Particle tracking and turbulent dispersion at high Reynolds number on Blue Waters

Dhawal Buaria  P. K. Yeung (PI)

dhawal817@gatech.edu  pk.yeung@ae.gatech.edu

Georgia Institute of Technology

Acknowledgments:
NSF: PRAC and Fluid Dynamics Programs
BW Team, Cray: Scaling, Reservations, Help Requests, Storage, Visualization
Co-PIs: A. Majumdar, R.D. Moser, D. Pekurovsky, J.J. Riley
Collaborators: B.L. Sawford, K.R. Sreenivasan
Other lab members: X.M. Zhai, M.P. Clay

Blue Waters Symposium, Sunriver, Oregon, June 13-15, 2016
Turbulence

- Most common state of fluid motion in nature and engineering
- Wide range of disorderly, non-linearly interacting scales in 3-D space and time, always at high Reynolds number \((Re = uL/\nu)\)
- Increased rates of dispersion and mixing; crucial in environmental problems and improved engineering devices.
- Direct numerical simulations (DNS): resolve all relevant spatial and temporal scales; cost \(\propto Re^3\) \(\rightarrow\) more CPU power needed
Turbulent dispersion

- Net spreading apart of some material by turbulent motion, e.g., dispersion of pollutants, cloud/vapor droplets, etc. in atmosphere

- Best studied in a Lagrangian frame of reference; follow motion of infinitesimal fluid elements/particles (Monin & Yaglom 1971)
  - Fluid particles have zero size and move with local flow velocity
  - Can also be extended to include effects of molecular diffusion (Brownian particles or ‘molecules’) or inertia (inertial particles)

- Motion of single particles; relative motion between two or more

- Pursuit of Kolmogorov similarity at high $Re$; also relevant in stochastic modeling (Sawford ARFM 2001)
Forward vs. Backward tracking

**Forward tracking:** where will material go?
- In both simulations and experiments, track a population of particles forward in time with the flow, from $t = 0$ to $t = T$
- Useful in understanding spreading of pollutants, contaminants, etc.

**Backward Tracking:** where did material come from?
- At observation time $t = T$, identify particles and study their past histories, i.e., track backwards in time ($t < T$)
- Relevant in turbulent mixing — $n^{th}$ moment of scalar field from backward statistics of $n$-particle cluster (Thomson 1990)
- Also useful in identifying origins of pollutants, pathogens, etc.
- Simple as forward in principle; very difficult to do because Navier-Stokes are time irreversible — massive detail in DNS allows a postprocessing approach (Buaria et al. PoF 2015)
Methods: Petascale DNS

- Navier-Stokes with constant density \((\nabla \cdot \mathbf{u} = 0)\)
  \[
  \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla \left( \frac{p}{\rho} \right) + \nu \nabla^2 \mathbf{u} + \mathbf{f}
  \]
- Isotropic turbulence in cubic domain; periodic boundary conditions
- Fourier pseudo-spectral algorithm; parallel 3D-FFTs
- Map \(N^3\) grid points onto 2D processor grid \((P_{row} \times P_{col})\)

Two ALLTOALLs (A2As); in row and column communicators
Optimal performance: $P_{row} \leq 32$ (no. of cores on single node)
   - Slab of data on each node; one A2A completely on node

Co-Array Fortran w/ favorable topology (Fiedler et al. CUG 2013)
   - $8192^3$ on 262,144 cores ($32 \times 8192$): 9 secs/step

Extreme events (earlier findings using $8192^3$ DNS)
   - Localized (in space)/short lived (in time) regions of extremely high rates of strain (dissipation) and/or rotation (enstrophy)
   - Very sensitive to $Re$; spatial structure of extreme events fundamentally different at high $Re$ (Yeung et al. PNAS 2015)
   - How do these extreme events affect turbulent dispersion?
   - Fluid elements may experience extreme local deformation; critical in flame propagation, cloud droplet clustering, etc.
Particle tracking and interpolation

- Particles initialized at $t = 0$; tracked forward with N-S equations
  - Integrate $dx^+/dt = u^+$, where $u^+ = u(x^+, t)$

- Cubic-spline interpolation (Yeung & Pope JCP 1988):
  - 4th order accurate, twice differentiable
    (important for velocity gradients and acceleration)

$$u^+ = \sum_{k=1}^{4} \sum_{j=1}^{4} \sum_{i=1}^{4} b_i(x^+) c_j(y^+) d_k(z^+) e_{ijk}(x)$$

- $(e_{ijk})$: $(N + 3)^3$ Eulerian spline coefficients

- $(b_i, c_j, d_k)$: basis functions at $4^3 = 64$ adjacent grid points
Parallel implementation for Particle tracking

- Large number of fluid particles; also distribute among MPI tasks
- \((N + 3)^3\) spline coefficients distributed like grid points (pencils)
  - Solve a system of tridiagonal equations
  - 3 computation cycles and 2 transposes (ALLTOALLVs)
- Particles wander randomly due to turbulence
- Neighboring grid points for interpolation keep changing
- Parallel interpolation not trivial
Global approach

- Each MPI task always responsible for the same set of particles
- Particles typically far from sub-domain present on MPI task
- Communicate with every other task for interpolation; collective communication calls (‘global’ communication)
- Each processor calculates a partial contribution for each particle (based on spline coefficients it has)

- Very generalized implementation; particles can move any distance in one time step — even load balance regardless of flow physics

- Reasonable performance up to $O(10^4)$ cores

- However collective communication calls very expensive at larger problem sizes / core counts

- Need better parallel implementation
Local approach

- Each MPI task responsible for dynamically evolving set of particles; interpolation stencil ‘local’ on same MPI task
- Some communication required for particles near the boundaries
- Transfer control of particles as they cross over to new sub-domains
- May restrict how far a particle can move in one time step
Local approach (contd.)

- Similar to spatial decomposition particle tracking in molecular dynamics (Plimpton JCP 1995); also used by many groups for particle tracking (Ireland et al. C&F 2013)

- Halo-exchange type communication (localized) instead of collective

- Typical method is to use ‘ghost’ layers; can be filled by a simple SEND+RECV between neighbors; 3 needed here

- However ghost layers not feasible at largest problem sizes
More on communication in Local approach

- Consider $8192^3$ grid using $32 \times 8192$ processors (for optimal ALLTOALLs); $8195^3$ spline coefficients

- Each processor has $8195 \times 256 \times 1$ ($8195 \times 257 \times 2$ on first three)

- $8195 \times 262 \times 7$ with ghost layer, more than 6 times the pencil size!

- Majority of ghost layer points are not even useful; consider 300M particles at $8192^3$ — one particle for thousands of grid points

- Fetch data directly from remote memory as needed; use one sided communication

- Co-Array Fortran (CAF) on Cray compiler!
Partitioned Global Address Space (PGAS) programming model using one-sided communication

FORTRAN + Co-Arrays: allocated in ‘global memory address space’, can be accessed by all ‘images’ (MPI tasks)

Currently fully supported only on Cray compiler

Significantly lower latency and smaller headers for small message sizes

Declare spline coefficients in global address space as a co-array

Direct remote memory access (no ghost layers needed); communication still between neighbors (exploits PGAS locality)
! N = no. of grid points in each direction
! p_row x p_col = proc. grid
ALLOCATE ( spline_coarray (N+3, N/p_row + 1, N/p_col + 1) [p_row*p_col] )
call populate_spline_coefficients (spline_coarray)

loop_ip: DO ip = 1, num_particles
posn_ip = x-y-z coordinate of particle ip
(bx(1:4), cy(1:4), dz(1:4)) = calculate_basis_functions (posn_ip)
(ix,iy,iz) = map_posn_to_global_index (posn_ip) ! 1 to N

loop_k: DO k=1,4
loop_j: DO j=1,4

!target_rank = map_array_indices_to_rank (iy+j,iz+k)
iy_new, iz_new = array_indices_based_on_target_rank (iy+j,iz+k)
spline_temp(1:4) = spline_coarray (ix+1:ix+4, iy_new, iz_new)[target_rank]

loop_i: DO i=1,4
velocity_ip = velocity_ip + bx(i)*cy(j)*dz(k)*spline_temp(i)

ENDDO loop_i; ENDDO loop_j; ENDDO loop_k; ENDDO loop_ip
Performance results on Blue Waters

- Massive improvement in interpolation cost (more than 2 orders); overall particle tracking 10X faster for 64M particles at $8192^3$
- Good scaling for any proc. grid (optimal 3D FFTs)
- $8192^3$ production with 300M particles (over 40X speedup)
Statistics of the (one-particle) Lagrangian velocity increment

\[ \Delta_\tau u^+ = u^+(t + \tau) - u^+(t) \]

Use of K41: \( \langle (\Delta_\tau u^+)^2 \rangle = C_0 \langle \epsilon \rangle \tau \) in inertial range \( \tau_\eta \ll \tau \ll T_L \); \( C_0 \) seems to converge slowly with Reynolds number, \( \approx 7 \).

How do higher-order structure functions scale?
— probably deviates from \( \langle (\Delta_\tau u^+)^{2p} \rangle \propto \langle \epsilon \rangle^{p} \), due to Lagrangian intermittency (use \( \langle \epsilon \rangle_\tau \), local average in time?)

How long is the memory of intense dissipation, enstrophy, etc? Do fluid particles get trapped in zones of such activity?

Statistics of \( \Delta_\tau u^+ \) conditioned upon low or high dissipation, etc, especially at small \( \tau \) (Yeung et al. JFM 2007)
Conditional Lagrangian
“flatness structure function”:

$$\frac{\langle (\Delta \tau u^+|Y^+(t))^4 \rangle}{\langle (\Delta \tau u^+|Y^+(t))^2 \rangle^2}$$

- Small $\tau$: related to (conditional) acceleration
- Lower at large $\epsilon^+$ or large $\Omega^+$ (Eulerian intermittency)
- Short inflexion for large $\Omega^+$
- Large $\tau$: towards Gaussian

Condition upon “extreme events”? More samples to be processed
Turbulent relative dispersion

- \( r(t) = x^{(1)}(t) - x^{(2)}(t) \); expected to grow on average, with dependence on \( r(0) \), without bounds (homogeneity)
- Established power laws for small, intermediate and large times

\[
\langle |r - r_0|^2 \rangle = (D_{LL} + 2D_{NN})t^2, \quad t \ll (t_\eta, t_0) \quad \text{(ballistic regime)}
\]
\[
\langle |r - r_0|^2 \rangle = g\langle \epsilon \rangle t^3, \quad (t_\eta, t_0) \ll t \ll T_L \quad \text{(Richardson’s scaling)}
\]
\[
\langle |r - r_0|^2 \rangle = 12\sigma_u^2 T_L t, \quad t \gg T_L \quad \text{(diffusive limit)}
\]

where, \( t_0 = (\langle r_0^2 \rangle / \epsilon)^{1/3} \) time for which \( r_0 \) is important

- Great deal of interest in quantification of Richardson’s constant \( g \) (Sawford 2001, Salazar and Collins 2009)
- Function of \( R_\lambda \ (= u\lambda / \nu) \), but generally accepted to asymptote at high \( R_\lambda \); \( g \approx 0.56 \) for forward, Sawford \ et al. \ 2008)
- Forward and backward in time different (Sawford \ et al. \ 2005)
Forward (solid) and backward (dotted) results overlap in ballistic and diffusive limit (expected for $R_{\lambda} = 1000$ run for a longer run)

- Stronger backward dispersion at intermediate times;
  faster approach to diffusive limit
Cubed-Local Slope (CLS)

- Richardson scaling

\[ \langle r^2(t) \rangle = g\langle \epsilon \rangle t^3, \quad (t_\eta, t_0) \ll t \ll T_L \]

- “Cubed-local slope” approach: (Sawford et al. 2008)

\[ \frac{1}{\langle \epsilon \rangle} \left( \frac{d}{dt} \left[ \langle r^2 \rangle^{1/3} \right] \right)^3 \quad \text{vs} \quad t/t_0 \]

- Look for a plateau for \( t/t_0 \gg 1 \); expect forward and backward to differ
Cubed-Local Slope (CLS)

- Taylor Reynolds nos. 390 (dotted), 650 (dashed), 1000 (solid)

Forward: $g_f \approx 0.55$ (0.56 in Sawford et al. 2008, $R_\lambda$ 650)

Backward: $g_b \approx 1.5$, but shorter scaling range; may need higher $R_\lambda$

More details in Buaria et al. PoF 2015
Molecular diffusion can be incorporated through the motion of molecules as Brownian particles (JFM: Saffman 1960, Pope 1998)

\[ dx^m(t) = u(x^m, t)dt + \sqrt{2\kappa} \, dW(t) \]

- backward statistics of molecules allows inference of Eulerian passive scalar variance at arbitrary (low or high) Schmidt number
- more results reported in Buaria et al. JFM 2016

Extension to tetrads of fluid particles
- minimum configuration to represent 3D geometry
- measures of size vs measures of shape (Hackl et al. PoF 2011)
- how do some tetrads become highly convoluted, or planar?
Turbulence as a grand challenge in science and HPC
- 3D, unsteady, non-linear, wide range of scales
- Want high $Re$ and better resolved small scales
- smaller simulations often compromised in physics or accuracy needed for applications where turbulence is the critical process

More than a decade between the first $4096^3$ (Japan 2002) and first $8192^3$ production run

Would be impossible if not for Blue Waters ...
- Sustained Petaflop performance
- dedicated, helpful and expert staff assistance
- Co-Array Fortran support on Cray compiler
- Nearly 1 PB of database (still growing !)
Concluding Remarks

- A large PRAC allocation on Blue Waters has made possible:
  - DNS with $8192^3$ grid points (Eulerian)
  - Corresponding Lagrangian data with over 300M particles

- New physical insights obtained in studies of extreme events associated with fine scale intermittency

- Improved understanding of turbulent dispersion and mixing via forward and backward tracking at high $Re$

- Ongoing work to investigate small scale intermittency from a Lagrangian perspective

- Long term: flows with more complex physics — high Schmidt number mixing, electromagnetic forces, solid-body rotation, etc.