS offers the potential to cope adequately with building performance related concerns. Its main purpose is to (i) analyze the energy consumption and comfort performance, and (ii) understand the relationship between the design parameters, the energy use, and the comfort characteristics of buildings. Experience shows that, if used properly, BPS can indeed result in a significant reduction of emission of greenhouse gases, and give substantial improvements in fuel consumption and comfort levels.

Due to the high time-cost of building performance prediction [3; 4; 5; 6] and the

low market interest, the significant effort required to develop BPS tools has been left to a fragmented research community. This has resulted in a slower development and inferior capabilities compared to simulation tools used in larger industry sectors, such as in the automotive, electronics or aerospace sector.

Due to the fragmented development of BPS tools and the rapid innovations in building and system technologies, state of the art BPS tools are not equally suited for modeling and simulation of the relevant building aspects. Moreover, the user's requirements often exceed the tool's functionality. As it has been previously argued [7; 8], in the area of system simulation there is still an enormous amount of work to be done.

The state of the art BPS tools are difficult and costly to extend. Adding new features requires from the tool developer to have in-depth knowledge of the programming languages used, of the underlying architecture, and of the tool-specific modeling strategies. Furthermore, switching to equation-based tools does not satisfy the near-term needs. Although they are better suited for rapid model prototyping than the BPS tools, they typically lack the vast range of state of the art models (e.g., for solar and wind processes) and domain-relevant input/output processing.

To successfully continue with developments of the tools that will drive innovation and help reaching the ambitious goal of reducing the emission of CO₂, the focus should be on enabling more analysis of innovative designs rather than investing in “reinventing the wheel.” An efficient way forward would be to provide a facility to combine features from different tools, sharing developments and reusing component models. A tool should be coupled with a complementary tool in such a way that the integrated result provides more value to the end user than the individual tool does itself.

One way to alleviate this problem is to use co-simulation. The co-simulation approach represents a particular case of simulation scenario where at least two simulators solve coupled systems of differential-algebraic equations and exchange data

that couples these equations during the time integration. It allows various aspects of building systems to be modeled and simulated in domain-specific tools and link them during simulation. Compared to the traditional approach, it offers increased functionality and more flexibility for integrated simulation-based analysis of innovative HVAC system technologies.

1.1. Co-simulation implementation strategies

In general, co-simulation can be implemented using different strategies, of which a few will be mentioned here. In particular, from the coupling point of view, the implementation can be done using strong or loose coupling, as illustrated in Figure 1.

- **Strong coupling** [9], also called fully-dynamic [10] or onion coupling [11], requires an iterative solution that involves both simulators. The iteration stops when the coupling data satisfy a convergence criteria. In strong coupling, simulators must have a mechanism to rewind their states if requested by a co-simulation manager. In addition, since each simulator may include iterative solutions of equations, strong coupling leads to nested iteration loops, consisting of an inner iteration within the individual simulators, and an outer iteration to achieve convergence of the coupled simulators. To ensure convergence, the inner iterations need to be solved at higher accuracy than the outer iterations. This may be impractical to accomplish in BPS tools that do not allow controlling the precision of the numerical error. Thus, to realize strong coupling, significant code modifications may be required.
- In **loose coupling** [9], also called quasi-dynamic [10] or ping-pong coupling [11], coupled simulators use the coupling data that is computed using only data from preceding time steps. There is no iteration between the coupled simulators. However, because of the time-lagged data, shorter coupling time

steps are needed compared to strong coupling. We distinguish two types of loose coupling strategies (see Figure 1):

- loose coupling with *sequential staggered* solution [12], also called zigzagged coupling, where the coupled simulators are executed in sequence, and
- loose coupling with *naive modification for parallel processing* [12], also called cross coupling, where the coupled simulators are executed in parallel.

The co-simulation discussed in this article implements zigzagged coupling, in which the sending and the receiving sequence differs between the coupled simulators. For that reason, we call the simulators the **base** and the **external** simulator. The base simulator starts the communication by sending the coupling data to the communication interface. The external simulator starts the communication by reading those data from the communication interface.

Depending on which data are delayed in time, [13] defines two partitioning strategies:

- **Implicit-implicit** - if the coupling data depends only on the state variables of the coupled subsystem.
- **Implicit-explicit** - if the coupling data depends on the state variables of both subsystems.

The experiments in this article will use two different system-decomposition strategies:

- **Intra-domain** system decomposition, in which the system is decomposed within one functional domain, such as within the HVAC domain only.
- **Inter-domain** system decomposition, in which the system is decomposed between different functional domains, such as between the building and the HVAC system domain.

1.2. Examples of co-simulation implementation

There are a few examples of co-simulation in the field of BPS. Examples include the integration of computational fluid dynamics simulation (FLUENT) with building energy simulation (ESP-r) [14]. In the domain of HVAC simulation tools, examples include the integration of TRNSYS with several other programs, such as MATLAB [<http://software.cstb.fr/>] and EES [15]. These BPS tools couple two simulators directly with each other, with one tool serving as the master and the other as the client. A different architecture has been implemented in the Building Controls Virtual Test Bed (BCVTB) that uses a middleware to manage the data exchange between different simulators, with each simulator acting as a client [16]. However until now, there exists no general standardized framework for integration of BPS simulators, nor do there exist guidelines for implementation of co-simulation with regards to stability and accuracy.

1.3. Co-simulation prototype

The co-simulation prototypes developed by the authors and reported earlier [1] implement both loose and strong coupling strategies. The prototypes are based on state of the art BPS tools such as TRNSYS and EnergyPlus. Flow-charts of the information flow in the prototypes are give in Figure 2 and 3.

In the case of loose coupling, the coupling data is exchanged only in the first iteration for the current time step in both simulators. Strong coupling requires an iteration between the simulators. In our prototype, simulator 1 in Figure 3 controls the iteration process. The iteration criterion is based on the difference between two subsequently received values of coupling data from TRNSYS. If the difference is greater than a specified value, EnergyPlus will request another iteration.

Numerical experiments [1] have shown that strong coupling allows using longer time steps than loose coupling at the same accuracy. It was also shown that loosely coupled co-simulation with sufficiently small time steps can generate results with the same accuracy as mono-simulation. However, based on the computation time

and ease of implementation, the loose coupling strategy with smaller time steps was recommended.

1.4. Problem statement

The BPS tools typically contain legacy code with more than 100 000 lines of code that mixes code to implement physical equations, data exchange and numerical solution algorithms. This makes it difficult to reinitialize state variables to previous values, which is necessary for strong coupling. However, loose coupling is easier to implement, but the time-delay of the coupling data causes the original numerical time integration methods to be modified. Consequently, the stability and accuracy properties of the original time integration method are no longer guaranteed.

The focus of this article is to investigate the characteristics of co-simulation that uses loose coupling. Although the stability and accuracy of different time integration methods are well understood [17; 18; 19], the stability and accuracy of the methods resulting from partitioning are not well analyzed. It is difficult, if not impossible, to determine these properties formally for a general class of problems [20].

In this article, the co-simulation problem is stated for a general class of problems that are encountered in BPS. Then the stability and accuracy properties are deduced formally for linear one-step numerical integration methods, applied to two coupled first-order linear initial value problems. Finally, numerical experiments are presented.

2. Numerical integration methods - Theoretical background

We consider problems defined by the first order initial value ordinary differential equation

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}, t), \tag{1a}$$

$$\mathbf{y}(a) = \boldsymbol{\eta}, \tag{1b}$$

where $\mathbf{f}: \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^m$, $t \in [a, b]$ for some $a, b \in \mathbb{R}$, with $a < b$, $m \in \mathbb{N}$ and $\boldsymbol{\eta} \in \mathbb{R}^m$. We assume $\mathbf{f}(\cdot, \cdot)$ is once Lipschitz continuously differentiable in \mathbf{y} and t . This ensures that a unique solution of equation (1) exists.

Numerical integration methods are used to approximate the solution of the initial value problem (1). In equidistant discretization, the continuous interval $[a, b]$ is replaced by the discrete point set $\{t^n \in \mathbb{R} \mid t^n = a + n\Delta t, n \in \{0, 1, 2, \dots, N\}, \Delta t = (b - a)/N\}$. A numerical integration method produces a sequence $\{\mathbf{y}^n \in \mathbb{R}^m\}_{n=0}^N$, which approximates the solution of equation (1) ($\{\mathbf{y}(t^n)\}_{n=0}^N$) at the discrete points $\{t^n\}_{n=0}^N$. The rule for computing \mathbf{y}^n is defined in terms of the values of \mathbf{y}^n at preceding discretization points. In a k -step numerical integration method, \mathbf{y}^n is computed using $\{\mathbf{y}^{n-1}, \dots, \mathbf{y}^{n-k}\}$.

To approximate the solution of the initial value problem represented by equation (1), we will use a k -step numerical integration method. The general form of the method is

$$\sum_{j=0}^k \alpha_j \mathbf{y}^{n+j} = \Delta t \phi_{\mathbf{f}}(\mathbf{y}^{n+k}, \dots, \mathbf{y}^n, t^n; \Delta t), \quad (2)$$

with $\mathbf{y}^\mu = \boldsymbol{\eta}^\mu$ and $\mu \in \{0, 1, \dots, k-1\}$ [18]. The subscript \mathbf{f} on the right hand side indicates that the function ϕ , which characterizes the particular method, depends on $\mathbf{f}(\cdot, \cdot)$.

For linear multi-step methods, the system represented by equation (2) can be written as

$$\sum_{j=0}^k \alpha_j \mathbf{y}^{n+j} = \Delta t \sum_{j=0}^k \beta_j \mathbf{f}(\mathbf{y}^{n+j}, t^{n+j}), \quad (3)$$

where $\{\alpha_j, \beta_j \in \mathbb{R} \mid j \in \{0, 1, \dots, k\}\}$ are method-specific parameters.

Let α be the implicitness factor of a linear one-step numerical integration method.

For members of the α - family of linear one-step numerical integration methods, equation (3) leads to

$$\mathbf{y}^{n+1} - \mathbf{y}^n = \Delta t[(1 - \alpha)\mathbf{f}(\mathbf{y}^n, t^n) + \alpha\mathbf{f}(\mathbf{y}^{n+1}, t^{n+1})], \quad (4)$$

where $0 \leq \alpha \leq 1$.

2.1. Consistency

Consistency of a numerical integration method is concerned with the truncation errors that result from the numerical integration method. The following definitions are taken from [18].

Definition 2.1 (Local truncation error). *The local truncation error is defined as the error produced in a single integration step starting from the exact solution.* \square

For the k -step numerical integration method (2), the local truncation error is $\mathbf{LTE}^{n+k}(\Delta t) = \sum_{j=0}^k \alpha_j \mathbf{y}(t^{n+j}) - \Delta t \phi_{\mathbf{f}}(\mathbf{y}(t^{n+k}), \dots, \mathbf{y}(t^n), t^n; \Delta t)$.

Definition 2.2 (Unit local truncation error). *The unit local truncation error is defined as $\mathbf{ULTE}^{n+k}(\Delta t) = \frac{\mathbf{LTE}^{n+k}(\Delta t)}{\Delta t}$.* \square

Definition 2.3 (Consistency). *A numerical integration method is said to be consistent if, for initial value problems of the form (1) that satisfy the Lipschitz condition, the unit local truncation error satisfies $\lim_{\Delta t \rightarrow 0} \mathbf{ULTE}^{n+k}(\Delta t) = 0$.* \square

2.2. Stability

In terms of stability properties of numerical integration methods, a distinction is made between zero-stability and absolute stability.

2.2.1. Zero-stability

Zero-stability is concerned with the asymptotic behavior in the limit as $\Delta t \rightarrow 0$. It is a property of the numerical integration method (2) and not of the differential

equation (1). Assuming that $\mathbf{f}(\cdot, \cdot)$ in equation (1) is once Lipschitz continuously differentiable, the linear numerical integration method (3) tends to the linear constant coefficient difference system, i.e., $\sum_{j=0}^k \alpha_j \mathbf{y}^{n+j} \rightarrow 0$, as $\Delta t \rightarrow 0$. Its characteristic polynomial, $\rho(r) = \sum_{j=0}^k \alpha_j r^j$, is the first characteristic polynomial of the numerical integration method. Let the roots of $\rho(r) = 0$ be $\{r_i \in \mathbb{C} \mid i \in \{1, 2, \dots, k\}\}$. The numerical integration method is said to be zero-stable if all the roots of the first characteristic polynomial satisfy $|r_i| \leq 1$, and any root for which $|r_i| = 1$ is simple.

2.2.2. Absolute stability

For some numerical integration methods, there exists a value Δt_0 of the step-length such that for fixed $\Delta t > \Delta t_0$ the method is unstable. Absolute stability considers the behavior of a numerical integration method for a finite value of Δt [17; 18; 21]. The absolute stability of the method (3) depends not only on the numerical integration method, but also on equation (1). The standard analysis of absolute stability of linear numerical integration methods is based on the second characteristic polynomial [17; 18; 21] and a constant matrix $\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{y}}$. However, co-simulation leads to a partitioning of the right-hand side of (2) and therefore the second characteristic polynomial cannot be used. Hence, to study absolute stability, we will use methods different from the ones in [17; 18; 21].

2.3. Convergence

We will now define convergence [18].

Definition 2.4 (Convergence). *Consider the problem represented by equation (1) and for $N \in \mathbb{N}$, $N > 0$, let $\tau_N \triangleq \{t^n \in \mathbb{R} \mid t^n = a + n \Delta t, n \in \{0, 1, \dots, N\}, \Delta t = (b-a)/N\}$. A numerical integration method is said to be convergent if for all $t^n \in \tau_N$,*

$$\lim_{N \rightarrow \infty} \mathbf{y}^n = \mathbf{y}(t^n). \quad (5)$$

□

A necessary and sufficient condition for a numerical integration method to be convergent is that it is both consistent and zero-stable [17; 18].

3. Consistency of co-simulation

In the co-simulation discussed in this article, the system of equations (2) is first partitioned algebraically, and then solved in coupled simulators. At the $(n+k)$ -th time step, where $n+k \leq N$, the coupling data depends on $\{\mathbf{y}^{n+j} \in \mathbb{R}^m \mid j \in \{0, 1, 2, \dots, k\}, n \in \{0, 1, \dots, N\}, N = (b-a)/\Delta t\}$. If the simulators are loosely coupled, the coupling data at t^{n+k} is not available to (both) coupled simulators and needs to be predicted based on the data of the preceding time steps. Thus, in the right-hand side of (2), the argument \mathbf{y}^{n+k} is replaced \mathbf{y}_P^{n+k} , which is the prediction of the state vector. Consequently, in loosely coupled co-simulation, equation (2) becomes

$$\sum_{j=0}^k \alpha_j \mathbf{y}^{n+j} = \Delta t \phi_{\mathbf{f}}(\mathbf{y}_P^{n+k}, \mathbf{y}^{n+k-1}, \dots, \mathbf{y}^n, t^n; \Delta t), \quad (6)$$

and equation (4) becomes

$$\mathbf{y}^{n+1} - \mathbf{y}^n = \Delta t [(1 - \alpha) \mathbf{f}(\mathbf{y}^n, t^n) + \alpha \mathbf{f}(\mathbf{y}_P^{n+1}, t^{n+1})]. \quad (7)$$

To determine the consistency of co-simulation, we directly use Definition 2.3. For the numerical approximation (7), the unit local truncation error is

$$\begin{aligned} \mathbf{ULTE}^{n+1}(\Delta t) &= \frac{1}{\Delta t} \left(\mathbf{y}(t^{n+1}) - \mathbf{y}(t^n) \right. \\ &\quad \left. - \Delta t [(1 - \alpha) \mathbf{f}(\mathbf{y}(t^n), t^n) + \alpha \mathbf{f}(\mathbf{y}_P(t^{n+1}), t^{n+1})] \right), \quad (8) \end{aligned}$$

where $\mathbf{y}_P(t^{n+1})$ is an approximation of $\mathbf{y}(t^{n+1})$ based on the known values $\{\mathbf{y}(t^{n-j}) \in \mathbb{R}^m \mid j \in \{0, 1, \dots, n\}\}$. Adding and subtracting $\alpha \mathbf{f}(\mathbf{y}(t^{n+1}), t^{n+1})$ to the right hand side and collecting terms that correspond to the unit local truncation error of the

original, non-partitioned numerical method, yields

$$\begin{aligned} \mathbf{ULTE}^{n+1}(\Delta t) &= \mathbf{ULTE}_{\text{non-partitioned}}^{n+1}(\Delta t) \\ &+ \alpha [\mathbf{f}(\mathbf{y}(t^{n+1}), t^{n+1}) - \mathbf{f}(\mathbf{y}_P(t^{n+1}), t^{n+1})]. \end{aligned} \quad (9)$$

Applying the norm on both sides of equation (9) yields

$$\begin{aligned} \|\mathbf{ULTE}^{n+1}(\Delta t)\| &\leq \|\mathbf{ULTE}_{\text{non-partitioned}}^{n+1}(\Delta t)\| \\ &+ \alpha L \|\mathbf{y}(t^{n+1}) - \mathbf{y}_P(t^{n+1})\|, \end{aligned} \quad (10)$$

where L is a Lipschitz constant.

To evaluate the order of the error, the exact solutions of the state vectors in the two subsequent time steps, $\mathbf{y}(t^{n+1})$ and $\mathbf{y}(t^n)$, is expressed using a Taylor series around $t^n + \alpha\Delta t$, for any $\alpha \in [0, 1]$. When substituted into equation (10) for the zero-order prediction $\mathbf{y}_P(t^{n+1}) = \mathbf{y}(t^n)$, one obtains

$$\|\mathbf{ULTE}^{n+1}(\Delta t)\| \leq \|\mathbf{ULTE}_{\text{non-partitioned}}^{n+1}(\Delta t)\| + \alpha L O(\Delta t). \quad (11)$$

It follows from equation (11) and from $\lim_{\Delta t \rightarrow 0} \alpha L O(\Delta t) = 0$ that if the original non-partitioned numerical method is consistent, i.e., $\lim_{\Delta t \rightarrow 0} \|\mathbf{ULTE}_{\text{non-partitioned}}^{n+1}(\Delta t)\| = 0$, then the partitioned numerical method is consistent as well. Furthermore, the unit local truncation error introduced by the partitioning is of order one. Hence, the order of first order accurate methods will not be changed by the partitioning. However, for the Crank-Nicholson method ($\alpha = 1/2$), which is a second order method, the order will be reduced by partitioning.

In Section 5, the influence of system parameters on the unit local truncation error introduced by partitioning will be further analyzed using a two-body problem.

4. Zero-stability and convergence of co-simulation

By inspection of the partitioned linear numerical integration method (7), it can be seen that the partitioning changes only the right-hand side of the equation. The first characteristic polynomial that determines zero-stability of a numerical integration method depends only on the coefficients on the left-hand side of equation (7) and thus remains unchanged. Consequently, the partitioning does not disturb the properties of the zero-stability of the non-partitioned numerical integration method.

Since by equation (11) it was shown that the numerical integration method (7) is consistent, and since the zero-stability of the original numerical method is not changed with partitioning, it follows that the numerical integration method (7) is convergent.

The absolute stability of the partitioned numerical integration method (7) will be examined in the following section using a specific two-body system.

5. Analysis of co-simulation for a two-body system

To gain insight into the solution characteristics of the partitioned numerical simulation of building and HVAC simulations, we will analyze the system shown in Figure 4. The example is a simplified representation (but still based on the same subset of physical laws) of problems tackled by BPS tools. It constitutes of two subsystems with lumped capacities. Each subsystem is represented by a single state variable. Each subsystem interacts with the surrounding environment through heat transfer by conduction and convection and with each other through fluid flow. The subsystems are connected by a fluid loop.

In case of co-simulation, the flow of information is as presented in Figure 2, where simulator 1 simulates the subsystem denoted with index 1 and simulator 2 simulates the subsystem denoted with index 2. For the analysis purpose of this paper both

subsystems are modeled and simulated in Mathematica 6¹.

5.1. Mathematical model

The system in Figure 4 is modeled as an initial value problem for $t \in [a, b]$:

$$\begin{aligned} \mathbf{C}\dot{\mathbf{T}}(t) + \mathbf{S}\mathbf{T}(t) &= \mathbf{g}(t), \\ \mathbf{T}(a) &= \mathbf{T}_0, \end{aligned} \tag{12}$$

where

$$\mathbf{T}(t) = \begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix}, \mathbf{C} = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}, \mathbf{S} = \begin{bmatrix} \dot{m}c_p + K_1 & -\dot{m}c_p \\ -\dot{m}c_p & \dot{m}c_p + K_2 \end{bmatrix}, \mathbf{g}(t) = \begin{bmatrix} g_1(t) \\ g_2(t) \end{bmatrix}.$$

$\mathbf{T}(\cdot)$ is a temperature vector, \mathbf{C} is a non-singular diagonal matrix for the capacity, \mathbf{S} is a matrix that couples the subsystems and $\mathbf{g}(\cdot)$ is a forcing function. As for problem (1), we assume that $\mathbf{g}(\cdot)$ is once Lipschitz continuously differentiable. The subscripts 1 and 2 refer to the subsystems in Figure 4, and the subscript ∞ refers to the environment, C_i is the overall heat capacity, $K_i \triangleq U_i A_i$, where U_i is the overall heat transfer coefficient (assumed invariant) and A_i is the heat exchange surface area, \dot{m} is the working fluid mass flow, c_p is the specific heat capacity of the working fluid and $g_i(t) = K_i T_\infty$ is the forcing function acting upon the i -th subsystem.

In general, the flow rate is a function of time and the heat transfer coefficient is a function of temperature and therefore the matrix \mathbf{S} is time variant. The matrices \mathbf{C} and \mathbf{S} are also temperature dependent. However, to simplify the analysis, the problem described by equation (12) is regarded as a linear time-invariant system. In Section 5.2.1, we comment on the situation where the coefficients of the matrix \mathbf{S} are time varying.

¹However, it should be noted that state of the art tools in BPS are used our prototypes that are used to simulate whole buildings[1].

5.1.1. Numerical approximation to the solution

Applying a linear one-step numerical integration method (4) to approximate the solution of (12), and rearranging equations to place the known terms on the right-hand side and the unknown terms on the left-hand side, yields

$$[\mathbf{C} + \alpha\Delta t\mathbf{S}]\mathbf{T}^{n+1} = \Delta t[\alpha \mathbf{g}(t^{n+1}) + (1 - \alpha) \mathbf{g}(t^n)] + [\mathbf{C} - (1 - \alpha)\Delta t\mathbf{S}]\mathbf{T}^n, \quad (13)$$

with initial condition $\mathbf{T}^0 = \mathbf{T}_0$.

5.1.2. Partitioning of the numerical integration method

In loosely coupled co-simulation, the numerical integration method (13) changes to

$$\begin{aligned} [\mathbf{C} + \alpha\Delta t\mathbf{S}_I]\mathbf{T}^{n+1} &= \Delta t[\alpha \mathbf{g}(t^{n+1}) + (1 - \alpha) \mathbf{g}(t^n)] \\ &+ [\mathbf{C} - (1 - \alpha)\Delta t\mathbf{S}]\mathbf{T}^n - \alpha\Delta t\mathbf{S}_E\mathbf{T}_P^{n+1} \end{aligned} \quad (14)$$

where \mathbf{T}_P^{n+1} is the predicted temperature vector at t^{n+1} , based on the known temperature vectors from the preceding steps, and \mathbf{S}_I and \mathbf{S}_E are partitions of the matrix \mathbf{S} . Different decomposition strategies lead to different partitioning of the matrix \mathbf{S} . Implicit-implicit partitioning leads to

$$\mathbf{S} = \mathbf{S}_I + \mathbf{S}_E = \begin{bmatrix} \dot{m}c_p + K_1 & -\dot{m}c_p \\ 0 & \dot{m}c_p + K_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\dot{m}c_p & 0 \end{bmatrix}, \quad (15)$$

whereas implicit-explicit partitioning leads to

$$\mathbf{S} = \mathbf{S}_I + \mathbf{S}_E = \begin{bmatrix} K_1 & -\dot{m}c_p \\ 0 & \dot{m}c_p + K_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\dot{m}c_p & \dot{m}c_p \end{bmatrix}. \quad (16)$$

Equations (15) and (16) will be used in Section 5.3 to define criteria for absolute stability.

5.2. Consistency

Analyzing the unit local truncation error $\mathbf{ULTE}^{n+1}(\Delta t)$ allows getting insights on how the systems characteristics, such as the thermal capacity, influence the error introduced by the partitioning. To accomplish this, first, the numerical integration method (14) is written in the form of equation (4), which gives

$$\begin{aligned} \mathbf{T}^{n+1} - \mathbf{T}^n &= \Delta t [(1 - \alpha) \mathbf{C}^{-1}(\mathbf{g}^n - \mathbf{S} \mathbf{T}^n) \\ &\quad + \alpha \mathbf{C}^{-1}(\mathbf{g}^{n+1} - \mathbf{S}_I \mathbf{T}^{n+1} - \mathbf{S}_E \mathbf{T}_P^{n+1})]. \end{aligned} \quad (17)$$

Then, using Definitions 2.1 and 2.2, the unit local truncation error of the numerical integration method (17) yields

$$\begin{aligned} \mathbf{ULTE}^{n+1}(\Delta t) &= \frac{1}{\Delta t} (\mathbf{T}(t^{n+1}) - \mathbf{T}(t^n) \\ &\quad - \Delta t [(1 - \alpha) \mathbf{C}^{-1}(\mathbf{g}(t^n) - \mathbf{S} \mathbf{T}(t^n)) \\ &\quad + \alpha \mathbf{C}^{-1}(\mathbf{g}(t^{n+1}) - \mathbf{S}_I \mathbf{T}(t^{n+1}) \\ &\quad - \mathbf{S}_E \mathbf{T}_P(t^{n+1}))]). \end{aligned} \quad (18)$$

Since we assumed $\mathbf{g}(\cdot)$ to be Lipschitz continuously differentiable, it follows that $\mathbf{T}(\cdot)$ is Lipschitz continuously differentiable. Hence, we will use the zero-order prediction $\mathbf{T}_P(t^{n+1}) = \mathbf{T}(t^n)$, and express $\mathbf{g}(t^{n+1})$ and $\mathbf{g}(t^n)$, and $\mathbf{T}(t^{n+1})$ and $\mathbf{T}(t^n)$, around time $t^n + \alpha\Delta t$, for any $\alpha \in [0, 1]$, by a Taylor series. Then, equation (18) yields

$$\begin{aligned} \mathbf{ULTE}^{n+1}(\Delta t) &= \mathbf{C}^{-1}(\mathbf{C} \dot{\mathbf{T}}(t^n + \alpha\Delta t) + \mathbf{S} \mathbf{T}(t^n + \alpha\Delta t) - \mathbf{g}(t^n + \alpha\Delta t)) \\ &\quad - \frac{1}{2}(1 - 2\alpha) \ddot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t \\ &\quad - \alpha \mathbf{C}^{-1} \mathbf{S}_E \dot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t + O(\Delta t^2). \end{aligned} \quad (19)$$

Substituting equation (12), evaluated at time $t^n + \alpha\Delta t$, into equation (19) the unit

local truncation error yields

$$\begin{aligned} \mathbf{ULTE}^{n+1}(\Delta t) &= \frac{1}{2}(1-2\alpha)\ddot{\mathbf{T}}(t^n + \alpha\Delta t)\Delta t \\ &\quad + \alpha\mathbf{C}^{-1}\mathbf{S}_E\dot{\mathbf{T}}(t^n + \alpha\Delta t)\Delta t + O(\Delta t^2). \end{aligned} \quad (20)$$

The term $\frac{1}{2}(1-2\alpha)\ddot{\mathbf{T}}(t^n + \alpha\Delta t)\Delta t + O(\Delta t^2)$ is caused by the original non-partitioned numerical integration method (13), while the term $\alpha\mathbf{C}^{-1}\mathbf{S}_E\dot{\mathbf{T}}(t^n + \alpha\Delta t)\Delta t + O(\Delta t^2)$ is introduced by the partitioning.

This shows that the unit local truncation error due to the partitioning is proportional to the factor α , and thus it is zero for the explicit numerical integration method, as expected. The greater the capacity of the external subsystem, the smaller the unit local truncation error is. The smaller the rate of change of the delayed data at $t^n + \alpha\Delta t$ and the smaller Δt , the smaller the unit local truncation error is.

The accuracy of the partitioned numerical integration method that uses a first-order prediction of the form (30) (see Section 5.5.1) for the coupling data can be evaluated in a similar way, by using the exact solutions of the temperature vectors in the two subsequent time steps, $\mathbf{T}(t^{n-1})$ and $\mathbf{T}(t^n)$, expressed around $t^n + \alpha\Delta t$ by means of a Taylor series. Substituting these expressions into the expression for $\mathbf{ULTE}^{n+1}(\Delta t)$, one obtains

$$\begin{aligned} \mathbf{ULTE}^{n+1}(\Delta t) &= \frac{1}{2}(1-2\alpha)\ddot{\mathbf{T}}(t^n + \alpha\Delta t)\Delta t \\ &\quad - \frac{1}{2}\alpha\mathbf{C}^{-1}(\alpha\mathbf{S} - \mathbf{S}_I + \mathbf{S}_E)\ddot{\mathbf{T}}(t^n + \alpha\Delta t)\Delta t^2 + O(\Delta t^3) \end{aligned} \quad (21)$$

Here, the term $\alpha\mathbf{C}^{-1}\mathbf{S}_E\ddot{\mathbf{T}}(t^n + \alpha\Delta t)\Delta t^2 + O(\Delta t^3)$ is introduced by the partitioning. The introduced error is of order two and it can be concluded that the prediction of the coupling data enhances the accuracy of co-simulation.

5.2.1. Influence of the time-varying matrix that couples the subsystems

We will now comment on the influence of time variant coefficients of the matrix \mathbf{S} to the unit local truncation error introduced by the partitioning. For this purpose, we will assume that the heat conduction system of equation (12) is coupled to an algebraic system of equations that is used to model the fluid flow rate. In particular, the fluid flow rate is modeled as a linear function of the subsystems' states. In co-simulation, this algebraic system of equations is partitioned as well.

Let $\mathbf{S}(t^{n+1}; \Delta t) \triangleq \mathbf{S}^*(t^{n+1}; \Delta t) + \Delta\mathbf{S}(t^{n+1}; \Delta t)$ be the unknown exact value and let $\Delta\mathbf{S}(t^{n+1}; \Delta t)$ contain the unit local truncation error introduced by the partitioning in one time step. Following the same procedure as before, the unit local truncation error yields

$$\begin{aligned} \mathbf{ULTE}^{n+1}(\Delta t) &= \frac{1}{\Delta t}(\mathbf{T}(t^{n+1}) - \mathbf{T}(t^n)) \\ &\quad - \Delta t[(1 - \alpha) \mathbf{C}^{-1}(\mathbf{g}^n - [\mathbf{S}(t^n; \Delta t)] \mathbf{T}(t^n)) \\ &\quad + \alpha \mathbf{C}^{-1}(\mathbf{g}^{n+1} - [\mathbf{S}_I^*(t^{n+1}; \Delta t) + \Delta\mathbf{S}_I(t^{n+1}; \Delta t)] \mathbf{T}(t^{n+1}) \\ &\quad - [\mathbf{S}_E^*(t^{n+1}; \Delta t) + \Delta\mathbf{S}_E(t^{n+1}; \Delta t)] \mathbf{T}_P(t^{n+1}))]. \end{aligned} \quad (22)$$

The error introduced by the partitioning can be written as

$$\mathbf{ULTE}_{\text{partitioning}}^{n+1}(\Delta t) = \alpha \mathbf{C}^{-1} \Delta\mathbf{S}(t^{n+1}; \Delta t) \mathbf{T}(t^n + \alpha \Delta t) + O(\Delta t) \quad (23)$$

and thus, for the unit local truncation error to vanish, the term $\Delta\mathbf{S}(t^{n+1}; \Delta t)$ need to vanish as $\Delta t \rightarrow 0$.

5.3. Absolute stability

As described in Section 2.2.2, due to the partitioning, the second characteristic polynomial cannot be used to determine the absolute stability of the partitioned integration method. Instead, the absolute stability of the mutated numerical integration methods (7) (used with the sequential staggered solution) will be analyzed

using the Routh-Hurwitz stability criterion as reported in [13] and [22].

The stability of the partitioned numerical integration method (14) can be examined by seeking nontrivial solutions for $\mathbf{g}(t^n) = 0, n \in \{1, 2, \dots, N\}$ and $\mathbf{T}_P^{n+1} = \mathbf{T}^n$ in the form $\mathbf{T}^{n+1} = \mathbf{A}\mathbf{T}^n$, or

$$\mathbf{T}^{n+1} = \gamma\mathbf{T}^n, \quad (24)$$

where γ represents one of the eigenvalues $\{\gamma_i \in \mathbb{C} \mid i \in \{1, 2, \dots, m\}\}$ of \mathbf{A} . The eigenvalue γ is also known as the solution amplification factor. The system is stable when $\max |\gamma_i| \leq 1$ for all $i \in \{1, \dots, m\}$. To analyze the Routh-Hurwitz criterion, γ_i is replaced by

$$\gamma_i = \frac{1 + z_i}{1 - z_i}, \quad (25)$$

where $z_i \in \mathbb{C}$ for $i \in \{1, 2, \dots, m\}$. Equation (25) maps the unit circle defined by $|\gamma_i| \leq 1$ into the negative real half-plane $Re(z) \leq 0$. This implies that as long as $Re(z) \leq 0$, the numerical integration method (14) will be stable.

Substituting (24) and (25) in equation (14) the following system is obtained:

$$\mathbf{J}(z)\mathbf{T}^n = 0, \quad (26)$$

where $\mathbf{J}(z) = [2\mathbf{C} + \alpha\Delta t(\mathbf{S}_I - \mathbf{S}_E) - (1 - \alpha)\Delta t\mathbf{S}]z + \Delta t\mathbf{S}$. To obtain nontrivial solutions of equation (26), $\det\mathbf{J}(z)$ needs to be zero. If the substitutions $\tau_1 = \dot{m}c_p + K_1$, $\tau_2 = \dot{m}c_p + K_2$, $\mu = \dot{m}c_p$, $\zeta_1 = C_1$, and $\zeta_2 = C_2$ are used for implicit-implicit partitioning (15), then equation (26) yields

$$\det \begin{bmatrix} [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1]z + \Delta t\tau_1 & (\Delta t\mu - 2\alpha\Delta t\mu)z - \Delta t\mu \\ \Delta t\mu z - \Delta t\mu & [2\zeta_2 + (2\alpha - 1)\Delta t\tau_2]z + \Delta t\tau_2 \end{bmatrix} = 0, \quad (27)$$

or

$$a_0 z^2 + a_1 z + a_2 = 0, \quad (28)$$

where

$$\begin{aligned}
a_0 &= [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1][2\zeta_2 + (2\alpha - 1)\Delta t\tau_2] + (2\alpha - 1)\Delta t^2\mu^2, \\
a_1 &= [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1]\Delta t\tau_2 + [2\zeta_2 + (2\alpha - 1)\Delta t\tau_2]\Delta t\tau_1 \\
&\quad + 2(1 - \alpha)\Delta t^2\mu^2, \text{ and} \\
a_2 &= \Delta t^2\tau_1\tau_2 - \Delta t^2\mu^2.
\end{aligned}$$

The Routh-Hurwitz criterion [23] defines conditions for the coefficients a_0 , a_1 , and a_2 , in order to have all the roots, z , of the equation (28) in the left half plane $Re(z) \leq 0$. The criterion is that all the roots of the polynomial (28) have negative real parts if and only if the inequalities $a_0 > 0$, $a_1 \geq 0$ and $a_2 \geq 0$ hold.

If $\alpha \geq 1/2$ then both a_0 and a_1 are positive. The criterion $\tau_1\tau_2 \geq \mu^2$ is additionally required in order to satisfy the third inequality. It is obvious that this is satisfied, as $(\dot{m}c_p + K_1)(\dot{m}c_p + K_2) \geq (\dot{m}c_p)^2$ always holds. Therefore, the co-simulation is unconditionally stable if $\alpha \geq 1/2$.

If $\alpha < 1/2$, the co-simulation is conditionally stable. The necessary stability criteria is $\Delta t_0^2 < \frac{4\zeta_1\zeta_2}{(1-2\alpha)^2(\tau_1\tau_2 - \mu^2)}$.

For implicit-explicit partitioning the stability criterion differs. The coefficient a_2 remains unchanged, while even for the time weighting factor $\alpha \geq 1/2$ the inequalities $a_0 > 0$, $a_1 \geq 0$ do not hold for some values of the coupling term $\dot{m}c_p$.

It can be shown that for $\alpha = 1/2$, the critical time step equals $\Delta t_0 = 2/\mu$. More general expressions for the critical time step for the implicit-explicit partitioning can be obtained by following a cumbersome calculation which does not result in an elegant formulae. Due to the partitioning of the diagonal term of the \mathbf{S} matrix, implicit-explicit partitioning has a more strict stability criterion.

If the subsystems are linked by a control loop, the absolute stability criterion is influenced by the control parameters. However, the stability criterion is difficult to express and we will not report it here.

5.4. Comparison to an analytical solution

To illustrate the behavior of the error introduced by the partitioning, a free-response system, represented by equation (12) in which the forcing function is set to $\mathbf{g}(t) = 0$ is used. The analytic solution is

$$\mathbf{T} = \mathbf{T}_0 e^{(t-a)\mathbf{C}^{-1}\mathbf{S}}. \quad (29)$$

The analytic solution is compared to the results obtained by co-simulation, represented by equation (14) with $\mathbf{g}(t) = 0$, using different coupling time steps. The results of one set of parameters² are shown in Figure 5. Increasing the coupling time step degrades the accuracy of co-simulation results as expected.

5.5. Measures to improve accuracy of co-simulation solution

We will now list possible measures that can be used to improve accuracy of loosely coupled co-simulation.

- Co-simulation can be executed with small time steps.
- Instead of loose coupling, co-simulation can be implemented using strong coupling. As discussed in the introduction, the strong coupling approach requires an increased effort for code modification to ensure the correct synchronization and rewinding of the state of the passively executed simulator. In addition, it requires that the simulators compute their iterative solutions at an accuracy that is higher than the accuracy that is used by the co-simulation manager to determine the iteration on the coupling variables. Such an implementation is discussed in [24].
- Co-simulation can run with a variable time step size. The time step could be calculated based on the unit local truncation error. Both coupled simulators

²The illustrative purpose of examples in this paper is in the relative difference between the presented examples. The specific sets of parameters are thus not of the importance for this purpose and will not be reported in the paper.

need to have additional features in order to change the simulation time step during the execution time. As this feature is available only in few state of the art tools from the BPS domain, it will not be discussed further.

- Instead of the zero-order prediction of the coupling data, a more accurate prediction can be used. This approach will be addressed further in more detail.

5.5.1. Predictions

The accuracy might be improved by using a more accurate prediction for the coupling data instead of the zero-order prediction. We define the approximation to the first order prediction as

$$\mathbf{T}_P^{n+1} \triangleq 2\mathbf{T}^n - \mathbf{T}^{n-1}. \quad (30)$$

Figure 6 shows comparison between results obtained by co-simulation with and without predictions given a set of subsystems' parameters³. The reference curve is obtained by co-simulation without predictions using significantly smaller time steps than in the other two co-simulations. The results of co-simulation using first-order prediction is significantly closer to the reference curve than the results of co-simulation using zero-order prediction.

Piperno [25] compared the performance of several predictions and found that the most accurate and stable prediction is the one that, adapted for the case studied here, has the form: $\mathbf{T}_P^{n+1} = \mathbf{T}^n + \Delta t(1.5\dot{\mathbf{T}}^n - 0.5\dot{\mathbf{T}}^{n-1})$, which can be approximated by $\mathbf{T}_P^{n+1} \approx 2.5\mathbf{T}^n - 2\mathbf{T}^{n-1} + 0.5\mathbf{T}^{n-2}$. Given another set of subsystems' parameters, the comparison between the simple first-order prediction and the prediction suggested by Piperno for larger time steps are shown in Figure 7. The reference curve is obtained by co-simulation without predictions using significantly smaller time steps

³Depending on the order of the prediction, there may be several coupling data that need to be initialized. In the first time step, the values of the coupling data in one or more preceding time steps are assumed to be equal to the initial value of that variable.

than in the other two co-simulations. The prediction suggested by Piperno is closer to the reference curve.

In the above numerical experiments, the coupling data changed little per time step. In more realistic simulations with fast transients in coupling data, we observed that higher-order predictions overestimated the future values, which led to a lower accuracy [26].

6. Conclusions

Due to the high effort required to develop building performance prediction tools and the low market interest, the significant effort required to develop BPS tools has been left to a fragmented research community. This has resulted in a slower development and inferior capabilities compared to simulation tools used in other industry sectors, such as the automotive, electronics or aerospace sector. State of the art BPS tools are not equally suited for modeling and simulation of the relevant building aspects. Moreover, the user's requirements often exceed the tool's functionality.

One way to alleviate this problem is to use co-simulation. The focus of this article was to investigate the characteristics of co-simulation that uses loose coupling, since this strategy can be most efficiently implemented using legacy BPS tools.

The study reported in this article was limited to partitioning of linear one-step numerical integration methods. The investigation showed that new numerical integration methods, obtained by the partitioning of the original numerical integration methods, are zero-stable and consistent, and thus convergent. The unit local truncation error of the original numerical integration method is changed. The order of the unit local truncation error due to the partitioning is one, which means that the order of the unit local truncation error of the second-order numerical integration methods is reduced by the partitioning.

To gain insights into how HVAC system characteristics influence the unit local

truncation error introduced by the partitioning, the unit local truncation error analysis was continued on a two-body system, described by a time-invariant system of linear first-order differential equations. It was shown that the unit local truncation error was related to the system characteristics as follows: The greater the heat capacity of the subsystem simulated in the external simulator, the smaller the unit local truncation error is. Also, the lower the rate of change of the delayed coupling and the smaller Δt , the smaller the unit local truncation error is.

An investigation of the unit local truncation error under the influence of time varying coefficients of the matrix \mathbf{S} showed the following: The smaller the unit local truncation error in the time varying coefficients, introduced by the partitioning in one simulation time step, i.e., $\Delta\mathbf{S}(t^{n+1}; \Delta t)$, the smaller the overall unit local truncation error is.

A stability analysis using the Routh-Hurwitz stability criterion was performed on the two-body system. The analysis resulted in a cumbersome calculation procedure already when small complexities were introduced to the problem. For the two-body time-invariant system represented by a system of linear first order differential equations, the partitioned numerical integration method is unconditionally stable for $\alpha \geq 1/2$.

To enhance accuracy, several measures can be taken, such as (i) the use of strong coupling, (ii) the use of a variable time step and (iii) the use of predictions for the coupling data. If changes in the coupling data are small, the use of a first-order prediction increases accuracy of loosely coupled co-simulation.

7. Nomenclature

7.1. Conventions

1. Vectors are typeset in bold fonts.
2. Superscripts denote the time step number.

3. $f(\cdot)$ denotes a function where (\cdot) stands for the undesignated variables. $f(x)$ denotes the value of $f(\cdot)$ for the argument x . $f: A \rightarrow B$ indicates that the domain of $f(\cdot)$ is in the space A , and that the image of $f(\cdot)$ is in the space B .
4. $y(t)$ is a state variable and $\dot{y}(t)$ denotes the time derivative of the state variable.
5. We say that a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is once Lipschitz continuously differentiable if $f(\cdot)$ is defined on \mathbb{R}^n , and if $f(\cdot)$ has a Lipschitz continuous derivative on \mathbb{R}^n .

7.2. Symbols

$a \in A$	a is an element of A
a	polynomial coefficients
A	area
C	heat capacity
\mathbf{S}	interaction matrix
c_p	specific heat capacity
E	explicit
\mathbf{g}	forcing function
i, j, k, n	counters
I	implicit
h	convective heat transfer coefficient
HG	heat gain
inf	infiltration
L	Lipschitz constant
LTE	local truncation error
\dot{m}	mass flow
\mathbb{N}	set of natural numbers, $\mathbb{N} = \{0, 1, 2, \dots\}$
\dot{Q}	heat rate
P	predicted
\mathbb{R}	set of real numbers
\mathbf{C}	diagonal matrix for the capacity
s	surface
sys	system related
U	specific heat capacity
ULTE	heat transfer coefficient
\mathbf{T}	temperature vector
t	time
z	zone, and complex number
$\alpha, \beta, \mu, \tau, \zeta$	scalar parameters
Δ	difference

8. Acknowledgement

This research was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technologies of the U.S. Department of Energy, under Contract No. DE-AC02-05CH11231.

References

- [1] M. Trcka, J.L.M. Hensen, M. Wetter, Co-simulation of innovative integrated HVAC systems in buildings, in *Journal of Building Performance Simulation*, 2 (3), pp. 209–230 (2009).
- [2] ASHRAE, ASHRAE 2020 vision, position, strategies, actions, Tech. rep., The American Society of Heating, Refrigerating and Air-Conditioning Engineers, Available at: http://www.ASHRAE.org/doclib/20080226_ASHRAEvision2020.pdf [Accessed: April, 2008] (2008).
- [3] K. Papamichael, P. Vineeta, Barriers in developing and using simulation-based decision-support software, in: *Proceedings of ACEEE 2002 summer study on energy efficiency in buildings*, American Council for an Energy-Efficient Economy, CA, USA., 2002.
- [4] A. J. Wright, D. Bloomfield, T. J. Wiltshire, Building simulation and building representation: Overview of current developments, *Building Serv. Eng. Res. Technol* 13 (1) (1992) 1–11.
- [5] V. C. Thomas, Using M-E design programs (some reasons for the lack of progress), in: *Proceedings of IBPSA-USA, International Building Performance Simulation Association*, Chicago, USA, 2006.
- [6] P. Sahlin, The methods of 2020 for building envelope and HVAC systems simulation - Will the present tools survive?, in: *CIBSE conference*, Dublin, Ireland, 2000.

- [7] J. L. M. Hensen, On the thermal interaction of building structure and heating and ventilating system, Ph.D. thesis, Eindhoven University of Technology (1991).
- [8] J. L. M. Hensen, J. A. Clarke, Integrated simulation for HVAC performance prediction: State of the art illustration, in: Proceedings of Int. ASHRAE/CIBSE Conf. “20-20 Vision”, Dublin, Ireland, 2000.
- [9] E. Struler, J. Hoefliger, L. V. Kale, M. Bhandarkar, A new approach to software integration frameworks for multi-physics simulation codes, in: Proceedings of IFIP TC2/WG2.5 Working Conference on Architecture of Scientific Software, Ottawa, Canada, 2000, pp. 87–104.
- [10] Z. Zhai, Developing an integrated building design tool by coupling building energy simulation and computational fluid dynamics programs, Ph.D. thesis, Massachusetts Institute of Technology, USA (2003).
- [11] J. L. M. Hensen, A comparison of coupled and de-coupled solutions for temperature and air flow in a building, ASHRAE Transactions 105 (1999) 962–969.
- [12] C. A. Felippa, K. C. Park, C. Farhat, Partitioned analysis of coupled mechanical systems, Tech. Rep. CU-CAS-99-06, Center for aerospace structures, University of Colorado (1999).
- [13] K. C. Park, Partitioned transient analysis procedures for coupled-field problems: Stability analysis, Journal of Applied Mechanics 47 (1980) 370–376.
- [14] E. Djunaedy, J. L. M. Hensen, M. G. L. C. Loomans, Towards external coupling of building energy and air flow modeling programs, ASHRAE Transactions 109 (2) (2003) 771–787.
- [15] W. Keilholz, TRNSYS World-wide, The journal of the international building performance simulation association, ibpsaNEWS 12 (1).

- [16] M. Wetter, P. Haves, A modular building controls virtual test bed for the integration of heterogeneous systems, in: Proceedings of SimBuild, 3rd National Conference of IBPSA-USA, International Building Performance Simulation Association, USA chapter, Bekeley, CA, USA, 2008, pp. 69–76.
- [17] C. W. Gear, Numerical initial value problems in ordinary differential equations, Prentice-Hall, Englewood Cliffs, N.J., 1971.
- [18] J. D. Lambert, Numerical methods for ordinary differential systems, the initial value problem, John Wiley & Sons, West Sussex, U.K., 1991.
- [19] G. H. Golub, J. M. Ortega, Scientific computing and differential equations, Academic Press, London, U.K., 1992.
- [20] R. E. A. Kubler, Two methods of simulator coupling, Mathematical and computer modeling of dynamic systems 6 (2) (2000) 93–113.
- [21] G. Hall, J. M. Watt, Modern numerical methods for ordinary differential equations, Clarendon press, Oxford, U.K., 1976.
- [22] P. H. Tseng, A. Sciortino, M. T. van Genuchten, A partitioned solution procedure for simulating water flow in a variably saturated dual-porosity medium, Advances in Water Resources 18 (6) (1995) 335–343.
- [23] F. R. Gantmacher, The theory of matrices, Chelsea Publishing Complaney, New York, 1974.
- [24] M. Trčka, Co-simulation for performance prediction of innovative integrated mechanical energy systems in buildings, Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands (2008).
- [25] S. Piperno, Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations, International journal for numerical methods in fluids 25 (1997) 1207–1226.

- [26] M. Radošević, J. L. M. Hensen, A. Wijsman, Implementation strategies for distributed modeling and simulation of building systems, in: Proceedings of 9th International IBPSA Conference, International Building Performance Simulation Association, Montreal, Canada, 2005, pp. 995–1002.

List of figures

Figure 1 Sequence of coupling data exchange. a) Time-state scheme of strong coupling; b) Time-state scheme of loose coupling with sequential simulators execution; and c) Time-state scheme of loose coupling with parallel simulators execution.

Figure 2 Flow-chart of the loosely-coupled implementation.

Figure 3 Flow-chart of the strongly-coupled implementation.

Figure 4 Two-body system.

Figure 5 Comparison between analytical solution and the results of co-simulation using different coupling time steps for the free-response system.

Figure 6 Comparison between co-simulation results with and without predictions.

Figure 7 Co-simulation results using different predictors plotted against the reference curve, obtained with the larger time step.

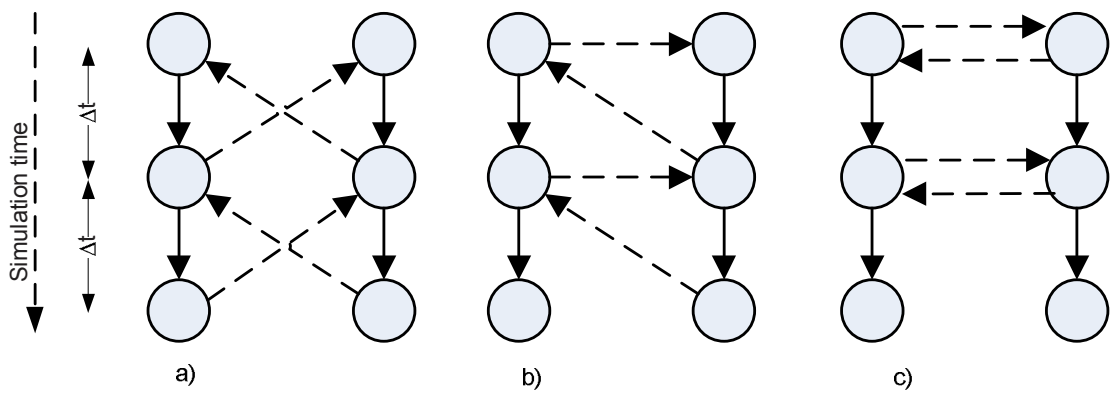


Figure 1: Sequence of coupling data exchange. a) Time-state scheme of strong coupling; b) Time-state scheme of loose coupling with sequential simulators execution; and c) Time-state scheme of loose coupling with parallel simulators execution.

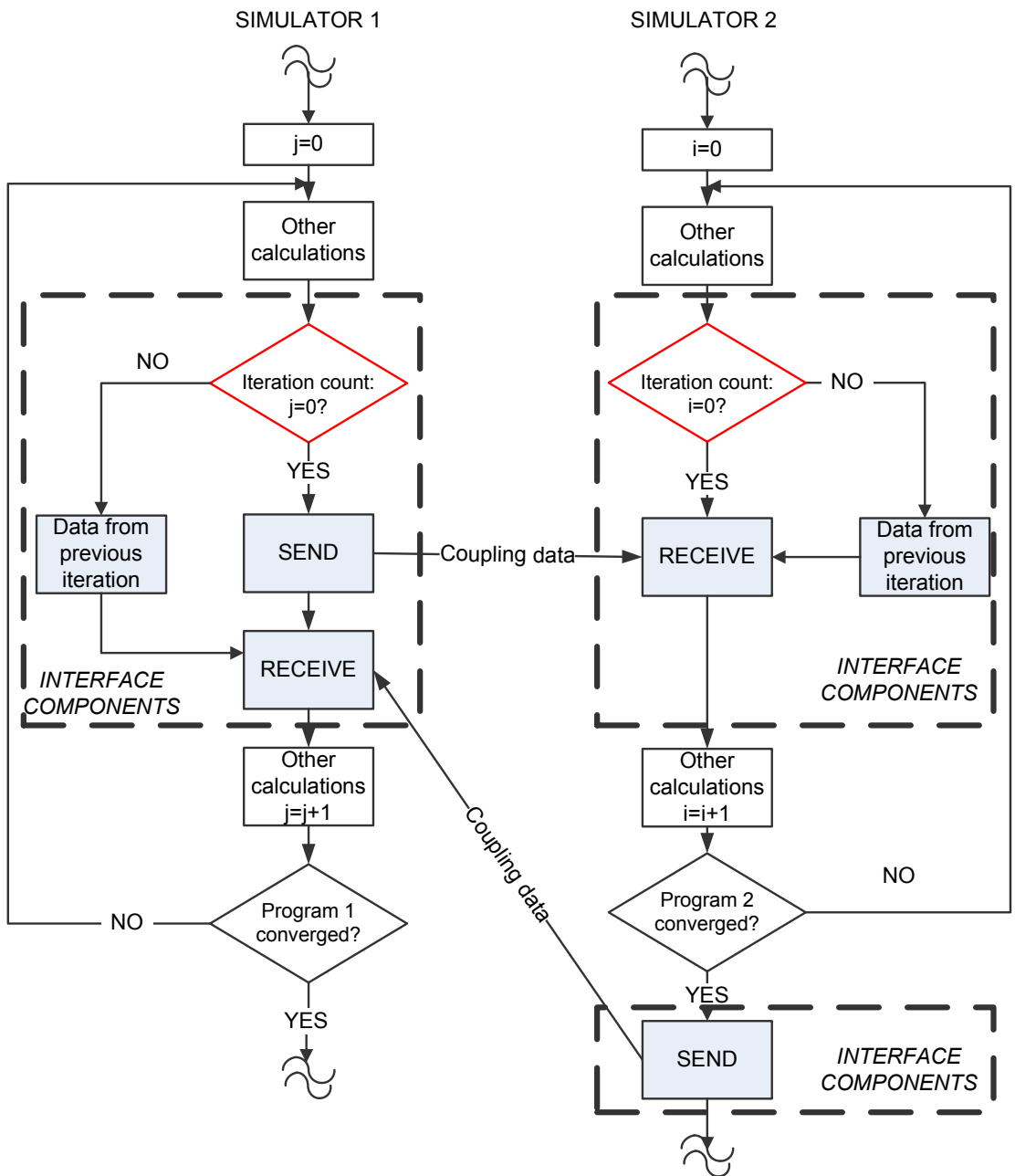


Figure 2: Flow-chart of the loosely-coupled implementation.

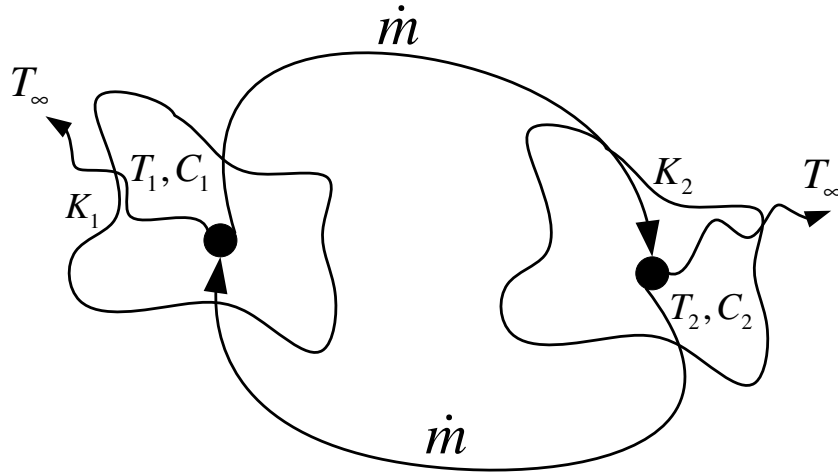


Figure 4: Two-body system.

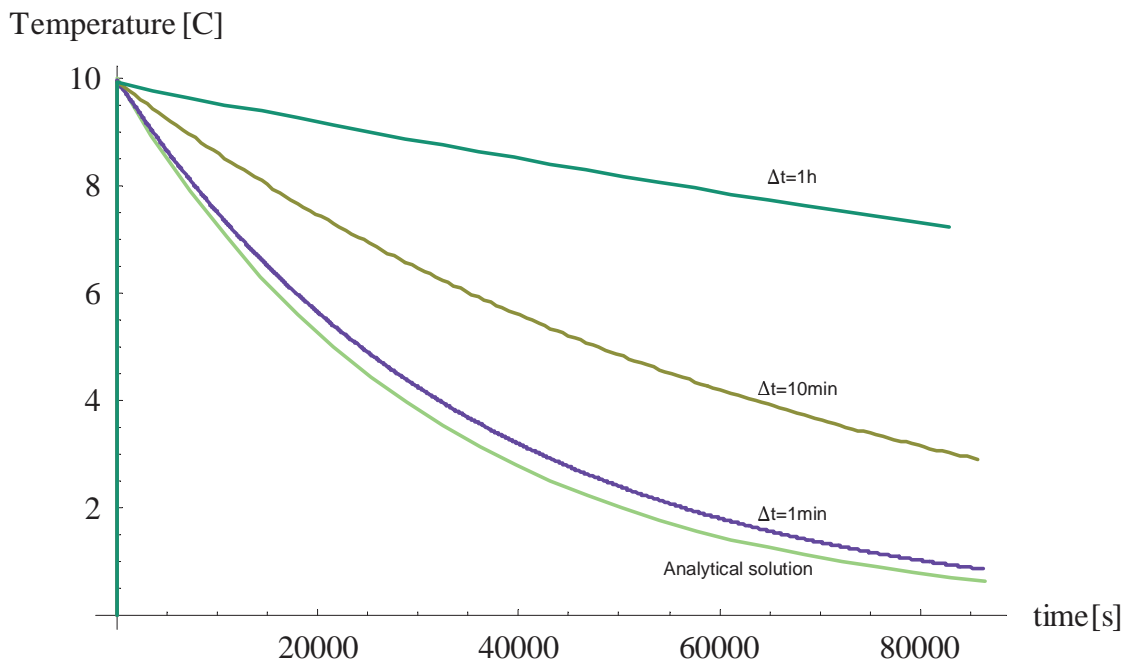


Figure 5: Comparison between analytical solution and the results of co-simulation using different coupling time steps for the free-response system. The temperature of the subsystem denoted with index 2.

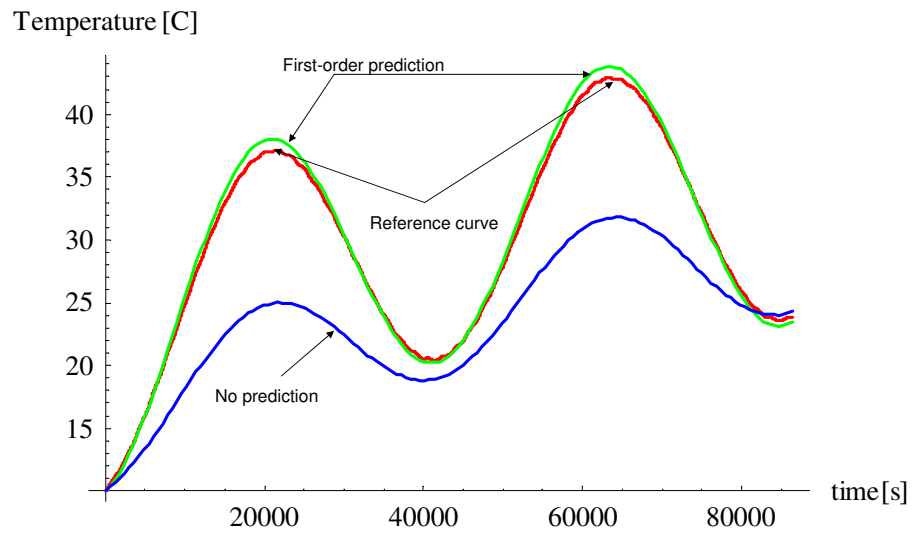


Figure 6: Comparison between co-simulation results with and without predictions. The temperature of the subsystem denoted with index 2.

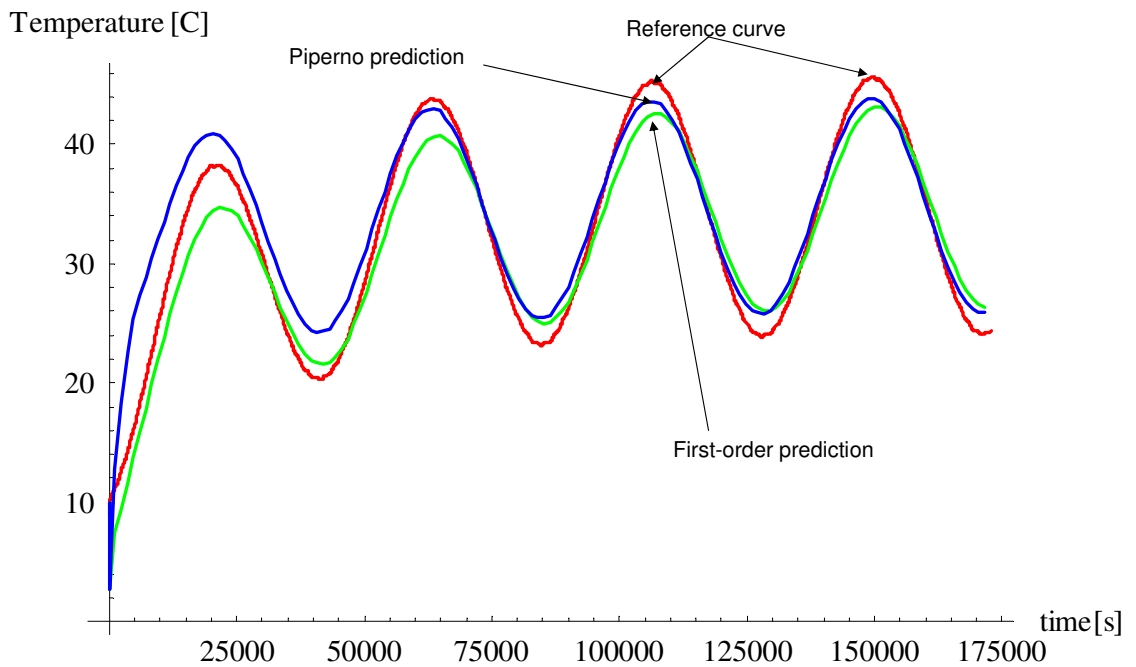


Figure 7: Co-simulation results using different predictors plotted against the reference curve, obtained with the larger time step. The temperature of the subsystem denoted with index 2.