INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2003; **58**:1397–1434 (DOI: 10.1002/nme.860)

Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications

Charbel Farhat[*,†] and Marion Chandesris

Department of Aerospace Engineering Sciences, Center for Aerospace Structures, University of Colorado, Boulder, CO 80309-0429, U.S.A.

SUMMARY

A methodology for squeezing the most out of massively parallel processors when solving partial differential evolution equations by implicit schemes is presented. Its key components include a preferred implicit time-integrator, a decomposition of the time-domain into time-slices, independent time-integrations in each time-slice of the semi-discrete equations, and Newton-type iterations on a coarse time-grid. Hence, this methodology parallelizes the time-loop of a time-dependent partial differential equation solver without interfering with its sequential or parallel space-computations. It is particularly interesting for time-dependent problems with a few degrees of freedom such as those arising in robotics and protein folding applications, where the opportunities for parallelization over the degrees of freedom are limited. Error and stability analyses of the proposed parallel methodology are performed for first- and second-order hyperbolic problems. Its feasibility and impact on reducing the solution time below what is attainable by methods which address only parallelism in the space-domain are highlighted for fluid, structure, and coupled fluid–structure model problems. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: massively parallel processing; parareal scheme; parallel time-integration; time-domain decomposition

1. INTRODUCTION

Most, if not all, time-dependent partial differential equation (PDE) solvers are implemented on parallel computers by mapping all requested processors on the space-domain for parallel computations within one time-step, and performing serially the time-integration of the underlying semi-discrete equations, from the initial condition to the last time-instance of interest. This

[*] Correspondence to: Charbel Farhat, Department of Aerospace Engineering Sciences, Center for Aerospace Structures, University of Colorado at Boulder, Campus Box 429, Boulder, CO 80309-0429, U.S.A.
[†] E-mail: charbel.farhat@colorado.edu

Contract/grant sponsor: National Science Foundation; contract/grant number: 008463
Contract/grant sponsor: Air Force Office of Scientific Research; contract/grant number: F49620-01-1-0129

Received 5 August 2002
Revised 14 March 2003
Accepted 25 April 2003

strategy is sound as long as it keeps all processors usefully busy and reduces the total solution time. It is motivated primarily by the fact that time-integration is inherently a sequential process in which the knowledge of the solution at a given time-instance requires the knowledge of the solution at previous time-instances. Other considerations such as granularity also favour in many cases assigning all the parallel processing resources to the spatial component of a partial differential evolution equation solver. As a result, space-domain decomposition techniques in general, and parallel equation solvers in particular—for the case of implicit time-integration which incurs at each time-instance the solution of an algebraic system of equations—have received considerable attention in the literature during the last two decades.

However, with the advent of massively parallel computers with hundreds and even thousands of processors such as the Department of Energy's ASCI machines, the parallelization strategy outlined above is not always optimal. This is particularly true when the main motivation for parallel processing is the reduction of the total solution time, irrespectively of the parallel efficiency (speed-up per processor). Indeed, it is well-known that because of hardware related issues such as memory bandwidth effects and interprocessor communication costs, the solution time of a given fixed size problem in space typically stagnates then increases when the number of processors N_p is increased beyond a critical value N_p^cr. In this sense, the smaller is N_p^cr compared to the maximum number of available processors $N_\mathrm{p}^\mathrm{max}$, the less optimal is the parallel implementation strategy outlined above. When the objective is to demonstrate parallel scalability up to as large a number of processors as possible, this issue is typically dealt with by increasing the size of the spatial component of the PDE problem to be solved in order to reduce the relative impact on the total solution time of interprocessor communication costs and other sources of parallel processing overhead. However, once a critical mesh resolution dictated by accuracy considerations is reached, increasing further the size in space of a given problem in order to enable a more effective use of a higher number of processors unnecessarily increases the total solution time. For some scientific fields such as turbulence, the critical mesh resolution often remains beyond the capabilities of the largest of the currently available supercomputers. However, in many other scientific and engineering fields, the mesh resolution dictated by accuracy is such that the problem size is often within the realm of a few dozen processors if the system has a distributed memory architecture, N_p^cr ranges from a few dozens to a few hundreds, whereas $N_\mathrm{p}^\mathrm{max}$ can be as large as several thousands. This scenario is more exacerbated for non-PDE-related time-dependent problems with only a handful degrees of freedom (dofs) as in robotics applications where computing the solution in real time is often desired, and protein folding applications where time-integration is carried over a very large time-domain. For such problems, the question is how to reduce further the solution time when desired? One possible approach is to superpose parallelism in the time-domain to parallelism in the space-domain, or in other words, to parallelize the time-loop of a PDE solver. This is a challenging task which to the best of our knowledge has not been addressed in the archival literature, except in a recent paper [1] by Lions, Maday, and Turinici. Indeed, these authors have recently exposed in Reference [1] a framework for domain decomposition in time featuring a fine as well as a coarse time-grid, and proposed an iterative algorithm for parallelizing the time-loop of a single-stage time-integrator. They labelled their resulting parallel ordinary differential equation (ODE) solver as a 'parareal' scheme, to underscore their long-term desire for achieving real-time performance with their approach. In this paper, we focus on the shorter term objective of superposing as efficiently as possible time-parallelism to space-parallelism. To this effect, we adopt as a starting point the framework of parareal algorithms proposed in Reference [1]. Using Newton's method for the continuous problem and the theory of distributions, we first present an original mathematical theory and justification for this framework, then generalize it to multi-stage time-integrators as these are often preferred for achieving second- and higher-order accuracy. We present error and stability analyses that provide further insight into the possibilities and limitations of the parareal concept, and prove its convergence. Most importantly, we also discuss feasibility and speed-up studies for three fluid, structure, and fluid–structure model problems. These studies identify both the physical and simulation parameters that can make or break the objective stated in this paper. They also highlight the potential of the proposed parallel computational technology for using meaningfully a higher number of processors than allowed by conventional approaches, thereby reducing the total solution time of time-dependent PDE problems below what parallelism exclusively in the space-domain can achieve.

Finally, we note that whereas recent works focusing on time-integrators have addressed multiple spatial scales in a fine time-scale partition (for example, see Reference [2]), to the best of our knowledge, Reference [1] and this paper study for the first time the use of multiple time-scales to integrate a particular spatial dof.

2. DISCONTINUOUS TIME-DECOMPOSED PARALLEL TIME-INTEGRATORS

2.1. Decomposition of the time-domain for doubly parallel computations

If $N_{\rm p}^{\rm cr} < N_{\rm p}^{\max}$, we begin by partitioning the target parallel computer into $N_{\rm c}$ clusters each with $N_{\rm p}^{\rm cr}$ processors. We assume that

$$N_{\rm c} = \left[\frac{N_{\rm p}^{\max}}{N_{\rm p}^{\rm cr}} \right] \geqslant 2 \tag{1}$$

Let

$$\begin{aligned} \frac{\mathrm{d}y}{\mathrm{d}t} &= F(y,t) + g(t) \\ y(t^0) &= y^0 \end{aligned} \tag{2}$$

denote the ODE and corresponding initial condition obtained after semi-discretizing a given time-dependent first-order hyperbolic, second-order hyperbolic, or parabolic, linear or non-linear PDE. We denote by $d \geqslant 1$ the number of dof's generated by the chosen semi-discretization procedure. If the PDE is first-order hyperbolic or parabolic, $y \in \mathbb{R}^d$. If it is second-order hyperbolic, then

$$y = \begin{pmatrix} \dfrac{\mathrm{d}q}{\mathrm{d}t} \\ q \end{pmatrix} \in \mathbb{R}^{2d} \tag{3}$$

We assume that F is sufficiently smooth, introduce the notation

$$F_y(y,t) = \frac{\partial F}{\partial y}(y,t) \tag{4}$$

and denote by ITA the sequential implicit time-integration algorithm chosen for solving the ODE (2) in the time-domain

$$\Omega_t = [t^0,\ t^{\mathrm{f}}] \tag{5}$$

For the sake of simplicity, we assume a constant time-step

$$\Delta t = t^{n+1} - t^n \tag{6}$$

Following the idea proposed in Reference [1], we partition Ω_t into N_{ts} time-slices $\{S^i = [T^i, T^{i+1}]\}_{i=0}^{N_{\mathrm{ts}}-1}$, with $T^0 = t^0$ and $T^{N_{\mathrm{ts}}} = t^{\mathrm{f}}$. We assume that

$$N_{\mathrm{ts}} = rN_{\mathrm{c}}, \quad r \in \mathbb{N},\ r \geqslant 1 \tag{7}$$

all time-slices have the same size

$$\Delta T = T^{i+1} - T^i = \frac{(t^{\mathrm{f}} - t^0)}{N_{\mathrm{ts}}} = \frac{(t^{\mathrm{f}} - t^0)}{rN_{\mathrm{c}}} \tag{8}$$

and that this size is a multiple of Δt—that is,

$$\Delta T = J\,\Delta t, \quad J \in \mathbb{N},\ J > 1 \tag{9}$$

While we are interested in relatively large values of J, say $J \geqslant 5$, we note that Equations (7)–(9) imply that J and N_{c} are constrained by

$$JN_{\mathrm{c}} = \frac{(t^{\mathrm{f}} - t^0)}{r\Delta t} \leqslant \frac{(t^{\mathrm{f}} - t^0)}{\Delta t} \tag{10}$$

Hence, for a given J, the free parameter r can be determined from Equation (10).

Our main idea is to assign a cluster of $N_{\mathrm{p}}^{\mathrm{cr}}$ processors to each time-slice S^i to perform in parallel all the computations in the space-domain that are required for advancing the solution of problem (2) from $t = T^i$ to $t = T^{i+1}$, and to process N_{c} time-slices simultaneously using the N_{c} clusters. This doubly parallel computational strategy is graphically depicted in Figure 1 where it is also contrasted with the conventional space-based parallel processing methodology. In this work, we do not address the space-based parallel component of this strategy because it is by now textbook material. We focus on its time-based parallel component and the technology of time-parallel implicit time-integrators which enables it, because these are still in their infancy. For this reason, and because it is convenient to do so, in the remainder of this paper we use simply the word parallel to refer to the time-based parallel—or parareal in the terminology of Reference [1]—component of the computational methodology graphically depicted in Figure 1(b).

As a preamble to the design of a parallel (or parareal) implicit time-integrator (PITA), we point out the following simple idea. Let Y^i denote the value of the exact solution of problem (2), y_{ex}, at the beginning of the time-slice S^i

$$Y^i = y_{\mathrm{ex}}(T^i), \quad 0 \leqslant i \leqslant N_{\mathrm{ts}} - 1 \tag{11}$$

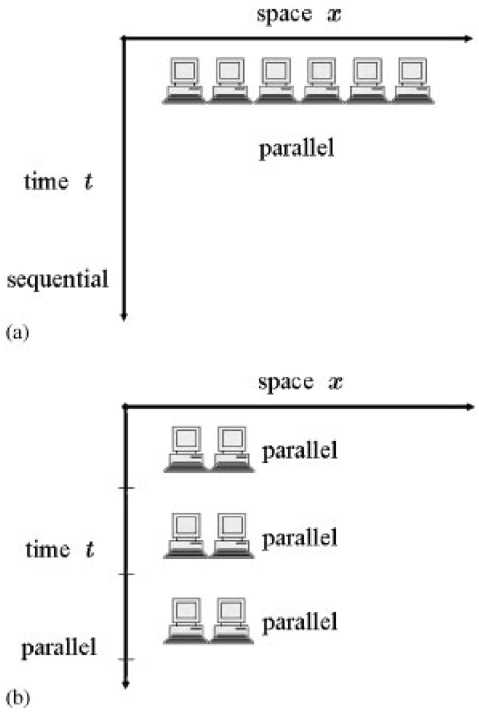

Figure 1. Doubly parallel vs simply parallel time-dependent computations: (a) space-based parallel implementation of a time-dependent PDE solver on a massively parallel computer; and (b) proposed parallel implementation for superposing space-and time-based parallelisms.

If the Y^i values were somehow known, a PITA could be simply designed by decomposing the execution of a given ITA as follows:

```
PITA-0
For $0 \leqslant i \leqslant N_{\mathrm{ts}} - 1$ do in parallel
    Initialize the ITA in each time-slice $S^i$ with $y(T^i) = Y^i$.
    Apply the ITA to the solution of problem (2) in the time-slice $S^i$.
End do in parallel
```

The above algorithm is embarrassingly parallel. Unfortunately, it is also unrealistic because the 'seed' information $\{Y^i\}_{i=1}^{N_{\mathrm{ts}}-1}$ is typically unavailable. Nevertheless, it motivates one approach for constructing a PITA which consists of generating a set of approximations of the Y^i values that are sufficiently accurate to enable the concept represented by the PITA-0. This approach was adopted in Reference [1] where the authors have outlined an iterative procedure for generating a sequence of increasingly accurate approximations of the Y^i values, assuming a single-stage time-integrator. In this paper, we *formally* derive the methodology presented in Reference [1] in order to gain a better insight into its potential and limitations, generalize it to multi-stage time-integrators, and analyse its performance for a more comprehensive family of problems.

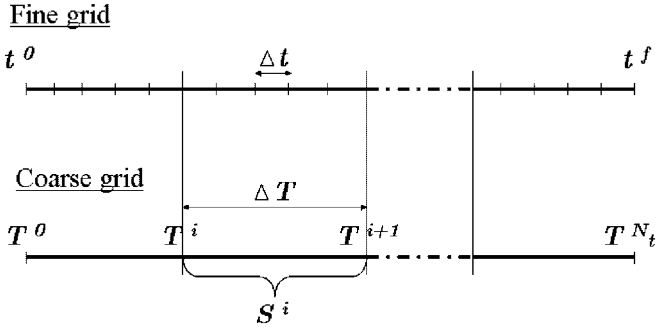

Figure 2. Fine and coarse time-grids.

2.2. *A Newton type, two-grid, parallel computational framework*

We refer to the time-discretization of Ω_t by samples of length Δt as the *fine* time-grid, and to the time-slicing of the same time-domain by samples of length ΔT as the *coarse* time-grid (see Figure 2). From Equation (9), it follows that the correspondence between these two time-grids is given by

$$T^i = t^{Ji}, \quad 0 \leqslant i \leqslant N_{\text{ts}} \tag{12}$$

The sequential ITA performs on the fine time-grid. The PITA to be presented in this paper is an *iterative* algorithm which performs on both the fine and coarse time-grids. For this reason, we begin by introducing the following notation:

- We designate by a subscript k a quantity associated with the kth iteration.
- We designate by a lower case and a superscript i a quantity evaluated on the fine time-grid at the time-instance t^i.
- We designate by an upper case and a superscript i a quantity evaluated on the coarse time-grid at the time-instance $T^i = t^{Ji}$.
- We designate by a lower case, superscript i, and a pair of parentheses ( ) a function defined in the time-slice S^i.
- We designate by the subscript ITA the result obtained by applying the ITA to the solution of problem (2) on the fine time-grid.
- We designate by the subscript ex the exact solution of problem (2).

Given for each Y^i value an initial approximation Y_0^i, our intent is to apply a Newton-type method to generate a sequence of increasingly accurate seed information $\{Y_k^i\}_{i=1}^{N_{\text{ts}}-1}$. To this effect, we first note that for any given set of Y_k^i values, a PITA which follows the concept illustrated by the PITA-0 replaces problem (2) defined in Ω_t and its solution y_{ex} by the N_{ts} problems

$$\begin{aligned} \frac{\mathrm{d}y}{\mathrm{d}t} &= F(y,t) + g(t) \quad \text{in } S^i \\ y(T^i) &= Y_k^i, \quad 0 \leqslant i \leqslant N_{\text{ts}} - 1 \end{aligned} \tag{13}$$

and their exact solutions which we denote here by $\{y_k^i(\ )\}_{i=0}^{N_{\rm ts}-1}$. These solutions satisfy

$$y_k^i(T^i) = Y_k^i, \quad 0 \leqslant i \leqslant N_{\rm ts} - 1 \tag{14}$$

and exhibit the following jumps at the time-instances T^i:

$$\Delta_k^i = y_k^{i-1}(T^i) - Y_k^i, \quad 1 \leqslant i \leqslant N_{\rm ts} - 1 \tag{15}$$

Let y_k denote the piecewise $\mathscr{C}^1$ function in Ω_t whose restriction to each time-slice S^i is equal to $y_k^i(\ )$. We note that the more accurate is each approximation Y_k^i of Y^i, the closer is y_k to satisfying Equations (2). Hence, we reformulate the problem of generating an improving sequence of Y_k^i values as that of generating by a Newton-type method an improving sequence of y_k functions. For this purpose, we introduce a correction function c_k which is also piecewise $\mathscr{C}^1$ in Ω_t, define

$$C_k^i = c_k(T^{i-}), \quad 1 \leqslant i \leqslant N_{\rm ts} - 1 \tag{16}$$

where T^{i-} is the right boundary of $S^{i-1} = [T^{i-1},\ T^i]$ (and T^{i+} is the left boundary of $S^i = [T^i,\ T^{i+1}]$), and plan on updating each Y_k^i value as follows:

$$Y_{k+1}^i = y_k^{i-1}(T^i) + C_k^i \tag{17}$$

Therefore, even though we introduce c_k in the entire domain Ω_t, Equation (17) implies that we are interested only in its values on the time-instances of the coarse time-grid. To determine c_k, we require that the corrected function $y_k + c_k$ satisfies Equations (2)—that is,

$$\begin{aligned} \frac{\mathrm{d}(y_k + c_k)}{\mathrm{d}t} &= F(y_k + c_k, t) + g(t) \\ (y_k + c_k)(t^0) &= y^0 \end{aligned} \tag{18}$$

where differentiation in time is performed in the *distribution* sense.

Linearizing the above equation around y_k gives

$$\frac{\mathrm{d}c_k}{\mathrm{d}t} = F_y(y_k, t)c_k - \left[\frac{\mathrm{d}y_k}{\mathrm{d}t} - F(y_k, t) - g(t)\right] \tag{19}$$

where the quantity between brackets is the residual of the approximate solution y_k. From the theory of distributions, it follows that

$$\frac{\mathrm{d}y_k}{\mathrm{d}t} = F(y_k, t) + g(t) - \sum_{i=1}^{N_{\rm ts}-1} \Delta_k^i \delta(T^i) \tag{20}$$

where δ is the Dirac function. Hence, from Equations (19) and (20) it follows that c_k is governed by

$$\begin{aligned} \frac{\mathrm{d}c_k}{\mathrm{d}t} &= F_y(y_k, t)c_k + \sum_{i=1}^{N_{\rm ts}-1} \Delta_k^i \delta(T^i) \\ c_k(T^0) &= 0 \end{aligned} \tag{21}$$

As stated above, the main purpose of c_k is to produce the values C_k^i needed for updating Y_k^i (see Equation (17)). Consequently, problem (21) needs be solved only on the coarse time-grid, using for example the same ITA. If J is sufficiently large and the ITA is sufficiently stable, this incurs a minimal computational cost. It is because of this stability requirement that we limit ourselves in this paper to implicit time-integrators.

The discretization of Equations (21) by the single-stage backward Euler scheme leads to

$$\begin{aligned}\frac{C_k^{i+1}-C_k^i}{\Delta T} &= F_y(y_k,t)C_k^{i+1}+\sum_{i=1}^{N_{\text{ts}}-1}\frac{\Delta_k^i}{\Delta T}\\ C_k^0 &= 0\end{aligned} \tag{22}$$

which is exactly the parareal algorithm proposed in Reference [1]. However, because of the second term on the right-hand side of the first of Equations (22), which results from the discretization of the Dirac term in problem (21), it is not clear how one can start from Equations (22) to construct a parallel multi-stage time-integrator. To address this issue, we invoke again the theory of distributions and re-write problem (21) in the following alternative form:

$$\begin{aligned}\frac{\mathrm{d}c_k}{\mathrm{d}t} &= F_y(y_k,t)c_k\\ c_k(T^0) &= 0\\ c_k(T^{i+}) &= c_k(T^{i-})+\Delta_k^i, \quad 1\leqslant i\leqslant N_{\text{ts}}-1\end{aligned} \tag{23}$$

This new form can be discretized by any multi-stage scheme without any difficulty because the jumps are converted from source terms to initial condition terms. It also highlights the role of the correction function c_k, which is to propagate jumps (15) (or errors) on the coarse time-grid, as typically done in classical multi-grid and multi-level domain decomposition methods.

Remark 1

If $F(y,t)$ is a linear function, then

$$F_y(y_k,t)c_k = F(c_k) \tag{24}$$

In summary, we adopt the following two-grid PITA for enabling the doubly parallel computational strategy highlighted in Figure 1. We designate by the symbol $\ominus$ the time-sequential steps performed on the coarse time-grid, and by the symbol $\oplus$ the parallel steps performed on the decomposed fine time-grid.

PITA

$\ominus$ Provide initial seed values Y_0^i, $0\leqslant i\leqslant N_{\text{ts}}-1$.
For $k=0,1,\ldots$

(1) $\oplus$ *Step 1*: Using the updated seed values as initial conditions, apply the ITA to the solution of problems (13) on the fine time-grid, which generates the numerical values y_k^i, $0\leqslant i\leqslant f$.

(2) $\oplus$ *Step 2*: Evaluate the jumps $\Delta_k^i, 1 \leqslant i \leqslant N_{\rm ts} - 1$ on the coarse time-grid Equation (15). Stop if all these jumps are sufficiently small.

(3) $\ominus$ *Step 3*: Apply the ITA to the solution of problem (23) on the coarse time-grid in order to compute the correction coefficients C_k^i, $1 \leqslant i \leqslant N_{\rm ts} - 1$ (Equation (16)).

(4) $\oplus$ *Step 4*: Update the seed values Y_{k+1}^i, $0 \leqslant i \leqslant N_{\rm ts} - 1$ (Equation (17)).

In this work, we compute the initial values Y_0^i by applying the same ITA of interest to the solution of the original problem (2) on the coarse time-grid. This step as well as Step 3 are time-sequential steps. However, they are carried out on the coarse time-grid. Hence, for a sufficiently large J-ratio (see Equation (9)) and a sufficiently stable ITA, their computational cost can be minimized.

2.3. Finite termination property and convergence proof

In Appendix A, we prove the following result.

Proposition 1

At each iteration k, the jumps, coarse time-grid corrections, and updates of the seed values generated by the PITA satisfy

$$\begin{aligned} \Delta_k^i &= 0, \quad 0 \leqslant i \leqslant k \\ C_k^i &= 0, \quad 0 \leqslant i \leqslant k+1 \\ Y_{k+1}^i &= y_{\rm ITA}^{Ji}, \quad 1 \leqslant i \leqslant k+1 \end{aligned}$$

From this result, it follows that:

- At each iteration k of the PITA, only the time slices $\{S^i\}_{i=k}^{N_{\rm ts}-1}$ need be processed.
- After $N_{\rm ts} - 1$ iterations, the numerical solution of problem (2) obtained by the PITA is identical to that obtained by the corresponding time-sequential ITA.

The first of the two properties stated above can be exploited for optimizing the computer implementation of the PITA. The second of these properties is a finite termination property which, for a sufficiently stable ITA, proves the convergence of the corresponding PITA. However, as it will be demonstrated in Section 6, the PITA described in Section 2.2 converges in practice in only a few iterations.

Next, we perform the speed-up analysis of the PITA in order to highlight its potential as well as its limiting factors.

3. SPEED-UP ANALYSIS

Let τ denote the CPU time associated with the execution of the chosen ITA over one time-step, including the overhead corresponding to parallel processing in the space-domain. The total CPU time associated with the time-sequential solution on the fine time-grid of problem (2) by this ITA is

$$T_{\mathrm{ITA}} = \frac{(t^{\mathrm{f}} - t^0)}{\Delta t}\tau \tag{25}$$

From the description of the PITA given in Section 2.2, the fact that at each iteration k its scope is restricted to the time-slices $\{S^i\}_{i=k}^{N_{\mathrm{ts}}-1}$ (see Section 2.3), and from Equations (7) and (9), it follows that the total CPU time associated with this parallel algorithm is

$$\begin{aligned} T_{\mathrm{PITA}} &= \sum_{k=0}^{N_{\mathrm{it}}-1} \left(\frac{(t^{\mathrm{f}} - t^0)}{\Delta t}\tau_k + \frac{(t^{\mathrm{f}} - t^0)}{\Delta T}\tau \right) + T_{\mathrm{ov}}(N_{\mathrm{c}}) \\ &= \left(\sum_{k=0}^{N_{\mathrm{it}}-1} \frac{(t^{\mathrm{f}} - t^0)}{\Delta t}\tau_k \right) + N_{\mathrm{it}} \frac{(t^{\mathrm{f}} - t^0)}{J\Delta t}\tau + T_{\mathrm{ov}}(N_{\mathrm{c}}) \end{aligned} \tag{26}$$

where N_{it} denotes the number of iterations for convergence of the PITA, the first term corresponds to parallel computations on the fine time-grid, the second term corresponds to sequential computations on the coarse time-grid, $T_{\mathrm{ov}}(N_{\mathrm{c}})$ denotes the overhead associated with parallel processing in the time-domain using N_{c} clusters, and

$$\tau_k = \left(\frac{N_{\mathrm{ts}} - iN_{\mathrm{c}}}{N_{\mathrm{ts}}} \right) \frac{\tau}{N_{\mathrm{c}}} = \frac{(r-i)}{rN_{\mathrm{c}}}\tau, \quad iN_{\mathrm{c}} \leqslant k \leqslant (i+1)N_{\mathrm{c}} - 1, \ 0 \leqslant i \leqslant r-1 \tag{27}$$

Let j^* be the unique integer satisfying

$$j^* \leqslant r - 1, \quad j^* N_{\mathrm{c}} \leqslant N_{\mathrm{it}} - 1 \leqslant (j^*+1)N_{\mathrm{c}} - 1 \tag{28}$$

From Equation (26) and definition (28), it follows that the total CPU time associated with the PITA can be written as

$$T_{\mathrm{PITA}}(r) = \frac{(t^{\mathrm{f}} - t^0)}{\Delta t} \left(N_{\mathrm{it}} - \frac{j^*}{r}(N_{\mathrm{it}} - j^* N_{\mathrm{c}}) - \sum_{l=1}^{j^*-1} \frac{l}{r} N_{\mathrm{c}} \right) \frac{\tau}{N_{\mathrm{c}}} + N_{\mathrm{it}} \frac{(t^{\mathrm{f}} - t^0)}{J\Delta t}\tau) + T_{\mathrm{ov}}(N_{\mathrm{c}}) \tag{29}$$

The two extreme implementation strategies of the PITA correspond to $r = 1$ (number of time-slices equal to number of clusters) and $r \gg 1$ (number of time-slices far greater than the number of clusters). From Equation (28), it follows that for $r = 1$, $j^* = 0$ and therefore

$$T_{\mathrm{PITA}}(r=1) = N_{\mathrm{it}} \left(\frac{(t^{\mathrm{f}} - t^0)}{\Delta t} \right) \left(\frac{1}{J} + \frac{1}{N_{\mathrm{c}}} \right) \tau + T_{\mathrm{ov}}(N_{\mathrm{c}}) \tag{30}$$

and for $r \gg 1$ and a reasonable number of iterations N_{it},

$$T_{\mathrm{PITA}}(r \gg 1) \approx N_{\mathrm{it}} \left(\frac{(t^{\mathrm{f}} - t^0)}{\Delta t} \right) \left(\frac{1}{J} + \frac{1}{N_{\mathrm{c}}} \right) \tau + T_{\mathrm{ov}}(N_{\mathrm{c}}) \tag{31}$$

Therefore, we adopt in all cases the following estimation of the parallel CPU time of the PITA:

$$T_{\mathrm{PITA}} = N_{\mathrm{it}} \left(\frac{(t^{\mathrm{f}} - t^0)}{\Delta t} \right) \left(\frac{1}{J} + \frac{1}{N_{\mathrm{c}}} \right) \tau + T_{\mathrm{ov}}(N_{\mathrm{c}}) \tag{32}$$

Furthermore, for a feasibility study, $T_{\text{ov}}(N_{\text{c}})$ can be neglected particularly in view of the little interprocessor communication required by the parallelization of the time-component of the PITA. Hence, using N_{c} clusters, the potential speed-up of the PITA over its underlying ITA is

$$S_{\text{p}}(N_{\text{c}}) = \frac{T_{\text{ITA}}}{T_{\text{PITA}}} = \frac{1}{N_{\text{it}}(1/N_{\text{c}} + 1/J)} \tag{33}$$

For a sufficiently large N_{c}, this speed-up can be approximated by J/N_{it}, which reveals that, as expected, the speed-up that can be provided by the PITA increases with J and decreases with N_{it}. We remind the reader that J is limited primarily by stability considerations on the coarse time-grid. Therefore, the PITA can be expected to perform best when the underlying ITA is unconditionally stable. We also point out that the PITA can be expected to require at least one corrective iteration for delivering the desired accuracy. In this presumably best case, the maximum theoretical speed-up is bounded by $N_{\text{c}}/2$, and therefore the maximum theoretical speed-up per cluster is 50%. For this reason, the PITA and the computational strategy outlined in Figure 1(b) may not necessarily optimize the traditional concept of parallel efficiency. However, they offer the potential for reducing the solution time by squeezing the most out of a given massively parallel processor, which is of great importance for many PDE applications. This potential is further analysed in Sections 4 and 5 before it is assessed in Section 6 for three fluid, structure, and fluid–structure model problems.

4. ERROR ANALYSIS FOR LINEAR PROBLEMS

Throughout this section, we assume that $F(y,t)$ is a linear function of y, and therefore problem (2) is linear.

First, we focus on the error analysis of the seed information generated by the PITA, and denote by

$$E_k^i = y_{\text{ex}}(T^i) - Y_k^i, \quad 0 \leqslant i \leqslant N_{\text{ts}} - 1 \tag{34}$$

the error for each seed value Y_k^i. By construction, $E_k^0 = 0$. In Appendix B.1, we prove that this error satisfies the following result.

Proposition 2

Given a p-order ITA that is stable on both the fine and coarse time-grids, there exists a constant a_k such that for all $1 \leqslant i \leqslant N_{\text{ts}} - 1$,

$$|E_k^i| \leqslant a_k \max(\Delta T^{(k+1)p},\ \Delta t^p) \tag{35}$$

This result shows that at each iteration k, the seed values generated by the PITA on the coarse time-grid improve by one order of accuracy.

Next, we turn our attention to the estimation of the error of the numerical solution computed by the PITA. For this purpose, we define

$$e_k^i(t^n) = y_{\text{ex}}^n - y_k^n, \quad t^n \in S^i, \ 0 \leqslant i \leqslant N_{\text{ts}} - 1 \tag{36}$$

We note that by design, $e_k^0(\ ) = O(\Delta t^p)$ where p is the order of accuracy of the chosen ITA. We prove in Appendix B.2 the following result.

Proposition 3
Given a p-order ITA that is stable on both the fine and coarse time-grids, there exists a constant b_k such that for all $1 \leqslant i \leqslant N_{\text{ts}} - 1$,

$$\|e_k^i(\ )\|_\infty \leqslant b_k \max(\Delta T^{(k+1)p},\ \Delta t^p) \tag{37}$$

The above result shows that the accuracy of the numerical solution of problem (2) computed by the PITA improves with the iteration count k, and is bounded by the order of accuracy of the underlying ITA. Furthermore, except for the fact that b_k depends on k, Equation (37) suggests that

$$N_{\text{it}} \approx \frac{\log \Delta t}{\log \Delta T} = \frac{\log \Delta t}{\log J \Delta t} \tag{38}$$

which shows that, as can be expected, the number of iterations required by the PITA for attaining the accuracy of the underlying ITA increases with J. This in turn shows that, given a sufficiently stable ITA, increasing the ratio of coarse and fine time-steps J can decrease the sought-after speed-up. However, we have shown in Section 3 that increasing J can also increase the PITA's potential speed-up because it decreases the CPU time associated with the repeated sequential solutions of the coarse problem (23). Hence, the optimal value of J is sufficiently large to reduce the CPU time associated with propagating the errors on the coarse time-grid, but sufficiently small to address any stability limit of the given ITA and limit the number of iterations required for convergence.

Finally, we consider the discrepancy between the numerical solution predicted by the PITA and that predicted by its underlying ITA. We denote this discrepancy by

$$\bar{e}_k^i(t^n) = y_k^n - y_{\text{ITA}}^n, \quad t^n \in S^i, \ 0 \leqslant i \leqslant N_{\text{ts}} - 1 \tag{39}$$

From Proposition 1 (see Section 2.3), it follows that

$$\bar{e}_k^i(\ ) = 0, \quad 1 \leqslant i \leqslant k - 1$$

In Appendix B.3, we prove the following result.

Proposition 4
Given a second-order ITA that is stable on both the fine and coarse time-grids, there exists a constant d_k such that for all $1 \leqslant i \leqslant N_{\text{ts}} - 1$,

$$\|\bar{e}_k^i(\ )\|_\infty \leqslant d_k i^{k+1} \Delta T^{3(k+1)} \tag{40}$$

This result suggests that long-term accuracy requires more iterations than short-term accuracy. Given Equation (33), it also follows that a long-term accuracy requirement could compromise the sought-after speed-up of the PITA.

In summary, the error analyses reported herein suggest that if the ITA is sufficiently stable, the PITA will exhibit the following behaviour:

- Its accuracy will increase with its iteration count until it reaches the accuracy of the underlying ITA.
- It will perform better for short-term simulations than for long-term ones.

5. LINEAR STABILITY ANALYSIS

Again, we consider here only the case where $F(y,t)$ is a linear function of y and therefore problem (2) is linear. We denote by $A^{\mathrm{F}}_{\mathrm{ITA}}$ the amplification matrix of the given ITA on the fine time-grid, and by $A^{\mathrm{C}}_{\mathrm{ITA}}$ its amplification matrix on the coarse time-grid. These matrices satisfy

$$\begin{aligned} y^{n+1}_{\mathrm{ITA}} &= A^{\mathrm{F}}_{\mathrm{ITA}} y^n_{\mathrm{ITA}}, \quad 0 \leqslant n \leqslant f-1 \\ Y^{i+1}_{\mathrm{ITA}} &= A^{\mathrm{C}}_{\mathrm{ITA}} Y^i_{\mathrm{ITA}}, \quad 0 \leqslant i \leqslant N_{\mathrm{ts}}-1 \end{aligned} \tag{41}$$

We assume that $A^{\mathrm{C}}_{\mathrm{ITA}}$ and $A^{\mathrm{F}}_{\mathrm{ITA}}$ commute and therefore are diagonalizable in the same basis. This assumption holds, for example, for the entire family of Newmark methods. In Appendix C.1, we prove the following result.

Proposition 5
Given an ITA with amplification matrices $A^{\mathrm{C}}_{\mathrm{ITA}}$ and $A^{\mathrm{F}}_{\mathrm{ITA}}$, the corresponding PITA admits a sequence of amplification matrices on the coarse time-grid denoted by $A^{\mathrm{C}}_{\mathrm{PITA}}(k,i)$ which satisfy for $k \geqslant 1$,

$$Y^{i+1}_k = A^{\mathrm{C}}_{\mathrm{PITA}}(k,i) Y^i_k, \quad 0 \leqslant i \leqslant N_{\mathrm{ts}}-2 \tag{42}$$

where

$$\begin{aligned} A^{\mathrm{C}}_{\mathrm{PITA}}(k,i) &= (A^{\mathrm{F}}_{\mathrm{ITA}})^J \quad \text{if } i<k \\ A^{\mathrm{C}}_{\mathrm{PITA}}(k,i) &= B_{k,i+1}(B_{k,i})^{-1} \quad \text{if } i \geqslant k \\ B_{k,i} &= (A^{\mathrm{C}}_{\mathrm{ITA}})^{i-k} \left(\sum_{l=0}^{k} g_{l,k}(i) (A^{\mathrm{F}}_{\mathrm{ITA}})^{Jl} (A^{\mathrm{C}}_{\mathrm{ITA}})^{k-l} \right) \\ g_{l,k}(i) &= (-1)^{l+k} \frac{\binom{k}{l}}{k!} \prod_{j=0, j\neq l}^{k} (i-j) \end{aligned}$$

This result shows that the PITA admits an amplification matrix, and that unlike $A^{\mathrm{F}}_{\mathrm{ITA}}$ and $A^{\mathrm{C}}_{\mathrm{ITA}}$, this amplification matrix depends on the time-instance where it is evaluated. This is a direct consequence of the propagation of the jumps on the coarse time-grid (see Equations (23)). Since the PITA is an iterative algorithm, its amplification matrix $A^{\mathrm{C}}_{\mathrm{PITA}}(k,i)$ also depends, as can be expected, on the iteration where it is evaluated. To gain some insight in the effect of the iterations on $A^{\mathrm{C}}_{\mathrm{PITA}}(k,i)$, we perform an eigenvalue analysis and derive directly from Proposition 5 the following result.

Proposition 6
If the ITA is chosen as the Newmark method with $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ (midpoint rule), the eigenvalues of the amplification matrix of the resulting PITA are given by

$$\lambda(A^{\mathrm{C}}_{\mathrm{PITA}}(k,i)) = \frac{\lambda(A^{\mathrm{C}}_{\mathrm{ITA}})(\sum_{l=0}^{k} g_{l,k}(i+1)(\lambda(A^{\mathrm{F}}_{\mathrm{ITA}}))^{Jl}(\lambda(A^{\mathrm{C}}_{\mathrm{ITA}}))^{k-l})}{\sum_{l=0}^{k} g_{l,k}(i)(\lambda(A^{\mathrm{F}}_{\mathrm{ITA}}))^{Jl}(\lambda(A^{\mathrm{C}}_{\mathrm{ITA}}))^{k-l}} \quad \text{if } i \geqslant k$$

$$\lambda(A^{\mathrm{C}}_{\mathrm{PITA}}(k,i)) = \lambda^{J}(A^{\mathrm{F}}_{\mathrm{ITA}}) \quad \text{if } i>k \tag{43}$$

where $\lambda(A^{\mathrm{C}}_{\mathrm{ITA}})$ and $\lambda(A^{\mathrm{F}}_{\mathrm{ITA}})$ denote the eigenvalues of the amplification matrices of the Newmark method with $\beta=\frac{1}{4}$ and $\gamma=\frac{1}{2}$ on the coarse and fine time-grids, respectively.

It is well-known that for a second-order system, the eigenvalues of the amplification matrix of the Newmark method with $\beta=\frac{1}{4}$ and $\gamma=\frac{1}{2}$ [3] are equal to 1 in absolute value when the system is undamped, and less than 1 in absolute value when the damping coefficient is less than its critical value. In Section 6.2, we show for a sample second-order linear system that $\lambda(A^{\mathrm{C}}_{\mathrm{PITA}}(k,i))$, Equation (43) is only slightly larger than one at $k=1$ and any time-instance in Ω_t, then decreases throughout the iterations to a value which is slightly below one in all Ω_t. This suggests that the effect of the iterations in the PITA is to damp the solution after it is slightly amplified at $k=1$, which is beneficial to numerical stability.

For other choices of the ITA, similar eigenvalue analysis can be performed for both first- and second-order systems.

6. FEASIBILITY STUDIES

The speed-up analysis performed in Section 3 shows that the potential of the PITA presented in this paper depends on its ability to converge in a very few iterations, while using a J ratio that is as large as possible. The former issue depends on the accuracy that can be achieved within a few iterations of the PITA. The latter issue depends on the stability of the ITA of interest and impacts the stability of the PITA. To this effect, both error and stability analyses summarized in Sections 4 and 5, respectively, reveal good accuracy and stability properties of the PITA. However, at this point only detailed studies can assess the feasibility of the PITA and its potential for speeding up applications that are parallelized only in the space-domain. As a first step of this effort, we report here on three different feasibility studies for fluid, structure, and coupled fluid–structure model problems. These studies are aimed at illustrating the behaviour and potential of the PITA, while identifying both the physical and simulation parameters that can make or break the main objective stated in this paper.

6.1. An unsteady flow model problem

We consider the one-dimensional piston problem described in Reference [4] and graphically depicted in Figure 3. We assume that the chamber has a constant cross-sectional area, the gas

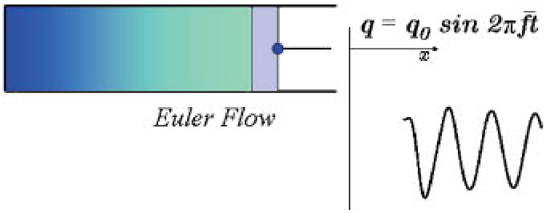

Figure 3. A piston flow problem.

is perfect, and that the flow is isentropic. We denote by l the length of the chamber, p the pressure field, ρ the density field, and a the speed of sound defined by

$$a^2 = \frac{\mathrm{d}p}{\mathrm{d}\rho}$$

We assume that the equilibrium of this system is defined by the uniform pressure $p_0 = 0.701 \times 10^5$ Pa inside and outside the piston chamber, the uniform density $\rho_0 = 0.909\ \mathrm{kg/m^3}$ (these values correspond to an altitude of 3000 m), the uniform flow velocity $u_0 = 0$ m/s, and $l_0 = 2$ m. We also assume that after this state of equilibrium is reached, the displacement q of the piston is forced into the harmonic motion

$$q = q_0 \sin 2\pi \bar{f} t$$

where q_0 denotes the amplitude of this motion and $\bar{f}$ its forcing frequency. Hence, the boundary conditions of this flow problem are

$$\begin{aligned} u(0,t) &= 0 \\ u(l_0 + q, t) &= \frac{\mathrm{d}q}{\mathrm{d}t} = 2\pi \bar{f} q_0 \cos 2\pi \bar{f} t \end{aligned} \tag{44}$$

and its initial conditions are

$$\begin{aligned} \rho(x,0) &= \rho_0 \\ u(x,0) &= u_0 = 0 \end{aligned} \tag{45}$$

As the flow has one moving boundary at $x = l$, we describe it by the arbitrary Lagrangian–Eulerian (ALE) form of the mass and momentum conservation equations

$$\begin{aligned} \frac{1}{\mathscr{J}} \frac{\partial}{\partial t}(\mathscr{J}\rho) + \frac{\partial}{\partial x}\left(\rho\left(u - \frac{\mathrm{d}\xi}{\mathrm{d}t}\right)\right) &= 0 \\ \frac{1}{\mathscr{J}} \frac{\partial}{\partial t}(\mathscr{J}\rho u) + \frac{\partial}{\partial x}\left(\left(\rho u\left(u - \frac{\mathrm{d}\xi}{\mathrm{d}t}\right) + p\right)\right) &= 0 \end{aligned} \tag{46}$$

where $\mathscr{J} = \det(\mathrm{d}x/\mathrm{d}\xi)$ denotes the Jacobian of the frame transformation $x \rightarrow \xi$.

We introduce the vector notation

$$W = \begin{pmatrix} \rho \\ \rho u \end{pmatrix}, \quad \mathscr{F}^{\mathrm{c}} = \begin{pmatrix} \rho\left(u - \dfrac{\mathrm{d}\xi}{\mathrm{d}t}\right) \\ \rho u\left(u - \dfrac{\mathrm{d}\xi}{\mathrm{d}t}\right) + p \end{pmatrix} \tag{47}$$

and re-write Equations (46) in vector form as follows:

$$\frac{1}{\mathscr{J}} \frac{\partial}{\partial t}(\mathscr{J} W) + \frac{\partial}{\partial x}(\mathscr{F}^{\mathrm{c}}(W)) = 0 \tag{48}$$

We semi-discretize the PDE (48) by the finite volume method and 20 cells, determine the values of $\mathrm{d}\xi/\mathrm{d}t$ by the method of transpiration [4], and finally obtain the following ODE problem:

$$\begin{aligned} \frac{\mathrm{d}Vw}{\mathrm{d}t} + F^{\mathrm{c}}(w) &= 0 \\ w(0) &= w^0 \end{aligned} \tag{49}$$

where V denotes the vector of volumes of the cells, and w results from the semi-discretization of W.

We choose as ITA on the fine time-grid the second-order three-point (or two-stage) backward difference scheme because of its accuracy and stability properties, as well as its popularity in computational fluid dynamics (CFD) [5]. Being a three-point scheme, this ITA requires a special initialization. On the coarse time-grid, the solution of problem (23) requires repeating the special initialization at each time-step. To avoid this computational issue, we switch on the coarse time-grid to the two-point variant of this ITA.

We consider two different forcing frequencies, $\bar{f} = 4$ and 10 Hz. These two frequencies delimit the frequency band containing the first dozen aeroelastic modes of many fighters such as the F-16 and F-18/A [6]. These modes usually dominate the aeroelastic behaviour of these aircraft. In each case, we sample the forcing period on the fine time-grid by 25 time-steps, which is more ambitious than usually done in realistic implicit CFD computations. This sampling strategy leads to two different time-steps on the fine time-grid: $\Delta t = 10^{-2}$ s for $\bar{f} = 4$ Hz, and $\Delta t = 4 \times 10^{-3}$ s for $\bar{f} = 10$ Hz. In both cases, we consider two different ratios of coarse and fine time-steps, namely, $J = 30$ and 90. We note that even for the smallest value, $J = 30$, neither of the two forcing periods are sampled on the coarse time-grid by even a single time-step, which is also quite challenging.

We assume that the maximum number of available clusters is $N_{\mathrm{c}} = 20$. This value of N_{c} is quite realistic for CFD applications because these typically involve a large number of dof's in the space-domain, and therefore are characterized by a large value of $N_{\mathrm{p}}^{\mathrm{cr}}$. For simplicity, we set $r = 1$. From Equation (10), it follows that if the PITA is to be used with $30 \leqslant J \leqslant 90$, it must demonstrate stability and accuracy during 72 periods for both $\bar{f} = 4$ and 10 Hz, before its potential for speed-up on such a large number of clusters can be evaluated.

In summary, except for its dimensionality, the unsteady flow model problem described above has many of the characteristics of the vibrating flow encountered in realistic aeroelastic simulations. Furthermore, the chosen values for J and $(t^{\mathrm{f}} - t^0)$ are large enough to test for a fluid flow problem both the stability and long-term accuracy issues addressed in Sections 5 and 4.

The results reported in Figure 4 show that for $\bar{f} = 10$ Hz, and $J = 30$ as well as $J = 90$, the PITA exhibits a remarkable stability and an impressive accuracy. More specifically, after only two iterations ($k = 1$), the PITA delivers the same pressure response as the ITA during all 72 periods. Similar results are obtained for $\bar{f} = 4$ Hz. From Equation (33), it follows that for this unsteady flow model problem for which it can withstand a ratio of coarse and fine time-steps $J = 90$, using 20 clusters, the PITA can deliver a speed-up factor equal to $(1/(2 \times (\frac{1}{20} + \frac{1}{90}))) = 8.2$. This highlights a significant potential of the PITA for reducing the solution time of a class of unsteady CFD applications.

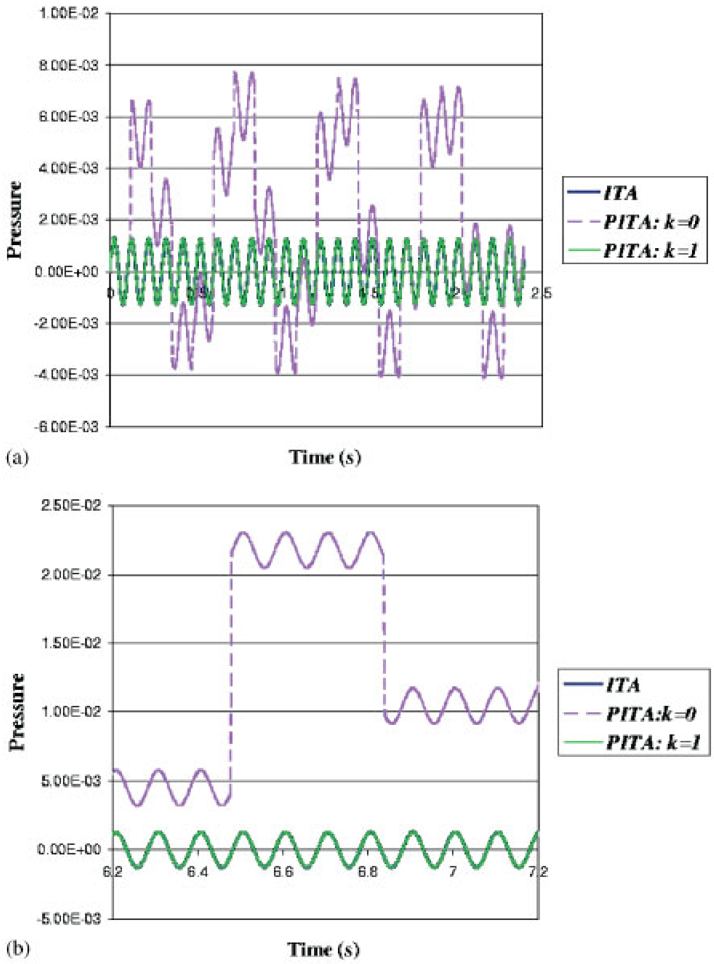

Figure 4. Unsteady flow model problem: PITA vs ITA for $\bar{f}=10$ Hz (a) $J=30$; and (b) $J=90$ (last 10 periods).

To illustrate what can be expected at a higher frequency, we consider a third case with $\bar{f}=50$ Hz. We sample the corresponding period on the fine time-grid also by 25 time-steps, which leads to $\Delta t=8\times 10^{-4}$ s. In this case, the results reported in Figure 5 show that using $J=30$, the PITA requires three iterations ($k=2$) to deliver the same result as the ITA, which lowers the potential speed-up factor to $(1/(3\times(\frac{1}{20}+\frac{1}{30})))=4$. Nevertheless, this speed-up factor can still be interesting for many CFD applications where faster turnaround is sought-after.

6.2. *A structural dynamics model problem*

Next, we consider a vibration problem for the bar shown in Figure 6. This bar has a length $L=1$ m, a cross-sectional area $A=2.5\times 10^{-5}$ m^2, a Young's modulus $E=2.0\times 10^8$ Pa,

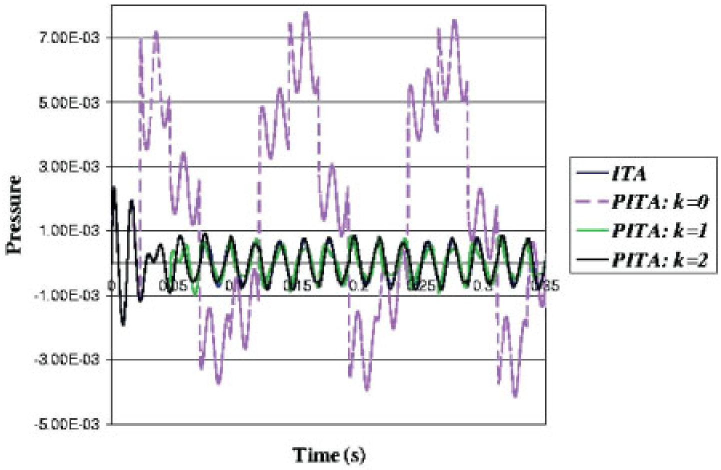

Figure 5. Unsteady flow model problem: PITA vs ITA for $\bar{f} = 50$ Hz and $J = 30$.

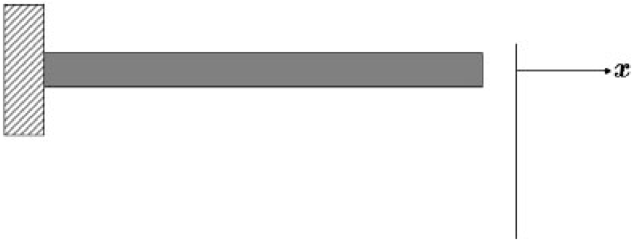

Figure 6. A structural dynamics model problem.

and a mass per unit length $\rho = 19.5$ kg/m. We discretize this bar by 20 linear finite elements and write the corresponding equations of dynamic equilibrium in first-order form as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \dfrac{\mathrm{d}q}{\mathrm{d}t} \\ q \end{pmatrix} + \begin{pmatrix} M^{-1}D & M^{-1}K \\ -I & 0 \end{pmatrix}\begin{pmatrix} \dfrac{\mathrm{d}q}{\mathrm{d}t} \\ q \end{pmatrix} = 0 \tag{50}$$

where q denotes here the vector of displacement dof's, M, D, and K denote the finite element mass, Rayleigh damping, and stiffness matrices, respectively, and I is the identity matrix. More specifically, we construct D as $D = aM + bK$ where a and b are chosen so that the critical damping coefficients of the first two modes of vibrations are both equal to 1%. We note that the first two undamped natural frequencies of this bar, as predicted by the finite element model, are $f_1 = 4$ Hz and $f_2 = 12$ Hz. Again, we remind the reader that our interest in these frequencies is motivated by the fact that they delimit the frequency band containing the first dozen aeroelastic modes of many fighters such as the F-16 and F-18/A [6].

We use the notation of Equation (3) and set the initial condition to

$$y(0) = 0.5X_1 + X_2 \tag{51}$$

where X_1 and X_2 are the M-normalized mode shapes associated with the frequencies f_1 and f_2.

For solving numerically the problem defined by Equations (50) and (51), we choose as ITA on both the fine and coarse time-grids the Newmark method with $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$. This scheme is second-order accurate, unconditionally stable, and popular in the structural dynamics community. We set the time step to $\Delta t = T_2/m$, where $T_2 = 1/f_2$ denotes the period of the second natural vibration mode. We vary m between 25 and 200 to consider two extreme types of structural dynamics applications: (a) low-frequency dynamic problems for which sampling T_2 by 25 time-steps is reasonable when using an unconditionally stable time-integrator, and (b) non-linear aeroelastic simulations of full fighter configurations for which, even when the midpoint rule is used to time-integrate Equations (50), sampling T_2 by 100–200 time-steps is often required to properly address important issues related to the fluid and moving fluid grid components of non-linear coupled aeroelastic formulations [7]. We note that in the latter case, implicit time-integration remains far more competitive than explicit time-integration. We consider two values of the ratio of coarse and fine time-steps, namely, $J = 10$ and 20. For $J = 10$ and $m = 25$, T_2 is sampled on the coarse time-grid by 2.5 time-steps only. For $J = 10$ and $m = 200$, T_2 is sampled on the coarse time-grid by 20 time-steps, and for $J = 20$ and $m = 200$, it is sampled by 10 time-steps.

The results reported in Figure 7 show that for $J = 10$, the PITA requires five iterations ($k = 4$) before it can deliver the same accuracy as the ITA during two cycles of the response when $m = 25$, and during three cycles of the response when $m = 50$. Those reported in Figure 8 show that also for $J = 10$, the PITA requires four iterations ($k = 3$) to reproduce the same result as the ITA during five cycles of the response when $m = 100$, but only two iterations ($k = 1$) to deliver the same accuracy as the ITA during six cycles of the response when $m = 200$. However, after two iterations, the PITA reproduces the same result as the ITA during only two cycles when $m = 200$ and $J = 20$ (see Figure 9). All these results indicate that, unlike for the fluid flow model problem of Section 6.1, for this structural dynamics model problem, the size of the time-step used on the coarse time-grid, which depends on m and J, governs the performance of the PITA. The larger this time-step and the larger the total simulation time, the more iterations the PITA needs to perform to deliver the same accuracy as the underlying ITA. As a result, using $N_c = 20$ clusters, the PITA can potentially achieve a speed-up factor ranging between 2.25 and 3.25 when m is varied between 50 and 100 and $J = 10$, and a speed-up factor ranging between 3.25 and 5 when m is varied between 100 and 200 and $J = 20$ (see Figure 10).

In summary, the above results indicate that while the PITA can deliver a decent speed-up factor for structural dynamics applications characterized by a large value of m, it does not appear to have in general the same potential for reducing the total solution time of structural dynamics applications as it has for CFD applications. To explain this difference in behaviour between CFD and structural dynamics problems, we report in Figure 11 the time-history of the solution on the coarse time-grid of the correction problem (23). This figure shows that at each iteration k, c_k exhibits the same frequency as the jump Δ_k which drives problem (23). Furthermore, this frequency is the same throughout the iterations and is equal to 10.4 Hz. Hence, it is close to the frequency of the second mode, $f_2 = 12$ Hz, which explains the onset

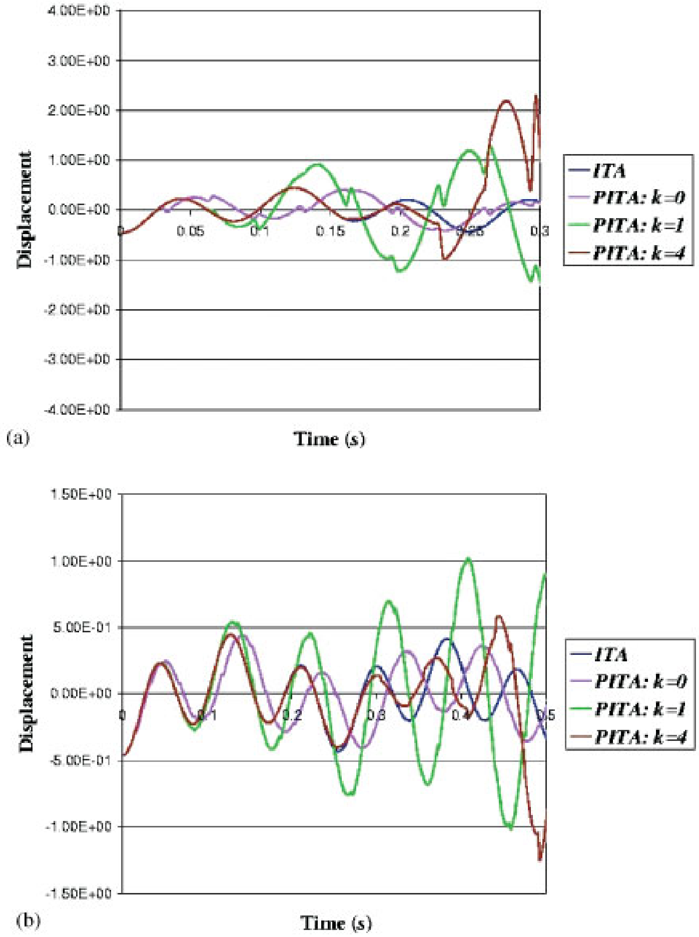

Figure 7. Structural dynamics model problem: PITA vs ITA for $J=10$: and (a) $m=25$; and (b) $m=50$.

of the 'beating' phenomenon observed in Figure 11. As shown in Figure 7(b), the beating is transmitted to the solution by PITA on the fine time-grid of the main problem by the update of the seed information (see Equation (17)). It does not affect the stability of the PITA, but its accuracy. Indeed, during a beating phenomenon involving a circular frequency $\bar{\omega}$ and a neighbouring one ω^*, the response of the system is characterized by both circular frequencies $(\bar{\omega}+\omega^*)/2$ and $(\bar{\omega}-\omega^*)/2$. A time-integrator is usually dispersive and therefore approximates $\bar{\omega}$ by $\bar{\omega}+\bar{\varepsilon}$, and ω^* by $\omega^*+\varepsilon^*$. In the case of the midpoint rule, $\bar{\varepsilon}=O(\Delta t^2)$ and $\varepsilon^*=O(\Delta t^2)$. Hence, a time-integrator approximates $(\bar{\omega}+\omega^*)/2$ by $(\bar{\omega}+\omega^*)/2+(\bar{\varepsilon}+\varepsilon^*)/2$, and $(\bar{\omega}-\omega^*)/2$ by $(\bar{\omega}-\omega^*)/2+(\bar{\varepsilon}-\varepsilon^*)/2$. When $\omega^* \to \bar{\omega}$, $(\bar{\omega}-\omega^*)$ becomes of the same order as the difference of errors $(\bar{\varepsilon}-\varepsilon^*)/2$, and therefore this circular frequency cannot be resolved accurately by a time-integrator unless Δt is made sufficiently small. This explains for the structural dynamics model problem discussed in this section the performance improvement

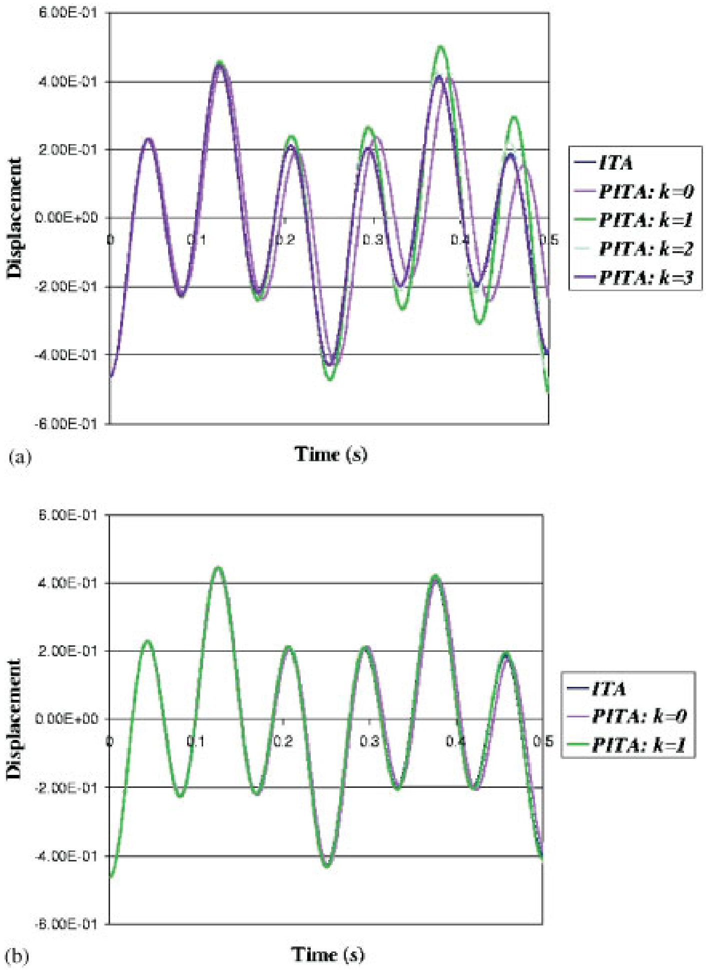

Figure 8. Structural dynamics model problem: PITA vs ITA for $J=10$ and: (a) $m=100$; and (b) $m=200$.

of the PITA observed when m is increased. For undamped systems, the beating phenomenon becomes a resonance phenomenon which can break the performance of the PITA. Given that both beating and resonance problems can happen only for second-order systems, this also explains why the PITA exhibits a better performance for the unsteady flow model problem of Section 6.1 than for the structural dynamics model problem discussed herein.

We also report in Figures 12 and 13 the time-histories of the first two eigenvalues of the amplification matrix of the PITA on the coarse time-grid, $A^{\mathrm{C}}_{\mathrm{PITA}}$, for $m=25$ and 100. Our focus on these two eigenvalues is justified by the fact that problem (50) is driven in Equation (51) by the first two natural modes of the bar. The reader can observe that, except for λ_2 and the case $m=25$, these eigenvalues reach initially a maximum that is slightly larger

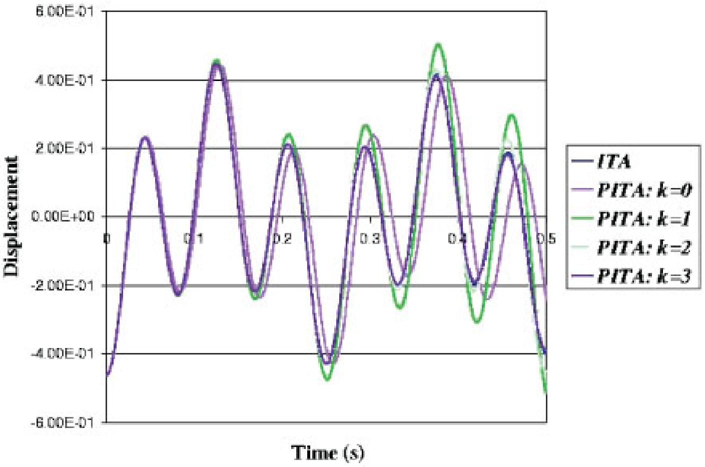

Figure 9. Structural dynamics model problem: PITA vs ITA for $m = 200$ and $J = 20$.

than one, then are attenuated throughout the PITA iterations to values that are slightly below one. Hence, these results highlight the rather good stability properties of the PITA for a given unconditionally stable ITA.

6.3. A dynamic fluid–structure interaction model problem

Finally, we consider again the piston described in Section 6.1. We assume that this piston is rigid, and denote by $\mathscr{M}$ and $\mathscr{K}$ its mass and stiffness. We set these values to $\mathscr{M} = 50$ kg and $\mathscr{K} = 10^5$ N/m so that the natural 'dry' frequency of this piston is 7.12 Hz. This frequency corresponds to the first dry torsional mode of an F-16 fighter [6]. Unlike in Section 6.1, we do not force the motion of this piston but represent it by the first-order form of its governing equation of dynamic equilibrium

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\frac{\mathrm{d}q}{\mathrm{d}t}\\ q\end{pmatrix} + \begin{pmatrix}0 & \mathscr{M}^{-1}\mathscr{K}\\ -I & 0\end{pmatrix}\begin{pmatrix}\frac{\mathrm{d}q}{\mathrm{d}t}\\ q\end{pmatrix} = \begin{pmatrix}p(l_0+q)-p_0\\ 0\end{pmatrix} \tag{52}$$

where q denotes again the displacement of the piston in the x direction, and all remaining variables have the same meaning as in Section 6.1. We allow the piston to freely interact with the fluid inside the chamber and therefore re-write Equations (44) as follows:

$$\begin{aligned} u(0,t) &= 0\\ u(l_0+q,t) &= \frac{\mathrm{d}q}{\mathrm{d}t} \end{aligned} \tag{53}$$

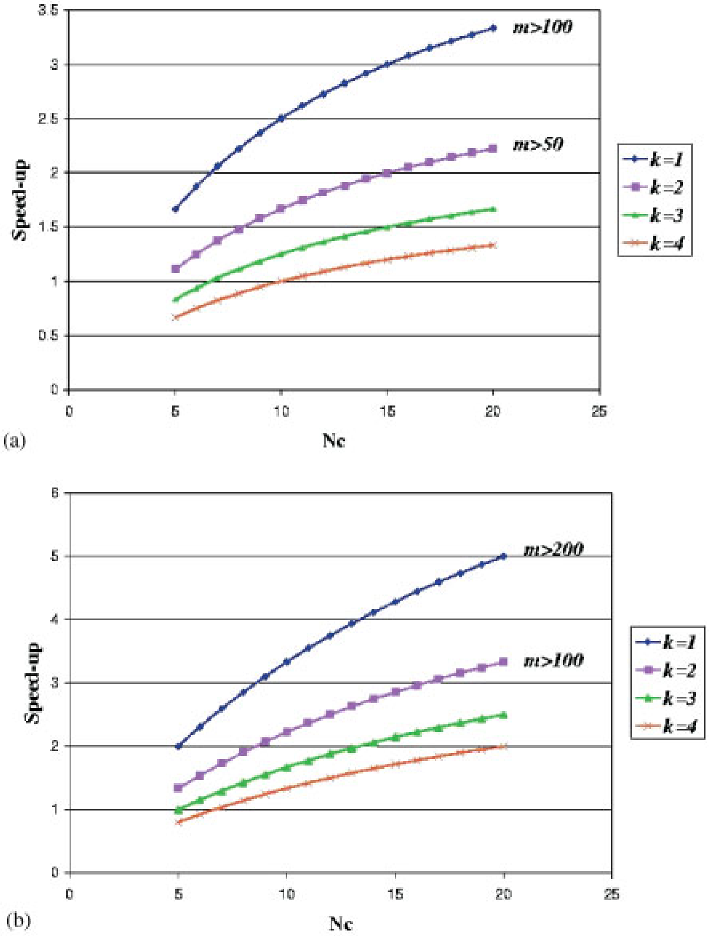

Figure 10. Structural dynamics model problem: potential speed-up of PITA for: (a) $J=10$; and (b) $J=20$.

As a model problem of computational aeroelasticity, we consider predicting the response of the coupled fluid–structure system to the small perturbation

$$\delta q(0) = -0.03\,\mathrm{m} \tag{54}$$

around the equilibrium position (ρ_0, u_0, c_0) defined in Section 6.1. For this purpose, we follow the procedure described in detail in Reference [4] to: (a) linearize Equation (48) around this position, (b) semi-discretize the resulting linear system by the finite volume method and 20 cells, and (c) use the method of transpiration to eliminate the $\mathrm{d}\xi/\mathrm{d}t$ unknowns. This transforms the coupled set of non-linear equations (48) and (52), the transmission conditions (53), and

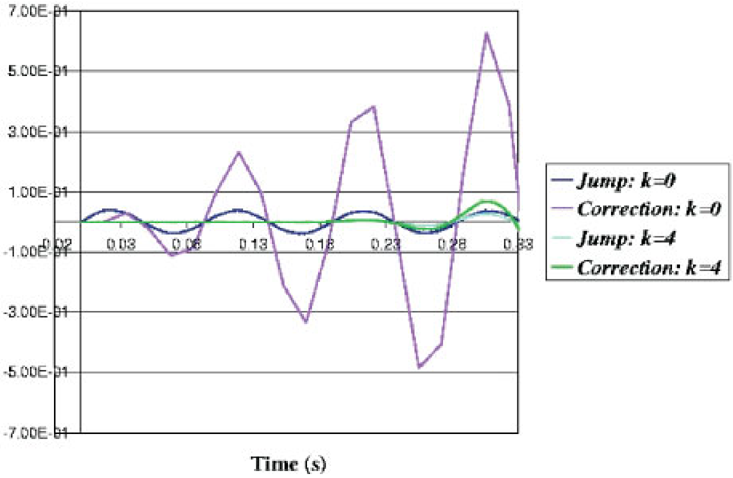

Figure 11. Structural dynamics model problem: $m = 50$ and $J = 10$—potential for resonance.

the initial conditions (45) and (54) into the following coupled system of ODEs:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\delta w\\ \delta y\end{pmatrix}=\begin{pmatrix}F_{\mathrm{FF}} & F_{\mathrm{FS}}\\ F_{\mathrm{SF}} & F_{\mathrm{FF}}\end{pmatrix}\begin{pmatrix}\delta w\\ \delta y\end{pmatrix}$$

$$\begin{pmatrix}\delta w\\ \delta y\end{pmatrix}=\begin{pmatrix}\delta w\\ \delta y\end{pmatrix}^{0} \tag{55}$$

where y has the same structure as in Equation (3), δy and δw are the perturbations of y and w around the aeroelastic equilibrium position of the piston, respectively, the subscripts F and S designate the fluid and structure subsystems, respectively, and the expressions of F_{FF}, F_{FS}, F_{SF}, and F_{FF} can be found in Reference [4].

For solving the above coupled problem (55) by the PITA, we choose as ITA on both the fine and coarse time-grids the partitioned (or staggered) algorithm described in Reference [8]. This algorithm couples the three-point backward difference scheme for time-advancing the solution of the fluid subsystem with the midpoint rule for time-advancing that of the structural subsystem, in a specific manner that maximizes accuracy and numerical stability. For the same reason specified in Section 6.1, we replace, however, on the coarse time-grid the three-point backward difference scheme by its two-point variant.

In References [6, 7], it is reported that a typical time-step for the transient non-linear aeroelastic simulation of an F-16 fighter using the partitioned algorithm outlined above varies between 0.5 and 1 ms. This time-step is still several orders of magnitude larger than the time-step dictated by the stability requirement of an explicit time-integration strategy. The values of 0.5 and 1 ms correspond to sampling the natural frequency of the piston ($f = 1/T = 7.12$ Hz) considered herein by 140 and 280 points. Hence, we set here the time-step to $\Delta t = T/m$ with $140 \leqslant m \leqslant 280$. Since the results reported in Sections 6.1 and 6.2 suggest that second-order structural problems challenge more the PITA than first-order fluid problems, we consider here the same values for the ratio of coarse and fine time-steps as in Section 6.2, namely, $J = 10$ and 20.

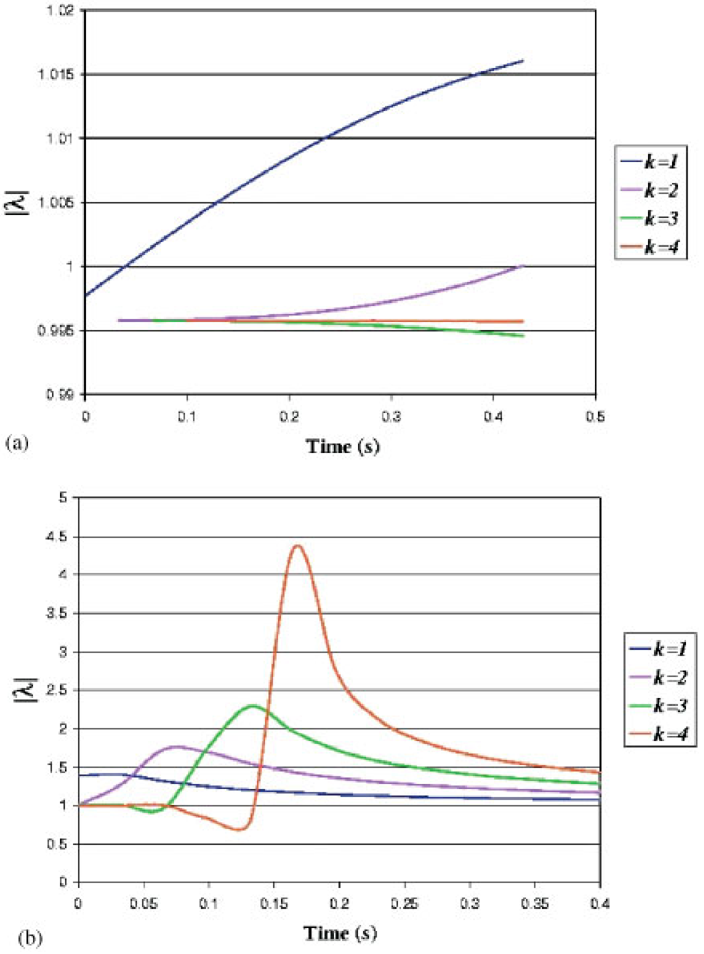

Figure 12. Structural dynamics model problem: first two eignevalues of the amplification matrix $A^{\mathrm{C}}_{\mathrm{PITA}}$ for $m=25$ and: (a) λ_1; and (b) λ_2.

The results reported in Figure 14 show that for $m=140$, the PITA requires three iterations ($k=2$) before it can deliver the same accuracy as the ITA during four cycles of the response when $J=10$, and five iterations ($k=4$) before it can deliver that accuracy during only two cycles when $J=20$. For $m=280$, the performance of the PITA is improved. Figure 15 shows that in this case, the PITA requires only two iterations ($k=1$) to reproduce the same result as the ITA during four cycles of the response when $J=10$, and three iterations ($k=2$) when $J=20$.

In summary, the performance results implied by Figures 14 and 15 suggest that, as it can be expected, the potential of the PITA for speeding up the simulation of fluid–structure interaction problems is governed by its potential for accelerating the solution of structural dynamics problems. Hence, this potential is governed by the same parameters and issues discussed in Section 6.2.

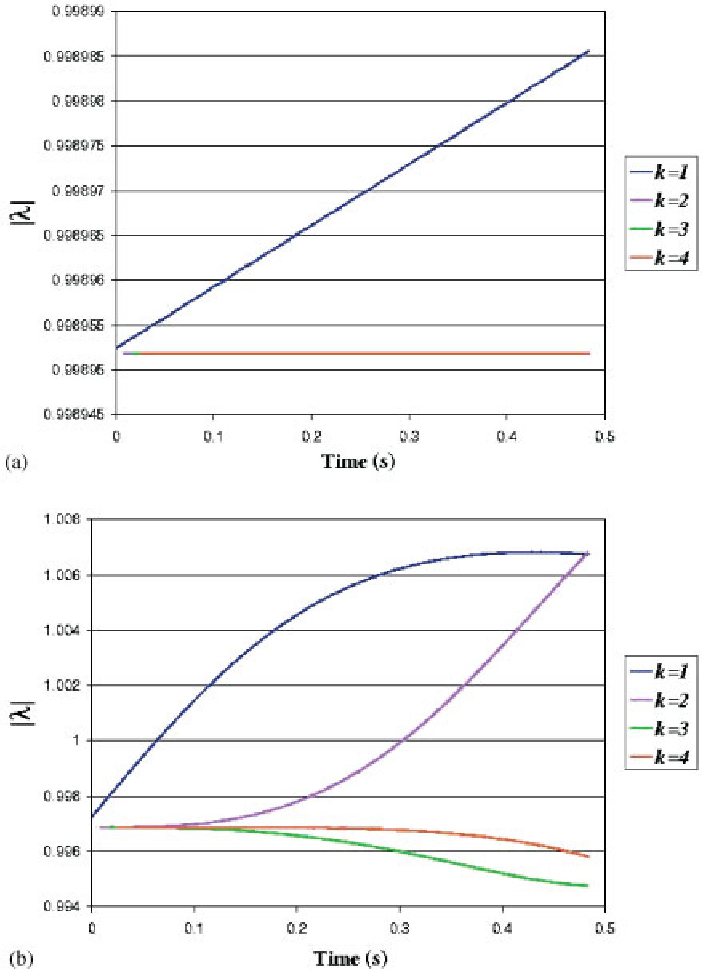

Figure 13. Structural dynamics model problem: first two eigenvalues of the amplification matrix $A^{\mathrm{C}}_{\mathrm{PITA}}$ for $m = 100$ and: (a) λ_1; and (b) λ_2.

7. CONCLUSIONS

We have presented a methodology for reducing the total solution time on a massively parallel processor of time-dependent PDE problems below what is achievable by current PDE solvers featuring parallel processing only in the space-domain. This methodology, which addresses so far only implicit time-integration schemes, relies on parallelizing the time-loop of a given PDE solver independently from the parallelization of its space-loop, by transforming its serial time-integrator into a parallel one. This transformation is based on the parareal concept outlined in Reference [1], which consists of decomposing the time-domain into time-slices, carrying out independent time-integrations in the time-slices, and as shown in this paper, correcting

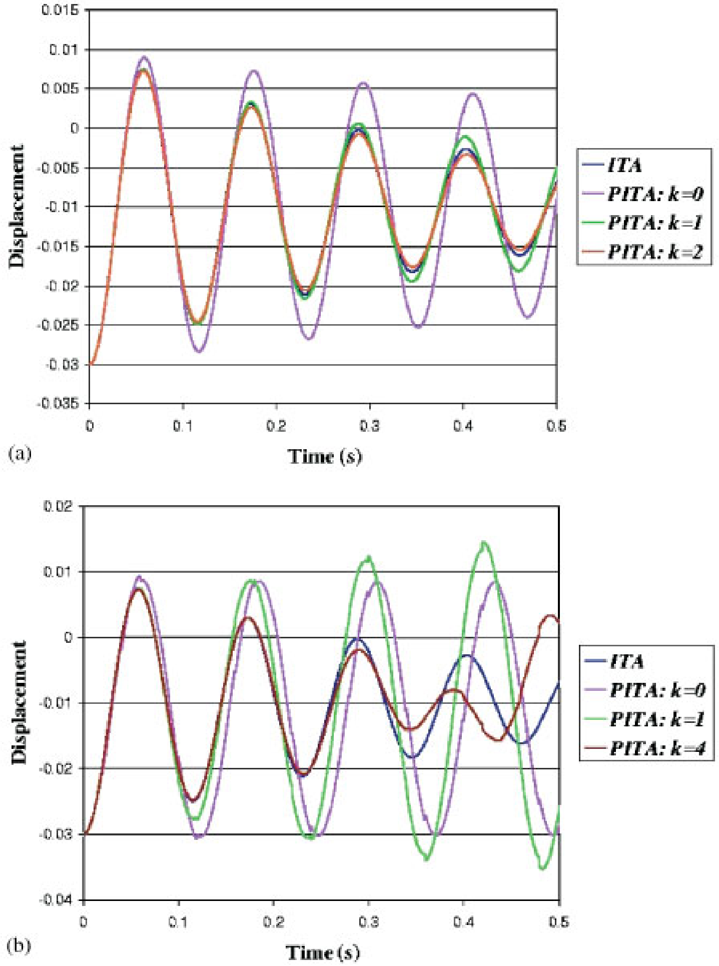

Figure 14. Fluid–structure interaction model problem: $m = 140$ and: (a) $J = 10$; and $J = 20$.

their outcome by Newton-like iterations on a coarse time-grid. We have performed error and stability analyses of this methodology that legitimize it and provide further insight into its key mechanisms and governing parameters. Using suitable model problems, we have also investigated its feasibility and potential for first-order fluid problems, second-order structural problems, and coupled fluid–structure interaction problems. The obtained results highlight an excellent potential of the described methodology for accelerating long-term simulations of first-order fluid problems and coupled fluid–structure interaction problems on massively parallel processors. For second-order structural dynamics problems, the proposed parallel methodology generates a parasitic beating phenomenon which limits its potential to shorter-term simulations requiring a time-step that is sufficiently small compared to the natural period of the dominating mode, and yet sufficiently large to warrant an implicit time-integration. Extending the scope of this potential to a larger class of long- and short-term simulations of second-order systems

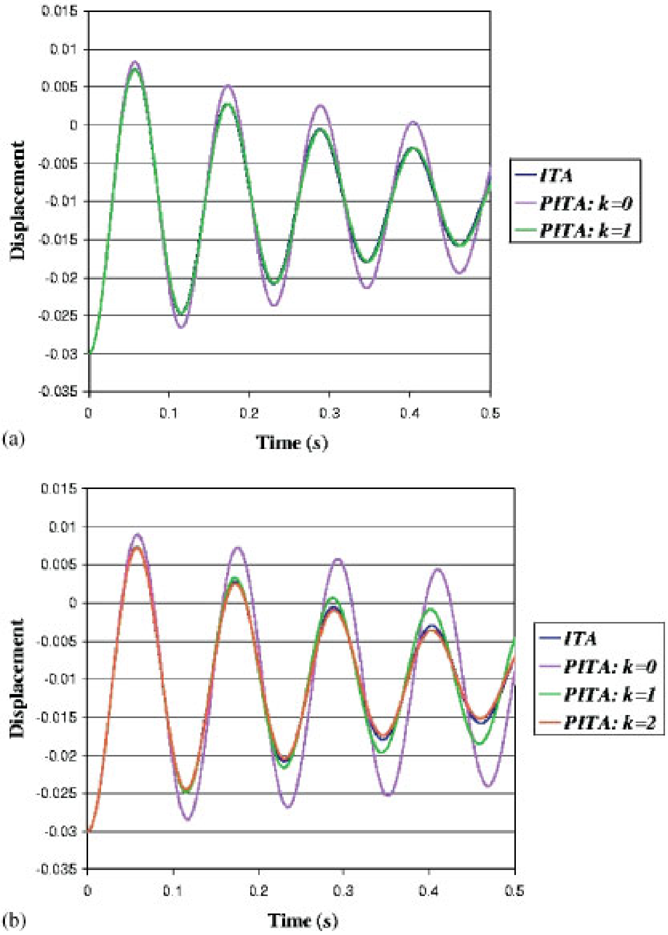

Figure 15. Fluid–structure interaction model problem: $m = 280$ and: (a) $J = 10$; and (b) $J = 20$.

appears to require addressing the parasitic beating phenomenon, which is the objective of our future work in this area.

APPENDIX A

Here, we prove the propositions presented in Sections 4 and 5. We remind the reader that these propositions are for the case where $F(y,t)$ is a linear function of y, and therefore the target problem (2) is linear. We begin by pointing out that from Equations (12) and (16), and the notation introduced in Section 2.2:

- Equation (15) can be re-written as

$$\Delta_k^i = y_k^{Ji-} - y_k^{Ji+}, \quad 1 \leqslant i \leqslant N_{\mathrm{ts}} - 1 \tag{A1}$$

- Equation (17) can be re-written as

$$Y_{k+1}^{i} = y_{k}^{Ji-} + C_{k}^{i} \tag{A2}$$

- The description of the PITA given in Section 2.2 and the first of Equations (41) imply

$$\forall 0 \leqslant i \leqslant N_{\text{ts}} - 1, \quad \begin{cases} y_{k}^{Ji+j+1} = A_{\text{ITA}}^{\text{F}} y_{k}^{Ji+j}, & 0 \leqslant j \leqslant J-1 \\ y_{k}^{Ji+} = Y_{k}^{i} \end{cases} \tag{A3}$$

- The solution of problem (23) implies

$$\forall 0 \leqslant i \leqslant N_{\text{ts}} - 1, \quad \begin{cases} C_{k}^{i+1} = A_{\text{ITA}}^{\text{C}} (C_{k}^{i} + \Delta_{k}^{i}) \\ C_{k}^{0} = 0 \end{cases} \tag{A4}$$

Appendix A.1

Proposition 1
At each iteration k, the jumps, coarse time-grid corrections, and updates of the seed values generated by the PITA satisfy

$$\Delta_{k}^{i} = 0, \quad 0 \leqslant i \leqslant k$$

$$C_{k}^{i} = 0, \quad 0 \leqslant i \leqslant k+1$$

$$Y_{k+1}^{i} = y_{\text{ITA}}^{Ji}, \quad 1 \leqslant i \leqslant k+1$$

Proof
We prove this proposition by induction.

By construction,

$$\Delta_{0}^{0} = 0, \quad C_{0}^{0} = 0, \quad \text{and} \quad y_{0}^{J-} = y_{\text{ITA}}^{J} \tag{A5}$$

From Equations (16) and (23), it follows that

$$C_{0}^{1} = 0 \tag{A6}$$

and from Equation (A2), it follows that

$$Y_{1}^{1} = y_{\text{ITA}}^{J} \tag{A7}$$

Hence, the proposition is verified for $k = 0$.

Now, suppose that it is verified until the kth iteration. Since the seeds satisfy

$$Y_{k+1}^{i} = y_{\text{ITA}}^{Ji}, \quad 1 \leqslant i \leqslant k+1 \tag{A8}$$

the ITA and PITA generate the same numerical solution for $0 \leqslant t \leqslant T^{(k+2)-}$, which implies that the solution computed by the PITA does not exhibit jumps in this time-interval, and therefore

$$\Delta_{k+1}^{i} = 0, \quad 0 \leqslant i \leqslant k+1 \tag{A9}$$

From Equations (16), (23) and (A2) it follows that

$$C_{k+1}^{i}=0, \quad 0\leqslant i\leqslant k+2$$
$$Y_{k+2}^{i}=y_{\mathrm{ITA}}^{Ji}, \quad 1\leqslant i\leqslant k+2$$

□

APPENDIX B

Appendix B.1

Proposition 2

Given a p-order ITA that is stable on both the fine and coarse time-grids, there exists a constant a_k such that for all $1\leqslant i\leqslant N_{\mathrm{ts}}-1$,

$$|E_k^i|\leqslant a_k \max(\Delta T^{(k+1)p}, \Delta t^p) \tag{B1}$$

Proof

We prove this proposition by induction.

If the initial seed values are generated by the ITA on the coarse time-grid, the proposition is verified for $k=0$.

Now, assume that this proposition is verified until the kth iteration. From Equations (A1)–(A4), and noting that by construction of the PITA

$$y_k^{J(i+1)-}=A_{\mathrm{ITA}}^{\mathrm{F}^J}Y_k^i \tag{B2}$$

it follows that the seed values verify

$$Y_{k+1}^i=A_{\mathrm{ITA}}^{\mathrm{F}^J}Y_k^{i-1}+\sum_{l=1}^{i-1}A_{\mathrm{ITA}}^{\mathrm{C}^{i-l}}(A_{\mathrm{ITA}}^{\mathrm{F}^J}Y_k^{l-1}-Y_k^l), \quad 2\leqslant i\leqslant N_{\mathrm{ts}}-1, \quad k\geqslant 0 \tag{B3}$$

(we address later the case $i=1$). From the definition of E_k^i given in Equation (34) and after some algebraic manipulations, it follows that

$$E_{k+1}^i=\sum_{l=1}^{i}A_{\mathrm{ITA}}^{\mathrm{C}^{i-l}}A_{\mathrm{ITA}}^{\mathrm{F}^J}E_k^{l-1}-\sum_{l=1}^{i-1}A_{\mathrm{ITA}}^{\mathrm{C}^{i-l}}E_k^l+R, \quad 2\leqslant i\leqslant N_{\mathrm{ts}}-1, \quad k\geqslant 0 \tag{B4}$$

where

$$R=\sum_{l=1}^{i-1}A_{\mathrm{ITA}}^{\mathrm{C}^{i-l}}(A_{\mathrm{ITA}}^{\mathrm{F}^J}y_{\mathrm{ex}}(T^{l-1})-y_{\mathrm{ex}}(T^l)), \quad 2\leqslant i\leqslant N_{\mathrm{ts}}-1 \tag{B5}$$

Since $E_k^0=0$, Equation (B4) can be re-written as

$$E_{k+1}^i=\sum_{l=1}^{i-1}A_{\mathrm{ITA}}^{\mathrm{C}^{i-l-1}}(A_{\mathrm{ITA}}^{\mathrm{F}^J}-A_{\mathrm{ITA}}^{\mathrm{C}})E_k^{l-1}+R, \quad 2\leqslant i\leqslant N_{\mathrm{ts}}-1, \quad k\geqslant 0 \tag{B6}$$

and therefore

$$|E_{k+1}^i|\leqslant\sum_{l=1}^{i-1}\|A_{\mathrm{ITA}}^{\mathrm{C}^{i-l-1}}\|\,\|(A_{\mathrm{ITA}}^{\mathrm{F}^J}-A_{\mathrm{ITA}}^{\mathrm{C}})\||E_k^{l-1}|+|R| \tag{B7}$$

To bound $\|(A^{\mathrm{F}^J}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}})\|$, we note that for all i,

$$|(A^{\mathrm{F}^J}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}})y_{\mathrm{ex}}(T^i)| \leqslant |A^{\mathrm{F}^J}_{\mathrm{ITA}}y_{\mathrm{ex}}(T^i) - y_{\mathrm{ex}}(T^{i+1})| + |A^{\mathrm{C}}_{\mathrm{ITA}}y_{\mathrm{ex}}(T^i) - y_{\mathrm{ex}}(T^{i+1})| \tag{B8}$$

If the ITA is p-order accurate and stable on both the fine and coarse time-grids, there exist two constants θ_1 and θ_2 associated with its accuracy on the coarse and fine time-grids, respectively, and a third constant θ_3 associated with its stability on the fine time-grid, such that

$$|A^{\mathrm{F}^J}_{\mathrm{ITA}}y_{\mathrm{ex}}(T^i) - y_{\mathrm{ex}}(T^{i+1})| + |A^{\mathrm{C}}_{\mathrm{ITA}}y_{\mathrm{ex}}(T^i) - y_{\mathrm{ex}}(T^{i+1})| \leqslant J\theta_2\theta_3\,\Delta t^{p+1} + \theta_1\,\Delta T^{p+1} \tag{B9}$$

From the inequality (B8) and the inequality (B9), it follows that

$$\|(A^{\mathrm{F}^J}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}})\| \leqslant J\theta_2\theta_3\,\Delta t^{p+1} + \theta_1\,\Delta T^{p+1} \tag{B10}$$

From Equation (B5), the fact that $i \leqslant N_{\mathrm{ts}}$, and the assumed accuracy and stability properties of the ITA on both the fine and coarse time-grids, it also follows that

$$|R| \leqslant \theta_4\theta_2\theta_3 N_{\mathrm{ts}} J\,\Delta t^{p+1} \tag{B11}$$

where θ_4 is the constant associated with the stability of the ITA on the coarse time-grid.

From Equations (B10) and (B11), the assumption of the recurrence, the stability of the ITA on the coarse time-grid, the fact that $i \leqslant N_{\mathrm{ts}}$, and recalling that $t^{\mathrm{f}} = N_{\mathrm{ts}}J\Delta t$, it also follows that:

- If $\Delta t^p \leqslant \Delta T^{(k+1)p}$, then $\forall 2 \leqslant i \leqslant N_{\mathrm{ts}} - 1$,

$$\begin{aligned} |E^i_{k+1}| &\leqslant \theta_4 N_{\mathrm{ts}}(\theta_1\,\Delta T^{p+1} + J\theta_3\theta_2\,\Delta t^{p+1})a_k\,\Delta T^{(k+1)p} + \theta_4\theta_2\theta_3 N_{\mathrm{ts}} J\,\Delta t^{p+1} \\ &\leqslant \theta_4 t^{\mathrm{f}}(\theta_1 + \theta_2\theta_3)a_k\,\Delta T^{(k+2)p} + t^{\mathrm{f}}\theta_4\theta_3\theta_2\,\Delta t^p \end{aligned} \tag{B12}$$

  Defining $a_{k+1} = t^{\mathrm{f}}\theta_A((\theta_1 + \theta_3\theta_2)a_k + \theta_3\theta_2)$ concludes in this case the proof of Proposition 2.
- If $\Delta t^p > \Delta T^{(k+1)p}$, then $\forall 2 \leqslant i \leqslant N_{\mathrm{ts}} - 1$,

$$\begin{aligned} |E^i_{k+1}| &\leqslant \theta_4 N_{\mathrm{ts}}(\theta_1\,\Delta T^{p+1} + J\theta_3\theta_2\,\Delta t^{p+1})a_k\,\Delta t^p + \theta_4\theta_3\theta_2 N_{\mathrm{ts}} J\,\Delta t^{p+1} \\ &\leqslant \theta_4 t^{\mathrm{f}}(\theta_1 + \theta_2\theta_3)a_k\,\Delta t^p + t^{\mathrm{f}}\theta_4\theta_3\theta_2\,\Delta t^p \end{aligned} \tag{B13}$$

  Here again, defining $a_{k+1} = t^{\mathrm{f}}\theta_4((\theta_1 + \theta_3\theta_2)a_k + \theta_3\theta_2)$ concludes in this case the proof of Proposition 2.

Finally, for the case $i = 1$, we have

$$E^1_{k+1} = y_{\mathrm{ex}}(T^1) - A^{\mathrm{F}^J}_{\mathrm{ITA}}y^0, \quad k \geqslant 0 \tag{B14}$$

where y^0 is the initial condition of problem (2). From the assumed accuracy and stability properties of the ITA, and assuming that $\Delta T = J\,\Delta t \leqslant 1$, it follows that

$$|E^1_{k+1}| \leqslant \theta_2\theta_3\Delta t^p, \quad k \geqslant 0 \tag{B15}$$

□

Appendix B.2

Proposition 3
Given a p-order ITA that is stable on both the fine and coarse time-grids, there exists a constant b_k such that for all $1 \leqslant i \leqslant N_{\text{ts}} - 1$,

$$\|e_k^i(\ )\|_\infty \leqslant b_k \max(\Delta T^{(k+1)p}, \Delta t^p) \tag{B16}$$

Proof
From Equations (36) and (A3) and after some algebraic manipulations, it follows that for $1 \leqslant i \leqslant N_{\text{ts}} - 1$ and $0 \leqslant j \leqslant J$,

$$e_k^i(t^{Ji+j}) = A_{\text{ITA}}^{\text{F}^j}(y_{\text{ex}}(t^{Ji}) - y_k^{Ji}) + \sum_{l=1}^{j} A_{\text{ITA}}^{\text{F}^{j-l}}(A_{\text{ITA}}^{\text{F}} y_{\text{ex}}(t^{Ji+l-1}) - y_{\text{ex}}(t^{Ji+l})) \tag{B17}$$

From the above equation, the result of Proposition 2, and the assumed accuracy and stability properties of the ITA, it follows that

$$\forall 1 \leqslant i \leqslant N_{\text{ts}} - 1, \quad \forall 0 \leqslant j \leqslant J \tag{B18}$$

$$\begin{aligned} |e_k(t^{Ji+j})| &\leqslant \theta_3 |E_k^i| + J\theta_2\theta_3\,\Delta t^{p+1} \\ &\leqslant \theta_3 a_k \max(\Delta T^{(k+1)p}, \Delta t^p) + \theta_2\theta_3\,\Delta t^p\,\Delta T \end{aligned} \tag{B19}$$

where θ_2 and θ_3 are the constants introduced in the proof of Proposition 2. If $\Delta T < 1$, then $\Delta t^p \Delta T$ is bounded by Δt^p and $b_k = \theta_3 a_k + \theta_2\theta_3$. □

Appendix B.3

Proposition 4
Given a second-order ITA that is stable on both the fine and coarse time-grids, there exists a constant d_k such that for all $1 \leqslant i \leqslant N_{\text{ts}} - 1$,

$$\|\bar{e}_k^i(\ )\|_\infty \leqslant d_k i^{k+1}\,\Delta T^{3(k+1)} \tag{B20}$$

Proof
From the definition of $\bar{e}_k^i(\ )$ (39), the first of Equations (41), and Equation (A3), it follows that for all $1 \leqslant i \leqslant N_{\text{ts}} - 1$,

$$\bar{e}_k^i(t^{Ji+j}) = A_{\text{ITA}}^{\text{F}^j}(Y_k^i - y_{\text{ITA}}^{Ji}) \quad 0 \leqslant j \leqslant J \tag{B21}$$

Let

$$\bar{E}_k^i = Y_k^i - y_{\text{ITA}}^{Ji} \tag{B22}$$

denote the discrepancy error on the coarse time-grid. Since the ITA is assumed to be stable on the fine time-grid, it suffices to show, in order to prove this proposition, that there exists a constant $\tilde{d}_k$ such that for all $1 \leqslant i \leqslant N_{\text{ts}} - 1$,

$$|\bar{E}_k^i| \leqslant \tilde{d}_k (i-1)^k \Delta T^{3(k+1)} \tag{B23}$$

To this effect, we first note that starting from Equation (B3), it can be shown after some algebraic transformations that:

$$Y^i_{k+1} = A^{\mathrm{C}^{i-1}}_{\mathrm{ITA}} A^{\mathrm{F}^J}_{\mathrm{ITA}} y^0 + \sum_{l=1}^{i-1} A^{\mathrm{C}^{i-1-l}}_{\mathrm{ITA}} (A^{\mathrm{F}^J}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}}) Y^l_k, \quad 2 \leqslant i \leqslant N_{\mathrm{ts}} - 1, \quad k \geqslant 0 \tag{B24}$$

$$A^{\mathrm{C}^{i-1}}_{\mathrm{ITA}} A^{\mathrm{F}^J}_{\mathrm{ITA}} y^0 - y^{Ji}_{\mathrm{ITA}} = -\sum_{l=1}^{i-1} A^{\mathrm{C}^{i-1-l}}_{\mathrm{ITA}} (A^{\mathrm{F}^J}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}}) y^{Jl}_{\mathrm{ITA}}, \quad 2 \leqslant i \leqslant N_{\mathrm{ts}} - 1 \tag{B25}$$

$$\bar{E}^i_{k+1} = \sum_{l=1}^{i-1} A^{\mathrm{C}^{i-1-l}}_{\mathrm{ITA}} (A^{\mathrm{F}^J}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}})(Y^l_k - y^{Jl}_{\mathrm{ITA}}), \quad 2 \leqslant i \leqslant N_{\mathrm{ts}} - 1, \quad k \geqslant 0 \tag{B26}$$

Next, we prove the error estimate (B23) by induction. For $k = 0$, we note that Equation (B22) implies

$$|\bar{E}^i_0| \leqslant \|(A^{\mathrm{C}^i}_{\mathrm{ITA}} - A^{\mathrm{F}^{Ji}}_{\mathrm{ITA}})\| \, |y^0|, \quad 1 \leqslant i \leqslant N_{\mathrm{ts}} - 1 \tag{B27}$$

and using the same approach for deriving Equation (B10), we obtain (for $p = 2$)

$$\|(A^{\mathrm{F}^{Ji}}_{\mathrm{ITA}} - A^{\mathrm{C}^i}_{\mathrm{ITA}})\| \leqslant i(J\theta_2\theta_3\,\Delta t^3 + \theta_1 \Delta T^3) \leqslant i(J\theta_2\theta_3 + \theta_1)\,\Delta T^3 \tag{B28}$$

which proves (B23) for $k = 0$. Then, we assume that the error estimate (B23) is verified until the kth iteration. From Equation (B26), it follows that

$$|\bar{E}^i_{k+1}| \leqslant \sum_{l=1}^{i-1} \|A^{\mathrm{C}^{i-1-l}}_{\mathrm{ITA}}\| \, \|(A^{\mathrm{F}^J}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}})\| \, |\bar{E}^k_l|, \quad 2 \leqslant i \leqslant N_{\mathrm{ts}} - 1, \quad k \geqslant 0 \tag{B29}$$

From the assumption of stability of the ITA on the coarse time-grid, the inequality (B28), and the assumption of the recurrence, it follows that the error estimate (B23) holds for iteration $k + 1$. □

APPENDIX C

Lemma

For any pair of integers k and i verifying $k + 1 \leqslant i - 1$, the following results hold:

(1)

$$\sum_{l=k+1}^{i-1} \left(\prod_{s=1}^{k} (l - s) \right) = \frac{1}{k+1} \prod_{s=1}^{k+1} (i - s) \tag{C1}$$

(2)

$$\sum_{l=k+1}^{i-1} \left(\prod_{s=0}^{k-1} (l - s) \right) = \frac{1}{k+1} \prod_{s=0}^{k} (i - s) - k! \tag{C2}$$

(3)

$$\sum_{l=k+1}^{i-1} \left(z \prod_{s=0, s\neq z-1}^{k} (l - s) + (k + 1 - z) \prod_{s=0, s\neq z}^{k} (l - s) \right) = \prod_{s=0, s\neq z}^{k+1} (i - s) \tag{C3}$$

Proof

To prove the first result of this lemma, we first note that

$$\prod_{s=1}^{k}(l+k-s)=k!\begin{pmatrix} l-1+k \\ k \end{pmatrix} \tag{C4}$$

which can be used to show that

$$\sum_{l=k+1}^{i-1}\left(\prod_{s=1}^{k}(l-s)\right)=k!\sum_{s=0}^{i-(k+2)}\begin{pmatrix} s+k \\ k \end{pmatrix} \tag{C5}$$

We also note the following classical result for binomial coefficients:

$$\sum_{s=0}^{i-(k+2)}\begin{pmatrix} s+k \\ k \end{pmatrix}=\begin{pmatrix} i-1 \\ k+1 \end{pmatrix} \tag{C6}$$

From the above results, it follows that

$$\begin{aligned}\sum_{l=k+1}^{i-1}\left(\prod_{s=1}^{k}(l-s)\right)&=\frac{1}{k+1}\frac{(i-1)!}{(i-(k+2))!}\\ &=\frac{1}{k+1}\prod_{s=1}^{k+1}(i-s)\end{aligned} \tag{C7}$$

To prove the second result of this lemma, we first note that

$$\prod_{s=0}^{k-1}(l+k-s)=k!\begin{pmatrix} l+k \\ k \end{pmatrix} \tag{C8}$$

which we exploit to find that

$$\sum_{l=k+1}^{i-1}\left(\prod_{s=0}^{k-1}(l-s)\right)=k!\left(\sum_{l=0}^{i-(k+1)}\begin{pmatrix} l+k \\ k \end{pmatrix}-1\right) \tag{C9}$$

We note again the following classical result for binomial coefficients:

$$\sum_{l=0}^{i-(k+1)}\begin{pmatrix} l+k \\ k \end{pmatrix}=\begin{pmatrix} i \\ k+1 \end{pmatrix} \tag{C10}$$

and finally obtain

$$\begin{aligned}\sum_{l=k+1}^{i-1}\left(\prod_{s=0}^{k-1}(l-s)\right)&=\frac{1}{k+1}\frac{i!}{(i-(k+1))!}-k!\\ &=\frac{1}{k+1}\prod_{s=0}^{k}(i-s)-k!\end{aligned} \tag{C11}$$

To prove the third result of this lemma, we first introduce

$$\begin{aligned} S &= \sum_{l=k+1}^{i-1} \left(z \prod_{s=0, s\neq z-1}^{k} (l-s) + (k+1-z) \prod_{s=0,s\neq z}^{k} (l-s) \right) \\ &= \sum_{l=1}^{i-(k+1)} \left(z \prod_{s=0, s\neq z-1}^{k} (l+k-s) + (k+1-z) \prod_{s=0,s\neq z}^{k} (l+k-s) \right) \end{aligned} \tag{C12}$$

and note that the following can be shown:

$$\begin{aligned} \prod_{s=0,s\neq z-1}^{k} (l+k-s) &= (z-1)!(k+1-z)! \binom{l+k}{z-1} \binom{l+k-z}{k-(z-1)} \\ \prod_{s=0,s\neq z}^{k} (l+k-s) &= z!(k-z)! \binom{l+k}{z} \binom{l+k-(z+1)}{k-z} \end{aligned} \tag{C13}$$

Next, we exploit the following properties of binomial coefficients:

$$\begin{aligned} \binom{l+k}{z-1} &= \binom{l+k+1}{z} - \binom{l+k}{z} \\ \binom{l+k-(z+1)}{k-z} &= \binom{l+k-z}{k+1-z} - \binom{l+k-(z+1)}{k+1-z} \end{aligned} \tag{C14}$$

to transform S into a telescoping series and finally obtain

$$\begin{aligned} S &= z!(k+1-z)! \left(\binom{i}{z} \binom{i-(z+1)}{k+1-z} \right) \\ &= \prod_{s=0,s\neq z}^{k+1} (i-s) \end{aligned} \tag{C15}$$

□

Proposition 5
Given an ITA with amplification matrices $A^{\mathrm{C}}_{\mathrm{ITA}}$ and $A^{\mathrm{F}}_{\mathrm{ITA}}$ which commute, the corresponding PITA admits a sequence of amplification matrices on the coarse time-grid denoted by $A^{\mathrm{C}}_{\mathrm{PITA}}(k,i)$ which satisfy for $k \geqslant 1$,

$$Y_k^{i+1} = A^{\mathrm{C}}_{\mathrm{PITA}}(k,i) Y_k^i, \quad 0 \leqslant i \leqslant N_{\mathrm{ts}} - 2 \tag{C16}$$

where

$$\begin{aligned} A^{\mathrm{C}}_{\mathrm{PITA}}(k,i) &= (A^{\mathrm{F}}_{\mathrm{ITA}})^J, \quad \text{if } i<k \\ A^{\mathrm{C}}_{\mathrm{PITA}}(k,i) &= B_{k,i+1} (B_{k,i})^{-1}, \quad \text{if } i \geqslant k \end{aligned}$$

$$B_{k,i} = (A^{C}_{\mathrm{ITA}})^{i-k} \left(\sum_{l=0}^{k} g_{l,k}(i)(A^{\mathrm{F}}_{\mathrm{ITA}})^{Jl}(A^{\mathrm{C}}_{\mathrm{ITA}})^{k-l} \right)$$

$$g_{l,k}(i) = (-1)^{l+k} \frac{\binom{k}{l}}{k!} \prod_{j=0, j\neq l}^{k} (i-j)$$

Proof
We prove this proposition by proving by induction that for $k \geqslant 1$,

$$\begin{aligned} Y_k^i &= A^{\mathrm{F}^{Ji}}_{\mathrm{ITA}} y^0 \quad \text{for } i \leqslant k \\ Y_k^i &= A^{\mathrm{C}^{i-k}}_{\mathrm{ITA}} \left(\sum_{l=0}^{k} g(l,k,i) A^{\mathrm{F}^{Jl}}_{\mathrm{ITA}} A^{\mathrm{C}^{k-l}}_{\mathrm{ITA}} \right) y^0 \quad \text{for } i > k \end{aligned} \tag{C17}$$

If the seed information is generated by the ITA on the coarse time-grid, Equation (B24) and the assumption that $A^{\mathrm{C}}_{\mathrm{ITA}}$ and $A^{\mathrm{F}}_{\mathrm{ITA}}$ commute lead to

$$Y_1^i = A^{\mathrm{C}^{i-1}}_{\mathrm{ITA}} (i A^{\mathrm{F}^{J}}_{\mathrm{ITA}} - (i-1) A^{\mathrm{C}}_{\mathrm{ITA}}) y^0 \tag{C18}$$

which shows that (C17) is verified for $k=1$. Next, we assume that this proposition is verified until the kth iteration. At iteration $k+1$, two cases can be distinguished:

- $i \leqslant k+1$: In this case, Equation (B24), the observation that $l \leqslant i-1$ and $i \leqslant k+1$ implies $l \leqslant k$, and the assumption of the recurrence lead after some algebraic manipulations to

$$Y_{k+1}^i = A^{\mathrm{F}^{Ji}}_{\mathrm{ITA}} y^0 \tag{C19}$$

- $i > k+1$: In this case, we split Equation (B24) as follows:

$$Y_{k+1}^i = A^{\mathrm{C}^{i-1}}_{\mathrm{ITA}} A^{\mathrm{F}^{J}}_{\mathrm{ITA}} y^0 + \sum_{l=1}^{k} A^{\mathrm{C}^{i-1-l}}_{\mathrm{ITA}} (A^{\mathrm{F}^{J}}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}}) Y_k^l + \sum_{l=k+1}^{i-1} A^{\mathrm{C}^{i-1-l}}_{\mathrm{ITA}} (A^{\mathrm{F}^{J}}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}}) Y_k^l \tag{C20}$$

exploit the assumption of the recurrence to show that

$$\sum_{l=1}^{k} A^{\mathrm{C}^{i-1-l}}_{\mathrm{ITA}} (A^{\mathrm{F}^{J}}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}}) Y_k^l = A^{\mathrm{C}^{i-(k+1)}}_{\mathrm{ITA}} A^{\mathrm{F}^{J(k+1)}}_{\mathrm{ITA}} - A^{\mathrm{C}^{i-1}}_{\mathrm{ITA}} A^{\mathrm{F}^{J}}_{\mathrm{ITA}} y^0 \tag{C21}$$

and the fact that $A^{\mathrm{C}}_{\mathrm{ITA}}$ and $A^{\mathrm{F}}_{\mathrm{ITA}}$ commute to conclude that

$$\sum_{l=k+1}^{i-1} A^{\mathrm{C}^{i-1-l}}_{\mathrm{ITA}} (A^{\mathrm{F}^{J}}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}}) Y_k^l = \sum_{l=k+1}^{i-1} A^{\mathrm{C}^{i-(k+1)}}_{\mathrm{ITA}} (A^{\mathrm{F}^{J}}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}}) \left(\sum_{z=0}^{k} g(z,k,l) A^{\mathrm{F}^{Jz}}_{\mathrm{ITA}} A^{\mathrm{C}^{k-z}}_{\mathrm{ITA}} \right) y^0 \tag{C22}$$

Next, we note that

$$(A^{\mathrm{F}^{J}}_{\mathrm{ITA}} - A^{\mathrm{C}}_{\mathrm{ITA}}) \left(\sum_{z=0}^{k} g(z,k,l) A^{\mathrm{F}^{Jz}}_{\mathrm{ITA}} A^{\mathrm{C}^{k-z}}_{\mathrm{ITA}} \right)$$

$$= g(k,k,l)A_{\mathrm{ITA}}^{\mathrm{F}^{J(k+1)}} - g(0,k,l)A_{\mathrm{ITA}}^{\mathrm{C}^{k+1}} + \sum_{z=1}^{k}(g(z-1,k,l)-g(z,k,l))A_{\mathrm{ITA}}^{\mathrm{F}^{Jz}}A_{\mathrm{ITA}}^{\mathrm{C}^{(k+1)-z}} \tag{C23}$$

From Equations (C22) and (C23), it follows that

$$\begin{aligned}\sum_{l=k+1}^{i-1} A_{\mathrm{ITA}}^{\mathrm{C}^{i-1-l}}(A_{\mathrm{ITA}}^{\mathrm{F}^{J}} - A_{\mathrm{ITA}}^{\mathrm{C}})Y_k^l = A_{\mathrm{ITA}}^{\mathrm{C}^{i-(k+1)}} \Bigg(&\sum_{l=k+1}^{i-1} g(k,k,l)A_{\mathrm{ITA}}^{\mathrm{F}^{J(k+1)}} - \sum_{l=k+1}^{i-1} g(0,k,l)A_{\mathrm{ITA}}^{\mathrm{C}^{k+1}} \\ &+\sum_{z=1}^{k}\sum_{l=k+1}^{i-1}(g(z-1,k,l)-g(z,k,l))A_{\mathrm{ITA}}^{\mathrm{F}^{Jz}}A_{\mathrm{ITA}}^{\mathrm{C}^{(k+1)-z}}\Bigg) y^0\end{aligned} \tag{C24}$$

From all of the above results, it follows that

$$Y_{k+1}^i = A_{\mathrm{ITA}}^{\mathrm{C}^{i-(k+1)}}\left(\sum_{z=0}^{k+1} g(z,k+1,i)A_{\mathrm{ITA}}^{\mathrm{F}^{Jz}}A_{\mathrm{ITA}}^{\mathrm{C}^{(k+1)-z}}\right) y^0 \tag{C25}$$

where

$$\text{for } z=0, \quad g(0,k+1,i) = -\sum_{l=k+1}^{i-1} g(0,k,l) \tag{C26}$$

$$\text{for } 1\leqslant z\leqslant k, \quad g(z,k+1,i) = \sum_{l=k+1}^{i-1}(g(z-1,k,l)-g(z,k,l)) \tag{C27}$$

$$\text{for } z=k+1, \quad g(k+1,k+1,i) = \sum_{l=k+1}^{i-1} g(k,k,l)+1 \tag{C28}$$

Finally, we prove also by induction the proposed expression for $g(l,k,i)$. This expression is correct for $k=1$. At iteration $k+1$, we note that:

- For $z=0$, from Equation (C26) and Lemma 1 it follows that

$$\begin{aligned} g(0,k+1,i) &= (-1)^{k+1}\frac{1}{k!}\sum_{l=k+1}^{i-1}\left(\prod_{s=1}^{k}(l-s)\right) \\ &= (-1)^{k+1}\frac{\begin{pmatrix}k+1\\0\end{pmatrix}}{(k+1)!}\prod_{s=1}^{k+1}(i-s)\end{aligned} \tag{C29}$$

- For $z=k+1$, from Equation (C28) and Lemma 2, it follows that

$$\begin{aligned} g(k+1,k+1,i) &= \frac{1}{k!}\sum_{l=k+1}^{i-1}\left(\prod_{s=0}^{k-1}(l-s)\right)+1 \\ &= \frac{\begin{pmatrix}k+1\\k+1\end{pmatrix}}{(k+1)!}\prod_{s=0}^{k}(i-s)\end{aligned} \tag{C30}$$

- For $1 \leqslant z \leqslant k$, from Equation (C27) and after some algebraic transformations, it follows that

$$g(z,k+1,i) = \frac{(-1)^{z+(k+1)}}{k!} \sum_{l=k+1}^{i-1} \left(\binom{k}{z-1} \prod_{s=0,s\neq z-1}^{k} (l-s) + \binom{k}{z} \prod_{s=0,s\neq z}^{k} (l-s) \right) \quad \text{(C31)}$$

Next, we rely on the following properties of the binomial coefficients:

$$\binom{k}{z-1} = \frac{z}{k+1}\binom{k+1}{z} \quad \text{and} \quad \binom{k}{z} = \frac{k+1-z}{k+1}\binom{k+1}{z} \quad \text{(C32)}$$

as well as on Lemma 3 to finally obtain

$$g(z,k+1,i) = (-1)^{z+(k+1)} \frac{\binom{k+1}{z}}{(k+1)!} \prod_{s=0,s\neq z}^{k+1} (i-s) \quad \text{(C33)}$$

□

ACKNOWLEDGEMENTS

This material is based upon work supported partially by the National Science Foundation under Grant No. 008463, and partially by the Air Force Office of Scientific Research under Grant F49620-01-1-0129. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Air Force Office of Scientific Research.

REFERENCES

1. Lions JL, Maday Y, Turinici G. Résolution d'EDP par un schéma en temps 'pararéel'. *Comptes Rendus de l'Académie des Sciences – Series I: Mathematics* 2001; **332**(7):661–668.
2. Combescure A, Gravouil A. A numerical scheme to couple subdomains with different time-steps for predominantly linear transient analysis. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:1129–1157.
3. Hughes TJR. Analysis of transient algorithms with particular reference to stability behavior. In *Computational Methods for Transient Analysis*, Belytschko T, Hughes TJR (eds). North Holland, 1983; 67–155.
4. Piperno S, Farhat C, Larrouturou B. Partitioned procedures for the transient solution of coupled aeroelastic problems—Part I: model problem, theory, and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering* 1995; **124**(1–2):79–112.
5. Koobus B, Farhat C. Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering* 1999; **170**:103–130.
6. Farhat C, Geuzaine P, Brown G. Application of a three-field non-linear fluid–structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter. *Computers and Fluids* 2003; **32**:3–29.
7. Farhat C, Geuzaine P, Grandmont C. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *Journal of Computational Physics* 2001; **174**:669–694.
8. Lesoinne A, Farhat C. A higher-order subiteration free staggered algorithm for nonlinear transient aeroelastic problems. *AIAA Journal* 1998; **36**(9):1754–1756.