Time-parallel algorithm for two phase flows simulation

Katia Ait-Ameur, Yvon Maday, Marc Tajchman

Time-parallel algorithm for two phase flows simulation

Katia Ait-Ameur, Yvon Maday and Marc Tajchman

Abstract In this paper, we will report our recent effort to apply the parareal algorithm to the time parallelization of an industrial code that simulates two phase flows in a reactor for safety studies. This software solves the six equation two-fluid model by considering a set of balance laws (mass, momentum and energy) for each phase, liquid and vapor, of the fluid. The discretization is based on a finite volume method on a staggered grid in space and on a multistep time scheme. Here, we apply a variant of the parareal algorithm on an oscillating manometer test case: the multistep variant allowing to deal with multistep time schemes in the coarse and/or fine propagators. Numerical results show that parareal methods offer the potential for an increased level of parallelism and is a good strategy to complement the current space domain decomposition implemented in the code.

1 Introduction

In the nuclear energy domain, computations of complex two-phase flows are required for the design and safety studies of nuclear reactors. System codes are dedicated to the thermalhydraulic analysis of nuclear reactors at the system scale by

Katia Ait-Ameur
Laboratoire Jacques Louis Lions (LJLL), Sorbonne Université, 75005 Paris, France
C.E.A, CEA Saclay - DEN/DANS/DM2S/STMF/LMES - 91191 Gif-Sur-Yvette Cedex, France
e-mail: aitameur.katia@gmail.com

Yvon Maday
Laboratoire Jacques Louis Lions (LJLL), Sorbonne Université, 75005 Paris, France
Institut Universitaire de France
e-mail: maday@ann.jussieu.fr

Marc Tajchman
C.E.A, CEA Saclay - DEN/DANS/DM2S/STMF/LMES - 91191 Gif-Sur-Yvette Cedex, France
e-mail: marc.tajchman@cea.fr

simulating the whole reactor. We are here interested in the Cathare code developed by the CEA. Like all system codes, Cathare essentially simulates assemblies of one-dimensional elements (pipes) and 3D elements (vessels). The discretization level of each element is kept intentionnaly at a coarse level to be able to handle whole systems simulations. Typical meshes used for the simulations are about 10^2 to 10^3 cells with a 3D element. Typical cases involve up to a million of numerical time iterations, computing the approximate solution during long physical simulation times. A space domain decomposition method has already been implemented and to improve the response time, we will consider a strategy of time domain decomposition, based on the *parareal method* ([13]). The paper is organized as follows: after the presentation of the six equation two-fluid model and the Cathare numerical scheme in section 2, the main aspects of the parareal methods will be recalled in section 3. The numerical convergence observed in our example is shown in section 4 followed by the performances we obtain by applying the multistep variant of the parareal algorithm.

2 Model

At the system scale, the finest details of the flow (description of the liquid-vapor interfaces for example) are not absolutely necessary to obtain a satisfactory macroscale description of the dynamics. For this reason macroscopic models have been developped that focus on the evolution of averaged quantities (see [12] and [7]). There are many different averaged models depending on the simplifying assumptions. The model used in Cathare is the 6 equation two-fluid model that considers a set of balance laws (mass, momentum and energy) for each phase, liquid and vapor. It assumes independent velocities and a pressure equality.
The unknowns are the volume fraction $\alpha_k \in [0,1]$, the pressure $p \geq 0$, the velocity u_k and the enthalpy H_k of each phase. The subscript k stands for l if it is the liquid phase and g for the gas phase. For the sake of simplicity, we write the terms of the model involved in our test case, studied in section 4.

$$\begin{cases} \partial_t(\alpha_k\rho_k) + \partial_x(\alpha_k\rho_k u_k) = 0 \\ \alpha_k\rho_k\partial_t u_k + \alpha_k\rho_k u_k\partial_x u_k + \alpha_k\partial_x p = \alpha_k\rho_k g + F_k^{\text{int}} \\ \partial_t\left[\alpha_k\rho_k\left(H_k + \frac{u_k^2}{2}\right)\right] + \partial_x\left[\alpha_k\rho_k u_k\left(H_k + \frac{u_k^2}{2}\right)\right] = \alpha_k\partial_t p + \alpha_k\rho_k u_k g \end{cases} \tag{1}$$

with $\alpha_g + \alpha_l = 1$ and the two equations of state : $\rho_k = \rho_k(p, H_k)$. The interfacial forces F_k^{int} are of 2 types. The first ensures hyperbolicity of the system (see [15] for the well-posedness of the 6 equation model). The second is the interfacial friction term which will be important in the sequel for our test case. In this configuration, the phases are separated which means that one of the two phases vanishes in some parts of the domain. It is numerically challenging to compute the velocity of the

ghost phase (see [16]). For this reason, the Cathare scheme forces the two velocities to be equal with this damping term.

2.1 Numerical method

The Cathare scheme is based on a finite volume method on a staggered grid (MAC scheme) and on a two step time scheme. In a staggered scheme the i-th component of the velocity is located at the center of the edge orthogonal to the i-th unit vector. Pressures, void fractions and enthalpies are cell-centered. Given a time discretization $T^0, T^1, T^2, \cdots$ of the full time interval $[0,T)$, we use the following notations: $(\alpha_k\rho_k)^n$ is an approximation of $(\alpha_k\rho_k)$ at time T^n. Here, we write the time discretization of the Cathare scheme:

$$\begin{cases} \frac{(\alpha_k\rho_k)^{n+1}-(\alpha_k\rho_k)^n}{\Delta t} + \partial_x(\alpha_k\rho_k u_k)^{n+1} = 0 \\ (\alpha_k\rho_k)^{n+1}\frac{u_k^{n+1}-u_k^n}{\Delta t} + (\alpha_k\rho_k u_k)^{n+1}\partial_x u_k^{n+1} + \alpha_k^{n+1}\partial_x p^{n+1} = (\alpha_k\rho_k)^{n+1} g + F_k^{n,n+1} \\ \frac{1}{\Delta t}\left[(\alpha_k\rho_k)^{n+1}\left(H_k+\frac{u_k^2}{2}\right)^{n,n+1} - (\alpha_k\rho_k)^n\left(H_k+\frac{u_k^2}{2}\right)^{n-1,n}\right] \\ +\partial_x\left[\alpha_k\rho_k u_k\left(H_k+\frac{u_k^2}{2}\right)\right]^{n+1} = \alpha_k^{n+1}\frac{p^{n+1}-p^n}{\Delta t} + (\alpha_k\rho_k u_k)^{n+1} g \end{cases} \tag{2}$$

Where the notation $F_k^{n,n+1}$ means that the discretization of F_k is a function of the approximate solution at times T^n and T^{n+1}. After discretization, the non linear system is solved by a Newton method. Here, we highlight some characteristics of the Cathare scheme, some advantages and limitations:

- The scheme must be accurate enough at the incompressible limit to be able to capture the correct streamlines and pressure fields. This characteristic was mainly studied in the monophasic case:
  - Staggered schemes enjoy good precision at the incompressible limit
  - However, Riemann solvers have poor precision in the incompressible limit. Corrections are proposed in [6] to overcome this issue
- The vanishing phase is a numerical challenge of two phase flows simulation. It is important to capture well the volume fraction since it governs the composition of the mixture and two-phase/single phase transition . An important issue is to guarantee the positivity of the volume fraction. Many schemes were designed to ensure this property (like [16] for two incompressible phases). Cathare uses a high interfacial friction to deal numerically with these transitions.

3 The Parareal algorithm

Several approaches have been proposed over the years to decompose the time direction when solving a partial differential equation (see [9] for an overview). Of these, the parareal in time algorithm, which performances we explore in this work, has received an increasing amount of attention in the last twenty years with many applications (see [3], [8], [17] among many others). In the sequel, we recall the classical parareal algorithm as initially proposed in [13], [3], [4] and the principle of the multistep variant we will apply in section 4.

3.1 Original parareal algorithm

After the discretization in space of a PDE with $\mathcal{N}$ the number of degrees of freedom:

$$\frac{\partial u}{\partial t} + A(t,u) = 0, \quad t \in [0,T], \quad u(t=0) = u_0 \tag{3}$$

$$A : \mathbb{R} \times \mathbb{R}^{\mathcal{N}} \to \mathbb{R}^{\mathcal{N}}, u \in \mathbb{R}^{\mathcal{N}} \tag{4}$$

We recall here the classical parareal algorithm as initially proposed in [13], [3], [4]. Let G and F be two propagators such that, for any given $t \in [0,T]$, $s \in [0,T-t]$ and any function w in a Banach space, $G(t,s,w)$ (respectively $F(t,s,w)$) takes w as an initial value at time t and propagates it at time $t+s$. The full time interval is divided into N^c sub-intervals $[T^n, T^{n+1}]$ of size ΔT that will each be assigned to a processor. The algorithm is defined using two propagation operators:

- $G(T^n, \Delta T, u^n)$ computes a coarse approximation of $u(T^{n+1})$ with initial condition $u(T^n) \simeq u^n$ (low computational cost)
- $F(T^n, \Delta T, u^n)$ computes a more accurate approximation of $u(T^{n+1})$ with initial condition $u(T^n) \simeq u^n$ (high computational cost)

Starting from a coarse approximation u_0^n at times $T^0, T^1, \cdots, T^{N^c}$, obtained using G, the parareal algorithm performs for $k = 0, 1, \cdots$ the following iteration:

$$u_{k+1}^{n+1} = G(T^n, \Delta T, u_{k+1}^n) + F(T^n, \Delta T, u_k^n) - G(T^n, \Delta T, u_k^n)$$

In the parareal algorithm, the value $u(T^n)$ is approximated by u_k^n at each iteration k with an accuracy that tends rapidly to the one achieved by the fine solver, when k increases. The coarse approximation G can be chosen much less expensive than the fine solver F by the use of a scheme with a much larger time step (even $\delta T = \Delta T$) $\delta T \gg \delta t$ (time step of the fine solver) or by using a reduced model. All the fine propagations are made in parallel over the time windows and the coarse propagations are computed in a sequential way but have a low computational cost. The main convergence properties were studied in [10] and stability analysis was made in [18],

[5]. We refer to [14] about the parallel efficiency of parareal and a recent work offering a new formulation of the algorithm to improve the parallel efficiency of the original one.

3.2 Multistep variant of parareal

This variant of the parareal algorithm was proposed in [1]. In the sequel, we will consider that the fine solver is based on a two step time scheme like the Cathare time scheme. Hence we will use the following notation for the fine solver that takes two initial values: $F(t,s,x,y)$, for $t \in [0,T]$, $s \in [0,T-t[$ and x,y in a Banach space.

Example

If one solves (3) with a multistep time scheme as fine propagator F like the order 2 BDF method:

$$\frac{3}{2}u^{j+1} - 2u^j + \frac{1}{2}u^{j-1} = -\delta t A(u^{j+1}, t^{j+1}), \quad j = 1, \cdots, N^f, t^{j+1} - t^j = \delta t$$

Here the fine solver reads: $u^{j+1} = F(t^j, \delta t, u^{j-1}, u^j)$. Now, we apply the parareal algorithm with a coarse grid: $T^0, \cdots, T^{N^c}$ where: $T^{n+1} - T^n = \Delta T = N^f \delta t$.
Then we can write: $u(T^n + j\delta t) \simeq u^{n,j}$, $j = 1, \cdots, N^f, n = 1, \cdots, N^c$.
In order to perform the fine propagation, in a given time window $[T^n, T^{n+1}]$, we only need the local initial condition u_k^n and a consistent approximation of $u(T^n - \delta t)$.

In [2], the authors propose a consistent approximation in the context of the simulation of molecular dynamics. The proposed method was linked to the nature of the model and the symplectic character of their algorithm is shown, which is an important property to verify for molecular dynamics.
In the context of our application to the thermalhydraulic code Cathare, we want to derive a multistep variant of parareal that will not be intrusive in the software. We seek a consistent approximation of $u(T^n - \delta t)$. The only fine trajectory at our disposal is $F(T^{n-1}, \Delta T, u_k^{n-2,N^f-1}, u_k^{n-1})$. Its final value at T^n is:
$F(T^{n-1}, \Delta T, u_k^{n-2,N^f-1}, u_k^{n-1})(T^n)$ from which we compute u_{k+1}^n by the parareal correction. Hence, we translate the solution:
$F(T^{n-1}, \Delta T - \delta t, u_k^{n-2,N^f-1}, u_k^{n-1})(T^n - \delta t)$ by the same correction:
$u_{k+1}^n - F(T^{n-1}, \Delta T, u_k^{n-2,N^f-1}, u_k^{n-1})$ and obtain the so called consistent approximation u_{k+1}^{n-1,N^f-1} to initialize the fine propagation in $[T^n, T^{n+1}]$. We now detail our algorithm:

$$\begin{cases} u_0^{n+1} = G(T^n, \Delta T, u_0^n), \quad 0 \le n \le N-1 \\ u_{k+1}^{n+1} = G(T^n, \Delta T, u_{k+1}^n) + F(T^n, \Delta T, u_k^{n-1,N^f-1}, u_k^n) \\ \qquad\qquad -G(T^n, \Delta T, u_k^n), \quad 0 \le n \le N-1, \quad k \ge 0 \\ u_{k+1}^{n,N^f-1} = F(T^n, \Delta T - \delta t, u_k^{n-1,N^f-1}, u_k^n) + u_{k+1}^{n+1} \\ \qquad\qquad -F(T^n, \Delta T, u_k^{n-1,N^f-1}, u_k^n), \quad 0 \le n \le N-1, \quad k \ge 0 \end{cases} \tag{5}$$

Remark: In order to perform the fine propagation, in a given time window $[T^n, T^{n+1}]$, at the first parareal iteration we need to choose a different consistent approximation of $u(T^n - \delta t)$, since we have not used the fine solver yet. To treat this, we could make one iteration with a Backward Euler method or one iteration with a second order Runge Kutta method. In the context of the application to the Cathare code, we choose a non intrusive initialization by imposing $u_0^{n-1,N^f-1} = u_0^n$.

4 Test case

Here we apply the multistep parareal algorithm to the resolution of an oscillating manometer. This test case is proposed in [11] for system codes to test the ability of each numerical scheme to preserve system mass and to retain the gas-liquid interface. In this test case, the phases are separated and the interfacial friction term will be important in this configuration.
Note that here we have used the same physical model and the same mesh (110 cells) for both the coarse and the fine solvers: the only difference is the size of the time steps, δt for F and ΔT for G. All calculations have been evaluated with a stopping criteria where the tolerance is fixed to the precision of the numerical scheme, $\varepsilon = 5 \cdot 10^{-2}$. With this threshold, parareal convergence is achieved after 2 or 3 iterations.
In the following subsections, after giving a numerical proof of the convergence of the parareal algorithm in our test case, some results about measured speed-up will be presented.

4.1 About the convergence

Figure 1 illustrates that the multistep parareal algorithm effectively converges when applied to the problem of the oscillating manometer. For a given time step T^n and parareal iteration k, the relative error in L^2 norm between the parareal solution and the sequential fine solver decreases beyond our given convergence threshold ε. In the figure, the test case has been solved with the multistep parareal algorithm when $\delta t = 10^{-5}$ and $\Delta T = 10\delta t$.

These results are obtained on 16 time windows and we will use this configuration in the sequel to study the performances.

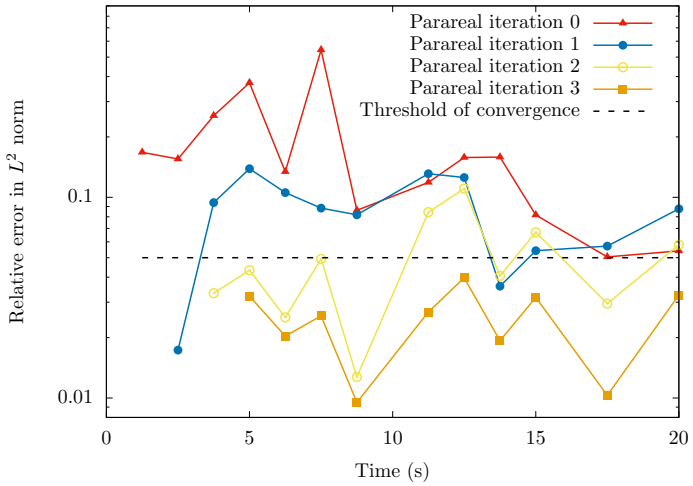

Fig. 1 Convergence of the multistep parareal algorithm when $\delta t = 10^{-5}$ and $\Delta T = 10\delta t$

4.2 Speed-up performances

In the following strong scaling tests, the same setting is used for the multistep parareal algorithm. The test case has been solved on an increasing number N_{proc} of processes $N_{proc} = 5, 10, 15, \cdots, 70$. In figure 2, with 25 processes, we obtain a speed up of 3.4 and of 3.7 with 50 processes. Here, we observe two global trends:

- For $N_{proc} = \{5, 10, 15, 20, 25, 40, 50\}$, the speed up first monotonically increases until reaching 25 processes and then increase again with 40 and 50 processes. This is due to the number of parareal iterations that is equal to 2 in this case
- For $N_{proc} = \{30, 35, 45, 55, 60, 65, 70\}$, the speed up is drastically reduced because the parareal algorithm converges in 3 iterations in this case

In the sequel, we highlight the well-known dependance of the computational cost of the parareal algorithm on the number of iterations. Let T_{fine} be the cpu time to run the fine solver in a sequential way on the whole time interval $[0, T)$. Since the coarse time step is ten times greater than the fine time step we suppose that the cpu time of the coarse solver $T_{coarse} = \frac{T_{fine}}{10}$. This ratio between coarse and fine solvers should be as high as possible to minimize the computational cost of the coarse solver which is launched in a sequential way. This aspect will be studied in a forthcoming work to obtain better speed-up performances by coarsening more the solver G. When the

algorithm converges in N_{it} iterations, the coarse solver is launched N_{it} times and the fine solver $N_{it}-1$ times in parallel over the number of processes N_{proc}. Hence, we can write the cpu time in parallel T_{para} in terms of T_{fine}:

$$T_{para} = (N_{it}-1)\frac{T_{fine}}{N_{proc}} + N_{it}T_{coarse} + \tau = \left(\frac{N_{it}-1}{N_{proc}} + \frac{N_{it}}{10}\right)T_{fine} + \tau$$

where τ contains the time of communication between processes and the cpu time for the computation of the parareal corrections and of the error. Now, we can deduce an upper bound of the speed up S when the parareal algorithm converges in 2 or 3 iterations by neglecting τ:

$$S = \frac{T_{fine}}{T_{para}} \leq \frac{1}{\frac{N_{it}-1}{N_{proc}} + \frac{N_{it}}{10}}$$

Example: On 25 processes, the algorithm converges in 2 iterations: $S \approx 4$ when the measured speed up is 3.4.
On 35 processes, the algorithm converges in 3 iterations: $S \approx 2.8$ when the measured speed up is 2.3.

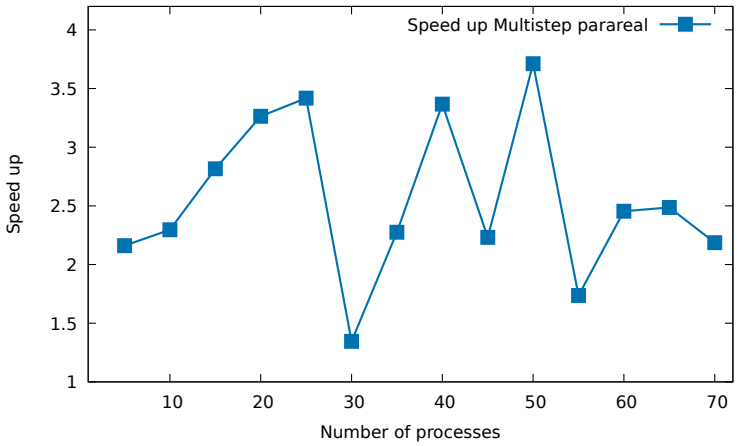

Fig. 2 Strong scaling results with the multistep variant of the parareal algorithm

Acknowledgements This research is partially supported by ANR project CINE-PARA (ANR-15-CE23-0019). We acknowledge the travel support provided by SMAI to attend the Spanish-French School Jacques-Louis Lions.

5 Conclusion

The results of this study show that the parareal algorithm can effectively speed-up two-phase flows simulations. Here we have tested the multistep variant of the parareal algorithm on a test case that is representative of the numerical challenges for two phase flows. These results can certainly be improved by using a new version of parareal (see [14]) that improves the parallel efficiency of the original parareal. This new algorithm proposes to solve the subproblems at increasing accuracy across the parareal iterations and should allow us to increase the speed up performances obtained on our test case.

References

1. K. Ait-Ameur, Y. Maday, M. Tajchman, Multi-step variant of the parareal algorithm, submitted
2. C. Audouze, M. Massot, S. Volz, Symplectic multi-time step parareal algorithms applied to molecular dynamics, http://hal.archives-ouvertes.fr/hal-00358459/fr/ (2009)
3. L. Baffico, S. Bernard, Y. Maday, G. Turinici, G. Zrah, Parallel-in-time molecular dynamics simulations, Physical Review E, 66, p. 057701 (2002)
4. G. Bal, Y. Maday, A "parareal" time discretization for non-linear PDE's with application to the pricing of an American put, Recent developments in domain decomposition methods, 23, pp. 189-202 (2002)
5. Bal, G., Parallelization in time of (stochastic) ordinary differential equations, http://www.columbia.edu/ gb2030/PAPERS/paralleltime.pdf (2003)
6. S. Dellacherie, Analysis of Godunov type schemes applied to the compressible Euler system at low Mach number, Journal of Computational Physics, 229(4), pp. 978-1016 (2010)
7. D.A. Drew, S.L. Passman, Theory of multicomponent fluids. Springer-Verlag, New-York (1999)
8. P.F. Fischer, F. Hecht, Y. Maday, A parareal in time semi-implicit approximation of the Navier-Stokes equations, In Kornhuber, R. and al, editors : Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, volume 40, pages 433-440, Springer (2004)
9. M.J. Gander, 50 years of time parallel time integration. In Carraro, T., Geiger, M., Krkel, S. and Rannacher, R., editors : Multiple Shooting and Time Domain Decomposition Methods, pages 69-114, Springer (2015)
10. M. J. Gander, S. Vandewalle, Analysis of the Parareal Time-Parallel Time-Integration Method, SIAM J. Sci. Comput., 29(2):556-578 (2007)
11. G.F. Hewitt, J.M. Delhaye, N. Zuber, Multiphase science and technology, vol. 6 (1991)
12. M. Ishii, Thermo-fluid dynamic theory of two-phase flow. Eyrolles, Paris (1975)
13. J.-L. Lions, Y. Maday, G. Turinici, Résolution par un schéma en temps "pararéel", C. R. Acad. Sci. Paris, 332(7):661-668 (2001)
14. Y. Maday and O. Mula. An adaptive parareal algorithm. Submitted, https://arxiv.org/pdf/1909.08333.pdf (2019)
15. M. Ndjinga, Influence of interfacial pressure on the hyperbolicity of the two-fluid model, C. R. Acad. Sci. Paris; Ser. I 344, pp. 407-412 (2007)
16. M. Ndjinga, T. P. K. Nguyen, C. Chalons, A 2x2 hyperbolic system modelling incompressible two phase flows: theory and numerics, Nonlinear Differential Equations and Applications (2017)

17. D. Samaddar, D. E. Newman, R. Sanchez, Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm, Journal of Computational Physics, 229(18):6558-6573 (2010)
18. G.A. Staff, E.M. Ronquist, Stability of the Parareal algorithm, In: Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, vol. 40, p. 449-456, Springer (2005)