

## Parallelization in time of numerical simulations of plasma turbulence

D. Samaddar<sup>1</sup>, D.E. Newman<sup>1</sup>, R. Sanchez<sup>2</sup>

<sup>1</sup> *University of Alaska, Fairbanks, AK, USA*

<sup>2</sup> *Oak Ridge National Laboratory, Oak Ridge, TN, USA*

Understanding turbulence and turbulent transport requires studying the evolution of the transport dynamics over very long time scales, typically thousands or tens of thousands of eddy decorrelation times, which is almost impossible with present computational techniques. This work successfully introduces a new and innovative approach to perform simulations of turbulent transport which might help overcome present difficulties. Turbulence simulations are extremely challenging from a computational view point due to the huge disparity (of the order of  $10^6$  -  $10^9$ ) of the timescales involved in the microturbulence that governs the dynamics and the plasma confinement times relevant in fusion research. Simplified approaches are often undertaken to overcome these difficulties. For example, the microturbulence is evolved through only a few tens of eddy decorrelation times and the plasma profiles are assumed constant, thus leading to a description of the dynamics using effective transport coefficients. However, even these simplified approaches are computationally very intensive [1].

With the advent of modern supercomputers, parallelizing these simulations over multiple processors allows some computational speedup. Prior to this work, parallelizing the space domain has been the only successful approach to utilize parallel processors. Such a technique has found limited success, as the state of the art fluid codes scale only up to a few thousand processors, while modern super computers offer more than a hundred thousand processors.

The parareal algorithm, first proposed by *Lions et al.* [2] in 2001, introduces a whole new avenue for parallelization, namely, parallelizing the time domain, to solve PDEs. This technique is extremely innovative and is very non intuitive for initial value problems of the kind discussed in this work. The algorithm requires two solvers, say F and G. F is the fine solver that is computationally expensive but gives a correct solution. G is the coarse solver that is computationally very fast with respect to F, but it gives a coarse and inaccurate solution to the same problem. G is always used in serial. The total simulation time T is split into mul-

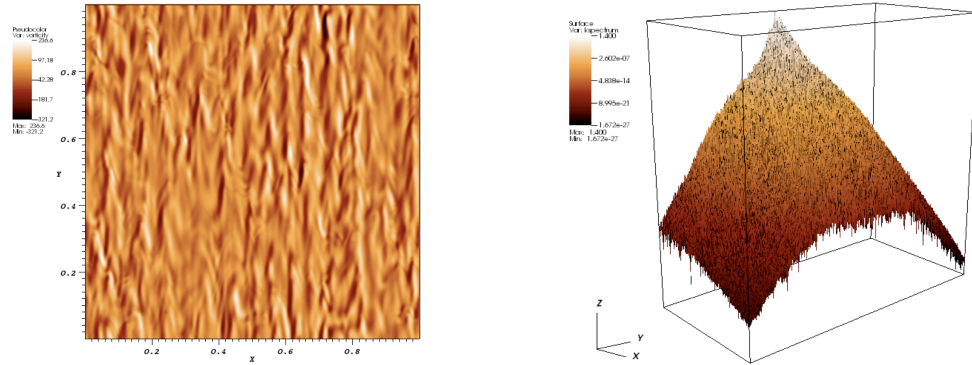
multiple time chunks  $\Delta T$  (so  $T = N\Delta T$ ,  $N$  being the total number of processors) and  $F$  is used in parallel by each processor to perform a simulation of length  $\Delta T$ . A detailed description of the algorithm is available in [2], [3]. Details of the application to this particular work are given in [4].

A dissipative trapped electron mode (DTEM) model with slab geometry is used in this application of the parareal algorithm. A uniform magnetic field along  $z$ -direction is assumed. The simulations are done using a pseudo spectral code with a resolution of  $385 \times 385$  (complex) modes in Fourier space ( $k$  space). The model used is described in Eq. 1.

$$\frac{\partial}{\partial t}(1 - \rho_s^2 \nabla_\perp^2) \tilde{\phi} + D \frac{\partial^2 \tilde{\phi}}{\partial y^2} + \frac{V_D}{2} \frac{\partial \tilde{\phi}}{\partial y} - \frac{4L_n D}{\epsilon^{1/2}} \left[ \nabla_\perp \left( \frac{\partial \tilde{\phi}}{\partial y} \right) \times z \right] \cdot \nabla_\perp \tilde{\phi} - \rho_s^3 C_s \nabla \tilde{\phi} \times z \cdot \nabla \nabla_\perp^2 \tilde{\phi} = 0 \quad (1)$$

where  $\tilde{\phi}$  is the fluctuating potential,  $V_D = (\rho_s C_s) L_n^{-1}$  is the effective diamagnetic drift velocity and  $T_i$  is the ion temperature.  $B$  represents the magnetic field strength.  $D = V_D^2 / 4\nu_{\text{eff}}$ , with  $\nu_{\text{eff}}$  being the effective collision frequency of ion electron collisions. The last two terms on the left hand-side of Eq. 1 represent the ExB and polarization non-linearities respectively. When Fourier transformed, the ExB and polarization nonlinear terms take the forms

$$\frac{4iL_n D}{\epsilon^{1/2}(1 + \rho_s^2 k_\perp^2)} \sum_{\mathbf{k}'} k'_y (\mathbf{k} \times \mathbf{k}' \cdot \mathbf{z}) \tilde{\phi}_{\mathbf{k}} \tilde{\phi}_{\mathbf{k}-\mathbf{k}'} \quad \text{and} \quad \frac{\rho_s^3 C_s}{2} \sum_{\mathbf{k}'} k^2 (\mathbf{k} \times \mathbf{k}' \cdot \mathbf{z}) \tilde{\phi}_{\mathbf{k}} \tilde{\phi}_{\mathbf{k}-\mathbf{k}'}, \text{ respectively.}$$

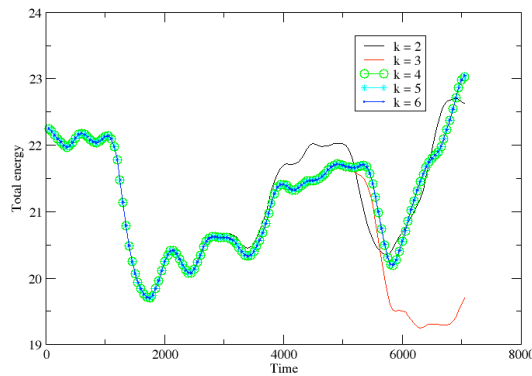


**Fig 1. (left) Vorticity field for a typical simulation with only E X B non linearity ; (right) Power spectrum (lin-log scale) for only E X B non linearity.**

This model has already been extensively studied in [5], [6], [7] and is used here as a test bed for studying the performance of the parareal technique for turbulence simulations. Multiple cases of this model have been used here where either one or both non linearities are present. Fig. 1 (left) shows the vorticity field of the turbulent system with only the E X B non linearity which consists of eddies of multiple scales that non linearly interact with each other.

Fig. 1 (right) is a plot of the power spectrum of the same case showing that most of the energy is concentrated at low  $k$  modes.

For the purpose of the application of the parareal application, multiple options for the coarse solver  $G$  were explored. Using a reduced grid size in Fourier space (201 X 201) and bigger time steps  $dt_G > dt_F$  ( $dt_G$  and  $dt_F$  being the timesteps in  $G$  and  $F$  respectively) gave the best gains. In addition, the VODPK solver package [8] used in  $F$  was replaced by 2<sup>nd</sup> and 4<sup>th</sup> order Runge Kutta solvers. Fig. 2 is an example of the successful application of the parareal algorithm to turbulence. It is a plot of the total energy of the system (with the  $E \times B$  non linearity) varying with time,  $t$  where the solutions converge with increasing parareal iterations,  $k$ . A gain of 8.8 was observed with  $N=88$ , which is remarkable for a system as complex as this. Similar convergence have been observed with cases where only the polarization non linearity or both non linearities were present.



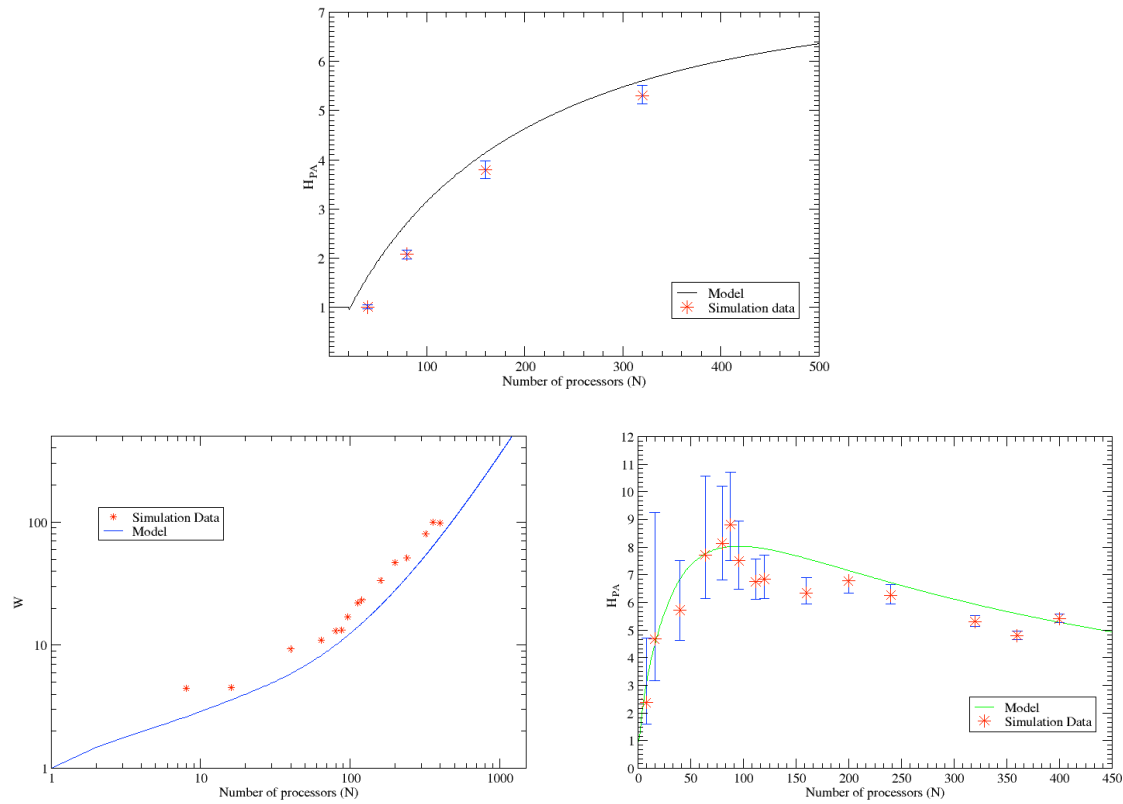
**Fig 2. Total energy as a function of time for a simulation with  $N = 88$ ,  $\Delta T = 80$  for different parareal iterations,  $k$ . The coarse solver  $G$  is a 4<sup>th</sup> order Runge-Kutta with  $dt_G=40dt_F$  and grid size in  $k$ space being 201X201. Convergence is observed in  $k=5$ .**

In order to attain a better understanding of the convergence of the solutions for a turbulent system, a model (discussed in [4]) was built to describe the computational gain,  $H_{PA}$ :

$$H_{PA}(N) = \frac{T_F^{ser}(T)}{T_{PA}} = \left[ k_s(N) \left( \frac{1}{\beta} + \frac{1}{N} \right) \right]^{-1} \quad (2)$$

where  $\beta = \frac{T_F^{ser}(T)}{T_G^{ser}(T)}$  with  $T_F^{ser}(T)$  and  $T_G^{ser}(T)$  being the computational time for using in serial, solvers  $F$  and  $G$  respectively.  $k(N)$  represents the number of cycles for convergence and has been modeled analytically by fitting simulation data. Both weak and strong scaling studies were done. Fig. 3 (top) represents the result for a strong scaling study where  $T$  was constant but  $\Delta T$  varied with  $N$ . Figs. 3 (bottom left) and 3 (bottom right) are the results for a weak scaling study where  $\Delta T$  was kept constant but  $T$  varied with  $N$ . W in Fig. 3 (bottom left) is the

work per processor and an analytic expression for it can be derived from the model. Both strong and weak scaling results show good agreement with the model.



**Fig 3. Results with coarse solver G being a 4<sup>th</sup> order Runge-Kutta with  $dt_G=40dt_F$  and grid size in kspace being 201X201 : (top)  $H_{PA}$  as a function of N for strong scaling study.; (bottom left)  $W$  as a function of N for weak scaling; (bottom right)  $H_{PA}$  as a function of N for weak scaling study.**

In conclusion, the parareal technique successfully works for a turbulent system and is accompanied by a sizable gain. This allows parallelization of the time domain, which may be combined with existing space parallelization to achieve optimum computational gain.

### References

- [1] J. Dahlburg J. Corones, D. Batchelor, R. Bramley, M. Greenwald, S. Jardin, S. Krashennikov, A. Laub, J. N. Leboeuf, J. Lindl, W. Lokke, M. Rosenbluth, D. Ross and D. Schnackl, *J. Fus. Energy* **20** (4), 135-196 (2001).
- [2] J. Lions, Y. Maday and G. Turinici, *C. R. Acad. Sci. paris-Serie*, **332** (7), 661-668 (2001).
- [3] L. Baffico, S. Bernard, Y. Maday, G. Turinici and G. Zerah, *Phys. Rev. Lett.*, **66** (5), (2002).
- [4] D. Samaddar, D. E. Newman and R. Sanchez, *J. Comp. Phys.*, doi:10.1016/j.jcp.2010.05.012 (2009).
- [5] Y. -M. Liang, P. H. Diamond, X. H. Wang, D. E. Newman and P. W. Terry, *Phys. Fluids*, **B5** (4), 1128-1139 (1993).
- [6] D. E. Newman, P. W. Terry and P. H. Diamond, *Phys. Fluids*, **B4** (3), 599-610 (1992).
- [7] D. E. Newman, P. W. Terry, P. H. Diamond and Y. M. Liang, *Phys. Fluids*, **B5** (4), 1140-1153 (1993).
- [8] A. Hindsmarsh, *Report UCID-30001, Rev. 3, LLNL*, (1974).