Proxy-equation paradigm: A strategy for massively parallel asynchronous computations

Ankita Mittal and Sharath Girimaji

Phys. Rev. E **96**, 033304 — Published 8 September 2017

DOI: 10.1103/PhysRevE.96.033304

"Proxy equation" paradigm - A strategy for massively-parallel asynchronous computations

Ankita Mittal* and Sharath Girimaji†
Department of Aerospace Engineering, Texas A&M University, College Station, Texas - 77843

Massively parallel simulations of transport equation systems call for a paradigm change in algorithm development to achieve efficient scalability. Traditional approaches require time synchronization of processing elements (PEs) which severely restricts scalability. Relaxing synchronization requirement introduces error and slows down convergence. In this paper, we propose and develop a novel 'proxy equation' concept for a general transport equation that (i) tolerates asynchrony with minimal added error, (ii) preserves convergence order and thus, (iii) expected to scale efficiently on massively parallel machines. The central idea is to modify *a priori* the transport equation at the PE boundaries to offset asynchrony errors. Proof-of-concept computations are performed using a one-dimensional advection (convection) diffusion equation. The results demonstrate the promise and advantages of the present strategy.

I. INTRODUCTION

Massively parallel computing capability has the potential to reduce the computational elapse time of simulating large-sized complex physical systems. However, computing time-evolution of transport equations represents a special challenge. Such systems require overhead communication for synchronization among processing elements (PEs). Although advances in modern hardware and software have made it possible to communicate asynchronously [1], mathematical level global synchronization is still a requirement for current numerical schemes. This imposed synchronization increases the idle wait time of PEs. Also, hardware and software failure rates increase with increasing number of PEs [2] further adding to the PE load imbalance. Thus, the requirement of global synchronization throughout the computational domain leads to poor scaling characteristics with increasing system size [3, 4]. These imbalances become especially critical at exascale computing where millions of cores are expected to operate synchronously [5]. Aditya et al. [6], show that the distribution of delay among processors widens as we increase the number of PEs. Therefore, it is important to develop computational strategies that tolerate asynchrony among PEs.

Typical synchronous computations (traditional methods) of transport equations incur truncation and round-off error. Asynchronous computations which relax the mathematical level synchronization develop additional error due to the delay at PE boundaries called the delay error, Fig. 1. Currently a few numerical schemes have been developed to improve the accuracy of asynchronous computations. Recently developed asynchrony-tolerant numerical scheme [7] and the delayed difference scheme [8] attempt to counteract the delay error due to asynchrony by modifying the discretization scheme. However, the delay error still continues to be significant. Moreover, these approaches present difficulty when adapting existing solvers to asynchronous computations. Modified numerical schemes also increase the stencil size which adds to the communication time.

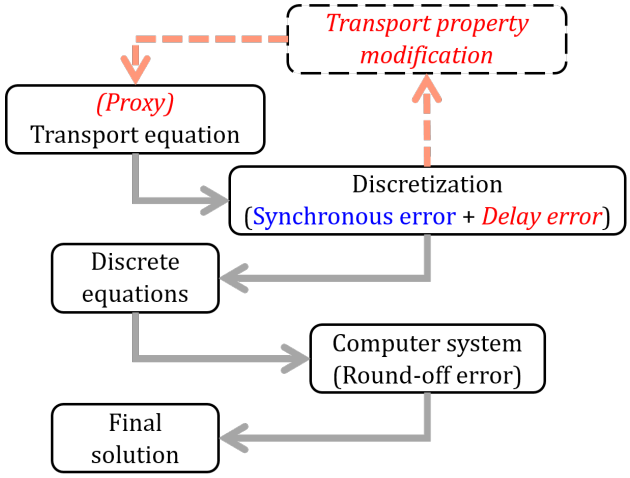


FIG. 1: Schematic of synchronous and asynchronous (in italics) errors. Logic of proxy equation is also shown.

In this paper, we present an alternate approach to mitigate the effect of asynchrony. We start with the tenet that the function of asynchrony-tolerant computational strategy is to render the delay error to be of a lower order than the synchronous discretization error. To accomplish this, we propose modifying the governing equation as a function of delay, rather than changing the numerical scheme. The proposed modification is conceptually similar to the work of Warming [9] and VonNeumann [10]. In that case, the modification was introduced to understand/improve robustness and stability of a synchronous scheme. In this paper, we develop the modified or *proxy* equation for the purpose of offsetting delay errors. The italicized (and red) parts of Fig. 1 identify the logic of the proxy equation approach.

* ankitami@tamu.edu
† girimaji@tamu.edu

II. PROXY EQUATION METHODOLOGY

The advection (convection) diffusion reaction equation represents one of the most common transport system in physics and engineering. Of these three effects, advection and diffusion include spatio-temporal communication and reaction is typically a local process. Thus, in this paper we will restrict ourselves to advection and diffusion phenomena as described by

$$\frac{\partial u_i}{\partial t} = -c_j \frac{\partial u_i}{\partial x_j} + \alpha \frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{1}$$

where 'c_j' represents the wave speed in each direction and 'α' represents the diffusion coefficient or viscosity. We will first consider the linear case wherein c_j is constant and then proceed to the non-linear case.

We analytically characterize the effect of asynchrony on advection and diffusion processes in isolation by examining the one-dimensional wave and diffusion equations

$$\begin{aligned} \frac{\partial u}{\partial t} &= -c\frac{\partial u}{\partial x} \ \text{ leading to } \ u(x,t) = Ae^{\iota\lambda(x-ct)} \\ \frac{\partial u}{\partial t} &= \alpha\frac{\partial^2 u}{\partial x^2} \ \text{ leading to } \ u(x,t) = Ae^{\iota\lambda x}e^{-\alpha\lambda^2 t}. \end{aligned} \tag{2}$$

The exact solutions of wave and diffusion equations are indicated for the initial condition of $u(x,0) = Ae^{\iota\lambda x}$. Here '$\lambda$' is the wavenumber of the initial field. Now, let us examine the computational solution of the above equations wherein a time delay of δt is introduced. Under delay conditions, the solution has to be inferred from the time instance $t - \delta t$. From the form of the analytical solution it is evident that the effect of the delay error can be completely offset, if the wave-speed c_* and diffusion coefficient α_* are redefined to satisfy

$$ct \equiv c_*(t - \delta t) \qquad \alpha t \equiv \alpha_*(t - \delta t) \tag{3}$$

leading to

$$c_* \equiv \frac{c}{1-D} \qquad \alpha_* \equiv \frac{\alpha}{1-D}. \tag{4}$$

Here, D is the delay correction factor. Then the proxy equation that can offset the delay effect can be written as

$$\frac{\partial u}{\partial t} = -c_*\frac{\partial u}{\partial x} \quad \text{and} \quad \frac{\partial u}{\partial t} = \alpha_*\frac{\partial^2 u}{\partial x \partial x}. \tag{5}$$

Clearly, the correction is a function of the degree of delay. This simple analysis demonstrates the manner of modification needed to mitigate advection and diffusion errors independently. We now proceed to derive the delay correction factor for coupled computations.

III. CORRECTION FACTOR DETERMINATION

The exact delay correction factor, D is determined by performing a truncation error analysis. Since the communication between PEs depend on the numerical scheme used, D will vary with the scheme used as well as other factors such as the problem parameters, degree of delay and grid size. To illustrate the process, we proceed with a one-dimensional advection (convection) diffusion equation

$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial t} + \alpha\frac{\partial^2 u}{\partial x^2} \tag{6}$$

using a basic forward in time and central in space (FTCS) scheme. The synchronous stencil is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -c\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} + \alpha\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + E_i \tag{7}$$

where E_i is the truncation error. To examine asynchronous effects we use the analytical approach as established in Donzis and Aditya [7]. Consider two processors PE0 and PE1 wherein PE1 is delayed by $\tilde{k}$ timesteps in comparison to PE0 (right processor delayed). The advection-diffusion operator now has the form

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -c\frac{u_{i+1}^{n-\tilde{k}} - u_{i-1}^n}{2\Delta x} + \alpha\frac{u_{i+1}^{n-\tilde{k}} - 2u_i^n + u_{i-1}^n}{\Delta x^2} + E_i^{\tilde{k}}. \tag{8}$$

Here $E_i^{\tilde{k}}$, given by

$$\begin{aligned} E_i^{\tilde{k}} = &\frac{\Delta t}{2}\ddot{u} + \left(c\frac{\Delta x^2}{6}u''' - \alpha\frac{\Delta x^2}{12}u''''\right) + \\ &\left(\tilde{k}\frac{\alpha\Delta t}{\Delta x^2} - \tilde{k}\frac{c\Delta t}{2\Delta x}\right)\dot{u} + ..., \end{aligned} \tag{9}$$

is the total error. In Eq. (9), $\dot{u}$ represents derivative in time and u' represents derivative in space. The number of dots/primes depicts the order of the derivative. The first part of the error is due to truncation while the second part is the delay error. The delay error $O(\Delta t/\Delta x^2)$ is of lower order than the truncation error $O(\Delta x^2), O(\Delta t)$ and must be eliminated or reduced. The leading order terms in the delay error involve $\dot{u}$ which also appears in the original equation. In order to reduce the delay error and improve accuracy, the original equation can be modified as

$$\left(1 - \tilde{k}\left(r_\alpha - \frac{r_c}{2}\right)\right)\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x} + \alpha\frac{\partial^2 u}{\partial x \partial x}. \tag{10}$$

If the exact delays are known then Eq. (10) presents one from of the proxy equation for Eq. (6). For a case where exact delays are unknown, a similar approach can be applied using a probabilistic value for the delay. Eq. (10) can be interpreted as either a system with added mass or modified time scale. Effectively, the effects of asynchrony are pre-modeled and corrected by adding inertia to the system or by modifying the time scale near PE boundaries. The added mass coefficient or the delay correction factor for a FTCS scheme is given by

$$D \equiv \tilde{k}\left(r_\alpha - \frac{r_c}{2}\right), \quad r_\alpha = \frac{\alpha\Delta t}{\Delta x^2}, \quad r_c = \frac{c\Delta t}{\Delta x}. \tag{11}$$

Another representation of the proxy equation can be obtained by dividing though by $(1-D)$ and can be written as

$$\frac{\partial u_i}{\partial t} = -\frac{c_j}{(1-D)}\frac{\partial u_i}{\partial x_j} + \frac{\alpha}{(1-D)}\frac{\partial^2 u_i}{\partial x_j \partial x_j}. \tag{12}$$

Eq. (12) represents the form of the proxy equation in line with the methodology discussed earlier. Here the transport parameters (i.e. advection speed and diffusion coefficient) are pre-corrected to reduce the errors due to asynchrony.

The above calculations are based on a right delay assumption. Performing a similar analysis for a left delay shows $D = \tilde{k}\left(r_\alpha + \frac{r_c}{2}\right)$. So the delay correction factor for a general 1D advection-diffusion equation solved using a FTCS scheme becomes

$$D \equiv \tilde{k}\left(r_\alpha \pm \frac{r_c}{2}\right). \tag{13}$$

For a three-dimensional transport equation Eq. (1), the delay correction factor can be found in a similar manner. For a FTCS scheme it is given by

$$D \equiv \sum_{r=1}^{d} \tilde{k}_r \left(r_{\alpha,r} \pm \frac{r_{c,r}}{2}\right), \quad r_{\alpha,r} = \frac{\alpha \Delta t}{\Delta x_r^2}, \quad r_{c,r} = \frac{c_r \Delta t}{\Delta x_r} \tag{14}$$

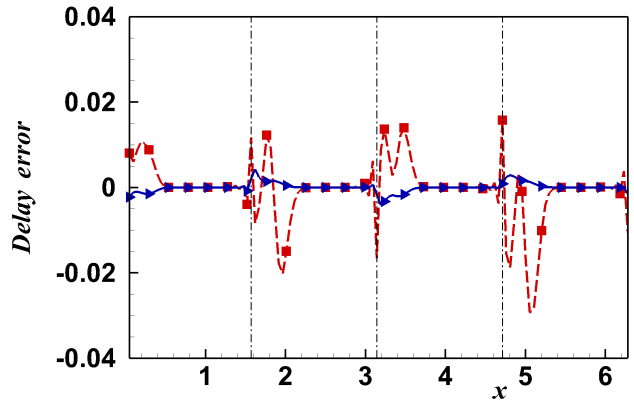
where 'd' is the spacial dimension of the problem. Since this approach removes the leading order delay error, the order of accuracy of asynchronous computations improve by one. Higher order corrections will be considered in the future. Also, the method as presented cannot be applied in the presence of sharp discontinuities like shocks. However, for those cases a similar philosophy can be used with a different pre-correction factor. In the present work we have only addressed well-behaved functions, as have previous works in literature.

IV. STABILITY ANALYSIS

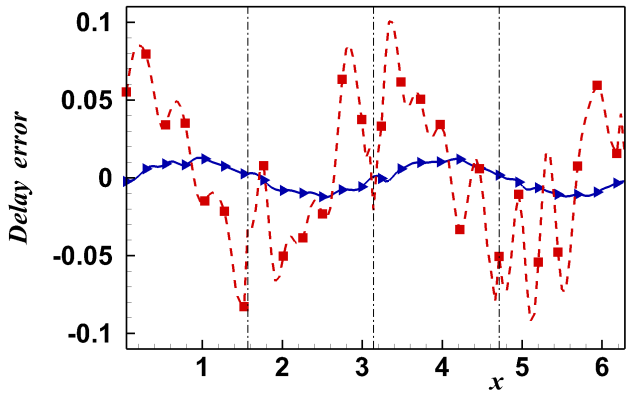
Since a delay appears only at PE boundaries, performing an exact stability analysis is difficult. In order to determine a hard upper bound we consider the limiting case where delay appears at all grid points. We will focus on the stability of the proxy equation under the ∞-norm [7].

The modified transport coefficients in the proxy equation are higher in magnitude compared to the original which leads to a restricted stability range. Performing some mathematical manipulation we can reduce the stability of the proxy equation (for the FTCS scheme) to be as follows

$$\begin{aligned} \frac{r_{c,r}}{2} &\leq r_{\alpha,r} \\ \max_{r=1,\ldots,d} r_{\alpha,r} &\leq \frac{1}{2d[1+\max_r(\tilde{k}_r)]}. \end{aligned} \tag{15}$$



(a)



(b)

FIG. 2: The delay error, $E_i^{\tilde{k}} - E_i$, for the original equation (red squares) and the proxy equation (blue triangles) at (a) $tc/l = 0.08$ (b) $tc/l = 1.0$. Simulation parameters: N = 128, P = 4, $c = 1$, $\alpha = 0.01$, $r_\alpha = 0.1$ with $\{p_0, p_1\} = \{0.3, 0.7\}$

For no delay or $\tilde{k}_r = 0$, we recover the original stability range. As the level of delay ($\tilde{k}$) increases, the stability range becomes more restricted which also sets a limit on the amount of delay that can be allowed. Other numerical approaches developed for asynchronous computations also present a similar issue with stability of the system. For stability, the upper bound in Eq. (15) is sufficient but not necessary. A true upper bound can only be determined numerically.

V. ILLUSTRATIONS

We now present proof-of-concept numerical simulations using the one-dimensional advection-diffusion equation, Eq. (6) and its proxy equation, Eq. (12) with the delay correction factor as in Eq. (13). Following Donzis and Aditya [7] a periodic domain of size $l = 2\pi$ is con-

sidered with an initial condition given by

$$u(x,0) = \sum_{\lambda} A(\lambda)\sin(\lambda x + \phi_\lambda) \tag{16}$$

i.e. a sum of sinusoidal waves. Here, λ is the wavenumber with $A(\lambda)$ and ϕ_λ the amplitude and phase angle of each wave. The analytical solution to Eq. (6) with initial condition as in Eq. (16) is given by

$$u_a(x,t) = \sum_{\lambda} e^{-\alpha\lambda^2 t} A(\lambda)\sin(\lambda(x-ct) + \phi_\lambda). \tag{17}$$

We analyze the asynchronous effects using a similar computational approach as established in [7]. The results are presented for different grid sizes $N = \{32, 128, 256, 512, 1024\}$ and number of processors $P = 4$. In [6], the authors examined the delay statistics showing the most probable delay to be one until 128 processors. Therefore, for these simulations the maximum allowable delay is restricted to one. As a result, we will have two delay levels i.e. a delay of one time step or no delay at all. The delay is simulated in a manner identical to [7] using random number generators: p_0 represents the probability of no delay and p_1 represents the probability of one timestep delay ($p_0 + p_1 = 1.0$). $p_0 = 1.0$ is the synchronous case and $p_0 = 0.0$ corresponds to the most asynchronous case. Results are presented for different values of $p_0 = \{1.0, 0.6, 0.3, 0.0\}$ in increasing order of asynchrony. Since the correction depends on the actual value of the delay, the outcome of this method is independent of the statistical description of the delay. Therefore, changing the approach to simulate delay would not have any effect on the findings of this paper. Ensemble averages are taken over different initial phases for estimating order of error. Please refer to the FORTRAN code provided as supplemental material for further implementation details [11].

We present results for wavenumber, $\lambda = 2$. The plots in Fig. 2 compare the delay error obtained when solving the original 1D advection-diffusion equation, Eq. (6) and its proxy equation Eq. (12) under asynchronous computing using FTCS scheme at different times. The results clearly show the delay error initially manifests as a spike at the PE boundaries, slowly diffusing to interior grid points after a few time steps. At initial times ($tc/l = 0.08$) the delay error for the proxy equation is much lower in magnitude, Fig. 2(a). Even when the simulation is run for considerably longer duration ($tc/l = 1.0$), the delay error for the proxy equation stays considerably small in comparison to the original, Fig. 2(b). In the present approach the delay error is neutralized as soon as it appears near the PE boundaries and thus, it does not diffuse to interior points at later time steps.

The scaling of average error (ensemble and space average) with increase in grid points (decreasing grid size) is shown in Fig. 3. The errors are evaluated at normalized time $tc/l = 1.0$. The synchronous case or $p_0 = 1.0$

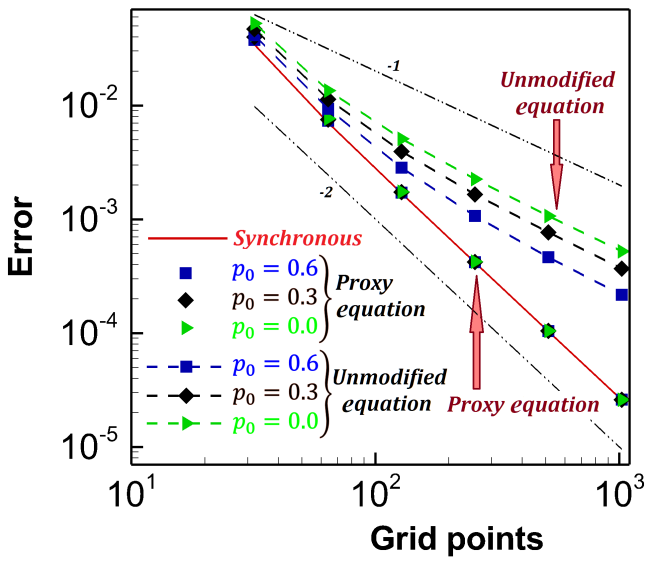


FIG. 3: Scaling of average error with number of grid points for proxy equation and unmodified equation at $tc/l = 1.0$ with $r_\alpha = 0.1$, $c = 1.0$, $\lambda = 2$ and $\alpha = 0.1$. All proxy equation solutions fall on the same line.

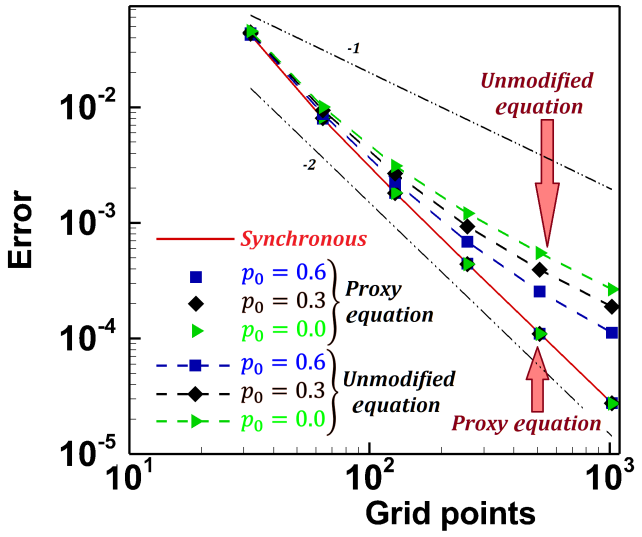
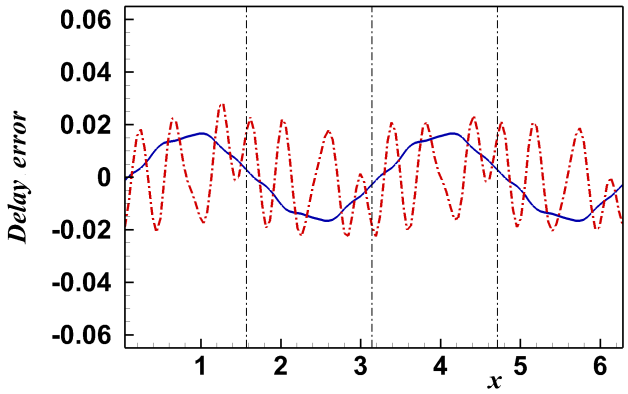
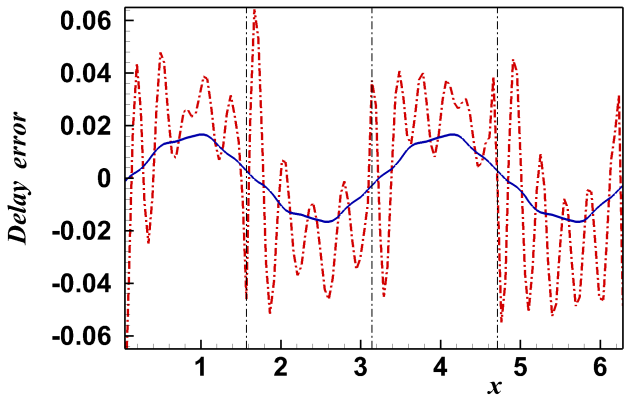


FIG. 4: Scaling of average error with number of grid points for proxy equation and unmodified equation at $tc/l = 1.0$ with $r_\alpha = 0.1$, $c = 1.0$, $\lambda = 8$ and $\alpha = 0.1$. All proxy equation solutions fall on the same line.

shows a second order scaling which is expected. Solving the original equation asynchronously without modification clearly shows a drop in the order of accuracy to first order. However, when the proxy equation is used, the order of accuracy remains second order recovering the order of the original scheme. Fig. 4 shows the scaling of average error at a higher initial wavenumber ($\lambda = 8$).



(a)



(b)

FIG. 5: Delay error comparisons at $tc/l = 1.0$ for the proxy equation (solid blue line) and the original equation with (a) Asynchrony tolerant scheme, (b) Delay difference scheme (dot dashed red). Simulation Parameters: N = 128, P = 4, $c = 1.0$, $\alpha = 0.01$, $r_\alpha = 0.1$ with $\{p_0, p_1\} = \{0.3, 0.7\}$.

It clearly shows that even at higher wavenumbers the proxy equation approach is valid and retains the order of accuracy for asynchronous computations.

A. Error Comparisons

Fig. 5(a) compares the delay error of the proxy equation with the asynchrony-tolerant numerical scheme [7]. It clearly shows the delay error from the proxy equation approach is much smother and also lower in magnitude. Similar observations can be made when comparing the delayed time difference scheme [8] with the proxy equation approach, Fig. 5(b). Overall, it can be seen that the proxy equation approach is significantly better at capturing the inherent physics of the problem.

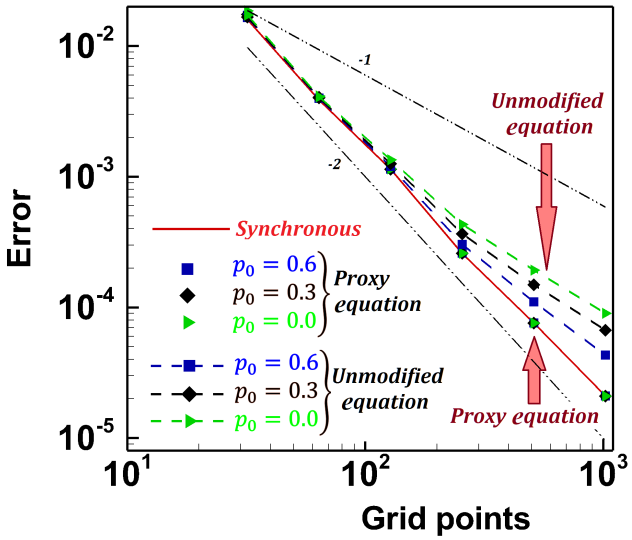


FIG. 6: Scaling of average error with number of grid points for proxy equation and unmodified Burgers equation at $tc/l = 1.0$ with $r_\alpha = 0.1$ and $\alpha = 0.1$. All proxy equation solutions fall on the same line.

B. Non-Linearity Effects

Now we examine the effectiveness of the proxy equation approach for a non-linear system using a one-dimensional viscous Burgers equation given by

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial t} + \alpha\frac{\partial^2 u}{\partial x^2}. \tag{18}$$

The corresponding proxy equation can be written as follows

$$\begin{aligned}(1-D)\frac{\partial u}{\partial t} &= -u\frac{\partial u}{\partial t} + \alpha\frac{\partial^2 u}{\partial x^2},\\ D &= k\left(r_\alpha \pm \frac{r_u}{2}\right).\end{aligned} \tag{19}$$

Fig. 6 shows average error scaling with grid resolution for the proxy equation Eq. (19) and the Burgers equation Eq. (18) under different levels of asynchrony. The results for the unmodified equation, Eq. (18) again reduce to first order while the ones for the proxy equation, Eq. (19) remain at second order as expected. The synchronous case is the red dashed line which is also second order.

This paper presents proof-of-concept simulations of the proxy-equation approach. Implementation of the approach on a massively parallel machine will be undertaken in collaboration with a computer science team lead by the developers of STAPL [12] (Standard Template Adaptive Parallel Library). STAPL is a framework for developing parallel programming on shared and distributed memory platforms. STAPL allows users to achieve high scalability while hiding many details specific to parallel programming.

VI. CONCLUSIONS

In summary, we developed a modified equation or *proxy equation* approach to mitigate the effects of asynchronous computations of transport equations on massively parallel computational systems. A physical framework has been established to determine the degree of modification as a function of time delay between the processing elements (PEs). In principle, the proposed approach is similar to the technique of [9, 10] but with the intent of improving accuracy in the face of asynchronous computing. The proxy equation approach eliminates the need for any changes at the numerical scheme level making it easier to extend existing solvers to asynchronous computations. The advantages of the approach are demonstrated for both linear and non-linear advection-diffusion systems.

[1] G. Shainer, T. Wilde, P. Lui, T. Liu, M. Kagan, M. Dubman, Y. Shahar, R. Graham, P. Shamis, and S. Poole, Computer Science - Research and Development **28**, 119 (2013).

[2] I. P. Egwutuoha, D. Levy, B. Selic, and S. Chen, Journal of Supercomputing **65**, 1302 (2013).

[3] S. Jagannathan and D. A. Donzis, in *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond* (ACM, 2012) p. 23.

[4] M. Lee, N. Malaya, and R. D. Moser, in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13 (ACM, New York, NY, USA, 2013) pp. 61:1–61:11.

[5] J. Dongarra, P. Beckman, T. Moore, P. Aerts, G. Aloisio, J.-C. Andre, D. Barkai, J.-Y. Berthou, T. Boku, B. Braunschweig, F. Cappello, B. Chapman, X. Chi, A. Choudhary, S. Dosanjh, T. Dunning, S. Fiore, A. Geist, B. Gropp, R. Harrison, M. Hereld, M. Heroux, A. Hoisie, K. Hotta, Z. Jin, Y. Ishikawa, F. Johnson, S. Kale, R. Kenway, D. Keyes, B. Kramer, J. Labarta, A. Lichnewsky, T. Lippert, B. Lucas, B. Maccabe, S. Matsuoka, P. Messina, P. Michielse, B. Mohr, M. S. Mueller, W. E. Nagel, H. Nakashima, M. E. Papka, D. Reed, M. Sato, E. Seidel, J. Shalf, D. Skinner, M. Snir, T. Sterling, R. Stevens, F. Streitz, B. Sugar, S. Sumimoto, W. Tang, J. Taylor, R. Thakur, A. Trefethen, M. Valero, A. Van Der Steen, J. Vetter, P. Williams, R. Wisniewski, and K. Yelick, International Journal of High Performance Computing Applications **25**, 3 (2011).

[6] K. Aditya, D. A. Donzis, and T. Hoefler, in *High Performance Computing, Networking, Storage and Analysis (SC)* (2013).

[7] D. A. Donzis and K. Aditya, Journal of Computational Physics **274**, 370 (2014).

[8] D. Mudigere, S. D. Sherlekar, and S. Ansumali, Physical Review Letters **113**, 218701 (2014).

[9] R. Warming and B. Hyett, Journal of Computational Physics **14**, 159 (1974).

[10] J. VonNeumann and R. D. Richtmyer, Journal of Applied Physics **21**, 232 (1950).

[11] See Supplemental Material at `http://` for the FORTRAN code implementation of proxy equation approach.

[12] `https://parasol.tamu.edu/stapl/`.