Applied Mathematics and Computation 227 (2014) 199–211

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Can complex systems really be simulated?

Tamas Kalmar-Nagy [a], Ilinca Stanciulescu [b,*]

[a] Mitsubishi Electric Research Laboratory, Cambridge, MA 02139, USA
[b] Rice University, Department of Civil and Environmental Engineering, Houston, TX 77005, USA

ARTICLE INFO

Keywords:
Stability
Simulation and co-simulation
Stability boundaries

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 behavior 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.

1. Introduction

Many place the origin of the theory of stability in the first theoretical study of unstable structural systems by Euler [1], who analyzed the behavior of a slender column under compressive loading. Others point 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, [3–5]. 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 modeling 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

* Corresponding author.
E-mail address: ilinca.s@rice.edu (I. Stanciulescu).

0096-3003/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2013.11.037

(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 modeling 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 stability* 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-and-play 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 splitting 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. 1(a)). The system is described by the continuous evolution equation

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

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. 1(b)). The equation governing the system can be rewritten as

$$\mathcal{D}\begin{pmatrix}\mathbf{x}(t)\\ \mathbf{y}(t)\end{pmatrix} = \begin{pmatrix}\mathbf{f}_1(\mathbf{x}(t),\mathbf{y}(t))\\ \mathbf{f}_2(\mathbf{x}(t),\mathbf{y}(t))\end{pmatrix}, \tag{2}$$

where $\mathbf{x}(t)$ and $\mathbf{y}(t)$ are the state vectors of 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. 1(c)) in an appropriate manner [14].