15th World Congress on Computational Mechanics (WCCM-XV)
8th Asian Pacific Congress on Computational Mechanics (APCOM-VIII)
Virtual Congress: 31 July – 5 August 2022
S. Koshizuka (Ed.)

A SIMPLE TEST CASE FOR CONVERGENCE ORDER IN TIME AND ENERGY CONSERVATION OF BLACK-BOX COUPLING SCHEMES

VALENTINA SCHÜLLER[1], BENJAMIN RODENBERG[1], BENJAMIN UEKERMANN[2] AND HANS-JOACHIM BUNGARTZ[1]

[1] Department of Informatics, Technical University of Munich
Boltzmannstr. 3, 85748 Garching, Germany
valentina.schueller@tum.de
{benjamin.rodenberg,bungartz}@in.tum.de

[2] Institute for Parallel and Distributed Systems, University of Stuttgart
Universitätsstr. 32, 70569 Stuttgart, Germany
benjamin.uekermann@ipvs.uni-stuttgart.de

Key words: equation coupling, order degradation, Strang splitting, multiphysics, waveform iterations, co-simulation

Abstract. *The most commonly used coupling schemes in partitioned multiphysics simulations suffer from a decrease in the order of convergence, specifically in the time domain; a phenomenon we call order degradation. This paper discusses when this issue arises and how it can be studied with a simple example. We present a simple mass-spring system of ordinary differential equations (ODEs) to analyze accuracy and energy conservation of different coupling schemes. The ability to restore higher order of convergence by using Strang splitting or waveform iterations is verified in the context of the presented example.*

1 INTRODUCTION

Partitioned simulations are broadly used in science and engineering. The partitioned approach tries to split large and complicated systems, such as the ones needed for climate modeling [1] or nuclear fusion [2], into independent subsystems. This approach follows the natural wish to reuse existing software solutions for the individual subsystems. The subsystems then need to be coupled via coupling schemes to solve the overall problem. Coupling schemes, however can introduce "subtle stability and accuracy issues" [3, p. 34].

In this paper we want to closely investigate such issues, specifically reduced convergence order (order degradation) and unphysical loss or increase of energy. We use a simple mass-spring oscillator example (Section 2) to investigate such issues. Our example is a system of ordinary differential equations (ODEs), while multiphysics problems are mostly described by systems of partial differential equations or even differential algebraic equations. This simplification, however, allows us to ignore further technicalities of the partitioned approach, such as the need for data mapping or fixed-point acceleration. Additionally, pure ODE problems allow us to ignore space discretization errors and to fully concentrate on errors originating from time stepping and coupling schemes.

We partition the mass-spring oscillator example, which allows us to consider the solver for each partition as a *black box* – an approach commonly used in software packages for coupling, such as preCICE

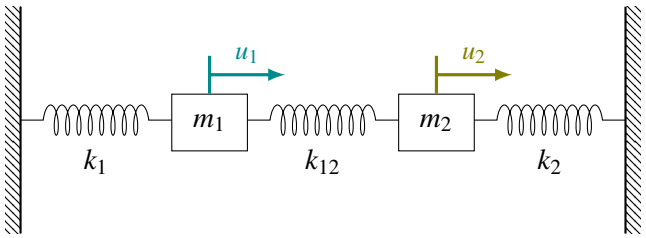


Figure 1: The mass-spring oscillator example analyzed in this paper.

[4]. In Section 3, we then review state-of-the-art black-box coupling schemes for such solvers. To assess the performance of these coupling schemes with respect to convergence order and energy conservation, we apply them to the mass-spring oscillator example in Section 4.

2 OSCILLATOR EXAMPLE

We analyze a simple mass-spring oscillator with two masses connected via three springs. The resulting ODE system is similar to the one introduced in [5, Section 6.1]. The system is visualized in Figure 1 and defined by the following initial value problem:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{pmatrix} \ddot{u}_1 \\ \ddot{u}_2 \end{pmatrix} + \begin{bmatrix} k_1 + k_{12} & -k_{12} \\ -k_{12} & k_2 + k_{12} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = 0 \,,$$

or, in matrix notation,

$$M\ddot{\mathbf{u}} + K\mathbf{u} = 0 \,,$$

with initial conditions for the position and velocity of the masses

$$\mathbf{u}(0) = \mathbf{u_0} \,, \qquad \dot{\mathbf{u}}(0) = \mathbf{v_0} \,.$$

The system is autonomous and linear and the analytical solution can be computed easily, which makes convergence analyses of numerical methods straightforward. The total energy of the system consists of kinetic and potential energy and reads

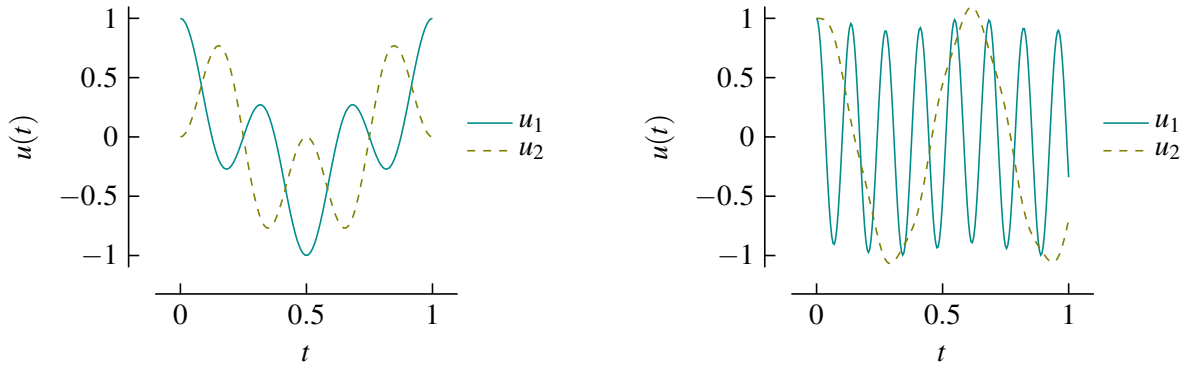
$$E = \frac{1}{2}\dot{\mathbf{u}}^T M\dot{\mathbf{u}} + \frac{1}{2}\mathbf{u}^T K\mathbf{u} \,.$$

We can easily show that

$$\frac{\mathrm{d}E}{\mathrm{d}t} = 0 \,.$$

The mass-spring oscillator can be partitioned by cutting through the connecting spring k_{12}. This results in interface forces and two decoupled initial value problems:

$$\begin{aligned} m_1\ddot{u}_1 &= -(k_1 + k_{12})u_1 + F_2(t) \,, \\ m_2\ddot{u}_2 &= -(k_2 + k_{12})u_2 + F_1(t) \,. \end{aligned}$$



(a) Parameters and initial conditions chosen such that the two oscillations have similar temporal scales: $k_1 = k_2 = 4\pi^2$, $k_{12} = 16\pi^2$, $m_1 = m_2 = 1$, $u_1 = 1$, $u_2 = \dot{u}_1 = \dot{u}_2 = 0$

(b) Parameters and initial conditions chosen such that the two oscillations have different temporal scales: $k_1 = 2000$, $k_2 = 10$, $k_{12} = 100$, $m_1 = m_2 = 1$, $u_1 = u_2 = 1$, $\dot{u}_1 = \dot{u}_2 = 0$

Figure 2: Different analytical solutions of the mass-spring oscillator. By tuning the spring stiffnesses, masses and initial conditions, the solution's properties can be altered to reflect specific effects of partitioned systems.

By adjusting the parameters m_i, k_i and the initial conditions, we can construct examples where the oscillations of the two masses are dominated by different frequencies, see Figure 2. In this paper, we study the case shown in Figure 2(a), a case with period $T = 1$ and the following analytical solution:

$$u_1(t) = \frac{1}{2}\left(\cos\left(2\pi t\right) + \cos\left(6\pi t\right)\right) \quad \text{and} \quad u_2(t) = \frac{1}{2}\left(\cos\left(2\pi t\right) - \cos\left(6\pi t\right)\right).$$

The case shown in Figure 2(b), on the other hand, could be used to study systems where subcycling or multirate time integration is required.

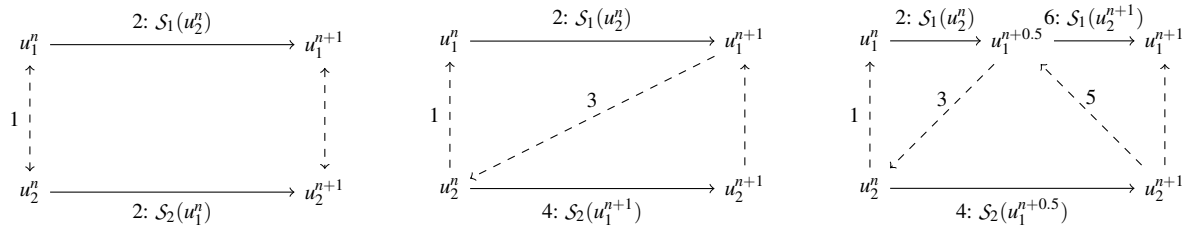
3 COMPONENT PARTITIONING AND COUPLING SCHEMES

Taking inspiration from the description in [3], we describe the oscillator example as a general time-dependent problem $\partial_t u = f(u)$, where $u = u(t)$ is the time-dependent solution and $f(u)$ the right-hand side. We call $\partial_t u = f(u)$ the monolithic system. To partition the monolithic system and obtain simpler subproblems, we assume that we can split f using component partitioning:

$$f(u) = \begin{pmatrix} f_1(u_1, u_2) \\ f_2(u_1, u_2) \end{pmatrix}.$$

The subproblems $\partial_t u_1 = f_1(u_1, u_2)$ and $\partial_t u_2 = f_2(u_1, u_2)$ are then solved separately to produce a solution for the unknown u of the monolithic system. The solvers for the two subproblems need to exchange information at regular intervals since the right-hand sides are functions of both u_1 and u_2. Of course, the total number of partitions depends on the given application, problem, and modeling approach; we chose two without loss of generality and for the sake of simplicity.

In this paper, we use the terms solver and time integration method interchangeably. For general multi-physics problems, the numerical solver is usually not only related to the temporal but also the spatial discretization of the problem at hand. We refer to the solver for each one of the two subproblems $\partial_t u_i = f_i(u_1, u_2)$ as $\mathcal{S}_i$ with $i = 1, 2$. Such a solver $\mathcal{S}_i$ applies a time stepping scheme to advance in



(a) The conventional parallel staggered (CPS) scheme, an explicit first order coupling scheme where all components are updated in parallel.

(b) The conventional serial staggered (CSS) scheme, an explicit first order coupling scheme where the components are updated sequentially using.

(c) Strang splitting, a second-order scheme for a partitioned system, originally designed for additive partitioning.

Figure 3: Three simple explicit coupling schemes for partitioned equations. Dashed arrows represent a communication of the interface values, solid lines represent a step of the subsystem's time integration method. The steps are numbered chronologically. A solver step $u_1^{n+1} = \mathcal{S}_1(\tilde{u}_2, u_1^n)$ is abbreviated as $u_1^{n+1} = \mathcal{S}_1(\tilde{u}_2)$ for more compact notation.

time from t^n to $t^{n+1} = t^n + \Delta t$, computing an approximate solution $u_i^{n+1} \approx u_i\left(t^{n+1}\right)$, given conditions $u_i^n = u_i(t^n)$. The solvers are considered *black boxes* in the sense that we cannot access the discretization or algorithmic details. We write one solver time step of $\mathcal{S}_1$ as

$$u_1^{n+1} = \mathcal{S}_1(\tilde{u}_2, u_1^n) \ .$$

Each solver expects the solution of the other solver as a time-dependent boundary condition. Since the exact solution is usually not available, we use an approximation $\tilde{u}_i \approx u_i$, instead. With these inputs each solver outputs its own approximate solution u_i^{n+1}.

The circular dependency of the two solvers on each other's output leads to a deadlock-like situation when trying to evolve the system in time. We resolve this by applying coupling schemes that tell us which known data we have to use for the approximation $\tilde{u}_i$. In the first part of this section, we introduce relatively simple coupling schemes that use a constant value, for example u_i^n or u_i^{n+1} for the approximation $\tilde{u}_i$. Later, we also see coupling schemes that use intermediate values or an interpolation for $\tilde{u}_i$. Coupling schemes furthermore define the order of solver evaluations, how data is communicated between solvers and, finally, whether solvers may be executed in parallel.

3.1 Simple coupling schemes

The simplest and most common explicit coupling schemes are the *conventional parallel staggered* (CPS) and *conventional serial staggered* (CSS) schemes (Figures 3(a) and 3(b)) [6]. They are both first-order convergent, meaning that even if the solvers $\mathcal{S}_1$ and $\mathcal{S}_2$ use, for instance, second-order time integration methods, the overall solution is still of order $O(\Delta t)$ [7]. This effect is what we refer to as *order degradation*: The coupling scheme is responsible for a drop in the achievable order of convergence; one cannot improve the accuracy significantly without moving to a different way of coupling the partitions.

Both explicit coupling schemes from above can be transformed into implicit coupling schemes by iteratively executing the coupling procedure until convergence. This typically leads to improved numerical stability while still showing order degradation. With each iteration cycle, we get a new iterate $\left(u_i^{n+1}\right)^k$ that we use as approximation $\tilde{u}_i$. Here, $(\cdot)^k$ denotes the k-iteration. The converged solution from the last

time step may be used as an initial guess for $\left(u_i^{n+1}\right)^{k=0}$, but there also exist alternatives, such as extrapolation strategies [8]. To measure convergence we can, for example, compare the results of subsequent iterations in the following way:

$$\left\| \left(u_i^{n+1}\right)^{k+1} - \left(u_i^{n+1}\right)^{k} \right\| \leq \varepsilon.$$

$\varepsilon > 0$ is the chosen tolerance. Since each solver evaluation is costly, one wants to keep the amount of iterations small, essentially reducing the computational cost of the scheme. By replacing the pure fixed-point approach with under-relaxation (e.g., the dynamic Aitken scheme) or quasi-Newton approaches, one can accelerate the convergence of the scheme significantly [8]. A discussion of acceleration techniques for implicit coupling schemes is, however, out of scope in this paper and fortunately we also do not need any acceleration for our relatively simple and computationally cheap example case.

3.2 Strang splitting and waveform iteration

Next, we want to discuss possible improvements of explicit and implicit coupling where the central goal is to improve the overall accuracy by creating higher order explicit coupling schemes. For explicit schemes, a common approach is Strang splitting or the *improved serial staggered* (ISS) scheme, where the two solvers update each other staggered in time by a half-step, to obtain a second order scheme (Figure 3(c)) [9, 6].

To improve the order of implicit schemes in a black-box fashion, one possibility is to use waveform iterations or waveform relaxation for coupling. This means moving away from single-value coupling, where the approximation $\tilde{u}_i$ is given as a specific value in time [10]. Instead, the solvers supply continuous functions of the interface values over the required time interval using interpolation[1]. The order of the overall solution depends on the degree of the interpolant [11]. In this paper we use linear interpolation for $\tilde{u}_i$. Just as for the previously discussed implicit coupling schemes, there are serial and parallel waveform iteration schemes. In this study we focus on the parallel (Jacobi-type) waveform iteration algorithm (Algorithm 1).

Algorithm 1 Parallel Waveform Iterations

$k \leftarrow 0$
Initial guess: Assume $(\tilde{u}_i)^0 = u_i^n$, constant for $t \in \left[t^n, t^{n+1}\right]$
while not converged **do**
  $\left(u_1^{n+1}\right)^{k+1} \leftarrow$ solve timestep $\mathcal{S}_1\left((\tilde{u}_2)^k, (u_1)^k\right)$
  $(\tilde{u}_1)^{k+1} \leftarrow$ interpolate u_1 over $[t^n, t^{n+1}]$
  $\left(u_2^{n+1}\right)^{k+1} \leftarrow$ solve timestep $\mathcal{S}_2\left((\tilde{u}_1)^k, (u_2)^k\right)$
  $(\tilde{u}_2)^{k+1} \leftarrow$ interpolate u_2 over $[t^n, t^{n+1}]$
  $k \leftarrow k+1$
end while

[1] Single-value coupling can also be considered as a special case of waveform iterations with a constant interpolant.

4 NUMERICAL EXPERIMENTS

We start off with an overview of the experimental setup and used software in Subsection 4.1. In Subsection 4.2, Subsection 4.3, and Subsection 4.4, we perform several convergence studies to experimentally determine the order of the time-stepping schemes when different coupling schemes are used. Another interesting measure of quality is energy conservation, which we investigate for the oscillator example in Subsection 4.5.

4.1 Setup

The results were computed with a Python implementation available on GitHub[2]. We show results using five different time integration methods. As examples for simple first-order and second-order methods, we use the semi-implicit Euler method and the implicit midpoint method, respectively. As an example for a higher-order timestepping method, we use the classical fourth-order Runge-Kutta method. These three methods all use the ODE system shown in Section 2 transformed into a system of first-order differential equations. The remaining two methods are specifically designed for structural dynamics problems of the form

$$M\ddot{\mathbf{u}} + C\dot{\mathbf{u}} + K\mathbf{u} = F(\mathbf{u}) :$$

the generalized-α method [12] and the Newmark-β method [13]. We use the following parameters for the Newmark-β method:

$$\beta = \frac{1}{4}, \quad \gamma = \frac{1}{2},$$

and for the generalized-α method:

$$\alpha_m = \frac{1}{5}, \ \beta = \frac{1}{4}(1 - \alpha_m + \alpha_f)^2, \ \alpha_f = \frac{1}{2}, \ \gamma = \frac{1}{2} - \alpha_m + \alpha_f .$$

This choice of parameters with $\alpha_m \leq \alpha_f \leq 0.5$ yields second-order, unconditionally stable numerical methods [12].

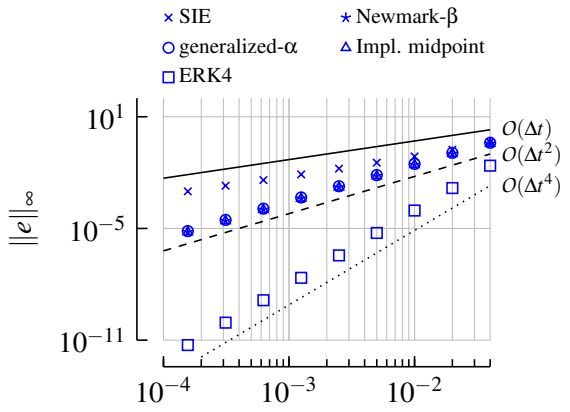
All experiments use a maximum simulation time of $T = 1$. This corresponds to one complete cycle of the oscillator example introduced in Section 2. We measure the error e by comparing displacements u_1, u_2 of the numerical and analytical solution for all discrete time steps. We want to get a worst-case approximation of the error over one complete cycle of the oscillator. Therefore, the overall error of an individual simulation run is estimated by taking the maximum norm $\|e\|_\infty$ over all time steps.

As a first test, we verify the monolithic system using the five aforementioned time integration methods (Figure 4(a)): The results confirm the theoretically achievable convergence orders.

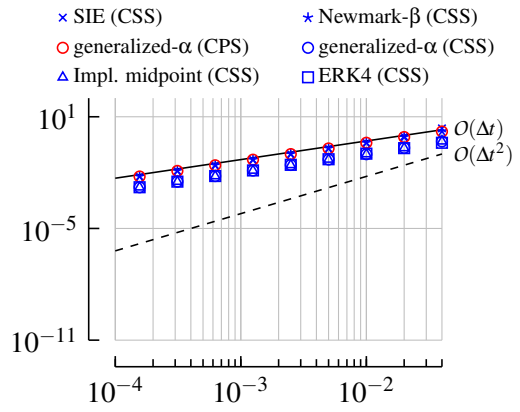
4.2 Order degradation for simple explicit and implicit coupling

We start with an analysis of the conventional parallel staggered (CPS) and conventional serial staggered (CSS) results, shown in Figure 4(b). For their implicit counterparts (Figure 4(c)), we use a convergence tolerance of $\varepsilon = 10^{-6}$, but achieve similar results with 10^{-4} and 10^{-8}. All explicit coupling schemes lead to a degradation of the numerical solution to first-order convergence, with the CSS variants achieving slightly better accuracy than the CPS scheme. This last aspect falls in line with the discussion of Gatzhammer [7, p. 36]. In the implicit case, the CPS and CSS schemes result in the same numerical

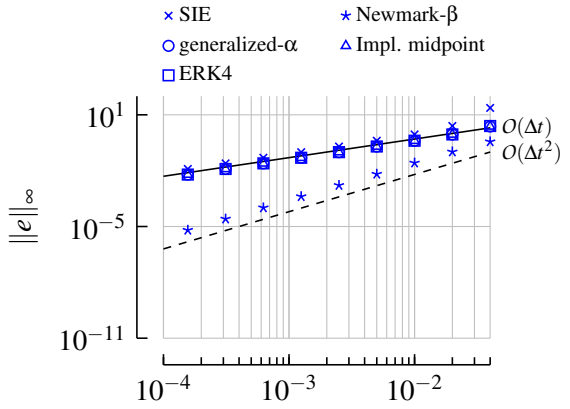
[2] https://github.com/valentinaschueller/oscillator-example/releases/tag/v1.0.0



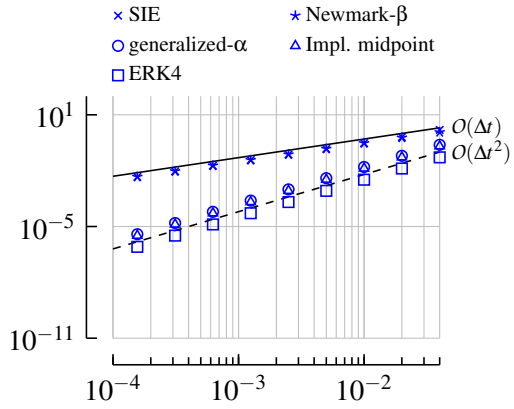
(a) Monolithic system.



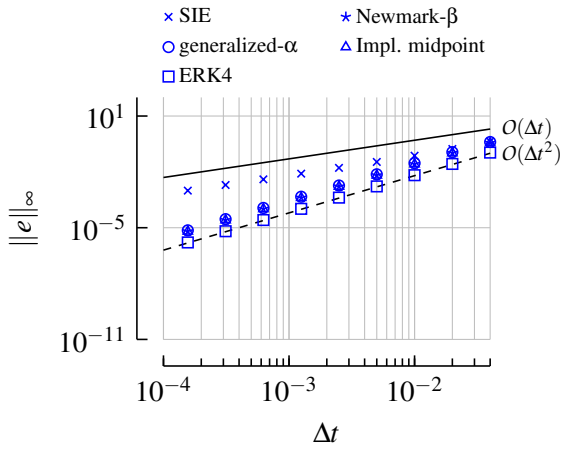
(b) Partitioned system with explicit CSS and CPS coupling schemes.



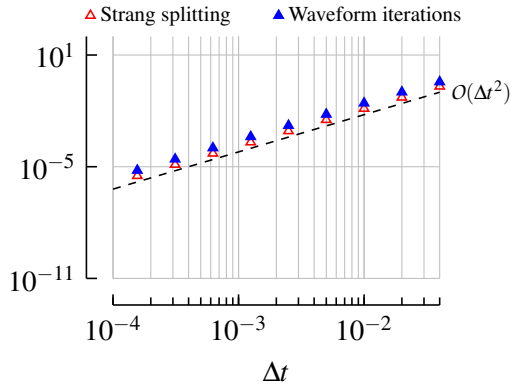
(c) Partitioned system with fixed-point iterations over the CPS solution.



(d) Partitioned system with Strang splitting.



(e) Waveform iterations and linear interpolation.



(f) Comparison of Strang splitting and waveform iteration with linear interpolation for implicit midpoint rule.

Figure 4: Convergence studies for the mass-spring oscillator (cf. Figure 2(a)). SIE is the semi-implicit Euler method, ERK4 the classical Runge-Kutta method.

error, which is why we only show results for CPS. Here, the Newmark-β implementation recovers its second-order convergence, whereas the Runge-Kutta scheme (ERK4 in the plots) and the generalized-α scheme still show the degradation to first-order convergence. The semi-implicit Euler scheme (SIE) keeps its first-order convergence in all configurations. None of the configurations lead to unstable results.

Interestingly, the implicit variants do not per se lead to higher accuracy compared to the explicit schemes. An exception is the behavior of the Newmark-β solution, a remarkable result to which we return in Subsection 4.6.

4.3 Strang splitting: a simple second order coupling scheme

Using Strang splitting, the classical Runge-Kutta scheme and the generalized-α scheme show second-order convergence (Figure 4(d)). The semi-implicit Euler variant stays first-order accurate, but at a slightly lower accuracy than in the monolithic case (Figure 4(a)). Somewhat surprisingly though, the Newmark-β implementation does not show second-order convergence when used with Strang splitting. We discuss a reasoning for this in Subsection 4.6.

4.4 High–order convergence with waveform iterations

Figure 4(e) presents the convergence results obtained with waveform iterations and linear interpolation. With the oscillator example and in the non-multirate setting we investigate in this paper, linear interpolation is sufficient to reach second order, if the time integration schemes in the partitions have sufficiently high order. As the numerical experiments show, the semi-implicit Euler solution is limited by its inherent first-order convergence while all other solvers achieve second-order accuracy.

By linearly interpolating the interface values, both the Newmark-β and generalized-α method recover second-order convergence without the issues encountered previously. We carried out additional experiments which verify that waveform iterations yield second-order convergence also for other values of α_f as long as $\alpha_m \leq \alpha_f \leq 0.5$ (results not shown).

In general, waveform iterations recover the convergence order of the numerical methods more reliably than Strang splitting for our example case. However, when Strang splitting recovers second-order convergence, we see that it produces results of slightly higher accuracy than waveform iterations with linear interpolation (Figure 4(f)). We saw similar results for the generalized-α and the classical Runge-Kutta method (results not shown).

4.5 Energy conservation

As mentioned in Section 2, the analytical solution for the oscillator example has a constant total energy. We now qualitatively study which coupling schemes are well-suited when an emphasis of a (e.g. long-running) simulation is energy conservation. Of the methods implemented in our example code, the semi-implicit Euler method, the implicit midpoint method, and the generalized-α method with the parameter set mentioned in Subsection 4.1 lend themselves for this task: The first two are symplectic methods which generally show good behavior regarding energy conservation [14, Section IX.8]. The family of generalized-α methods has been designed to yield stable numerical methods for systems such as our oscillator example.

We visualize the results in this section using phase plots, an example of which is shown in Figure 5. The plots show the trajectory of m_1 in its position-velocity space (u_1 and $\dot{u}_1$). Since the total energy only depends on the positions, velocities, as well as the mass and stiffness constants, the phase plots contain

all the necessary information to see significant increases or decreases in the total energy. We only show the trajectories for m_1; the numerical results for m_2 are analogous. In all of the phase plots, we show the results after $T = 5s$, i.e., five full periods of the oscillation.

Figure 5 shows the behavior of the analytical solution (dashed line) and a numerical solution of the coupled system which shows bad energy conservation behavior: By using the generalized-α method with the implicit CPS scheme, the trajectory in phase space clearly deviates from the analytical solution; moreover, we see that it moves outwards. This corresponds to an increase in amplitude (and energy) when visualizing u_1 as shown in Figure 2.

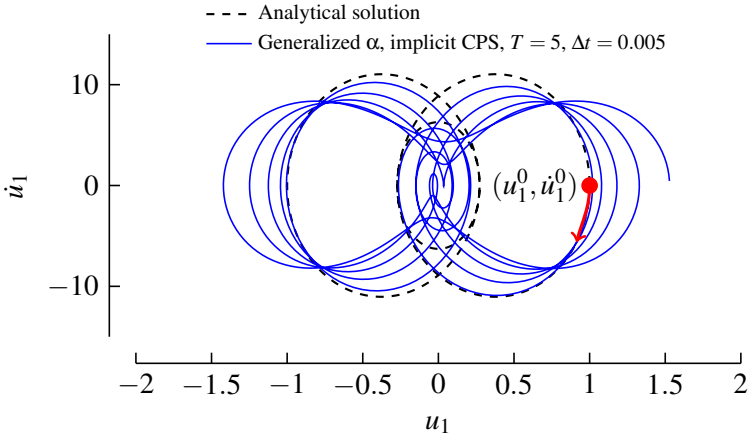


Figure 5: The position and velocity of m_1 in phase space. We plot the analytical solution as a dashed reference and an exemplary numerical result in blue to present the effect of a net energy increase on the trajectory. The red dot and arrow show the initial condition and direction of the trajectory.

Figure 6 presents an overview of all examined coupling schemes. For the simple coupling schemes – CSS, CPS and implicit CPS – there are only three configurations in which we see good energy conservation: the semi-implicit Euler method coupled using the CPS scheme, as well as the generalized-α and implicit midpoint method with the CSS scheme. Of these three examples, the semi-implicit Euler method case has the trajectory which most closely resembles the trajectory of the analytical solution. We note here that none of the simple coupling schemes recovers numerical solutions with good energy conservation for all of the three integration methods, but similarly to the results of the convergence studies, a coupling scheme can sometimes positively interfere with the numerical method. For instance, we get surprisingly good results regarding energy conservation when using the CSS coupling scheme with generalized-α or implicit midpoint time stepping.

Moving on to the more advanced coupling schemes – Strang splitting and waveform iterations – we observe better results: The only trajectory which is significantly different from the analytical solution is the semi-implicit Euler method coupled using Strang splitting. The worse behavior of this specific case falls in line with the higher error we observed in the corresponding convergence study (compared to the monolithic experiment, see Figure 4(d) and Figure 4(a)). Moreover, Strang splitting yields slightly "sharper" trajectories (with less deviations across the periods) when looking very closely at the data. However, only waveform iterations consistently manage to restore the energy conservation properties for all used time integration methods.

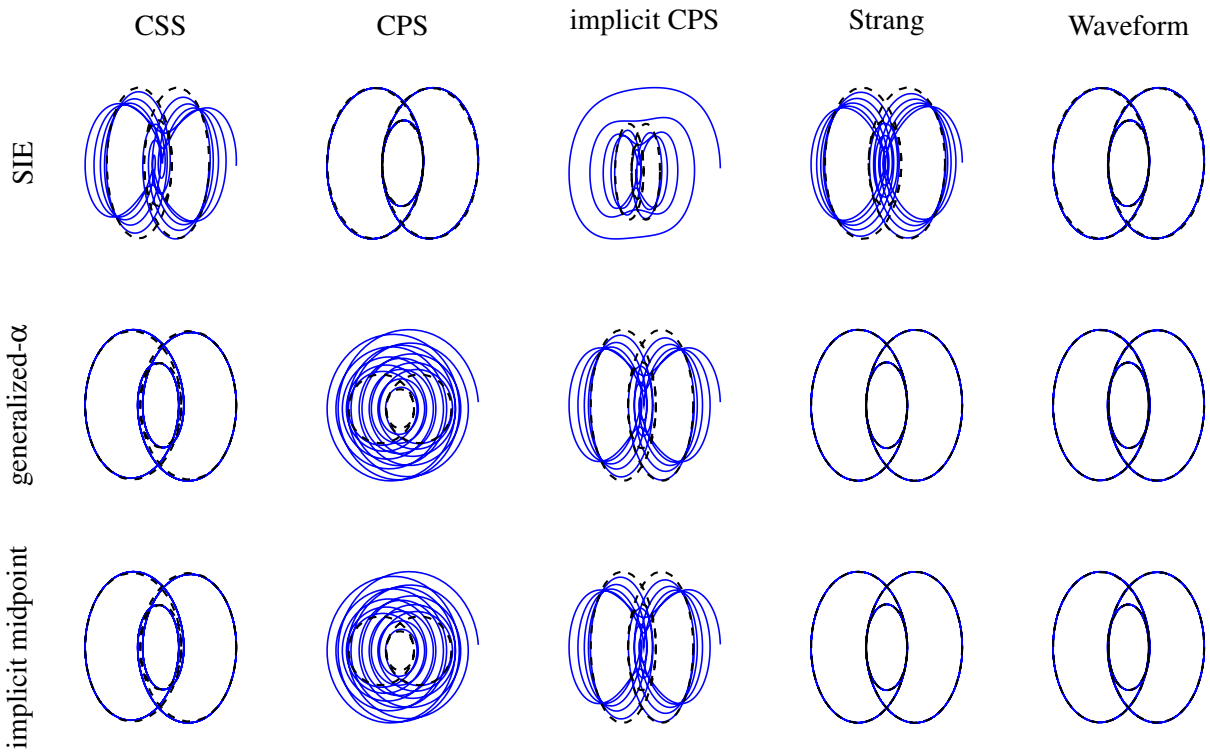


Figure 6: The position and velocity of m_1 in phase space for the simple coupling schemes (CSS, CPS, and implicit CPS) and high order coupling schemes (Strang splitting and waveform iteration). The dashed trajectory from Figure 5 is provided as a reference.

Comparing these results to the results of Figure 4, we can say that order degradation and loss of energy conservation do not necessarily go hand in hand. For example, the implicit midpoint method coupled with the CSS scheme yields only first-order convergence, but is still able to conserve energy.

4.6 The behavior of generalized-α and Newmark-β

We see peculiar behavior for the generalized-α and the Newmark-β method in Subsections 4.2 and 4.3. It is therefore worth having a closer look. We begin with the second-order convergence of the Newmark-β method combined with the implicit CPS scheme (Figure 4(c)):

Both the generalized-α scheme and its special case, the Newmark method, linearly interpolate the interface force $F_i = k_{12}u_i$ in the interval $[t^n, t^{n+1}]$. This interpolation is controlled by the parameter α_f:

$$F(t^{n+1-\alpha_f}) = \alpha_f F(t^n) + (1-\alpha_f)F(t^{n+1}) \,.$$

We chose $\alpha_f = 0.5$ for the generalized-α scheme. The interface force should thus be evaluated in the middle of the time step to achieve the desired convergence. In the Newmark-β method, $\alpha_f = 0$, which means that the force should be evaluated at $t = t^{n+1}$ to reach second-order convergence.

Repeated fixed-point iterations over the CPS scheme mean that the interface values used to compute F_i

approach the value of u_i at time t^{n+1} (cf. Figure 3). The convergence conditions are thus fulfilled for the Newmark method and implicit coupling, but not with the simple explicit schemes (Figure 4(b)). On the other hand, the generalized-α solutions consistently use the wrong value of u_i to compute the interface force, reducing the order of convergence to $O(\Delta t)$.

Turning to Strang splitting (Figure 4(d)), we can use a similar reasoning: The interface value used by solver $\mathcal{S}_2$ in Figure 3(c) is evaluated in the middle of the time step $[t^n, t^{n+1}]$. For its second-order convergence, however, the Newmark scheme requires the interface force to be given at t^{n+1}. With $\alpha_f = 0.5$ in the generalized-α scheme, this is no issue for $\mathcal{S}_2$. $\mathcal{S}_1$ gets the interface values at the boundaries of the smaller time interval, which perhaps makes it seem like the overall order should degrade to $O(\Delta t)$ again. The apparent contradiction is resolved by viewing the two half-steps of $\mathcal{S}_1$ in the interval $[t + \frac{\Delta t}{2}, t + \frac{3\Delta t}{2}]$ as one full step, essentially making Strang splitting a variant of the CSS scheme where one solver is shifted by $\frac{\Delta t}{2}$. With this, we can use the same argument for $\mathcal{S}_1$ as for $\mathcal{S}_2$ and thus explain the second-order convergence of the generalized-α method with the given parameters. We verified this line of thought by using other values for α_f without modifying α_m or violating the stability conditions. With $\alpha_m \leq \alpha_f < 0.5$, the generalized-$\alpha$ method only achieves first-order convergence with Strang splitting (results not shown).

5 CONCLUSION

The mass-spring oscillator example allows us to observe order degradation and loss of energy in an illustrative, simple, yet versatile way. This makes the mass-spring oscillator a suitable test case to closely investigate the interaction of time stepping and (black-box) coupling schemes. It is important to note that our example considers only a few aspects of the overall coupling, while for a multiphysics simulation one also has to consider a more complex physical model and the effect of additional components in the toolchain.

The numerical results of Section 4 show pitfalls of the classical conventional parallel and serial staggered coupling schemes. The generalized-α and the Newmark-β time-stepping methods show, however, that simple coupling schemes can yield higher order under certain conditions. In particular, this shows that interactions between coupling schemes and time-stepping methods can become complicated and that simple coupling schemes are not generally reliable.

The higher-order Strang splitting shows good performance in certain situations, but can also interfere with the time integration method in other cases. For the semi-implicit Euler method, for instance, the accuracy of the method is not fully restored although the coupling scheme has a sufficiently high convergence order in theory. Waveform iterations, on the other hand, are a reliable coupling scheme to recover second-order convergence and to conserve total energy for the mass-spring oscillator.

Waveform iterations also lend themselves to simulations employing subcycling or multirate time stepping – more than Strang splitting – as waveform iterations do not require an exact halving of the time step size. We, therefore, suggest as next step to have a closer look at waveform iterations combined with subcycling and different time scales. Fortunately, the mass-spring oscillator also allows studying such scenarios (Figure 2(b)). The test case even enables easy construction of higher-order interpolation using velocity $\dot{u}$ or acceleration values $\ddot{u}$.

Acknowledgements: This paper originated as part of the seminar "Partitioned Fluid-Structure Interaction and Multiphysics Simulations" at the Technical University of Munich. We want to thank everyone involved with the seminar for the resources and feedback. The numerical experiments were carried out using Python 3.8 and the li-

braries NumPy, SymPy, and Pandas. Thank you to the many developers involved in providing these tools alongside excellent documentation.

B.U.'s research was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016. We further acknowledge the support by the Stuttgart Center for Simulation Science (SimTech).

REFERENCES

[1] Gross, M., et al. Physics-Dynamics Coupling in Weather, Climate, and Earth System Models: Challenges and Recent Progress. *Mon. Weather Rev.* (2018) **146(11)**:3505-3544. DOI: 10.1175/MWR-D-17-0345.1

[2] Choi, J.Y., et al. Coupling Exascale Multiphysics Applications: Methods and Lessons Learned. *2018 IEEE 14th International Conference on e-Science (e-Science)* (2018) 442-452. DOI: 10.1109/eScience.2018.00133

[3] Keyes, D.E., et al. Multiphysics simulations: Challenges and opportunities. *Int. J. High Perform. C.* (2013) **27(1)**:4-83. DOI: 10.1177/1094342012468181

[4] Chourdakis, G., et al. preCICE v2: A sustainable and user-friendly coupling library [version 1; peer review: awaiting peer review]. *Open Res. Europe* (2022) **2**:51 DOI: 10.12688/openreseurope.14445.1

[5] Prakash, A. and Hjelmstad, K.D. A FETI-based multi-time-step coupling method for Newmark schemes in structural dynamics. *Int. J. Numer. Meth. Engng.* (2004) **61**:2183-2204. DOI: 10.1002/nme.1136

[6] Farhat, C. and Lesoinne, M. Higher-order staggered and subiteration free algorithms for coupled dynamic aeroelasticity problems. *36th AIAA Aerospace Sciences Meeting and Exhibit* (1998) DOI: 10.2514/6.1998-516

[7] Gatzhammer, B. Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions. Dissertation. Technical University of Munich (2014)

[8] Degroote, J., Bathe, K.J. and Vierendeels J. Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Comput. Struct.* (2009) **87(11-12)**:793-801. DOI: 10.1016/j.compstruc.2008.11.013

[9] Strang, G. On the Construction and Comparison of Difference Schemes. *SIAM J. Numer. Anal.* (1968) **5(3)**:506-517. DOI: 10.1137/0705041

[10] Gander M.J. Waveform Relaxation. *Engquist B. (eds) Encyclopedia of Applied and Computational Mathematics.* (2015) DOI: 10.1007/978-3-540-70529-1

[11] Rüth, B., et al. Quasi-Newton waveform iteration for partitioned surface-coupled multiphysics applications. *Int. J. Numer. Methods Eng.* (2021) **122**:5236-5257. DOI: 10.1002/nme.6443

[12] Chung, J. and Hulbert, G.M. A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: The Generalized-α Method. *ASME. J. Appl. Mech.* (1993) **60(2)**:371–375. DOI: 10.1115/1.2900803

[13] Newmark, N.M. A method of computation for structural dynamics. *J. Eng. Mech. Div.-ASCE* (1959) **85(3)**:67-94. DOI: 10.1061/JMCEA3.0000098

[14] Hairer, E., Hochbruck, M., Iserles, A. and Lubich, C. Geometric numerical integration. *Oberwolfach Reports* (2006) **3(1)**:805-882. DOI: 10.4171/owr/2006/14