Can complex systems really be simulated?

Tamas Kalmar-Nagy^a, Ilinca Stanciulescu^{b,*}

^aMitsubishi Electric Research Laboratory, Cambridge, MA, 02139 ^bRice University, Department of Civil and Environmental Engineering, Houston, TX, 77005, U.S.A

Abstract

The simulation of complex systems is important in many fields of science and in real-world applications. Such systems are composed of many interacting subsystems. There might exist different software packages for simulating the individual subsystems and co-simulation refers to the simultaneous execution of multiple interacting subsystem simulators. Simulation or co-simulation, if not designed properly, can return misleading numerical solutions (unstable numerical solutions for what is in fact a stable system or vice versa). To understand the cause of these numerical artifacts, we first propose a simple mathematical model for co-simulation, and then construct stability charts. These charts shed light on transitions between stable and unstable behaviour in co-simulation. Our goal is to understand the stability properties of the simulated and co-simulated representation of the continuous system. We will achieve this goal by expressing the trace and determinant of the discretized system in terms of the trace and determinant of the continuous system to establish stability criteria.

Keywords: stability, simulation and co-simulation, stability boundaries

1. Introduction

Many place the origin of the theory of stability in the first theoretical study of unstable structural systems by Leonhard Euler [1], who analyzed the behaviour of a slender column under compressive loading. Others point

Preprint submitted to Applied Mathematics and Computation

^{*}Corresponding author. Tel.: +1713 348 4704; fax: +1713 348 5268.

Email address: ilinca@rice.edu (Ilinca Stanciulescu)

out that the interest for these concepts existed well before that time, even during the Medieval and Renaissance periods [2], with early studies being traced back to ancient Greece. Now, more than two centuries after Euler, there is a vast literature dedicated to structural stability. Several methods exist for obtaining the stability criteria; see, for instance, [3], [4], and [5]. Soon after Euler, Lagrange was probably the first to formalize the framework for stability analysis in his study of the stability of planetary orbits (1776). More recently, in the late 19th century, Lyapunov set the solid mathematical foundations on which today's stability theory rely on. Today, we also talk about numerical stability, a property associated with numerical algorithms.

From a computational modelling perspective, answering "Is the system stable?" is challenging due to a variety of issues. Among these are the interaction between multiple scales, fields, and processes. More open questions emanate from concerns regarding numerical stability in that the method employed may itself introduce instability or even mask the true system stability. The goal of this article is to propose techniques that will allow analysts to chose algorithmic frameworks adequate for the specific problem to be solved.

Metrics of system performance have classically been obtained through dynamic simulation of the system as a whole. In such monolithic simulations the numerical schemes rely on solving for the coupled variables simultaneously. Such solutions can be computationally inefficient and also prone to ill-conditioning. As modern engineering systems are becoming increasingly complex, it is now more common to solve large systems by decomposition and using specific solvers for the different partitions (while continuously updating in each the information from the other) and iterating until numerical convergence is reached. The decomposition can be motivated mathematically (equations of different types, elliptic, parabolic, etc.) or by the presence of subsystems described by different areas of physics (a coupled flow-structure problem, or thermo-mechanical problem for example). One problem plaguing this area of research is that different communities tend to use different terminology for concepts that are very much related: staggered methods (schemes), decomposition, operator split, co-simulation, etc. In the context of this paper, simulation refers to a numerical solution (discretization) of a continuous system. Gu and Asada [6] describe the concept of *co-simulation* as simultaneous execution of multiple interacting subsystem simulators. Co-simulation techniques are widely adopted to interface different modelling and simulation domains and tools. The Functional Mock-up Interface (FMI) is a tool independent standard to support both model exchange and co-simulation of dynamic models using a combination of xml-files and compiled C-code (www.fmi-standard.org). In general the xml-file is used in combination with an executable binary and the solver is already embedded in the binary. Regardless of the motivating factor for decomposing the system (mathematical or physical), we need to be careful when performing the analysis of a coupled/partitioned system. Even if high fidelity subsystem models are available, integrating the subsystem simulators to predict overall system behavior is still a challenging task. Problems include numerical instabilities, spurious solutions, etc. It is possible that the main system is stable while the simulation shows instability or it may identify an unstable system as stable. For such systems, we can no longer discuss the unconditional sta*bility* of the time integration, we rather need to enforce the condition that the simulation of the system preserves the stability properties of the system under study. In other words, we need time integrators and staggered schemes for which the range of parameters of physical interest place the system that is integrated exclusively in the areas where the stability, or lack thereof, of the continuous and discretized systems coincide.

Wang et al [7] introduced a gluing algorithm that couples "in a plug-andplay manner" different component models in the simulation of multibody systems. Their algorithm assumes that partitioning is not controlled by the user and focuses exclusively on how to make the partitions communicate (i.e., exchange interface information) during simulations. More recently, an iterative scheme for the coupling of the subsystem at predefined time points is introduced in [8].

In the literature specific to the finite element community, the term *staggered scheme* is often encountered. Traditional finite element staggered time stepping schemes are computationally efficient when the iterations converge to the monolithic solution but often times these methods are only conditionally stable. Alternatives that keep the unconditional stability of monolithic schemes were proposed in the literature for various types of coupled problems. Armero and Simo [9] propose for instance a fractional step method based on a two phase operator split for the solution of nonlinear thermomechanical problems and use the same approach for thermoplasticity [10]. Growth in biological tissue is modelled as a coupled problem by Garikipati and co-workers [11] and also solved by means of a staggered scheme based on operator split-ting techniques. In such fractional step methods, the splitting errors are often hard to control and sometimes make the methods inefficient (in particular in the case of strongly coupled fields). Vijalapura and Govindjee [12] propose

an adaptive hybrid approach to control the errors and improve the numerical performance. A coupling for heterogeneous time integrators in a domain decomposition framework whose interface is not affecting global stability is proposed in [13].

In this paper we use the Runge-Kutta method to simulate (i.e., numerically integrate) a simple two-dimensional linear system and its co-simulated counterparts. Our goal is to contrast the stability properties of the continuous and simulated/co-simulated (discrete) systems. We will achieve this goal by expressing the trace and determinant of the discretized system in terms of those of the continuous system to establish stability criteria. Accuracy of co-simulation schemes will not be discussed in this paper.

The paper is organized as follows: in section 2 we discuss simulation and co-simulation, give examples co-simulation strategies, and introduce the main concepts by means of specific examples of algorithmic expressions using as test case the Runge-Kutta integrator. In section 3 we introduce the framework for the comparison between the stability of continuous and discrete systems. Sections 4 and 5 demonstrate the stability properties of the simulated and co-simulated systems through a case study focused on the most basic system for which we can discuss co-simulation: a linear two-dimensional system. Section 6 completes our paper with a discussion of the findings.

2. Simulation and Co-simulation

To understand the concept of co-simulation, consider the fluid-structure interaction of a wing in airflow (Fig. 1a). The system is described by the continuous evolution equation

$$\mathcal{D}\left(\mathbf{z}\left(t\right)\right) = \mathbf{f}(\mathbf{z}\left(t\right)),\tag{1}$$

where $\mathbf{z}(t)$ is the state vector and \mathcal{D} is a differential operator. The full state vector can be partitioned into one composed of the states of the wing and airflow (Fig. 1b). The equation governing the system can be rewritten as

$$\mathcal{D}\left(\begin{array}{c}\mathbf{x}\left(t\right)\\\mathbf{y}\left(t\right)\end{array}\right) = \left(\begin{array}{c}\mathbf{f}_{1}\left(\mathbf{x}\left(t\right),\mathbf{y}\left(t\right)\right)\\\mathbf{f}_{2}\left(\mathbf{x}\left(t\right),\mathbf{y}\left(t\right)\right)\end{array}\right),\tag{2}$$

where $\mathbf{x}(t)$ and $\mathbf{y}(t)$ are the state vectors of the the wing and airflow, respectively. The system can then be co-simulated by separately computing

the deformation of the wing and the velocity/pressure fields of the airflow and then coupling them (Fig. 1c) in an appropriate manner [14].



Figure 1: Fluid structure interaction

In the context of this paper, *simulation* of a system refers to the approximation of the solution by numerical integration of the equations, i.e., the approximation of the state vector at discrete time steps. *Co-simulation* refers to *partitioning* a coupled system into *subsystems* that are separately simulated (numerically integrated) with a suitable exchange of states at predefined time instances to account for the coupling. We now explain these various concepts. First, let us consider Eq. (1) where \mathcal{D} is now replaced with a simple differential operator, i.e.,

$$\frac{d}{dt}\mathbf{z}\left(t\right) = \mathbf{f}(\mathbf{z}\left(t\right)),\tag{3}$$

subject to the initial condition $\mathbf{z}(t_0) = \mathbf{z}_0$. Without loss of generality we discuss the case of a system with two partitions, $\mathbf{z}(t) = \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix}$. Eq. (3) can now be expressed as $\frac{d}{dt}\mathbf{z}(t) = \frac{d}{dt}\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{y}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1(\mathbf{x}(t), \mathbf{y}(t)) \\ \mathbf{f}_2(\mathbf{x}(t), \mathbf{y}(t)) \end{pmatrix}$ or equivalently

$$\dot{\mathbf{x}}(t) = \mathbf{f}_{1}\left(\mathbf{x}(t), \mathbf{y}(t)\right), \qquad (4)$$

$$\dot{\mathbf{y}}(t) = \mathbf{f}_2(\mathbf{x}(t), \mathbf{y}(t)).$$
(5)

When the first subsystem (Eq. (4)) is simulated separately, $\mathbf{y}(t)$ is no longer a state variable, but rather an explicitly time-dependent term, an *input*. This input is not fully known (since it also depends on $\mathbf{x}(t)$!), so some form of *prediction* for $\mathbf{y}(t)$ is needed. Let us denote this prediction by $\mathbf{v}(t)$. Similarly, if the second subsystem (Eq. (5)) is simulated separately, a prediction $\mathbf{u}(t)$ is needed for its input $\mathbf{x}(t)$. The system can be written as

$$\dot{\mathbf{x}}(t) = \mathbf{f}_1(\mathbf{x}(t), \mathbf{v}(t)), \qquad (6)$$

$$\dot{\mathbf{y}}(t) = \mathbf{f}_2(\mathbf{u}(t), \mathbf{y}(t)).$$
(7)

The form $\mathbf{u}(t)$ and $\mathbf{v}(t)$ take, e.g., the input-output relationship of the subsystems, along with the iterative scheme (if any) determine the *strategy* or *type* of the co-simulation [15].

2.1. Simulation

Once again, *simulation* refers to the numerical approximation (by numerical integration) of the state $\mathbf{z}(t)$ of the system. Consider the initial value problem (this really is eq. (6))

$$\dot{\mathbf{z}}(t) = \mathbf{g}(\mathbf{z}(t), \mathbf{w}(t)), \tag{8}$$

$$\mathbf{z}(0) = \mathbf{z}_0. \tag{9}$$

A widely used numerical integration method is the 4^{th} order Runge-Kutta method with step size h. A straightforward extension of the basic Runge-Kutta integration for systems with input (i.e., the system (8-9)) approximates the state vector at $t_{n+1} = (n+1)h$ as

$$\mathbf{z}_{n+1} := \mathbf{z} \left(t_{n+1} \right) = \mathbf{z}_n + \frac{1}{6} h(\mathbf{k_1} + 2\mathbf{k_2} + 2\mathbf{k_3} + \mathbf{k_4}),$$

where $\mathbf{k_1} = \mathbf{g}(\mathbf{z}_n, \mathbf{w}(t_n)), \mathbf{k_2} = \mathbf{g}(\mathbf{z}_n + \frac{1}{2}h\mathbf{k_1}, \mathbf{w}(t_n + \frac{h}{2})), \mathbf{k_3} = \mathbf{g}(\mathbf{z}_n + \frac{1}{2}h\mathbf{k_2}, \mathbf{w}(t_n + \frac{h}{2})), \mathbf{k_4} = \mathbf{g}(\mathbf{z}_n + h\mathbf{k_3}, \mathbf{w}(t_{n+1}))$. The Runge-Kutta approximation for system (8) can therefore be represented as the discrete map

$$\mathbf{z}_{n+1} = \mathcal{R}\mathcal{K}_{\mathbf{g},h}(\mathbf{z}_n, \mathbf{w}_n, \mathbf{w}_{n+\frac{1}{2}}, \mathbf{w}_{n+1}),$$
(10)

$$\mathbf{z}_0 = \mathbf{z}(0),\tag{11}$$

where $\mathbf{w}_{n+\frac{1}{2}} = \mathbf{w}(t_n + \frac{h}{2}).$

2.2. Co-simulation

The co-simulation of system (4)-(5) is now formally written as

$$\mathbf{x}_{n+1} = \mathcal{RK}_{\mathbf{f}_1,h}(\mathbf{x}_n \mathbf{v}_n, \mathbf{v}_{n+\frac{1}{2}}, \mathbf{v}_{n+1}); \quad \mathbf{y}_{n+1} = \mathcal{RK}_{\mathbf{f}_2,h}(\mathbf{y}_n, \mathbf{u}_n, \mathbf{u}_{n+\frac{1}{2}}, \mathbf{u}_{n+1}).$$

Note that any numerical integration method seeks to find the states at specific time instances, in this case (e.g., when using equal time steps), the states are evaluated just at time instants t = ih for $i \in \mathbb{N}$. Therefore, there is no calculated value for $x(t_n + \frac{h}{2})$ and $y(t_n + \frac{h}{2})$ nor are the states at $t = t_{n+1}$ available before the calculation of \mathbf{x}_{n+1} and \mathbf{y}_{n+1} . Since these states are needed (as inputs for another partition of the system), they are estimated based on different strategies. Each strategy, (along with the partitioning choice and the techniques for exchange of information between partitions) leads to a specific type of co-simulation. Here, attention is focused on three such types illustrated in Fig. 2, namely cross, zigzag and strong co-simulation.



Figure 2: Three types of co-simulation

2.2.1. Cross coupling

With this strategy, the coupled partitions are simulated in parallel (Fig. 2a). The prediction is simple: we consider the inputs piecewise constant, i.e.,

$$\mathbf{v}_n = \mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_{n+1} = \mathbf{y}_n,\tag{12}$$

$$\mathbf{u}_n = \mathbf{u}_{n+\frac{1}{2}} = \mathbf{u}_{n+1} = \mathbf{x}_n,\tag{13}$$

and thus $\mathbf{x}_{n+1} = \mathcal{RK}_{\mathbf{f}_1}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{y}_n, \mathbf{y}_n)$, and $\mathbf{y}_{n+1} = \mathcal{RK}_{\mathbf{f}_2}(\mathbf{y}_n, \mathbf{x}_n, \mathbf{x}_n, \mathbf{x}_n)$.

2.2.2. Zigzag coupling

In this approach, the partitions are simulated in sequence (Fig. 2b). In other words, first the value of state vector \mathbf{x} at the $(n+1)^{th}$ time step, \mathbf{x}_{n+1} , is determined by means of \mathbf{x}_n and \mathbf{y}_n . Then \mathbf{x}_{n+1} , \mathbf{x}_n and \mathbf{y}_n are utilized to calculate \mathbf{y}_{n+1} . Since \mathbf{x}_{n+1} is calculated first, $\mathbf{v}_{n+\frac{1}{2}}$ and \mathbf{v}_{n+1} are chosen the same way as in the cross coupling procedure (Eq. (12)). On the other hand, because \mathbf{y}_{n+1} is calculated after \mathbf{x}_{n+1} , the value of \mathbf{x}_{n+1} is available when determining \mathbf{y}_{n+1} and so $\mathbf{u}_{n+1} = \mathbf{x}_{n+1}$. In addition, the value at the midpoint is estimated as the average of \mathbf{u}_n and \mathbf{u}_{n+1} , i.e., $\mathbf{u}_{n+\frac{1}{2}} = (\mathbf{u}_n + \mathbf{u}_{n+1})/2 = (\mathbf{x}_n + \mathbf{x}_{n+1})/2$. Accordingly, the state vectors are approximated as

$$\mathbf{x}_{n+1} = \mathcal{RK}_{\mathbf{f}_1,h}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{y}_n, \mathbf{y}_n),$$

$$\mathbf{y}_{n+1} = \mathcal{RK}_{\mathbf{f}_2,h}(\mathbf{y}_n, \mathbf{x}_n, \frac{\mathbf{x}_n + \mathcal{RK}_{\mathbf{f},h}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{y}_n, \mathbf{y}_n)}{2}, \mathcal{RK}_{\mathbf{f},h}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{y}_n, \mathbf{y}_n)).$$

2.2.3. Strong coupling

This co-simulation strategy relies on an iterative sequence to continuously improve on the estimates of the inputs from the two partitions:

$$\mathbf{x}_{n+1}^{l+1} = \mathcal{RK}_{\mathbf{f}_1,h}(\mathbf{x}_n, \mathbf{y}_n, \mathbf{y}_{n+\frac{1}{2}}^l, \mathbf{y}_{n+1}^l),$$
$$\mathbf{y}_{n+1}^{l+1} = \mathcal{RK}_{\mathbf{f}_2,h}(\mathbf{y}_n, \mathbf{x}_n, \mathbf{x}_{n+\frac{1}{2}}^l, \mathbf{x}_{n+1}^l),$$
(14)

with initial values specified as $\mathbf{y}_{n+1}^0 = \mathbf{y}_n$, $\mathbf{x}_{n+1}^0 = \mathbf{x}_n$. The intermediate values of the inputs are estimated as the averages $\mathbf{x}_{n+\frac{1}{2}}^l = \frac{\mathbf{x}_n + \mathbf{x}_{n+1}^l}{2}$ and $\mathbf{y}_{n+\frac{1}{2}}^l = \frac{\mathbf{y}_n + \mathbf{y}_{n+1}^l}{2}$. The converged values of the states (provided the iteration is convergent, see section 5.3) are then taken for \mathbf{x}_{n+1} and \mathbf{y}_{n+1} , i.e., $\mathbf{x}_{n+1} = \lim_{l \to \infty} \mathbf{x}_{n+1}^l$ and $\mathbf{y}_{n+1} = \lim_{l \to \infty} \mathbf{y}_{n+1}^l$.

3. Stability Properties of Continuous and Discrete Two-Dimensional Linear Systems

Our goal is to better understand the operational stability of the cosimulation strategies described in Section 2. The "minimal" system for which co-simulation is meaningful is the simple two-dimensional linear system

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z}.\tag{15}$$

In sections 4 and 5 we shall characterize and compare the stability properties of the continuous system (15) and of its discrete counterparts arising from simulation and from co-simulations with cross, zigzag, and strong coupling.

First we summarize the known stability results for the two-dimensional system (15). Its $\mathbf{z} = \mathbf{0}$ solution is asymptotically stable if and only if

$$\Gamma = \operatorname{Tr}(\mathbf{A}) < 0, \tag{16}$$

$$\Delta = \text{Det}(\mathbf{A}) > 0. \tag{17}$$

The zero solution is marginally stable if (Γ, Δ) belongs to the stability boundary

$$\{(\Gamma, \Delta) | \Gamma = 0, \Delta > 0 \text{ or } \Gamma < 0, \Delta = 0\}.$$

The region defined by $\Gamma \leq 0$, $\Delta \geq 0$ is the stability region of the continuous system (15) (Figure 3a).



Figure 3: Stability regions of the continuous and discrete systems

The $\mathbf{z} = \mathbf{0}$ solution of the two-dimensional discrete map

$$\mathbf{z}_{n+1} = \mathbf{D}\mathbf{z}_n,\tag{18}$$

is asymptotically stable if and only if ([16], p. 153)

$$|\operatorname{Tr}(\mathbf{D})| - 1 < \operatorname{Det}(\mathbf{D}) < 1.$$
(19)

The zero solution is marginally stable if either of these equalities hold

$$\gamma_1 : \operatorname{Det}(\mathbf{D}) = 1, \tag{20}$$

$$\gamma_2 : \operatorname{Det}(\mathbf{D}) = \operatorname{Tr}(\mathbf{D}) - 1, \tag{21}$$

$$\gamma_3 : \operatorname{Det}(\mathbf{D}) = -\operatorname{Tr}(\mathbf{D}) - 1.$$
(22)

The stability region for the discrete system (18) is shown in Figure 3b.

To compare the stability properties of the continuous system, its simulation and co-simulation, we will express the trace and determinant of the discretized system in terms the trace and determinant of the continuous system to establish stability criteria.

4. Stability Property of the Simulated System

We are now in the position to study the stability of the simulated linear system (15). The coefficient matrix of the continuous system is

$$\mathbf{A} = \left(\begin{array}{cc} a_c & b_c \\ c_c & d_c \end{array}\right).$$

The trace and determinant of \mathbf{A} are given by $\Gamma_c = a_c + d_c$ and $\Delta_c = a_c d_c - b_c c_c$. The state vector is approximated by the 4th-order Runge-Kutta method as $\mathbf{z}_{n+1} = \mathcal{RK}_{Az}(\mathbf{z}_n, t_n, t_{n+\frac{1}{2}}, t_{n+1}) = \mathbf{z}_n + \frac{1}{6}h(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$, with the coefficients $\mathbf{k}_1 = \mathbf{A}\mathbf{z}_n$, $\mathbf{k}_2 = \mathbf{A}(\mathbf{z}_n + \frac{1}{2}h\mathbf{k}_1)$, $\mathbf{k}_3 = \mathbf{A}(\mathbf{z}_n + \frac{1}{2}h\mathbf{k}_2)$, and $\mathbf{k}_4 = \mathbf{A}(\mathbf{z}_n + h\mathbf{k}_3)$. The coefficient matrix of the discrete system (18) is then identified as $\mathbf{D} = \sum_{k=0}^{4} \frac{(\mathbf{A}h)^k}{k!} = \frac{1}{24}(\mathbf{A}^4h^4 + 4\mathbf{A}^3h^3 + 12\mathbf{A}^2h^2 + 24\mathbf{A}h + 24\mathbf{I})$. Notice that \mathbf{D} is a function of $\mathbf{A}h$. Consequently we introduce the scaled variables $a = a_ch, b = b_ch, c = c_ch, d = d_ch$, as well as the scaled trace and determinant $\Gamma = \Gamma_c h, \Delta = \Delta_c h^2$. The case b = c = 0 corresponds to a trivial (uncoupled) system that is not of interest here. Without loss of generality, we assume $c \neq 0$. Utilizing the relations $d = \Gamma - a$ and $b = \frac{a(\Gamma - a) - \Delta}{c}$ we can express the trace and determinant of **D** as

$$\operatorname{Tr}(\mathbf{D}) = \left[\frac{\Gamma^4}{24} - \frac{\Gamma^2 \Delta}{6} + \frac{\Delta^2}{12} + \frac{(\Gamma^3 - 3\Gamma\Delta)}{6} + \frac{(\Gamma^2 - 2\Delta)}{2} + \Gamma + 2\right],$$
$$\operatorname{Det}(\mathbf{D}) = \frac{1}{576} \left[\Delta^4 + 4\Gamma\Delta^3 + 4\left(3\Gamma^2 - 2\Delta\right)\Delta^2 + 24\Gamma\left(\Gamma^2 - \Delta\right)\Delta + 24\Gamma^4 + 96\Gamma^3 + 288\Gamma^2 + 576\Gamma + 576\right].$$

The three stability boundaries (Eqs (20)-(22)) are then given by the implicit analytical expressions in terms of Δ and Γ

$$\gamma_1 : \Delta^4 + 4(\Gamma - 2)\Delta^3 + 12\Gamma(\Gamma - 2)\Delta^2 + 24\Gamma^3\Delta + 24\Gamma = 0,$$
(23)

$$\gamma_{2} : \Delta \left[\Delta^{3} + 4(\Gamma - 2)\Delta^{2} + 12(\Gamma^{2} - 2\Gamma - 4)\Delta + 24 \right] = 0, \tag{24}$$

$$\gamma_{2} : \Delta^{4} + 4(\Gamma - 2)\Delta^{3} + 12(\Gamma^{2} - 2\Gamma + 4)\Delta^{2} + 24(\Gamma^{3} - 4\Gamma^{2} - 12\Gamma - 24)\Delta$$

$$\gamma_3 : \Delta^4 + 4(\Gamma - 2)\Delta^3 + 12(\Gamma^2 - 2\Gamma + 4)\Delta^2 + 24(\Gamma^3 - 4\Gamma^2 - 12\Gamma - 24)\Delta$$
(25)

$$+48\Gamma(\Gamma^{3}+4\Gamma^{2}+12\Gamma+24)+2304=0,$$

The stability boundaries are shown in Fig. 4a and the stability region (satisfying Eq. (19)) is shaded.



Figure 4: Stability chart (a) and types of stability regions: continuous vs. discrete respresentation (b) for the Runge-Kutta simulation

The curve corresponding to γ_1 (Eq. (23)) is the heart-shaped curve. The

stability boundary γ_2 consists of the two lines: $\Delta = 0$ and $\Delta = p\Gamma + q$, where

$$p = \frac{1}{3} \left(-4 - 10 \left(\frac{2}{9\sqrt{29} - 43} \right)^{1/3} + 2^{2/3} \left(9\sqrt{29} - 43 \right)^{1/3} \right) \approx -2.78529,$$
(26)

$$q = \frac{2}{3} \left(4 - 26 \left(\frac{2}{173 - 27\sqrt{29}} \right)^{1/3} - 2^{2/3} \left(173 - 27\sqrt{29} \right)^{1/3} \right) \approx -7.75786,$$
(27)

are the solutions of

$$24 + 12p + 4p^2 + p^3 = 0, (28)$$

$$576 - 48q - 8q^2 + q^3 = 0. (29)$$

The second line intersects with the Δ axis at $\Delta = -\frac{q}{p}$. Only the two pairs $(\Gamma = 0.4389, \Delta = 6.1753)$ and $(\Gamma = -4.4389, \Delta = 7.7729)$ satisfy Eq. (25) for γ_3 . This can be seen by finding the extrema of the left hand side of (25). Elementary, but lengthy calculations (not reproduced here) show that there are three minima, one positive and two zeroes that also belong to γ_3 .

The trace – determinant parameter plane can be decomposed into four fundamental domains according to the behavior of the continuous system vs. that of the simulated system. These domains are shown in Fig. 4b and are denoted by UU (unstable system - unstable simulation), US (unstable system - stable simulation), SU (stable system - unstable simulation), SS (stable system - stable simulation). We found the areas of the stability regions SS and US to be $A_{SS} = 32.5732$, $A_{US} = 2.00072$. Clearly the areas in the original Γ_c , Δ_c variables scale with h^{-3} . Note that $\frac{A_{US}}{A_{SS}-A_{US}} = 0.057$ indicating that 5.7% of the cases identified in the simulation to correspond to stable systems are misleading since in fact the original (continuum) system is unstable.

We note that while in the SS and UU regions the accuracy of the simulation results is controlled by the size of the time step (studying the accuracy of solutions is beyond the scope of this work), in the SU and US regions the numerical results and the exact solutions are *qualitatively different*. In other words this means that in the SU and US cases, simulation introduces a large enough artificial damping (positive or negative) to alter the qualitative nature of the solution. To illustrate via time series representations the misleading na-

ture of the numerical solutions, we show particular instances of such results in Fig. 5. The SU case corresponds to a = 10, b = 3000, c = -2767, d = -110 and the US case to a = 10, b = 3000, c = -2000, d = 290.



Figure 5: Qualitative differences between continuous systems (left) and their discrete representations (right) for instances in the SU region (top) and US region (bottom)

5. Stability Properties of Co-simulation Strategies

In this section we analyse the stability regions of different co-simulation strategies.

5.1. Cross-Coupling

The discrete system describing the co-simulation of the linear system (15) by cross-coupling is given by

$$\mathbf{z}_{n+1} = \mathbf{D}\mathbf{z}_n,\tag{30}$$

$$\mathbf{D} = \begin{pmatrix} 1+a+\frac{a^2}{2}+\frac{a^3}{6}+\frac{a^4}{24} & b\left(1+\frac{1}{2}a+\frac{1}{6}a^2+\frac{1}{24}a^3\right) \\ c\left(1+\frac{1}{2}d+\frac{1}{6}d^2+\frac{1}{24}d^3\right) & 1+d+\frac{d^2}{2}+\frac{d^3}{6}+\frac{d^4}{24} \end{pmatrix}.$$
 (31)

The trace and determinant are

$$\operatorname{Tr}(\mathbf{D}) = \frac{\Gamma^4}{24} + \frac{(1-a)\Gamma^3}{6} + \frac{(a^2 - 2a + 2)\Gamma^2}{4} - \frac{(a^2 + 2a - 2)\Gamma}{2} + \frac{(a^4 + 2a^2 + 24)}{12},$$

$$Det(\mathbf{D}) = \frac{1}{576} \left[24\Gamma^4 + \left(12a(-8+\Delta) + 4a^2\Delta + a^3\Delta + 24(4+\Delta) \right) \Gamma^3 + \left(4a^2(36-5\Delta) - 8a^3\Delta - 3a^4\Delta + 96(3+\Delta) - 24a(12+\Delta) \right) \Gamma^2 + \left(16a^3(\Delta-6) + 4a^4\Delta + 3a^5\Delta + 288(2+\Delta) + (24a^2 - 48a)(12+\Delta) \right) \Gamma - 8a^4(\Delta-6) - a^6\Delta + 576(1+\Delta) + 48a^2(12+\Delta) \right].$$

Let $f(a) = (24 + 12a + 4a^2 + a^3)$. The stability boundary γ_2 is given by $\text{Det}(\mathbf{D}) = \text{Tr}(\mathbf{D}) - 1$ (Eq. (24)), i.e.,

$$f(a)\left(\Gamma^{3} + (4 - 3a)\Gamma^{2} + (12 - 8a + 3a^{2})\Gamma + 24 - 12a + 4a^{2} - a^{3}\right)\Delta = 0.$$
(32)

If $24 + 12a + 4a^2 + a^3 \neq 0$, the boundary γ_2 consists of the two lines

$$\Delta = 0, \tag{33}$$

$$\Gamma = a + p. \tag{34}$$

Let $g(a, \Gamma) = -\Gamma^3 + (-4 + 3a) \Gamma^2 + (-12 + 8a - 3a^2) \Gamma - 24 + 12a - 4a^2 + a^3$. The stability boundary γ_1 is given by

$$f(a)g(a,\Gamma)\Delta = 24\Gamma^4 + 96(1-a)\Gamma^3 + 144\left(2-2a+a^2\right)\Gamma^2 \qquad (35)$$

+ 96\left(6-6a+3a^2-a^3\right)\Gamma+48a^2\left(12+a^2\right),

describing a hyperbola. Since the denominator is the same as the left hand side of (32), the asymptotes are the lines given by (33) and (34). When

a = 0, expression (35) reduces to the line $\Delta = -\Gamma$. The stability boundary γ_3 is a rational function with denominator equal to $f(a)g(a,\Gamma)$. This curve does not influence the region of asymptotic stability.



Figure 6: Stability charts for cross coupling

The stability regions are shown in Figs. 6a-6c for a = -0.05, 0, 0.05, respectively. The stability region of the continuous system is always in the upper left quadrant of the trace-determinant plane. Note however that for negative a the discrete system has regions of stability below the Γ axis. These correspond to cases where the continuous system is unstable and misleading results (US) returned by the discretization algorithm (Figs. 7a-7b). Note that in some cases (e.g., a = -3) the entire stability region of the discretized system is in the unstable domain of the continuous system.

Figure 8 shows the area of the SS stability region as a function of a over the range of a where the area is non-zero and finite. For $a \in (-\infty, -3.1605) \cup (0.548155, \infty)$, $A_{SS} = 0$, and for $a \in (-3.1605, 0)$, $A_{SS} = \infty$.



Figure 7: Stability charts for cross coupling: cases exhibiting inconsistent (US) results



Figure 8: Area of SS for cross-coupling as a function of a.

5.2. Zigzag Coupling

Discretization of the linear system (15) for co-simulation with zigzag coupling results in the discrete system

$$\mathbf{z}_{n+1} = \mathbf{D}\mathbf{z}_n, \quad \mathbf{D} = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}, \tag{36}$$

where

$$d_{11} = 1 + a + \frac{a^2}{2} + \frac{a^3}{6} + \frac{a^4}{24},$$

$$d_{12} = b\left(1 + \frac{1}{2}a + \frac{1}{6}a^2 + \frac{1}{24}a^3\right),$$

$$d_{21} = c\left(\frac{(12 + 4d + d^2)(24a + 12a^2 + 4a^3 + a^4 + 24d)}{576} + 1\right),$$

$$d_{22} = \frac{(24 + 12a + 4a^2 + a^3)bc(12 + 4d + d^2) + 24(24 + 24d + 12d^2 + 4d^3 + d^4)}{576}.$$

The trace and determinant are too cumbersome to display. The stability



Figure 9: Stability charts for zigzag coupling

regions (satisfying (19)) are shown in Figs. 9a-9c for a = -0.5, 0, 0.5, respectively. Similarly to cross-coupling, γ_2 describes the same two lines. The stability region for a = 0.5 is as usual in the upper left quadrant of the tracedeterminant plane (e.g., it is fully in the SS area). However, similar to the simulated system, the zigzag coupling can also lead to stability areas below the Γ axis, i.e., US areas for a = 2 and a = -3 shown in Figs. 10a-10b. The areas of the SS and US regions are represented in Figure 11.

5.3. Strong Coupling

We first establish the conditions for convergence of the iterative sequence (14). By defining

$$\mathbf{r}^{l} = \left(\begin{array}{c} x_{n+1}^{l} \\ y_{n+1}^{l} \end{array}\right),$$



Figure 10: Stability charts for zigzag coupling: cases exhibiting inconsistent (US) results



Figure 11: Areas of stability regions of zigzag coupling

we recast Eq. (14) as the affine map

$$\mathbf{r}^{l+1} = \mathbf{M}\mathbf{r}^{l} + \mathbf{N}\mathbf{r}^{0}, \qquad \mathbf{r}^{0} = \begin{pmatrix} x_{n} \\ y_{n} \end{pmatrix}, \mathbf{M} = \begin{pmatrix} 0 & b\left(\frac{1}{2} + \frac{1}{6}a + \frac{1}{24}a^{2}\right) \\ c\left(\frac{1}{2} + \frac{1}{6}d + \frac{1}{24}d^{2}\right) & 0 \end{pmatrix}, \mathbf{N} = \begin{pmatrix} 1 + a + \frac{1}{2}a^{2} + \frac{1}{6}a^{3} + \frac{1}{24}a^{4} & b\left(\frac{1}{2} + \frac{1}{3}a + \frac{1}{8}a^{2} + \frac{1}{24}a^{3}\right) \\ c\left(\frac{1}{2} + \frac{1}{3}d + \frac{1}{8}d^{2} + \frac{1}{24}d^{3}\right) & 1 + d + \frac{1}{2}d^{2} + \frac{1}{6}d^{3} + \frac{1}{24}d^{4} \end{pmatrix}$$

Solving the recurrence relationship we obtain:

$$\begin{pmatrix} x_{n+1}^{l+1} \\ y_{n+1}^{l+1} \end{pmatrix} = \left[\mathbf{N} \left(\mathbf{I} + \mathbf{M} + \mathbf{M}^2 + \dots + \mathbf{M}^l \right) + \mathbf{M}^{l+1} \right] \begin{pmatrix} x_n \\ y_n \end{pmatrix}.$$
(37)

The sequence of iterations described by Eq. (37) is convergent if matrix **M** is stable (has its eigenvalues inside the unit circle). In this case

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \lim_{l \to \infty} \mathbf{r}^l = \left(\mathbf{I} + \mathbf{M} + \mathbf{M}^2 + \dots \right) \mathbf{N} \mathbf{r}^0 = \underbrace{(\mathbf{I} - \mathbf{M})^{-1} \mathbf{N}}_{\mathbf{D}} \begin{pmatrix} x_n \\ y_n \end{pmatrix}.$$

This discrete map is stable if \mathbf{D} is stable. Therefore, to compute the stability region of the strong coupling algorithm we require that both \mathbf{M} and \mathbf{D} are stable. To enforce these conditions we calculate the trace and determinant of



Figure 12: Stability charts for strong coupling

 \mathbf{M} , $\operatorname{Tr}(\mathbf{M}) = 0$, and $\operatorname{Det}(\mathbf{M}) = -bc\left(\frac{1}{2} + \frac{1}{6}a + \frac{1}{24}a^2\right)\left(\frac{1}{2} + \frac{1}{6}d + \frac{1}{24}d^2\right)$, as well as the trace and determinant of \mathbf{D} (too cumbersome to display) and then we utilize the stability criteria given by Eq. (19). The resulting stability charts are shown in Figs. 12a-12c for a = -0.5, 0, 0.5, respectively. The area of the stability region is effectively zero for $a \in (-\infty, -2.79) \cup (1.75, \infty)$. Unlike the simulation and the cross and zigzag co-simulations, in this case no US regions are observed. Fig. 13 shows the area of the SS stability region as a function of a.



Figure 13: Area of the SS region for strong coupling

5.4. Comparison between methods



Figure 14: Stability charts for strong coupling

The stability regions between the four methods are compared in Figs. 14a-14c for a = -0.3, 0, 0.3. The simulation results (blue, horizontal) do not depend on a and, as discussed before, they exhibit inconsistent results in the form of stable simulation for unstable systems (US). The cross coupling co-simulation (black) also exhibits inconsistent behaviour (US) for a = -0.3 (and in fact for any negative a) and has the smallest SS region of all the methods discussed here. The zig-zag (purple, $3\pi/4$ angle) and strong (green, $\pi/4$ angle) co-simulation results are qualitatively similar: they do not exhibit inconsistent US results and the areas of convergence are approximately the same (and smaller than the area of the simulated results for all a values). Based on these observations we conclude that the zig-zag and strong coupling have the best stability behaviour of all the methods examined here (larger SS areas with no US inconsistencies). Figure 15 shows the area of the SS stability region as a function of a for the four methods.



Figure 15: Area of the SS region for all methods

6. Discussion

Co-simulation, while a very useful computational tool, needs to be performed carefully. Computations of the stability boundaries for various cosimulaton strategies show that even for a simple linear system there are regions of the parameter space where the stability of the co-simulated system is different from that of the continuous system. To characterize these regions, their areas and scaling properties (with the stepsize) have been computed.

Acknowledgement

This material is based upon work partially supported by the National Science Foundation under Grant No. 0846783. We thank Moein Soodavi for valuable discussions. We thank the reviewer for helping us improve the manuscript.

References

[1] L. Euler, Methodus inveniendi lineas curvas maximi minimive proprietas gaudentes, 1744.

- [2] L. A. Godoy, Structural stability concepts in medieval and renaissance mechanics, Latin American Journal of Solids and Structures 8 (2011) 83–105.
- [3] G. J. Simitses, An Introduction to the Elastic Stability of Structures, Prentice-Hall, Inc, Englewood Cliffs, New Jersey, 1976.
- [4] Z. Bazant, L. Cedolin, Stability of Structures, Oxford University Press, 1991.
- [5] H. Ziegler, Principles of Structural Stability, Blaisdell Publishing Company, 1968.
- [6] B. Gu, H. H. Asada, Co-simulation of algebraically coupled dynamic subsystems without disclosure of proprietary subsystem models, Journal of Dynamic Systems, Measurement, and Control 126 (2004) 1–13.
- [7] J. Wang, Z.-D. Ma, G. Hulbert, A Gluing Algorithm for Distributed Simulation of Multibody Systems, Nonlinear Dynamics 34 (2003) 159– 188.
- [8] P. Tomulik, J. Frcazek, Simulation of multibody systems with the use of coupling techniques: A case study, Multibody System Dynamics 25 (2011) 145–165.
- [9] F. Armero, J. Simo, A new unconditionally stable fractional step method for nonlinear coupled thermomechanical problems, International Journal for Numerical Methods in Engineering 35 (1992) 737–766.
- [10] F. Armero, J. Simo, A priori stability estimates and unconditionally stable product formula algorithms for nonlinear coupled thermoplasticity, International Journal of Plasticity (1993).
- [11] K. Garikipati, E. Arruda, K. Grosh, H. Narayanan, S. Calve, A continuum treatment of growth in biological tissue: The coupling of mass transport and mechanics, Journal Of The Mechanics And Physics Of Solids 52 (2004) 1595–1625.
- [12] P. Vijalapura, S. Govindjee, An adaptive hybrid time-stepping scheme for highly non-linear strongly coupled problems, International Journal for Numerical Methods in Engineering 64 (2005) 819–848.

- [13] N. Mahjoubi, A. Gravouil, A. Combescure, N. Greffet, A monolithic energy conserving method to couple heterogeneous time integrators with incompatible time steps in structural dynamics, Computer Methods in Applied Mechanics and Engineering 200 (2011) 1069–1086.
- [14] Y. Lian, W. Shyy, D. Viieru, B. Zhang, Membrane wing aerodynamics for micro air vehicles, Progress in Aerospace Sciences 39 (2003) 425–465.
- [15] M. Trčka, J. Hensen, M. Wetter, Co-simulation for performance prediction of integrated building and HVAC systems - An analysis of solution characteristics using a two-body system, Simulation Modelling Practice and Theory 18 (2010) 957–970.
- [16] J. Thompson, H. Stewart, Nonlinear dynamics and chaos, John Wiley & Sons Inc, 2002.