XPS: FULL: DSD: Asynchronous PDE Algorithms for Turbulent Flows at Exascale

D. A. Donzis[1], L. Rauchwerger[2], R. Bhattacharya[1], S. Girimaji[1], N. Amato[2]

[1]Department of Aerospace Engineering, Texas A&M University
[2]Department of Computer Science, Texas A&M University

Turbulence and Advanced Computations Lab (TACL)
Parasol Laboratory
Laboratory for Uncertainty Quantification

NSF XPS Workshop, Arlington, VA
May 31-June 2, 2015

Acknowledgments: NSF, XSEDE (TACC/NICS), NERSC, ORNL, ANL

Introduction

- **Turbulence**: most common state of fluid motions in nature and engineering
  - aerodynamics, clouds, ocean, reacting flows, pollutant dispersion, origin of universe, etc.
  - *"Last unsolved problem in classical physics"* - Richard Feynman
- Unsteady, 3D, stochastic, wide range of nonlinearly interacting scales
- Direct Numerical Simulations: resolve all scales in time and space
  - Computationally extremely expensive
  - Communication/sync: 50-80% time on current PF machines[1]

[1] Jagannathan & Donzis (XSEDE 2012), Sankaran et al. (SC 2012), Lee et al. (SC 2013)

Turbulence simulations at Exascale

Issues with current methods: finite diff./vol./elem., spectral,...

- Communications
- Synchronizations at multiple levels (especially math!)
- Variability in PEs performance, system failures, ...

Need new approach

- Can we relax data synchronization at mathematical level?
- Asynchronous numerical schemes: virtually no work on PDEs...
- **Objective: trade-off accuracy and performance quantitatively and predictably**

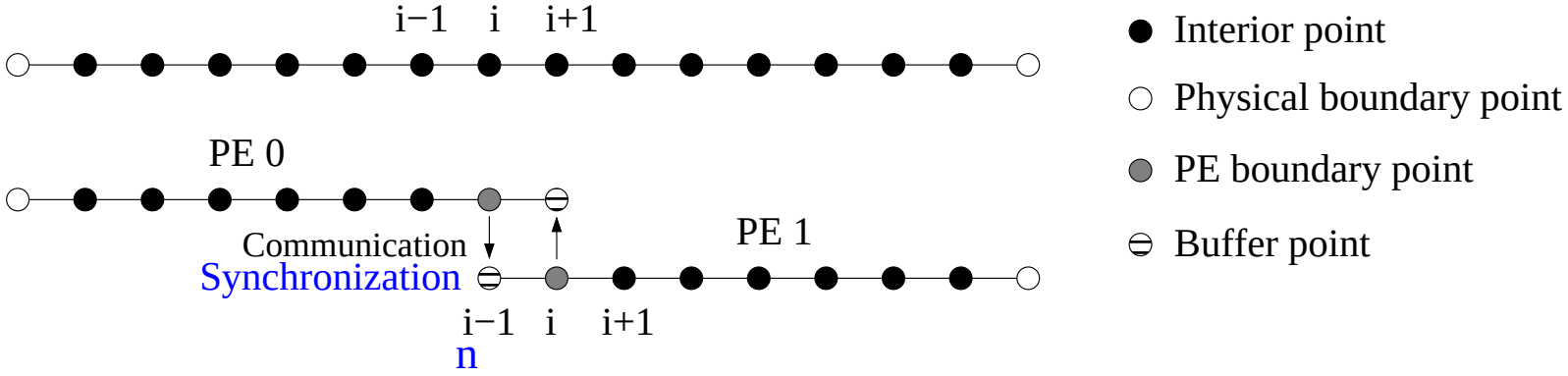
Main idea

Consider the simple 1D heat equation

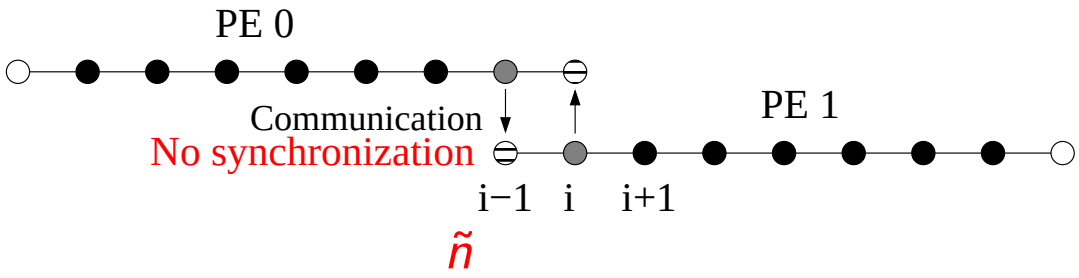
$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Discretized equation - forward in time and central in space

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \mathcal{O}(\Delta t, \Delta x^2)$$



Relax synchronizations

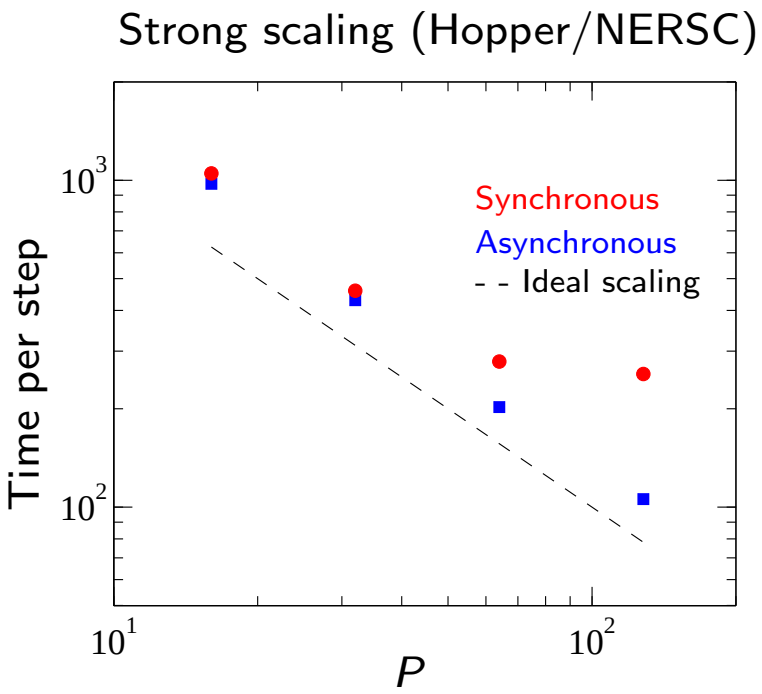


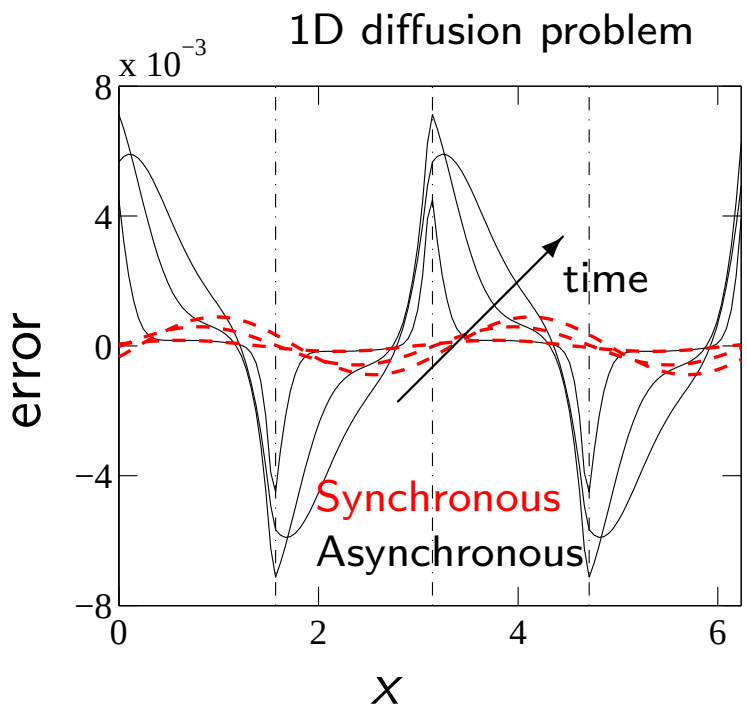
- $u_{i-1}^{\tilde{n}}$ is the available data at $(i-1)$th grid point

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^{\tilde{n}}}{\Delta x^2} + \mathcal{O}(\Delta t^?, \Delta x^?)$$

- PE boundary points are at $\tilde{n} = n - \tilde{k}$ th level: random!
- **Properties of such a scheme? Stable? Consistent? Accurate? How to characterize errors? Implementation?...**
- Need strong collaboration: numerical analysis, dynamical systems, computer science, physics of fluids

Results





Parameters:

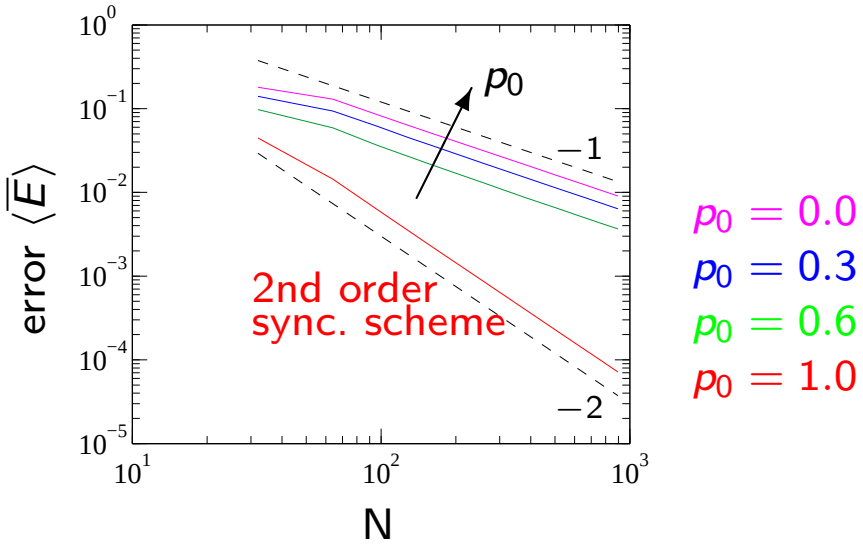
- # of PEs: $P = 4$
- Allowed delays: $L = 2$
- probabilities: $p_k = \{0.6, 0.4\}$

Accuracy - Overall

- How to define trunc. error? Not homogeneous in space & random
  - Statistically: spatial ($\langle E \rangle$) and ensemble ($\overline{E}$) averages

- We've shown that asymptotic truncation error with asynchrony is

$$\langle \overline{E} \rangle \sim \overline{\tilde{k}} \frac{P}{N} \sim \overline{\tilde{k}} P \Delta x$$



- Error depends on standard numerical parameters (e.g. N) but also on (i) system details (delays stats.), (ii) how problem is scaled up

- Strong scaling: $\langle \overline{E} \rangle \sim \mathcal{O}(\Delta x)$ — Weak scaling: $\langle \overline{E} \rangle \sim \mathcal{O}(1)$
  - Verified by numerical experiments

(Donzis, Aditya JCP 2014)

Asynchrony-Tolerant (AT) schemes

- Idea: study truncation error and eliminate terms that affect accuracy

Order recovery in space (2nd order)

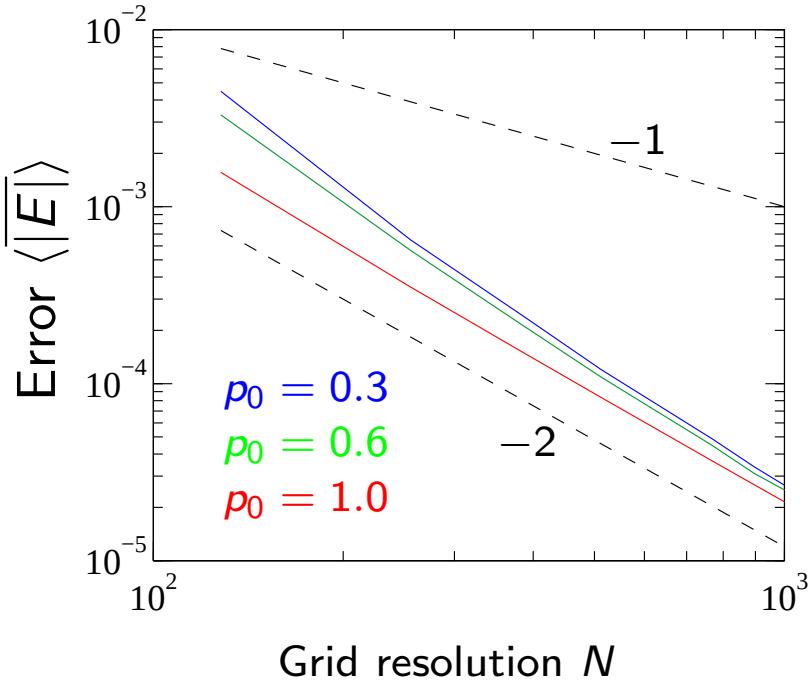
$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+2}^{n-k} - u_{i+1}^{n-k} - u_i^n + u_{i-1}^n}{2\Delta x^2}$$

Order recovery in time (4th order)

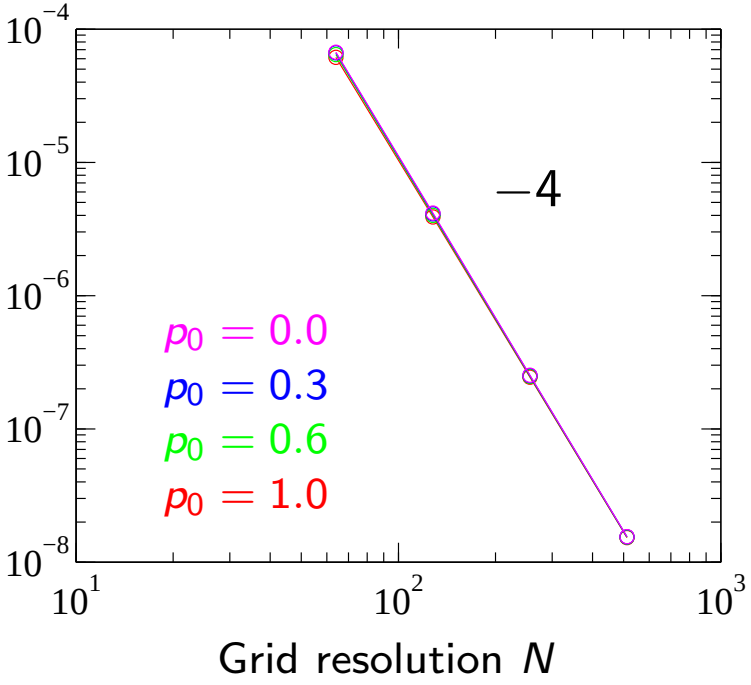
$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} \approx \frac{1}{24\Delta x^2}\Big[&(k+1)(k+2)(-u_{i-2}^{n-k} + 16u_{i-1}^{n-k} - 30u_i^n + 16u_{i+1}^n - u_{i+2}^n) \\ &-k(k+2)(-u_{i-2}^{n-k-1} + 16u_{i-1}^{n-k-1} - 30u_i^n + 16u_{i+1}^n - u_{i+2}^n) \\ &+k(k+1)(-u_{i-2}^{n-k-2} + 16u_{i-1}^{n-k-2} - 30u_i^n + 16u_{i+1}^n - u_{i+2}^n)\Big]\end{aligned}$$

AT schemes: numerical experiments

2nd order AT scheme in space



4nd order AT scheme in time



Implementation

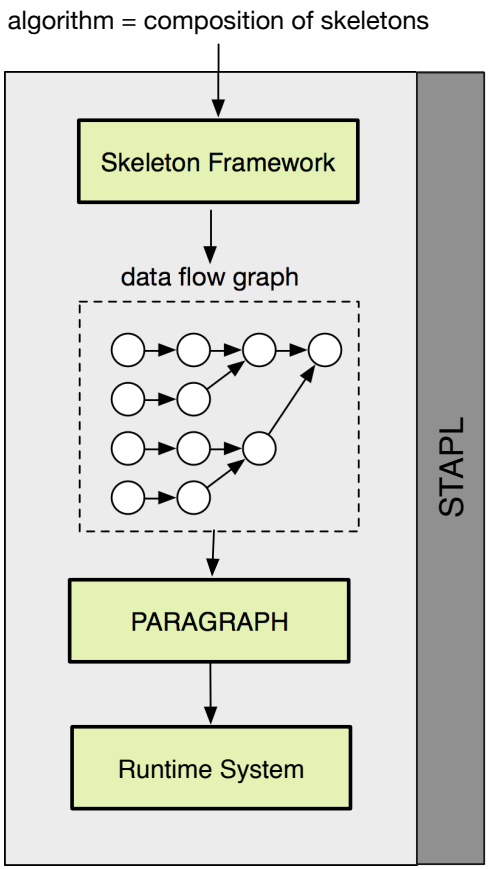
- **Requirements**:
  - "Ghost" cells: value should be either old or new value
  - Need timestamp information
  - Error control knob: $\langle \overline{E} \rangle \propto \overline{\tilde{k}}$
    - Enforce total/partial synchronization when $k = L$

- Can in principle be done with e.g. MPI
  - Two-sided: Isend/Irecv/Test
  - One-sided: Get/Put/Accumulate (fences, locks, ...)
  - Difficult to generalize: 1D/2D/3D, different schemes or stencils, different control knobs, etc

- Programming very challenging for application users
  - Need to study/experiment different schemes
  - Interested in physics

Need a general framework to express asynchronous schemes

Algorithmic skeletons

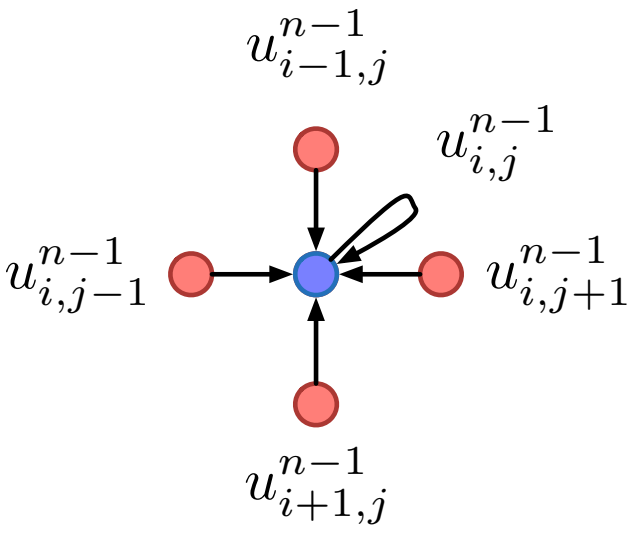


- Algorithmic skeletons
  - portable and implementation-independent specification of algorithms
  - polymorphic higher-order functions
    map $f[a_1, a_2, ., a_n] = [f(a_1), f(a_2), ., f(a_n)]$
  - inherently composable
  - allow formal analysis and transformations
- STAPL translates them to data flow graphs[1]
  - suitable representation for parallel execution
  - composition
    - point-to-point dependencies
    - no global synchronizations

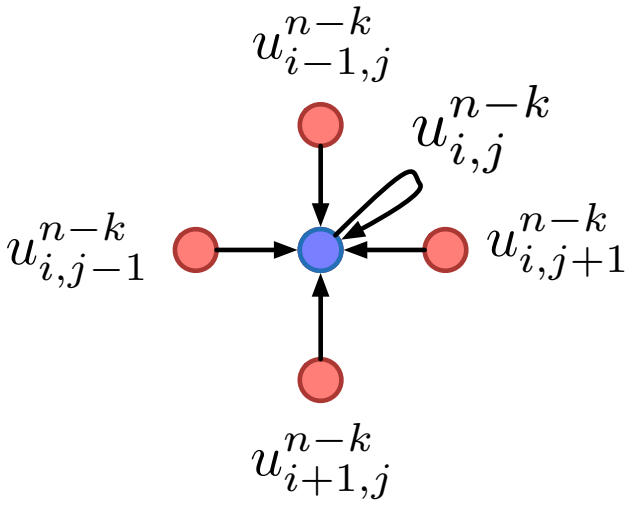
[1] Zandifar etal LCPC 2014; Zandifar etal ICS 2015

Stencil computations

- **skeleton::stencil<5>(op)**
  - each iteration can start only if previous is done
  - limits scalability



- k iteration relaxation – **skeleton::stencil<5>(op,L)**
  - each iteration can start if result from any of the previous $k \leq L$ levels is ready
  - more asynchrony

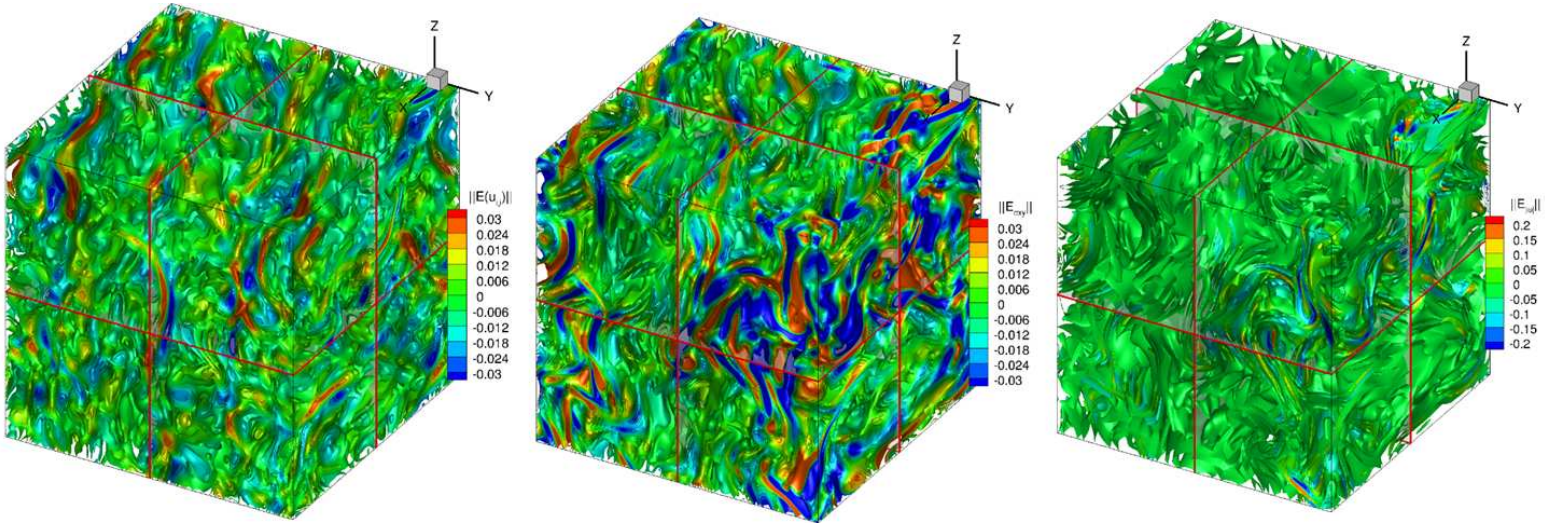


Note: k could be different on each side

Effect on realistic flows

- Need to understand propagation of perturbations in high-Reynolds number turbulence
  - Velocity, velocity gradients, pressure
  - Different regimes (sub/supersonic, strong shear): PDE changes type (propagation properties)
- Preliminary results: simulated delays at a given instant, then continue simulations

Relative error between delayed and non-delayed simulation



Larger error on vortical structures – how to avoid/minimize it?

Asynchronous schemes as a dynamical system

Approach: Switched System Framework

$$U^{(n+1)} = A_{\sigma_k} U^{(n)}, \quad \sigma_n \in \{1, 2, \ldots, m\}$$

$\pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$: switching probability for each A_{σ_n}
$m = L^{2(P-2)}$: total number of switching modes

- Monte Carlo simulations: intractable!
  - E.g.: $L = 2$, $P = 100$ leads to $m \sim O(10^{60})$ matrices!
- Instead, use Lyapunov theory for dynamical systems (1D)

$$||\bar{E}^{(n)}||^2 < C\,(1 - 1/\lambda_{max}(P_m))^n$$

- Can obtain PDF/moments of error given the PDF of delays $\tilde{k}$

(Lee etal ACC 2014, Autom. 2015; Lee & Bhattacharya DSCC 2014,ACC 2015)

Summary and outlook

- Turbulence simulations at exascale: critical to fundamental understanding
  - Challenges in scalability with current methods
- **Asynchronous schemes**: feasible if we can show stability, consistentcy and accuracy
  - Gained significant understanding, but mainly in a mean sense
  - Can now design schemes with desired mean properties
  - Some preliminary time-dependent bounds; need to tighten them
- Complexities of asynchronous implementations: skeletons/STAPL
  - High-level interfaces for application users
  - Readily implement/change numerical schemes