Lagrangian Intermittency in Fluid Turbulence: Science Goals and Algorithms of High Scalability

P.K. Yeung (Georgia Tech)
E-mail: pk.yeung@ae.gatech.edu

Rohini Uma-Vaideswaran (Georgia Tech), Kiran Ravikumar (HPE)
Michael Wilczek (U. Bayreuth, Germany)

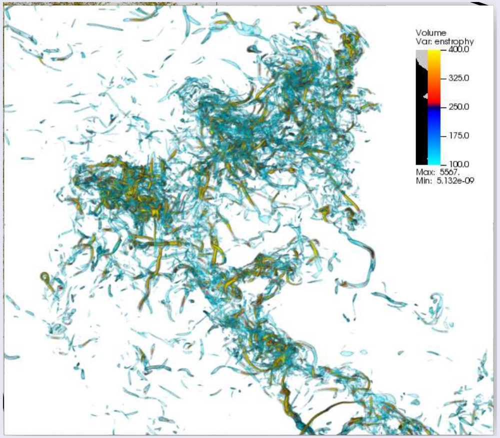
TACC Frontera Users Meeting, Austin, TX. August 3-4, 2023

NSF Grants 1953186 and LRAC Allocation 2023

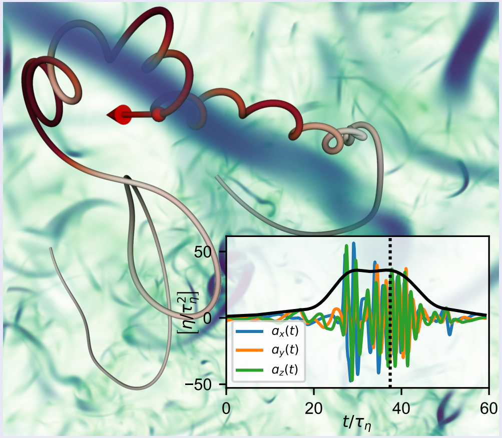
CSA Subaward on Fluid Turbulence (w/ Diego Donzis, TA&M)

Thanks: Lars Koesterke, John Cazes, Hanning Chen

What is Lagrangian Intermittency?



Eulerian: e.g. convoluted vortex "filaments" of high intensity, localized in time and (3D) space [Viz. experts at Oak Ridge]



Lagrangian: sample trajectory and acceleration of fluid element passing through high-rotation zone [CO-PI M. Wilczek]

The nature of the Lagrangian problem

Observer moving with the instantaneous fluid flow

- With velocity field from direct numerical simulation: equation of motion is just

$$d\mathbf{x}^+/dt = \mathbf{u}^+(t) \ ; \quad \text{where} \quad \mathbf{u}^+(t) = \mathbf{u}(\mathbf{x}^+(t), t)$$

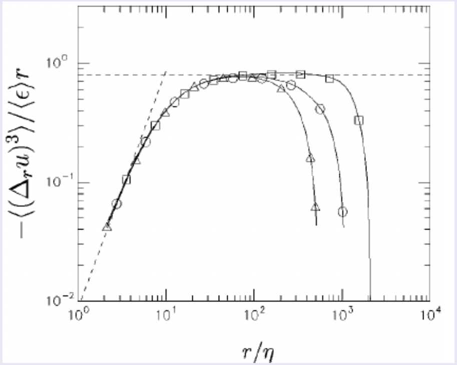
  interpolate at instantaneous particle position from neighboring grid points
- Very difficult in laboratory measurements, while DNS is helpful for modeling

Complexities and importance:

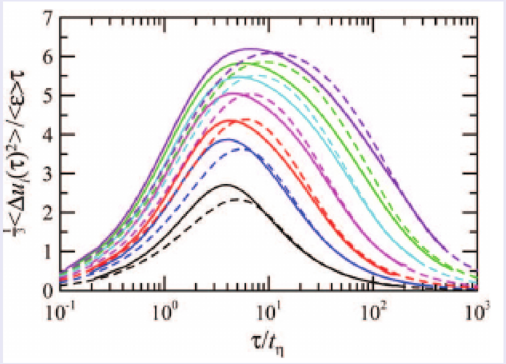
- Turbulent dispersion: e.g how far does a cloud of contaminants or disease agents travel, and how widely does it spread, as function of time
- Possible extreme events: can toxic material be found at some unlikely place?
- Forward in time (natural for PDEs) or backward (where did something came from?)

Tests of classical similarity, based on "structure functions"

Statistical moments of velocity increments over spatial separation r or time lag τ



Eulerian: a scaling "plateau" is well attained for Taylor-scale Reynolds number ≈ 650 and higher.

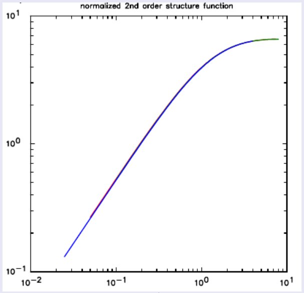


Lagrangian: clear scaling plateau is elusive, even at higher Reynolds number. Peak value seems to occur at short time lags.

Some high-resolution results on Frontera

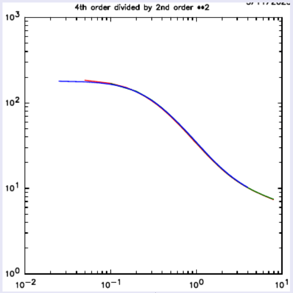
Consider structure function of order m i.e. $D_L^m(\tau) = \langle [\Delta_\tau u^+(t)]^m \rangle$ for $m = 2, 4, 6...$
Theory: scales as τ^m if τ is small, but as $\langle (\epsilon^{m/2}) \rangle \tau^{m/2}$ at intermediate $\tau_\eta \ll \tau \ll T_L$

$D_L^2(\tau)/(\langle \epsilon \rangle \tau)$



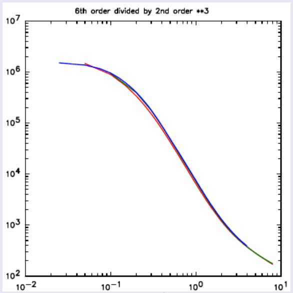
τ/τ_η

$D_L^4(\tau)/[D_L^2(\tau)]^2$



τ/τ_η

$D_L^6(\tau)/[D_L^2(\tau)]^3$



τ/τ_η

$R_\lambda \approx 1000$ at different spatial and temporal resolutions (blue line from 12288^3 DNS).
$D_L^2(\tau)$ and $D_L^4(\tau)$ depart strongly from theory, while showing signs of very strong intermittency (e.g. flatness factor of $\Delta_\tau u^+$ nearly 200 in data shown).

A spatial-temporal perspective

- $\mathbf{u}^+(t+\tau) \neq \mathbf{u}^+(t)$ because particle moves to a different location as flow evolves.
- Analyze the (non-unique) decomposition

$$\Delta_\tau \mathbf{u}^+(t) = [\mathbf{u}(\mathbf{x}^+(t+\tau), t+\tau) - \mathbf{u}(\mathbf{x}^+(t), t+\tau)] + [\mathbf{u}(\mathbf{x}^+(t), t+\tau) - \mathbf{u}(\mathbf{x}^+(t), t)]$$

  - First square bracket is a spatial increment; the second is temporal
  - At small τ the increment to proportional to the fluid particle acceleration,

$$\mathbf{a} = \mathbf{a}_C + \mathbf{a}_L$$

    where the convective and local accelerations are subject to strong mutual cancellation (Tennekes 1975, Tsinober, Vedula & Y, 2001)

- Proceeding to study spatial and temporal increments at higher R_λ. Present code can get $\mathbf{u}(\mathbf{x}^+(t), t+\tau)$ quite easy, at almost no extra cost.
  — expect correlation coefficient to evolve towards -0.5 in long-time limit

Particle Tracking Algorithms: Requirements and Objectives

- Within a Fourier pseudo-spectral DNS code that uses a 2D (rows and columns) domain decomposition for massive parallelism
- Cubic spline interpolation (Yeung & Pope 1988): 4th order accurate and twice differentiable (reduces noise in computing acceleration by differentiating velocity)
- Form $(N+3)^3$ cubic spline coefficients from N^3 instantaneous velocity field
- Reduce or eliminate communication costs associated with access to spline coefficients for particles tracked by MPI processes
- Enable particle counts that are large (tested up to 1 billion in some cases) although still small compared to the number of grid points
- I/O and file management, plus relative ease of post-processing also relevant.

Strategies Adopted (successfully) on Frontera

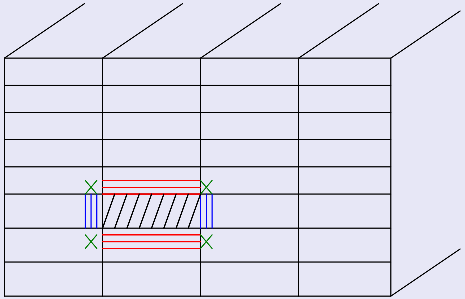
- Machine has 56 cores per node: let N be a multiple of $48 = 3 \times 2^4$, likewise for row dimension of 2D processor grid — each row communicator to have its own node.
- Dynamic local particle decomposition coupled to ghost layers for spline coefficients
  - Particles wander: but interpolation easiest if MPI task holding the particles has all $4^3 = 64$ spline coefficients needed, based on particle position
  - Each time step: if a particle migrates to a neighboring sub-domain, transfer its properties to MPI process holding all the splines coefficients needed.
  - Form ghost layers around spline coefficients held by each MPI process

Formation of spline coefficients followed by adding ghost layers are the most costly operations. However these items are independent of particles, allowing the code to meet objective of large particle counts at very little extra cost.

Ghost layers and One-sided MPI

- MPI_WIN_CREATE: time-consuming, call just once at beginning of code
- Three buffers: E-W (row comm), N-S (col comm), (all 4) Corners (world)
- Call MPI_WIN_FENCE (....)
  Call MPI_GET (....)
  Call MPI_WIN_FENCE (....)
- Message sizes dependent on $P_r \times P_c$. Some packing/unpacking required (less so for $N - S$, which however are likely to have the largest messages)

Ex: lower half of 4×16 decomposition in $x - z$ directions, with full pencil in y



When ratio of P_c / P_r is large, N-S comm. between the nodes is dominant.

Migration, Output, and Post-Processing

Transfer of migrating particles data to/from 8 neighboring MPIs

- MPI_ISEND and MPI_IRECV on particle ID, position, velocity.
- Only particles within 1 grid spacing of sub-domain boundaries can migrate. For homogeneous turbulence any load imbalance is generally minimal.

Management of output files holding particle data (avoid too many files)

- At output step: collect data from subsets of MPI processes, write 1 file per subset

Post-processing of particle data (some caveats)

- A re-sorting is necessary to collect data at all time instants for each particle, since they are spread over multiple files due to dynamic decomposition.
- Distribution of particle coordinates within each subset is not fully statistically uniform. But, due to homogeneity, this does not matter much.

Overall DNS Performance

Data quoted from Texscale Days July 2022

Grid (N^3)	# nodes	N_p (10^6)	Wall time per step						% Weak Scaling	
			Eul.	Splines	Ghost	Spcal	Migrate	Tot.	Eul.	Tot.
1536^3	4	22	11.6	3.06	0.29	0.26	0.02	15.1	-	-
3072^3	32	88	12.4	3.20	0.48	0.14	0.01	16.2	93.5	93.2
6144^3	256	352	15.5	4.52	0.59	0.07	0.03	20.7	80.0	78.3
12288^3	2048	1057	17.5	8.69	1.61	0.04	0.11	28.0	88.6	73.9

- Eulerian part (mostly 3D FFT) shows approx 90% weak scaling for each 2X in N
- Present implementation does allow high particle counts with little extra cost.
- But, code struggles with 12288^3 on 4096 nodes. with variability in inter-node MPI. Presumably hitting the limit of large MPI process count and small messages ...

A twin-communicator approach?

- Let the cost of code at a given problem size be dominated by two operations, called A and B (for Eulerian and Lagrangian)
- Code runs at elapsed wall time equal to $t_{1A} + t_{1B}$ on P_1 nodes. Double the node count to $P_2 = 2P_1$. If strong scaling is 100% then cost is proportional to

$$2(t_{2A} + t_{2B}) = (t_{1A} + t_{1B})$$

- If the code scales well at P_1 nodes but not P_2: divide latter into two equal halves, each handling their respective businesses, with two equal-sized MPI communicators. If completely independent, cost would be proportional to $2 \times \max\{t_{1A}, t_{1B}\}$.
- Depends on how much overlapping we can get, and overhead in communication needed between A and B. (Point-to-point, single message, faster than alltoalls?)
- Also depends on how poorly the single-communicator code scales at P_2.

  We have constructed a minimalist code for testing and optimization.

Summary and Next Steps

The science: Lagrangian intermittency in turbulence

- More complex than Eulerian counterpart. More demanding in computational resources, grid resolution, and Reynolds number.
- Active investigations of Lagrangian velocity increments and (with Co-PI Wilczek) acceleration in context of extreme events
- Problem size and particle count probably exceed most work in literature

High-performance code capable of huge particle counts

- (a) Let each row communicator occupy a node, (b) one-sided MPI for ghost layers and (c) dynamic mapping between particles and parallel processes
- Twin-communicator approach for strong scaling at largest problem sizes?