MULTISTEP INTERFACE COUPLING FOR HIGH-ORDER ADAPTIVE BLACK-BOX MULTIPHYSICS SIMULATIONS

L. François*, M. Massot†

*ONERA - The French Aerospace Lab
6 Chemin de la Vauve aux Granges
91123, Palaiseau, France
e-mail: laurent.francois@onera.fr

†CMAP/CNRS - École Polytechnique
Institut Polytechnique de Paris, Route de Saclay
91120 Palaiseau Cedex, France
e-mail: marc.massot@polytechnique.edu

Key words: Code coupling, Adaptivity, Implicit, Conjugate Heat Transfer, Stability

Abstract. Many multiphysics problem can be described by the coupling of several models through physical surfaces. Relying on existing model-specific solvers is very desirable, however they must be coupled in a way that ensures an accurate and stable coupled simulation. In this contribution, we present a multistep coupling scheme which relies on the history of the exchanged quantities to enable a high-order accurate coupling with time adaptation. Explicit and implicit variants are discussed in details. Numerical experiments conducted with an open-source demonstrator on a conjugate heat transfer problem show that high-order convergence is attained, and that stability is favourable compared to other classical approaches.

1 INTRODUCTION

Multiphysics problems often arise in engineering and involve various models, corresponding to different physical phenomena (e.g. solid mechanics, fluid flow) and time scales. Let us consider the case were each model is a set of evolutive partial differential equations (PDEs) whose solution depends on time and space. Often, each model is defined on its own specific spatial domain, and the models are only coupled through lower-dimensional interfaces (e.g. solid-gas contact surface), as opposed to volume-coupled models, whose domains may overlap. Implementing a solver that handles all models in a monolithic manner is a complex and time-consuming endeavour, which may suboptimal and plagued with numerical difficulties due to the multiscale character of the coupled equations. Usually, each of these models is well-known and has been implemented in one or multiple solvers, either open-source or commercial. These physics-specific solvers often rely on numerical methods especially suited to the problem considered and have been extensively validated. It is therefore very attractive to find a way to couple these solvers so as to a realise a fully-coupled simulation of the multiphysics problem of interest. This however requires that the solvers be adequately coupled to correctly capture the multiphysical and multiscale interactions. In particular, so-called *coupling variables* must be exchanged (e.g. heat fluxes) in a manner such that the coupling is accurate in time and stable.

Let us consider a formulation representative of most coupled problems. We assume that we possess N coupled PDE models. Let us denote as z_i the state vector associated with the semi-discretisation in space of the i-th subsystem. This vector is subject to the following ODE:

$$\partial_t z_i \;=\; f_i(z_i, u_i) \qquad i = 1 \ldots N \tag{1}$$

where u_i denotes the input of the subsystem (e.g. boundary conditions), and t is the physical time. We denote as coupling variables the gathering of all the inputs u_i. In its most general form, u_i is an arbitrary function of time and of all the state vectors $z_1, \ldots, z_N$:

$$u_{i,n} \;=\; h_i(z_1, \ldots, z_N) \qquad i = 1 \ldots N \tag{2}$$

It is instructive to conduct a short overview of the various methods presented in the literature for this kind of problem. The simplest strategy, ubiquitous in the literature, is to use existing solvers for each subsystem, with the coupling variables being exchanged at each coupling time step. The coupling time step can be adapted based on heuristic criteria, e.g. CFL-condition obtained from one subsystem. Although the approach allows for dynamic substepping of the subsystems, the exchanged values are frozen in time during the temporal integration of each subsystem, therefore the overall order of accuracy cannot exceed one. Also, the explicit nature of the exchange can hinder the stability of the overall computation.

One other strategy is to solve the problem in a monolithic manner by applying a single time integration scheme for all the solvers. Explicit algorithms are simple but often inadequate due to stability constraints. Implicit variants can be obtained by forming and solving an overall implicit system to be solved involving all subsystem state vectors. This approach is however highly complex, intrusive, and the associated linear system may be poorly conditioned [1, 2] because of both the multiscale character and the potentially different discretisations. Alternatively, the monolithic solution can be obtained without forming the fully-implicit global system, but by iteratively solving a fixed-point problem on the coupling variables, each iteration requiring the solution of each subsystem computed in a partitioned manner [3, 4]. This may however suffer from convergence issues. In both cases, traditional time step selection strategies based on embedded error estimates [5] may be used. However, the monolithic implicit approach makes it difficult to use different time schemes for each model, and substepping is not possible, which may limit the computational efficiency. A variant of this strategy relies on implicit-explicit (IMEX) schemes [6]: each subsystem is solved implicitly with a common implicit scheme, but their interactions via the coupling variables are integrated with an explicit companion scheme. This approach may however suffer from instability when the coupling is strong [4].

A second strategy to compute the coupled solution is based on the idea of waveform relaxation [7]. The exchanged coupling variables are not considered to be frozen in time over one coupling time step, instead they are functions of time, usually spline or polynomial interpolants of their values obtained at various substeps. This approach enables a higher-order convergence [8]. Some methods [9] allow for a dynamic adaptation of the number of interpolation points over a coupling time step, however dynamic selection of the coupling time step itself is not considered. Note that most of these strategies are implemented for cases where u_i only depends on a single state vector $z_j, j \neq i$. This is for instance the case for fluid-structure interactions, where the pressure imposed on the mechanical structure is computed based on the fluid pressure field only and, conversely, the displacements imposed on the fluid side is determined solely from the structure state. The simpler formulation of the input as $u_i = g_i(t, z_j)$ may however not be possible to obtain in all cases, e.g. for some specific conjugate heat transfer formulations [10, Chap. 8]. Finally, such waveform relaxation approaches are by essence implicit and iterative, i.e. no explicit variants exist. In cases where the coupling does not require the inherent stability of this implicit scheme, being forced to use such an iterative process may be suboptimal in terms of computational efficiency.

Another strategy that enables the use of separate solvers for each subsystem with substepping is that of splitting [11]. The first-order Lie splitting is often referred to as conventional serial staggered scheme (CSS), while the second-order Strang splitting is referred to as improved serial staggered scheme (ISS) [12]. This class of methods is simple to implement, yet it cannot go beyond second-order accuracy without the use of complex or negative time steps within the splitting scheme. Also, dynamic adaptation of the splitting time step is difficult [13].

Finally, a last strategy has been introduced in the last two decades for the simulation of coupled multibody mechanical systems [14, 15]. Based on the values of the coupling variables over a few previous steps, polynomial interpolants are built and used as predictions (extrapolations) of the input of each subsystem, which will be used in place of the true inputs during the integration of each subsystem. High-order convergence has been demonstrated, as well as the ability to dynamically adapt the coupling time step [16]. Implicit variants also exist, which greatly improve the stability of the coupled computation [15]. The name *co-simulation* is sometimes used for this approach. It is however slightly misleading, since this term also refers more generally to the process of coupling existing codes. This strategy has however been almost exclusively used on mechanical systems, assuming a different formulation of the coupling variables compared to Equation (2), which is not always suited to the framework of coupled partial differential equations.

We may list a few desirable features of an ideal coupling strategy. It should use existing solvers for each subsystem, requiring as few changes as possible to be introduced in these solvers. It should be able to perform an explicit coupling, or an efficient implicit coupling if stability demands it. To cope with changing physical time scales and guarantee both stability and accuracy, the coupling time step should be dynamically adapted. Finally, low to high orders of convergence in time should be attainable to meet the various accuracy requirements imposed by the user.

We believe that the last approach is the most suited to meet all these requirements. Therefore in this work, we present a generic coupling strategy inspired by this last approach. We do not focus on mechanical systems, but rather demonstrate its applicability on PDE-related applications, following works initiated in [10, Chap. 8]. We refer to this approach as *multistep coupling* due to its similarity with the well-known class of multistep time integration schemes. We first describe the general framework and give some theoretical results. We then present the Rhapsopy Python package [17], an open-source framework to demonstrate the applicability of this method. The proposed coupling strategy is tested on a conjugate heat transfer problem. High-order convergence in time is confirmed. Stability limits are investigated numerically and compared with those of other approaches (e.g. IMEX), showing the potential of the method.

2 FORMULATION AND ANALYSIS

In this section, we introduce the coupled strategy for the system (1)-(2) and provide an overview of its properties.

2.1 Explicit algorithm

The basic explicit coupling scheme for that system reads:

$$\begin{aligned} z_{i,n+1} &= z_{i,n} + \Delta t\ \Phi_i(z_{i,n}, u_{i,n}; \Delta t) \qquad i = 1 \dots N & (3)\\ u_i &= h_i(z_{1,n}, \dots, z_{N,n}) & (4) \end{aligned}$$

where Φ_i denotes the time integrator of the i-th subsystem. Even if the latter is of high order, the overall coupled result is only first-order accurate due to the frozen values of u_i. A natural idea is to replace these values by functions of time $\hat{u}_i$, referred to as *predictions*, which more accurately reflect the evolution of the coupling variables. Since the previous scheme is equivalent to assuming a constant extrapolation in time of the coupling variables over one coupling step, a way to accomplish this is to build higher-degree polynomial extrapolations based on values from past coupling times $t_{n-j}, j \geq 0$. Using a Newton-type formulation, this reads:

$$\hat{u}_i(t) = u_{i,n} + \sum_{j=1}^{p} \delta^j u_i[t_n, .., t_{n-j}] \prod_{k=0}^{j-1} (t - t_{n-k}) \tag{5}$$

where $\delta^j u_i$ denotes the j-th divided difference of u_i, defined by the following recurrence formula:

$$\delta^j u_i[t_n, .., t_{n-j}] = \frac{\delta^{j-1} u_i[t_{n-1}, .., t_{n-j}] - \delta^{j-1} u_i[t_n, .., t_{n-j+1}]}{t_{n-j} - t_n}, \text{ with } \quad \delta^0 u_i[t_n] = u_{i,n} \tag{6}$$

2.2 Implicit algorithm

The previous algorithm is explicit in nature, even if the subsystems are integrated implicitly, since the coupling variables are extrapolations based on past data points. This may hinder the stability of the overall coupled computation. A remedy consists in finding a way to let the future value u_i^{n+1} intervene in the definition of $\hat{u}_i$.

2.2.1 Predictor-corrector approach

For a given coupling step, a first pragmatic strategy is to perform the explicit coupling once to obtain new subsystems state z_i^{n+1}. From these, the inputs u_i^{n+1} can be evaluated with the functions h_i. Interpolations of the previous coupling values and of these newly obtained ones can be formed. The integration of the subsystems may be performed once again with these interpolations as inputs. The whole process can be repeated until convergence. Let us introduce $U = (u_1, \dots, u_N)^t$ and $Y = (z_1, \dots, z_N)^t$. Figure 1 depicts the procedure. Note that, at the end of the first iteration, the sampling points for the polynomial approximations shift from $t_n \dots t_{n-p}$ to $t_{n+1} \dots t_{n-p+1}$ to maintain the same polynomial degree. We can associate the operators F to the subsystems integration, H to the coupling variables update, and Ψ to the construction of the interpolants. The $(k+1)$-th iteration of the complete procedure can be formulated in terms of the coupling variables at time t_{n+1}:

$$U_{n+1}^{k+1} = (\Psi \circ H \circ F)(U_{n+1}^{k}) \tag{7}$$

which is a simple fixed-point algorithm applied to the problem:

$$G(U_{n+1}) \equiv U_{n+1} - (\Psi \circ H \circ F)(U_{n+1}) = 0 \tag{8}$$

2.2.2 Convergence of the fixed-point problem

It can be shown that the operator $(\Psi \circ H \circ F)$ has a Lipschitz constant which is proportional to Δt, due to the temporal integration that F represents. Hence, it is a contraction only if Δt is sufficiently small. Conversely, the convergence of the simple fixed-point algorithm (7) usually limits the attainable time step to a value which is close to the maximum stable time step allowed for the explicit coupling, i.e. it does not provide any additional stability.

Dynamic relaxation methods can increase the convergence radius. To improve further, a Newton method can be applied directly to Equation (8). There has been a growing interest in Newton-type approaches during the last decade for the solution of such systems arising from strongly-coupled simulations. In particular, interface quasi-Newton (IQN) methods, which dynamically construct an approximation of $\partial_U G$, have proved valuable for large-scale problems [18, 19]. For the low-dimensional test case presented in this paper, we use a simple damped Newton method, where the Jacobian of G is directly approximated via finite-differences.

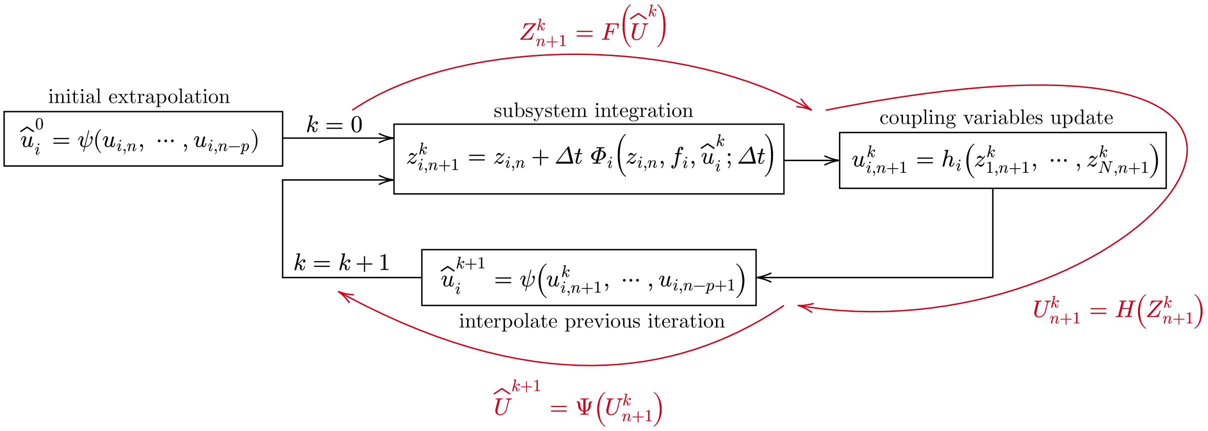
$Z^k_{n+1} = F\left(\widehat{U}^k\right)$
initial extrapolation
$\widehat{u}_i^0 = \psi(u_{i,n}, \cdots, u_{i,n-p})$
$k = 0$
subsystem integration
$z^k_{i,n+1} = z_{i,n} + \Delta t\ \Phi_i\left(z_{i,n}, f_i, \widehat{u}_i^k; \Delta t\right)$
coupling variables update
$u^k_{i,n+1} = h_i\left(z^k_{1,n+1}, \cdots, z^k_{N,n+1}\right)$
$k = k+1$
$\widehat{u}_i^{k+1} = \psi\left(u^k_{i,n+1}, \cdots, u_{i,n-p+1}\right)$
interpolate previous iteration
$U^k_{n+1} = H\left(Z^k_{n+1}\right)$
$\widehat{U}^{k+1} = \Psi\left(U^k_{n+1}\right)$

Figure 1: Fixed-point procedure for the implicit coupling

2.3 Local truncation error

A simplified analysis can show the dependence of the local error on the degree of the input approximation. We assume that the integration of the subsystems generates a temporal error negligible when compared to that induced by the approximation of the coupling variables, i.e. they closely match the exact integration result:

$$z_{i,n+1} = z_{i,n} + \int_{t_n}^{t_{n+1}} f_i\left(z_i(t), \hat{u}_i(t), t\right) dt \tag{9}$$

We assume that the functions f_i are Lipschitz-continuous with respect to z_i and u_i:

$$||f_i(z_a, u_a, t) - f_i(z_b, u_b, t)|| \leq L_{i,z}||z_a - z_b|| + L_{i,u}||u_a - u_b|| \tag{10}$$

with $L_{i,z}$ and $L_{i,u}$ positive real constants, and the norm $||\cdot||$ is any suitable norm, e.g. L_∞-norm. Now, let us denote the exact solutions of the coupled problems as $(z_i(t), u_i(t))$, and the coupled simulation result as $(\tilde{z}_i(t), \tilde{u_i}(t))$. We consider the time step from time t_n to $t_{n+1} = t_n + \Delta t$, with initial conditions $\tilde{z}_i(t_n) = z_i(t_n)$. We can apply Grönwall's lemma and obtain:

$$||\tilde{z}_i(t_{n+1}) - z_i(t_{n+1})|| \leq \frac{L_{i,u}}{L_{i,z}} \left(\exp(L_{i,z}\Delta t) - 1\right) \max_{t\in[t_n,t_{n+1}]} ||\hat{u}_i(t) - u_i(t)|| \tag{11}$$

Assuming $\hat{u}_i$ is a polynomial approximation on $[t_n, t_{n+1}]$ of u_i with degree p_i, i.e. using $p_i + 1$ points, the last term is $\mathcal{O}(\Delta t^{p_i+1})$. Taking the first-order Taylor expansion of the exponential, we obtain the following upper bound:

$$||\hat{z}_i(t_{n+1}) - z_i(t_{n+1})|| = \mathcal{O}(\Delta t^{p_i+2}) \tag{12}$$

The coupling variables u_i are updated via Equation (2) at time t_{n+1}, the errors of each subsystem are involved. Thus, the error on the new coupling variables $u_{i,n+1}$ satisfies:

$$||u_{i,n+1} - u_i(t_{n+1})|| = \mathcal{O}(\Delta t^q) \quad \text{with } q = \min_j(p_j + 2) \tag{13}$$

Therefore, the local error of the coupled solution is of order q, and the global error is of order $q - 1$, equal to the (minimum) number of points used to construct the polynomials. Some refined analysis for the case of mechanical systems are available [15, 16, 20].

The previous proof is valid for both explicit and implicit simulations. However, it is worth noting that the overall error constant is not the same. Indeed, it is known that the polynomial approximation error is proportional to $\prod_{i=0}^{N}(t - t_i)$ with $t_0, \ldots, t_N$ the sampling times. Let us consider the L$_1$-norm of the approximation error on the current coupling time step: $(1/\Delta t) \int_{t_n}^{t_{n+1}} |u(t) - \hat{u}(t)| dt$. In the interpolation case the error is bounded, whereas in the extrapolation case the error tends to diverge. Comparing this norm for extrapolation (explicit coupling) and interpolation (implicit coupling) shows that the implicit error constant is approximately $1 + 4.5p$ times lower than the explicit one, with p the degree of the approximation polynomials, e.g. for degrees $p = 0$ to 5, the implicit coupling is more precise by a factor 1, 5, 9, 13.2, 17.6, 22.1, respectively. This trend will be observed numerically in Section 3.1. This assumes a constant coupling step size. The situation is even more favourable to implicit coupling when the time step is gradually increased between the sampling points. Note that the exact same ratios are found when comparing Adams-Bashforth and Adams-Moulton methods, which is the motivation behind well-known predictor-corrector schemes [21].

2.4 Adaptation of the coupling time step

Typical coupled simulations are performed with a constant time step, or a time step which is driven by one of the solver, e.g. based on a fluid model CFL-condition. For the integration of ODEs, it is well known that dynamic adaptation of the time step provides substantial gains in computational time and robustness [5]. To the best of our knowledge, very few studies have focused on providing time adaptation capabilities in multi-physics coupling schemes, which may be partially explained by the fact that one solver typically dominates the computational time (typically a fluid flow solver), whereas other solvers and the coupling algorithm itself have a negligible cost. In that case, the costly solver usually dictates the coupling time step as a multiple of its own internal time step, and performing an unnecessarily high number of coupling time steps has a negligible cost overall.

There may however be cases where each solver can be costly and have sufficiently different time scales. Then, to avoid an excessive number of solver calls and synchronisations, we should try to maximise the coupling time step as long as a certain coupling accuracy requirement is met. Some research [15, 16] has been presented in the co-simulation literature for mechanical systems. In this section, we introduce two generic error estimates that can be used to drive the coupling time step: a natural explicit error estimate, and a second estimate inspired by embedded methods [5] which is especially suited to implicit multistep coupling.

2.4.1 Natural explicit error estimate

A pragmatic error estimate is built by assessing how well the updated coupling variables $u_{i,n+1}$ at time t_{n+1} have been predicted. Let us apply the triangular inequality on the prediction error with respect to the exact solution:

$$||\hat{u}_i(t_{n+1}) - u_i(t_{n+1})|| = ||\hat{u}_i(t_{n+1}) - (u_{i,n+1} + \mathcal{O}(\Delta t^q))|| \leq \underbrace{||\hat{u}_i(t_{n+1}) - u_{i,n+1}||}_{\epsilon^u_{i,n+1}} + \mathcal{O}(\Delta t^q) \quad (14)$$

For simplicity, let us assume $p_i = p \ \forall i \in [1, N]$. Then, $q = p + 2$ and $||\hat{u}_i(t_{n+1}) - u_i(t_{n+1})||$ is of the form $\mathcal{O}(\Delta t^{p+1})$, thus we see that $\epsilon^u_{i,n+1}$ will converge towards the former true error.

Intuitively, an error control based on the agreement between the polynomial extrapolations of the coupling variables and their updated values after a coupling time step is reassuring. Indeed, the accuracy of the predicted evolution of the coupling variables is ensured, therefore the coupled dynamics will also be correctly computed. However, it is possible that this approach is too conservative, as we cannot be certain that an error in the coupling variables will translate into a similar error in the coupled dynamics. We can imagine a situation where some coupling variables only have a weak impact on the subsystems solutions but may still vary rapidly, i.e. if $||\partial F(U_{n+1})/\partial U_{n+1}||$ is low while $||\partial H(Y_{n+1})/\partial Y_{n+1}||$ is large. Thus forcing a low-coupling time step to correctly capture their variations, even though the true dynamics of the coupled system is already well captured. This issue could be circumvented by computing errors on the subsystems state vectors z_i, however these are usually not accessible directly from the solvers.

With the error estimate obtained for a given coupling step size Δt, the "optimal" time step $\Delta t^\star$ can be computed, such that the error exactly matches the prescribed tolerance:

$$\Delta t^\star = \kappa \min_{i \in [1,N]} \Delta t \left| \frac{\epsilon^u_{i,n+1}(\Delta t)}{atol + rtol|u_{i,n+1}|} \right|^{\frac{-1}{p_i+1}} \quad (15)$$

where κ is a safety factor (typically 0.9), *atol* and *rtol* are absolute and relative error tolerances. Use of a PI- or PID-controller to smoothen the time step evolution across multiple steps can improve robustness [5] but will not be considered in this article.

2.4.2 Embedded error estimate

The previous error estimate can be computed for free. However it suffers from a major drawback when used in conjunction with an implicit coupling. Indeed, it has the same stability as the explicit coupling. In the stiff case, the time step will therefore be restricted by the stability of the explicit extrapolation, not by the actual accuracy of the implicitly-coupled result. A remedy we propose is to compare two implicit solutions $\bar{u}_{n+1}$ and $\bar{\bar{u}}_{n+1}$ obtained with approximations of degrees $\bar{p}$ and $\bar{\bar{p}} = \bar{p} + 1$. This is a typical approach for ODE integrators, e.g. embedded Runge-Kutta methods [5]. This second error estimate takes the form:

$$\epsilon^u_{i,n+1} = ||\bar{\bar{u}}_{i,n+1} - \bar{u}_{i,n+1}|| = \mathcal{O}\left(\Delta t^{\bar{p}+2}\right) \tag{16}$$

3 NUMERICAL EXPERIMENTS

We have developed an open-source Python demonstrator RHAPSOPY, available online [17], which allows the behaviour of the previous coupling strategy to be investigated on small-scale test problems. Tutorials in the form of Jupyter notebooks allow for a progressive introduction to the framework. In this section, we propose to test the coupling strategy on a simple yet representative application. We focus on a one-dimensional conjugate heat transfer problem, where two solid slabs exchange heat through an interface, as depicted in Figure 2.

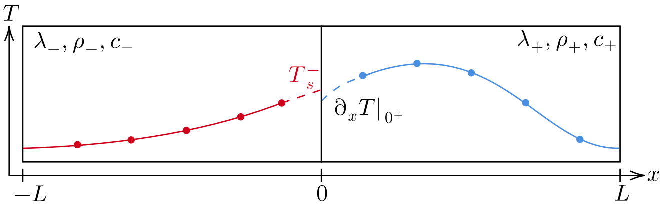

Figure 2: Conjugate heat-transfer problem

Each slab has uniform material properties. The thermal profile $T(x,t)$ is subject to the following evolution equation, with the continuity of the heat flux across the surface:

$$\rho_\pm c_\pm \partial_t T = \lambda_\pm \partial_{xx} T \tag{17}$$

$$\lambda_- \partial_x T|_{0-} = \lambda_+ \partial_x T|_{0+} \tag{18}$$

where ρ is the density, c the specific heat, λ the thermal conductivity. We impose zero-flux Neumann conditions at $x = \pm L$. We semi-discretise in space with a second-order centered finite-volume scheme with a uniform mesh of 50 cells in each slab. One heat conduction solver is attributed to each slab. We apply the classical Dirichlet-Neumann coupling at the interface as sketched in Figure 2: the right slab (+) receives a Dirichlet condition (equivalent to u_2 in our initial formulation), whose temperature is obtained by extrapolation of the internal temperature profile of the left slab (−). The left slab receives a Neumann (heat flux) condition (u_1), which is obtained by extrapolating the internal heat flux field $\lambda_+ \partial_x T$ of the right slab. We set $L = 1$, $\lambda_\pm = 1$, $c_\pm = 1$, $\rho_+ = 1$ and $\rho_- = 5$. The initial state is such that $T(x < 0) = 0$ and $T(x > 0) = 1$. We compute a monolithic quasi-exact reference solution with the high-order implicit RADAU5 scheme [5] and a relative error tolerance of $rtol = 10^{-12}$. The obtained evolution of the surface temperature T_s is plotted in Figure 3.

3.1 Convergence

We now verify the convergence order of the coupled strategy for various polynomial degrees. We focus on the initial part of the transient, up to $t_f = 0.04$ s. For each order and solution mode (explicit or implicit), we perform fixed time step simulations with 10 to 1000 steps. To

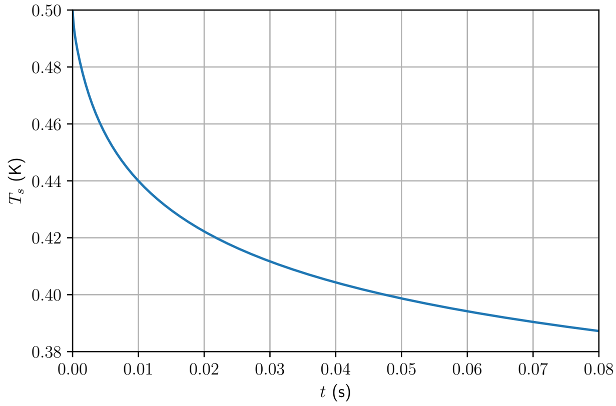

Figure 3: Reference evolution of T_s

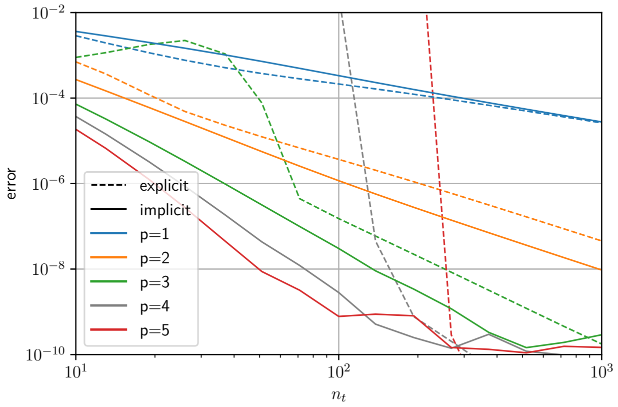

Figure 4: Global error on T_s

avoid larger errors caused by the rapid initial evolution of T_s, we start from $t = 0.01$ s and use the quasi-exact solution as initial condition. On each coupling interval, the left and right subsystems are integrated separately with RADAU5 and $rtol = 10^{-12}$. For the implicit coupling, the fixed-point problem is solved with a damped Newton method with a relative convergence criteria of 10^{-9} on the norm of the Newton step. The global error is measured with respect to the reference solution: $err = (1/t_f) \int_0^{t_f} |T_s(t) - T_{s,ref}|(t)dt$. An iterative starting procedure is used to ensure that the polynomial approximations reach their full order directly at first step.

Figure 4 shows the resulting convergence curves. We observe that many explicit simulations diverge when the time step is not small enough, and this divergence occurs sooner with higher-degree extrapolation polynomials. The asymptotic orders of convergence are reached as expected, thus demonstrating the potential of the strategy. In particular, we highlight the fact that the traditional first-order coupling is far less accurate than the coupling with linear predictions. In the asymptotic regime, for a given prediction degree, the ratio of the error between explicit and implicit coupling is very close to the ratio discussed in Section 2.3, e.g. we obtain ratios of 1, 5, 9 for degree-0, 1 and 2 predictions, respectively. Note that the error level cannot go below roughly 10^{-10} due to the prescribed accuracy for the solution of the fixed-point problem.

3.2 Stability

The previous convergence results show that, at least for the explicit case, instability may occur. A thorough theoretical investigation of stability in the context of coupled systems is a difficult task, and usually no analytical stability limit can be derived, even for simplified low-dimensional systems [15]. Therefore, we propose instead to investigate the stability of our coupling strategy numerically. To do so, we consider a similar test case as previously. For each order and each mode (explicit or implicit), we perform various adaptive coupled simulations with different error tolerances. As an example, we perform a few coupled simulations with a 3^{rd}-order explicit coupling with $rtol \in [10^{-8}, 10^{-1}]$ and we stop after 200 steps. Figure 5 shows the time step used in each simulation. After an initial transient, the time step settles on a nearly constant value. The interesting observation is that larger error tolerances lead to the same limit time step, i.e. for $rtol \leq 10^{-4}$, the time step becomes stability-limited. This limit value of the coupling step size will be referred to as maximum stable time step. In the context of ODE integration, this stability-limited behaviour is well known, e.g. [5].

We have used the same approach for other coupling orders and for the implicit case. To investigate how the coupling strategy behaves with various coupling strengths, we vary the ratio of densities ρ_-/ρ_+ from 0.01 to 1000. We note that a Neumann-Dirichlet coupling is only recommended when $\rho_- > \rho_+$ [22]. For comparison, we consider fully-coupled IMEX schemes as described in [6] which has been discussed in the introduction. We use the KenCarp3 (3^{rd}-order) and KenCarp58 (5^{th}-order) schemes of the Julia ODE library [23]. We also include the stability limit obtained when integrating the coupled problem monolithically with the well-known explicit RK45 method.

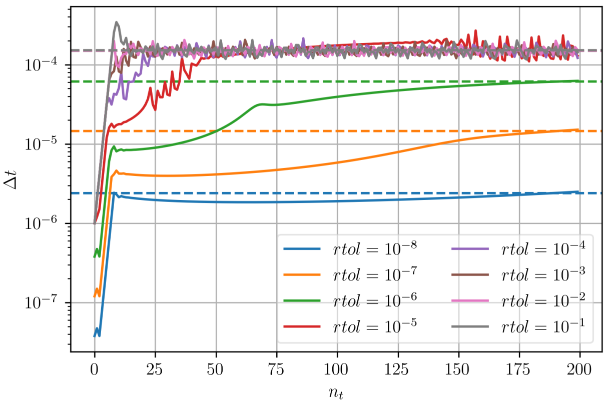

Figure 5: Effect of *rtol* on the coupling time step ($p = 3$, explicit, $\rho_+ = \rho_-$)

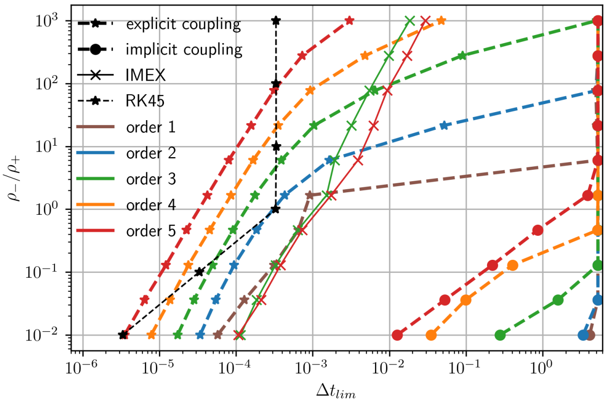

Figure 6: Largest stable time steps for various methods and orders

The obtained maximum stable time steps are plotted in Figure 6. RK45 closely follows the stability limit that can be computed analytically for an explicit Euler scheme, i.e. it is limited by the largest thermal diffusivity. For $\rho_-/\rho_+ < 10$, the explicit multistep coupling is not advantageous over other methods. For higher density ratios, it however becomes very attractive. The stability limit of the explicit multistep coupling decreases as its order increases, which is expected due to its similarity with Adams-Bashforth multistep schemes. We note that IMEX methods produce unstable results for $rtol > 10^{-8}$, i.e. their error estimates seem untrustworthy for such a test case. IMEX schemes are only marginally better than our explicit multistep coupling strategy for low to intermediate density ratios. Also, they cannot be applied as easily to any coupled system, since they require using a similar implicit scheme in the submodels, and having access to a split formulation which may be cumbersome to produce from existing codes.

The implicit multistep coupling strategy is much more stable than the other approaches. For $\rho_- > \rho_+$ the stability limit becomes practically unlimited. When investing the evolution of the time steps in each implicit simulation, no upper asymptotic limit is reached, the time step simply grows continuously until the maximum allowed physical time is reached. Even in the case where $\rho_- < \rho_+$ (the coupling should be reversed to a Dirichlet-Neumann condition), the implicit multistep coupling remains robust, though its stability decreases. We also note that high-order implicit coupling is less stable. Overall, the ratio of the maximum stable time step is on the order or 100 to 10^4 compared to the other approaches, which demonstrate the adequacy of our proposed strategy.

4 CONCLUSION

We have proposed a new coupling strategy focused on multiphysics applications involving the use of existing model-specific solvers. The strategy relies on multistep polynomial approximations of the coupling variables, which are used as dynamic inputs for the coupled solvers. High-order is attainable, and time adaptation is possible thanks to objective and reliable error estimates. Numerical experiments with the demonstrator code RHAPSOPY on a conjugate heat transfer problem have verified the convergence rates and shown that the stability limit is advantageous over other classical methods, in particular for the implicit variant of our strategy.

Future work will focus on applying the strategy to other problems of larger-scale, with an implementation of the coupling algorithm in the high-performance coupling library CWIPI developed at ONERA. Efficient use of interface quasi-Newton methods will be investigated to improve the applicability to large-scale cases. Finally, a theoretical analysis of the mathematical properties of the proposed multistep coupling will be conducted to improve the understanding of its behaviour.

REFERENCES

[1] F. Verdugo and W.A. Wall. Unified computational framework for the efficient solution of n-field coupled problems with monolithic schemes. *Computer Methods in Applied Mechanics and Engineering*, 310:335–366, 2016.

[2] M.M. Hopkins, H.K. Moffat, B. Carnes, R.W. Hooper, and R.P. Pawlowski. Final report on LDRD project : coupling strategies for multi-physics applications. Technical report, Sandia National Laboratories, 2007.

[3] P. Birken, K.J. Quint, S. Hartmann, and A. Meister. A time-adaptive fluid-structure interaction method for thermal coupling. *Computing and visualization in science*, 13(7):331–340, 2010.

[4] V. Kazemi-Kamyab, A.H. Van Zuijlen, and H. Bijl. Analysis and application of high order implicit Runge-Kutta schemes for unsteady conjugate heat transfer: A strongly-coupled approach. *Journal of Computational Physics*, 272:471–486, 2014.

[5] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.* Springer-Verlag Berlin Heidelberg, 2nd edition, 1996.

[6] V. Kazemi-Kamyab, A.H. Van Zuijlen, and H. Bijl. A high order time-accurate loosely-coupled solution algorithm for unsteady conjugate heat transfer problems. *Computer Methods in Applied Mechanics and Engineering*, 264:205–217, 2013.

[7] J. White, F. Odeh, A.L. Sangiovanni-Vincentelli, and A. Ruehli. Waveform Relaxation: Theory and Practice. Technical Report UCB/ERL M85/65, EECS Department, University of California, Berkeley, Jul 1985.

[8] B. Rüth, B. Uekermann, M. Mehl, P. Birken, A. Monge, and H. Bungartz. Quasi-newton waveform iteration for partitioned surface-coupled multiphysics applications. *International Journal for Numerical Methods in Engineering*, 122(19):5236–5257, 2021.

[9] P. Meisrimel, A.R. Monge, and P. Birken. A time adaptive multirate Dirichlet–Neumann waveform relaxation method for heterogeneous coupled heat equations. *Zeitschrift für Angewandte Mathematik und Mechanik*, page e202100328, 2023.

[10] L. François. *Multiphysics modelling and simulation of solid rocket motor ignition.* PhD thesis, Institut Polytechnique de Paris, 2022.

[11] G. Strang. On the construction and comparison of difference schemes. *SIAM journal on numerical analysis*, 5(3):506–517, 1968.

[12] M. Lesoinne and C. Farhat. Higher-order subiteration-free staggered algorithm for nonlinear transient aeroelastic problems. *AIAA Journal*, 36(9):1754–1757, 1998.

[13] S. Descombes, M. Duarte, T. Dumont, V. Louvet, and M. Massot. Adaptive time splitting method for multi-scale evolutionary partial differential equations. *Confluentes Mathematici*, 03(03):413–443, 2011.

[14] R. Kübler and W. Schiehlen. Two methods of simulator coupling. *Mathematical and computer modelling of dynamical systems*, 6(2):93–113, 2000.

[15] M. Busch. *Zur effizienten Kopplung von Simulationsprogrammen.* PhD thesis, Universität Kassel, 2012.

[16] T. Meyer, J. Kraft, and B. Schweizer. Co-simulation: Error estimation and macro-step size control. *Journal of Computational and Nonlinear Dynamics*, 16(4):041002, 2021.

[17] L. Francois. Rhapsopy GitHub page: https://github.com/hpc-maths/Rhapsopy.

[18] B. Uekermann. *Partitioned Fluid-Structure Interaction on Massively Parallel Systems.* PhD thesis, Technische Universität München, 2016.

[19] A. Santiago, M. Zavala-Aké, R. Borrell, G. Houzeaux, and M. Vázquez. HPC compact quasi-Newton algorithm for interface problems. *Journal of Fluids and Structures*, 96:103009, 2020.

[20] M. Arnold, C. Clauß, and T. Schierz. Error analysis and error estimates for co-simulation in FMI for model exchange and co-simulation V2. 0. In *Progress in Differential-Algebraic Equations*, pages 107–125. Springer, 2014.

[21] J. Butcher. *Numerical Methods for Ordinary Differential Equations.* John Wiley & Sons, Ltd, 2016.

[22] M.B. Giles. Stability analysis of numerical interface conditions in fluid-structure thermal analysis. *International journal for numerical methods in fluids*, 25(4):421–436, 1997.

[23] C. Rackauckas and Q. Nie. DifferentialEquations.jl–a performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, 5(1), 2017.