Time parallelization of advanced operation scenario simulations of ITER plasma

D. Samaddar¹, T.A. Casper¹, S.H. Kim¹, L.A. Berry², W.R. Elwasif², D. Batchelor², W.A. Houlberg¹
1. ITER Organization, Route de Vinon sur Verdon, 13115 Saint Paul Lez Durance, France
2. Oak Ridge National Laboratory, Oak Ridge, USA

Abstract. This work demonstrates that simulations of advanced burning plasma operation scenarios can be successfully parallelized in time using the parareal algorithm. CORSICA - an advanced operation scenario code for tokamak plasmas is used as a test case. This is a unique application since the parareal algorithm has so far been applied to relatively much simpler systems except for the case of turbulence. In the present application, a computational gain of an order of magnitude has been achieved which is extremely promising. A successful implementation of the Parareal algorithm to codes like CORSICA ushers in the possibility of time efficient simulations of ITER plasmas.

1. Introduction
With the construction of the ITER project well under way, understanding and predicting burning plasma scenarios have gained tremendous importance. These studies are used for designing of components for the world’s largest tokamak and contribute towards its successful operation. Plasma scenario studies are carried out by codes such as CORSICA [1, 2]. CORSICA employs a free or fixed boundary 2D equilibrium package that is coupled with various transport and source models. The advanced capabilities of these simulations allow studies with realistic physics and engineering constraints. CORSICA has also been used to predict and analyse plasma operations in the existing tokamak DIII-D[3] which served as a means of validation and verification.

It must be borne in mind that simulations of plasma operation scenarios require extremely long wallclock time. Space parallelization is a common approach to achieve computational speed-up although this technique experiences saturation while reduction in computation time is still desired. The option of time parallelization introduces a new opportunity, although this approach has not been explored much in the past for these kinds of simulations. The parareal algorithm [4] lays out a scheme for the application of temporal parallelization to simulations. The algorithm has attracted significant attention in recent times and has undergone modifications to maximize computational speed-up and efficiency [5, 6, 7, 8].

Except for turbulence [5, 9] which is a complex system with high dimensional chaos, the parareal algorithm has so far been applied to relatively simple problems [10, 11, 12, 13, 14]. This work demonstrates that the parareal algorithm may be successfully applied to CORSICA to attain computational speed-up for plasma scenario simulations. The coupling of the equilibrium solver with the transport and source modules makes this a unique application compared to previous attempts. These simulations are characterized by quasi steady states interrupted by
events which may be results of MHD activity, pellet injection or an introduction of ELMs. The success of temporal parallelization in such simulations ushers the possibility of time efficient simulations of ITER relevant plasmas.

The parareal algorithm has been applied using the Parareal framework developed at ORNL as part of the SWIM-IPS project [6].

2. The Parareal algorithm

The parareal algorithm allows the splitting of a time series into multiple time slices that are solved in parallel. This unique technique is described in more detail in [5, 4]. Two solvers - one coarse (G) and one fine (F) - are required for this purpose. G is used to evolve the time series to obtain an approximate estimate of the fine solution. G is characterized by high computational speed and is always applied as a serial process. F is always applied in parallel, is computationally slow and yields a solution whose accuracy is higher than that obtained by the coarse solver G.

At the first parareal iteration, G is used to generate an estimate for an initial value for each time slice. F is then applied in parallel to evolve that state. G and F then alternate across successive iterations until parareal convergence is obtained. At each parareal iteration, k, for the (i + 1)th time slice, the initial state $\lambda_{i+1}^k$ is computed using the parareal correction (eq.1) after the state $\lambda_i^k$ has been evolved using $F(\lambda_i^k)$ and $G(\lambda_i^k)$. The parareal correction is defined as

$$\lambda_{i+1}^k = F(\lambda_i^k) + G(\lambda_i^k) - G(\lambda_i^{k-1})$$

The Parareal framework is used to apply the event driven parareal technique, which has been described in [6]. Using the framework allows better computational gain and efficiency. Moreover, since the choice of the coarse solver for a case like CORSICA is non trivial, experimenting with different options becomes relatively simpler with the framework.

2.1. Convergence

A parareal convergence is said to occur when the relative error between two successive fine solutions (k and k − 1) on a given time slice (i) corresponding to a given processor reduces below a prescribed tolerance. In case of CORSICA, two parameters $\beta_i$ and $l_i$ were used to define the convergence metric. $\beta_i$ is the normalized stored energy of the system and $l_i$ is the internal inductance whose variation is a measure of the evolution of the plasma current density profile. If $\beta_{p(i)}^{(k)}$ and $l_{i(i)}^{(k)}$ represent the variables for the $i^{th}$ time slice at the $k^{th}$ parareal iteration, the relative errors are given by $Err_{\beta(i)}^{(k)} = \frac{\beta_{p(i)}^{(k)} - \beta_{p(i-1)}^{(k-1)}}{\beta_{p(i-1)}^{(k-1)}}$ and $Err_{l(i)}^{(k)} = \frac{l_{i(i)}^{(k)} - l_{i(i-1)}^{(k-1)}}{l_{i(i-1)}^{(k-1)}}$.

The simulation across the processor solving the $i^{th}$ time slice is said to have converged if it satisfies the convergence criterion given by $\xi_i^k \leq tolerance$ where $\xi_i^k = Err_{\beta(i)}^{(k)} + Err_{l(i)}^{(k)}$.

3. Results

The parareal algorithm has been applied to a prescribed boundary simulation of CORSICA with neutral beam injection as a source term. The introduction of the NBI using the Nfrea package makes the simulation realistic in terms of ITER operational scenarios. NBI is the injection of high energy neutral particle beams across the magnetic field to heat the plasma to the temperature required for fusion reaction. The Monte-Carlo method is used to statistically simulate the localized ionisations of the beam particles along its trajectory. The parareal algorithm has been applied to CORSICA simulations without realistic source modules in [15], but this application with the NBI significantly enhances the complexity.

The choice of the coarse solver is always important for the success of the parareal algorithm. In case of CORSICA, selecting the optimum coarse solver becomes critical. Since the ITER NBI system injects 33MW of power, a fast analytic source term was introduced in the coarse
solver to simulate power of the same order. This point is illustrated in Fig.(1). Care was also taken to see that the total radial plasma current had a similar profile as is shown in Fig.(2).

The use of an analytic source term simplifies the calculation thus requiring lesser computational time. However, a similarity in the current profiles and the total power between the coarse and the fine solvers allow parareal convergence in lesser number of iterations. In addition to using an analytic source model, the size of the timesteps were increased in the coarse solver. The CORSICA solver uses a variable timestep and the size varied between 0.001 and 0.01 seconds for the fine solution. The coarse solver on the other hand allowed a variation of 0.004 to 0.6 seconds.

**Figure 1.** The powers generated by the analytic source term (red) in the coarse solver and the realistic source module (green) in the fine solver are chosen such that their values are similar.

**Figure 2.** The radial profiles of the total plasma current for the coarse (red) and fine (green) solvers are similar, so convergence is achieved in the least number of parareal iterations.

A typical serial run using the fine solver took 10.5 hours wallclock time to complete a 35 second simulation. A computational gain of 10.13 was achieved with 32 processors on application of the parareal algorithm. The plots in Fig.(3) represent solutions obtained using a single processor with the fine solver as well as that obtained using the parareal scheme. For the parareal solution in Fig.(3), a time slice of size 5 seconds was solved on each processor across a total of 16 processors. The different colors represent solutions across multiple processors. For this case, convergence was attained in 2 parareal iterations with an error tolerance of $5e - 3$.

**4. Summary and future work**

The parareal algorithm has been successfully applied to simulations of plasma scenarios which are highly complex systems with coupled nonlinearities. The approach of time parallelization has been applied to this type of simulations for the first time and the results are very promising. The reduced wall time makes studies of burning plasma scenarios at ITER more tractable.

Future studies may include more complex cases of plasma scenarios. The inclusion of free-boundary equilibrium control, ray tracing electron-cyclotron heating and full-wave ion-cyclotron heating serve as examples. Since temporal parallelization adds an extra domain of parallelization, it may be combined with spatial parallelization to attain maximum computational speed-up. The
(a) Single processor using the fine solver. (b) Parareal solution using 16 processors.

**Figure 3.** Variation of the poloidal beta $\beta_p$ with time. The different colors on the parareal solution represent solutions obtained across individual processors.

...reduced simulation times will allow for further systematic parameter variation studies needed for scenario evolution and discharge development and validation.

5. Acknowledgements

The simulations were performed at the NERSC supercomputing facility Carver. The authors are thankful for the computer time allocated.

6. Disclaimer

The views and opinions expressed here do not necessarily represent those of the ITER Organization.

References

[1] Crotinger J A et al, LLNL Report UCRL-ID-126284; NTIS PB2005-102154. (1997).
[2] Casper T A et al, 23rd Int. Conf. on Fusion Energy (Daejeon, Korea) ITR/P1-19. (2010).
[3] Casper T A et al, Nucl. Fusion 47 (2007) 825-832
[4] Lions J., Maday Y. and Turinici G., CR Acad. Sci. I - Math. 332 (7), (2001) pp. 661-668.
[5] Samaddar D., Newman D. E. and Sanchez R., J. Comp. Phys 229 (18), (2010) pp. 6558-6573.
[6] Berry L. A., W. Elwasif, J.M Reynolds-Barredo, D. Samaddar, R. Sanchez and D. E. Newman, J. Comp. Phys 2012 (In Press).
[7] Elwasif W.R et al, 4th IEEE Workshop on Many-Task Computing on Grids and Supercomputers, MTAGS 2011, (2011).
[8] Aubanel E. et al, Scheduling of tasks in the parareal algorithm, Parallel Computing 37 (3), (2011).
[9] Reynolds-Barredo J.M et al, J. Comp. Phys 2012 (In Press).
[10] Baffico L., Bernard S., et. al, Phys. Rev. E 66 (5), 057706 (2002).
[11] Bal G., Maday Y., Lecture Notes in Computer Science and Engineering, vol. 23, Springer, Berlin, 2002.
[12] Garrido I., Espedal M.S., Fladmark G.E., Proceedings of 15th International Conf. on Dom. Decom. Methods, vol. 40, Springer, Berlin, pp. 469-476.
[13] Fischer P.F., Hecht F., Maday Y., Proceedings of 15th International Conf. on Dom. Decom. Methods, vol. 40, Springer Verlag, 2004, (2004), pp. 433-440.
[14] Gander M.J., Haire E., Dom. Decom. Methods in Sc. and Eng. XVII, vol. 60, Springer, (2008) pp. 45-56.
[15] Samaddar D. et al, 39th EPS Conf. & 16th Int. Cong. on Plasma Phys., Stockholm, Sweden (2012) P2.140