GenIDLEST Co-Design

Danesh Tafti & Amit Amritkar

Collaborators

Wu-chun Feng, Paul Sathre, Kaixi Hou, Sriram Chivukula, Hao Wang,

Eric de Sturler, Kasia Swirydowicz

Virginia Tech

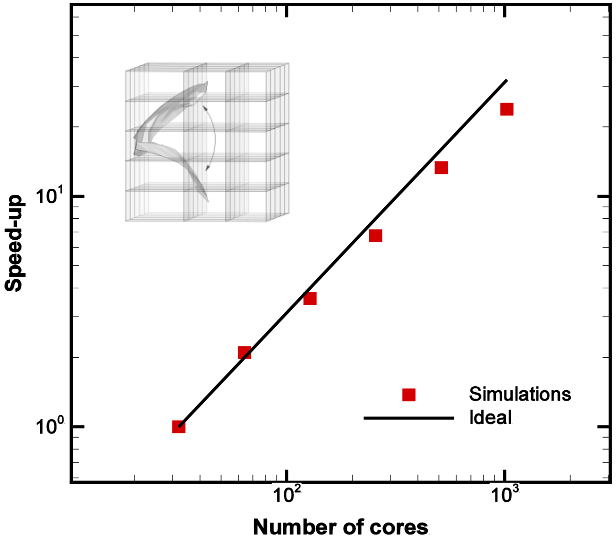
AFOSR-BRI Workshop
Feb 7 2014

Outline

- GenIDLEST Code
  - Features and capabilities
- Initial Code porting
  - Strategy
- Co-design optimization
- Validation
- Application
  - Bat wing flapping
- Solver co-design
- Future work

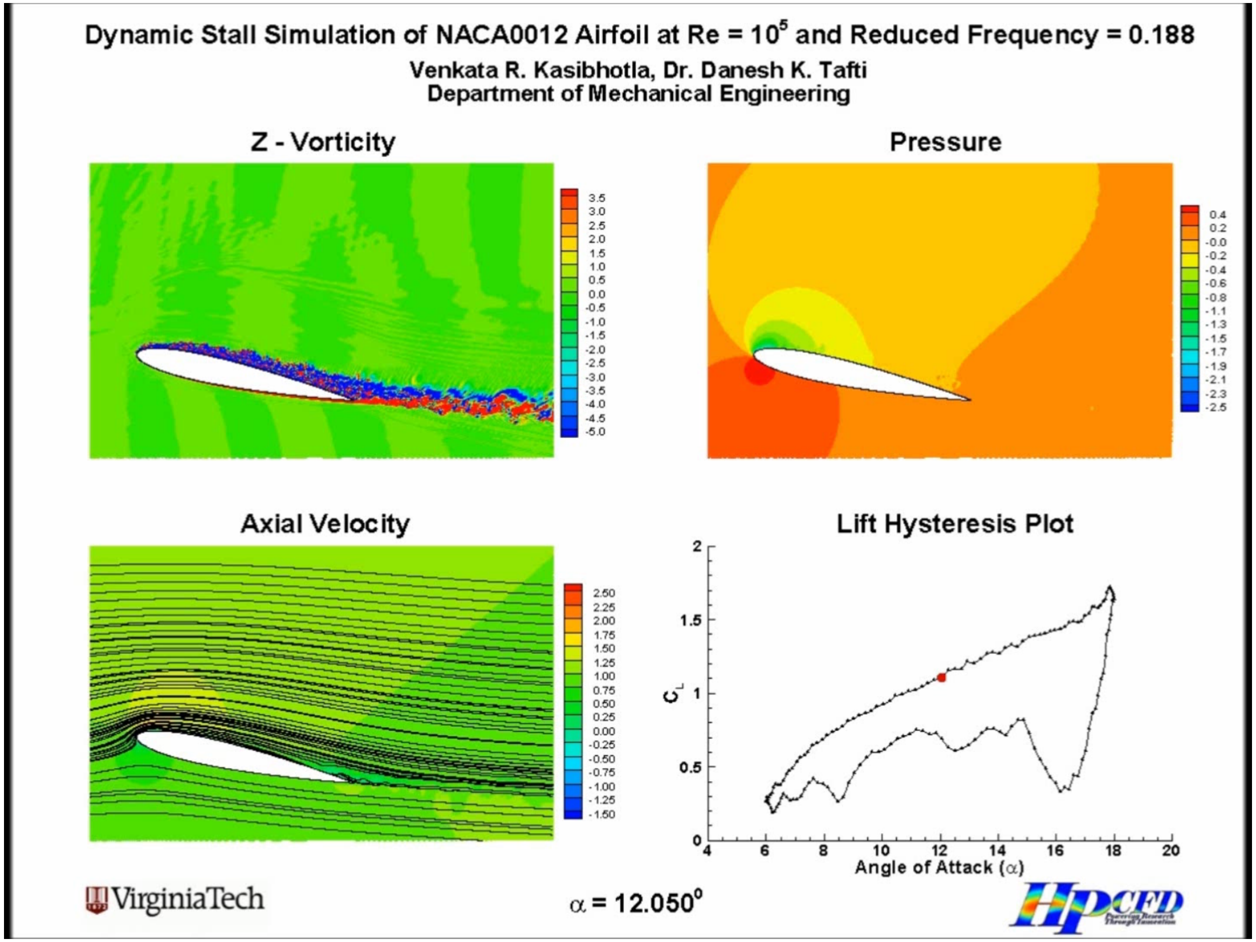
Generalized Incompressible Direct and Large-Eddy Simulations of Turbulence (GenIDLEST)

- **Background**
  - Finite-volume multi-block body fitted mesh topology.
  - Grid has structured (i,j,k) connectivity but blocks can have arbitrary connectivity (unstructured)
  - Pressure based fractional step algorithm
  - MPI, OpenMP or MPI-OpenMP parallelism
  - OpenMP scalability as good as MPI (tested up to 256 cores)
- **Capability**
  - Wall resolved and wall modeled LES in com
    - Dynamic Smagorinsky with zonal two-layer
  - Hybrid URANS-LES
    - K-ω based
  - Dynamic Grids (ALE)
  - Immersed Boundary Method (IBM) on gene
  - Discrete Element Method (DEM)



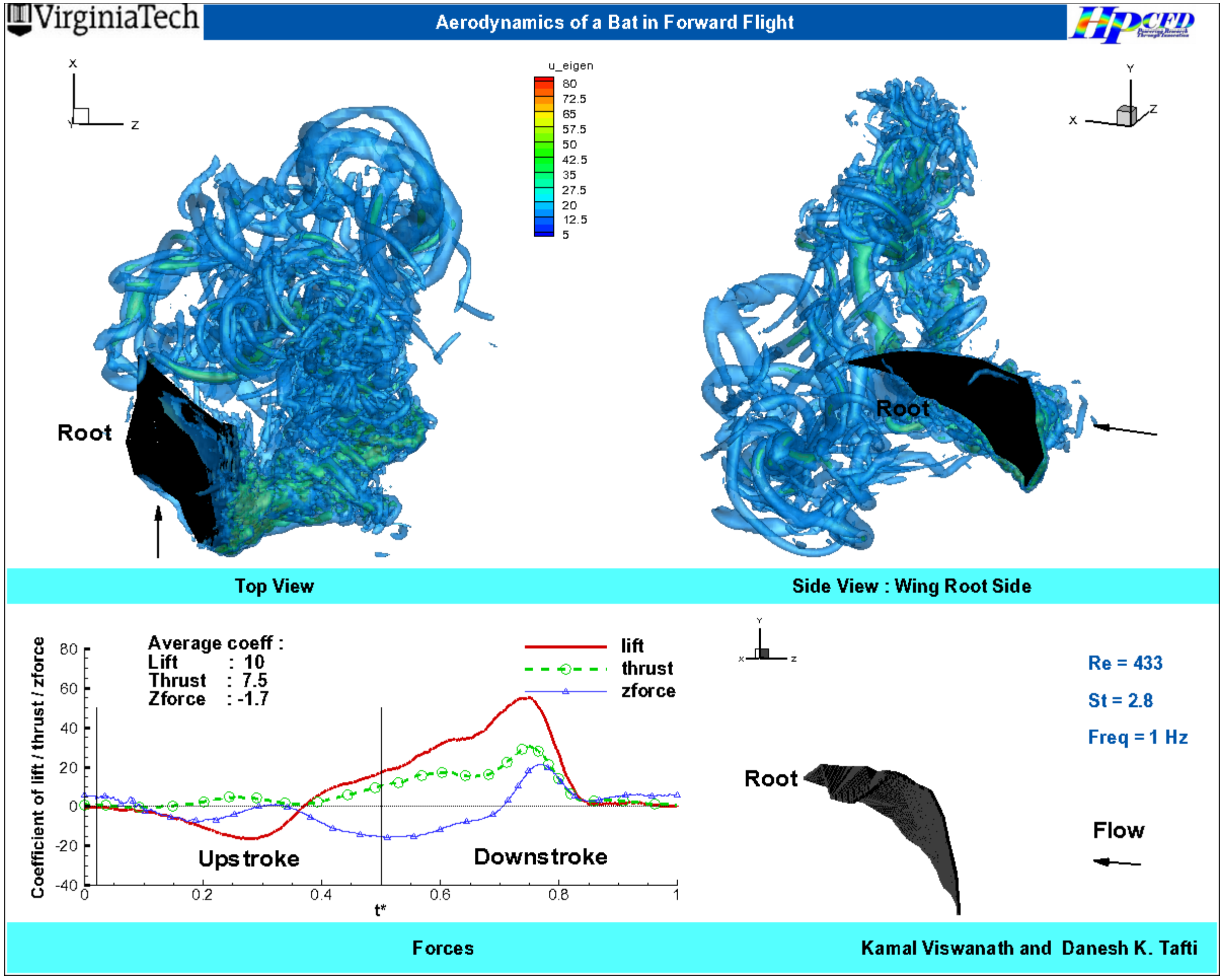
Generalized Incompressible Direct and Large-Eddy Simulations of Turbulence (GenIDLEST)

- **Dynamic stall on pitching airfoil (LES+ALE) -45 million cells-304 cores**



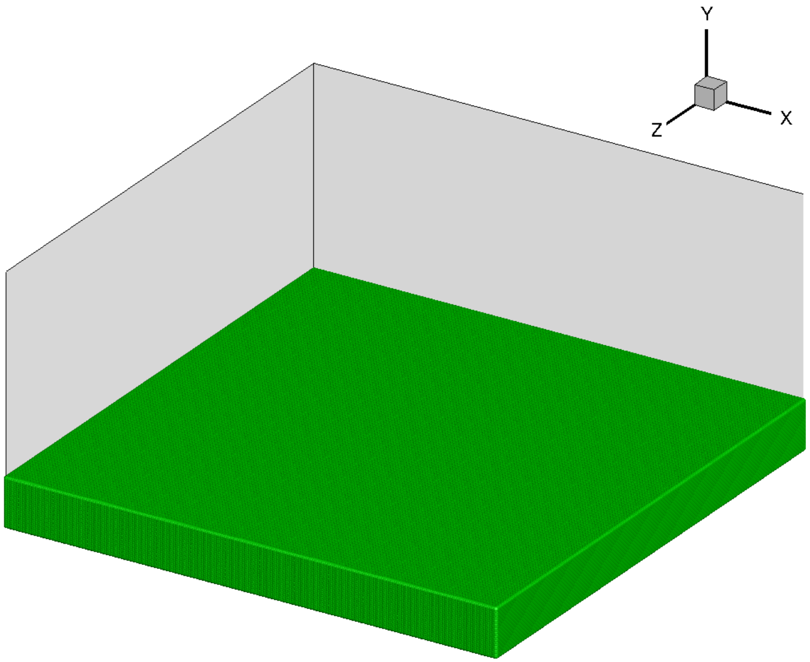
Generalized Incompressible Direct and Large-Eddy Simulations of Turbulence (GenIDLEST)

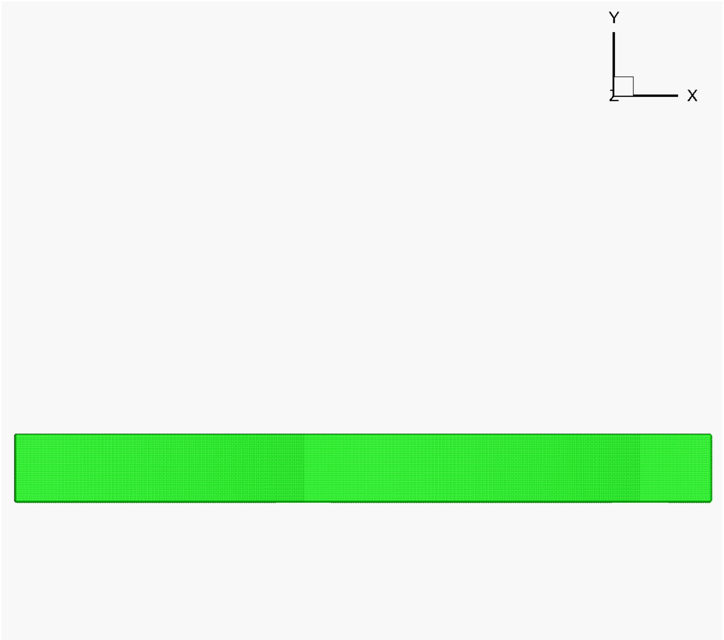
- **Flapping Bat wing (IBM) - 25 million cells – 60 cores**



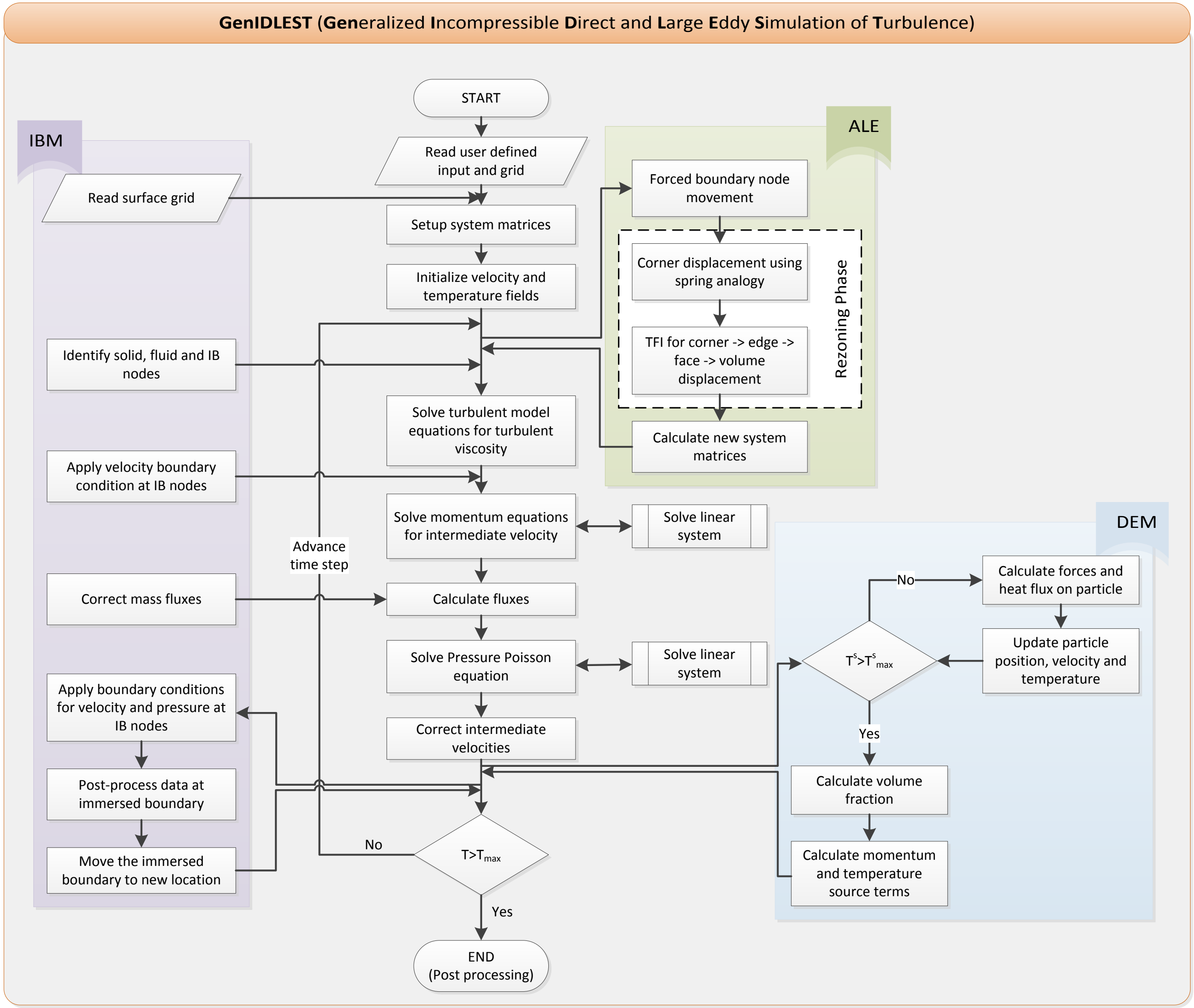
Generalized Incompressible Direct and Large-Eddy Simulations of Turbulence (GenIDLEST)

- **DEM in a fluidized bed (5.3 million particles, 1 million fluid grid cells – 256 cores)**





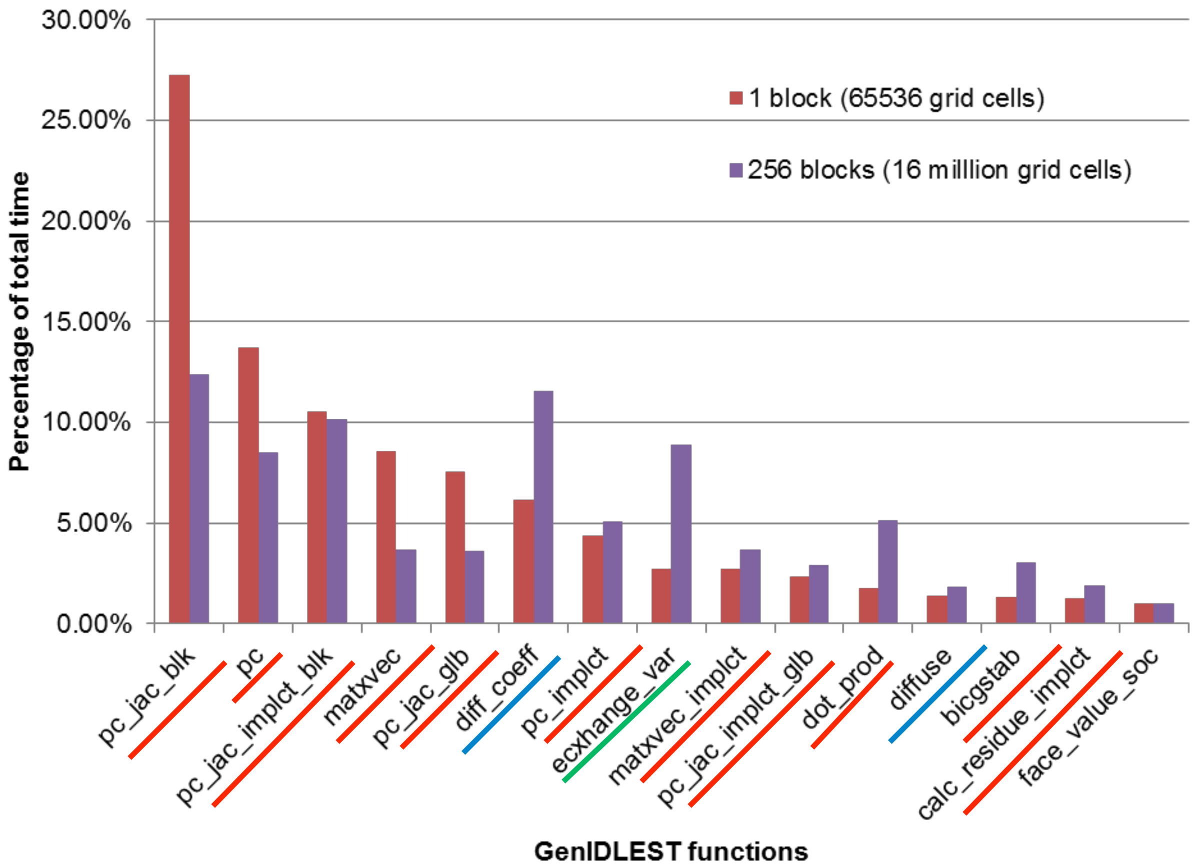
GenIDLEST Flow Chart



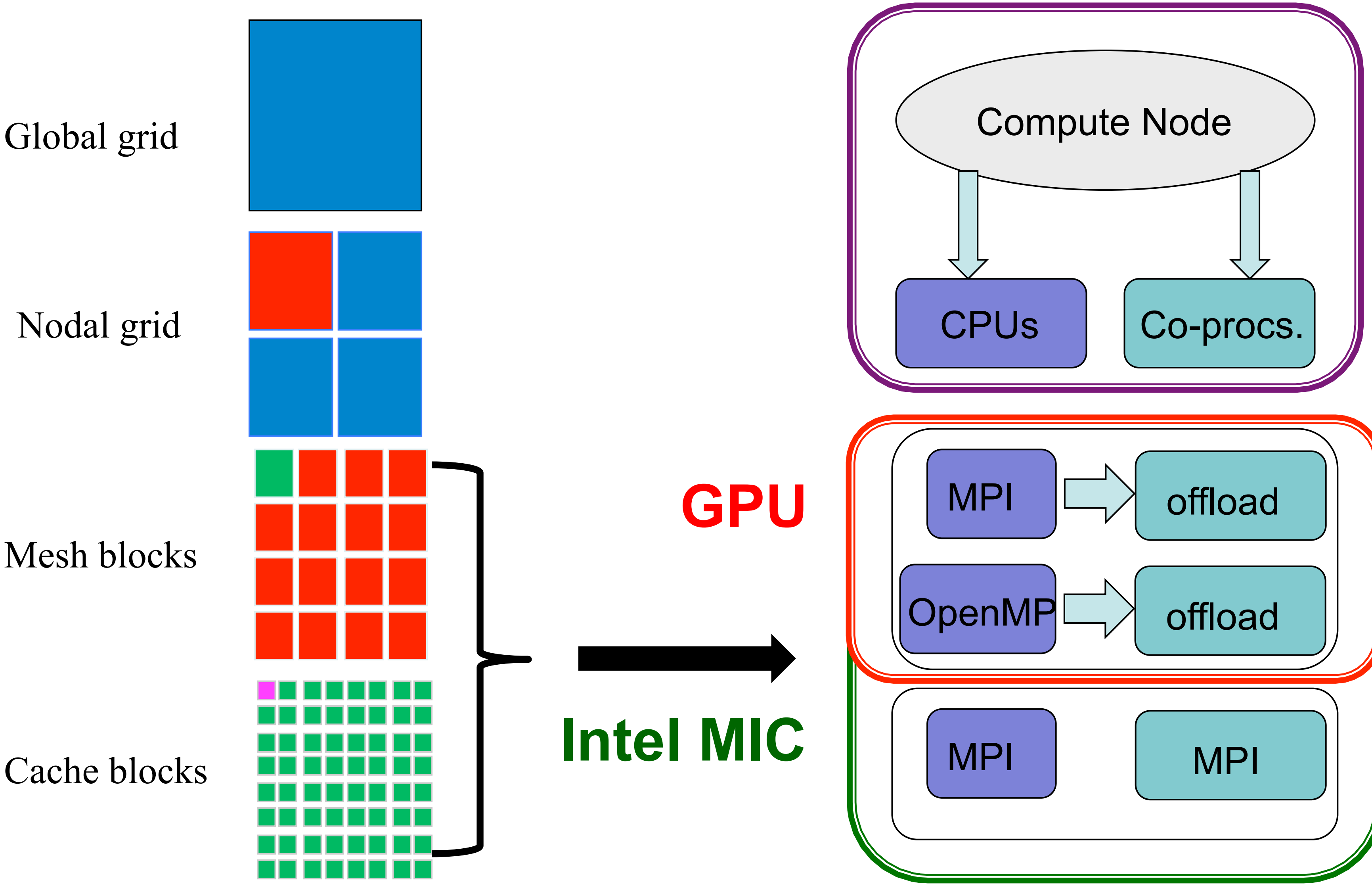
GenIDLEST-Linear Systems

- Sparse
  - Sparsity pattern depends on B.C.s.
  - Nonorthogonal grid - Non-symmetric - 18 off-diagonal elements
  - Orthogonal grid – Symmetric – 6 off-diagonal elements
- Use CG for symmetric system, and BiCGSTAB or GMRES(m) for nonsymmetric non-orthogonal systems.
- Preconditioners: Domain decomposition - Additive Schwarz Richardson or SSOR iterations applied to sub-structured overlapping "cache" blocks – SIMD parallel

GenIDLEST – Time Consuming Routines on CPU



Data Structures and Mapping to Co-Processor Architectures



Challenges in GPU Co-Processing Implementation

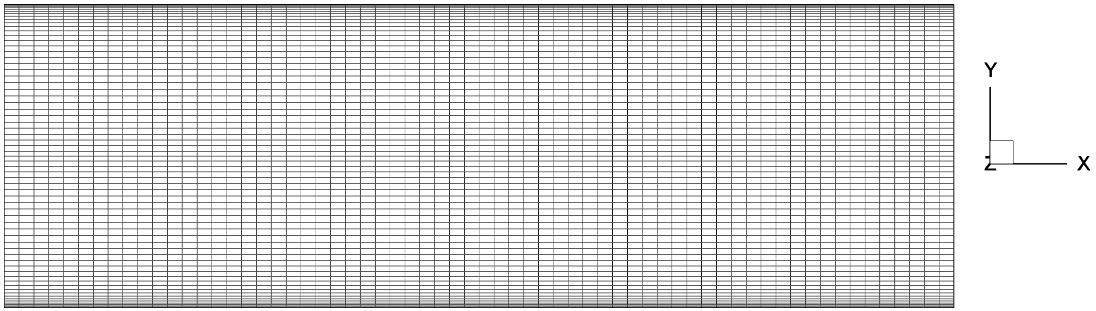
- **Large memory footprint for problems of practical interest limits GPU use to subset of code**
  - **Added overhead of getting data on and off GPU in time-integration loop**
- **All computations on GPU have to be data parallel – anything that deviates from this paradigm is not easy to implement**
- **Message passing for global block boundary synchronization between processors requires additional data transfer from and to GPU**

- **Implementation Strategy**
  - **Port large functional blocks of subroutines to GPU without overwhelming GPU memory while keeping CPU-GPU and GPU-CPU data transfers to a minimum.**
- **Over 75 CUDA-Fortran kernels written**
  - **Calculation of diffusion coefficients for all transport equations**
  - **Solution of all linear systems in transport equations and pressure equation which consume between 50-90% of computation time on CPU**
    - **2 *global* sparse matrix-vector multiplies, 4 *global* inner products or *global* reductions, 2 preconditioner calls and 4 *global* block boundary updates per BiCGSTAB iteration.**
  - **Message passing packing and unpacking**

Co-design with computer science team

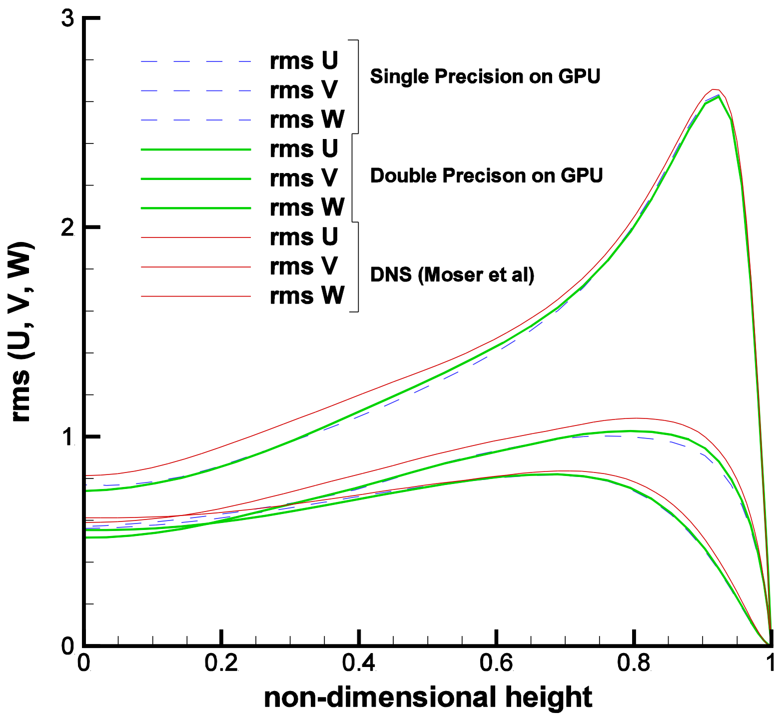
- **Approach**
  1. The program was profiled to identify performance bottlenecks.
  2. Accelerator architecture aware optimizations including coalesced memory access, shared memory usage, double buffering and pointer swapping, and removal of redundant synchronizations were applied.
  3. Kernel modifications were performed to suit the GPUs. This involved optimizations such as elimination of trivial memory accesses by performing in situ computations and dot products plus reductions which are ghost cell aware.
- **GPU code optimization**
  - Improved performance from 3x to 6x

- Turbulent flow through channel $Re_\tau = 180$
- Orthogonal Mesh
  - 7 point stencil
- 64^3 grid nodes
- Single mesh block



Channel flow - validation

- **BiCGSTAB solver**
  - additive Schwarz block pre-conditioner
    - Richardson iterative smoothing
- **Time averaged turbulence statistics**
  - Root mean squared flow velocities
- **Single vs Double precision**
  - More SP cores on GPUs



DNS study by – Moser, R. D., Kim, J., and Mansour, N. N., 1999, "Direct numerical simulation of turbulent channel flow up to Re= 590," Physics of fluids, 11, p. 943

Channel flow – performance results

- **Cache block configurations**
  - 3D data blocking
  - 25 iterations /time step - pressure Poisson equation

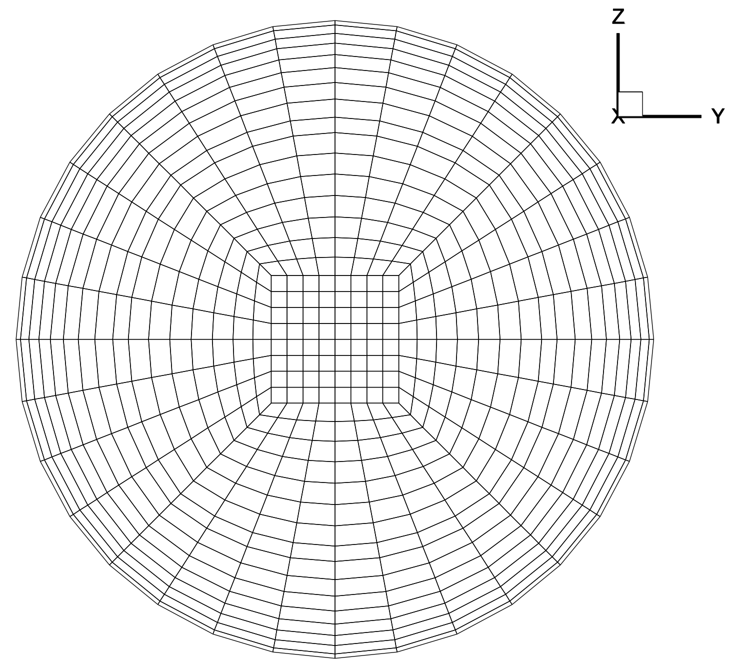
Layout	Total threads	Time (s)
8x4x4	128	98.7
8x8x2	128	97.9
8x8x4	256	101.1
16x8x2	256	103.6
32x4x2	256	104
64x2x2	256	103.5
64x4x2	512	96.3
32x8x2	512	97.5
16x16x2	512	98.4
8x8x8	512	107.5

- **Different GPUs**
  - 200 time steps
  - Time in time integration loop

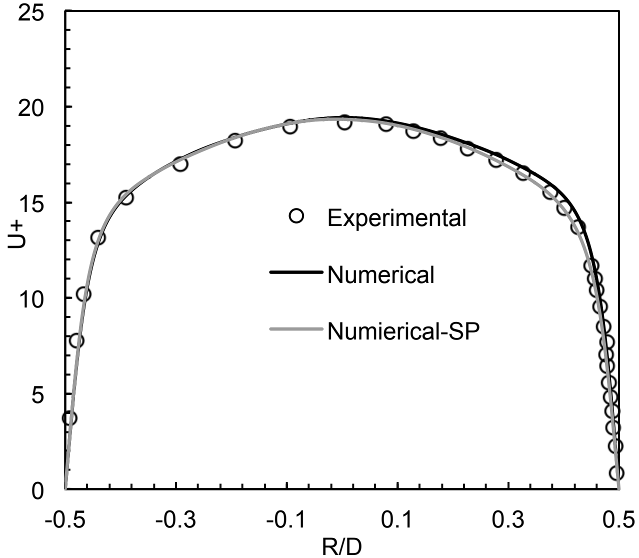
CPU (s)	Nvidia Tesla GPU (s)			
	K20c		C2050	
DP	SP	DP	SP	DP
504.5	39.7	63.1	54.2	96.3

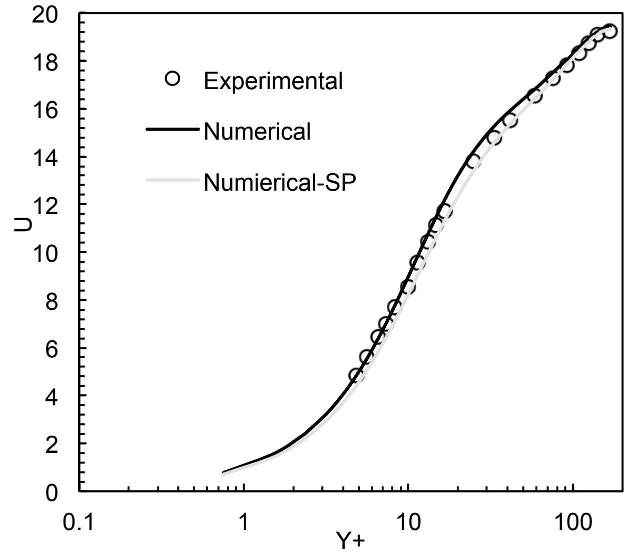
Turbulent Pipe flow

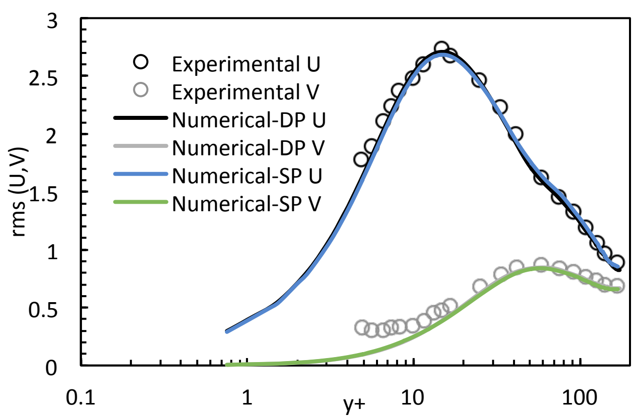
- **Turbulent flow through pipe $Re_\tau = 338$**
- **Non-orthogonal body fitted mesh**
  - 19 point stencil
- **1.78 million grid nodes**
- **36 mesh blocks**
- **Krylov solvers**
  - Sensitive to round off
  - Single precision calculations
    - Relaxed convergence criterion
    - Restricted maximum number of iterations

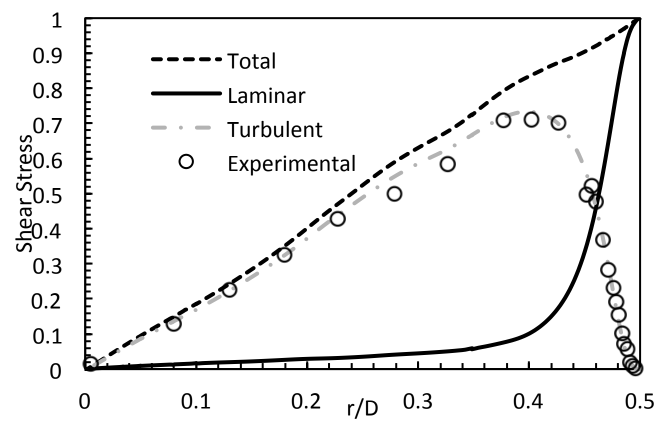


Pipe flow – validation









Experiments by – den Toonder, J. M. J., and Nieuwstadt, F. T. M., 1997, "Reynolds number effects in a turbulent pipe flow for low to moderate Re," Physics of Fluids (1994-present), 9(11), pp. 3398-3409

Pipe flow – performance results

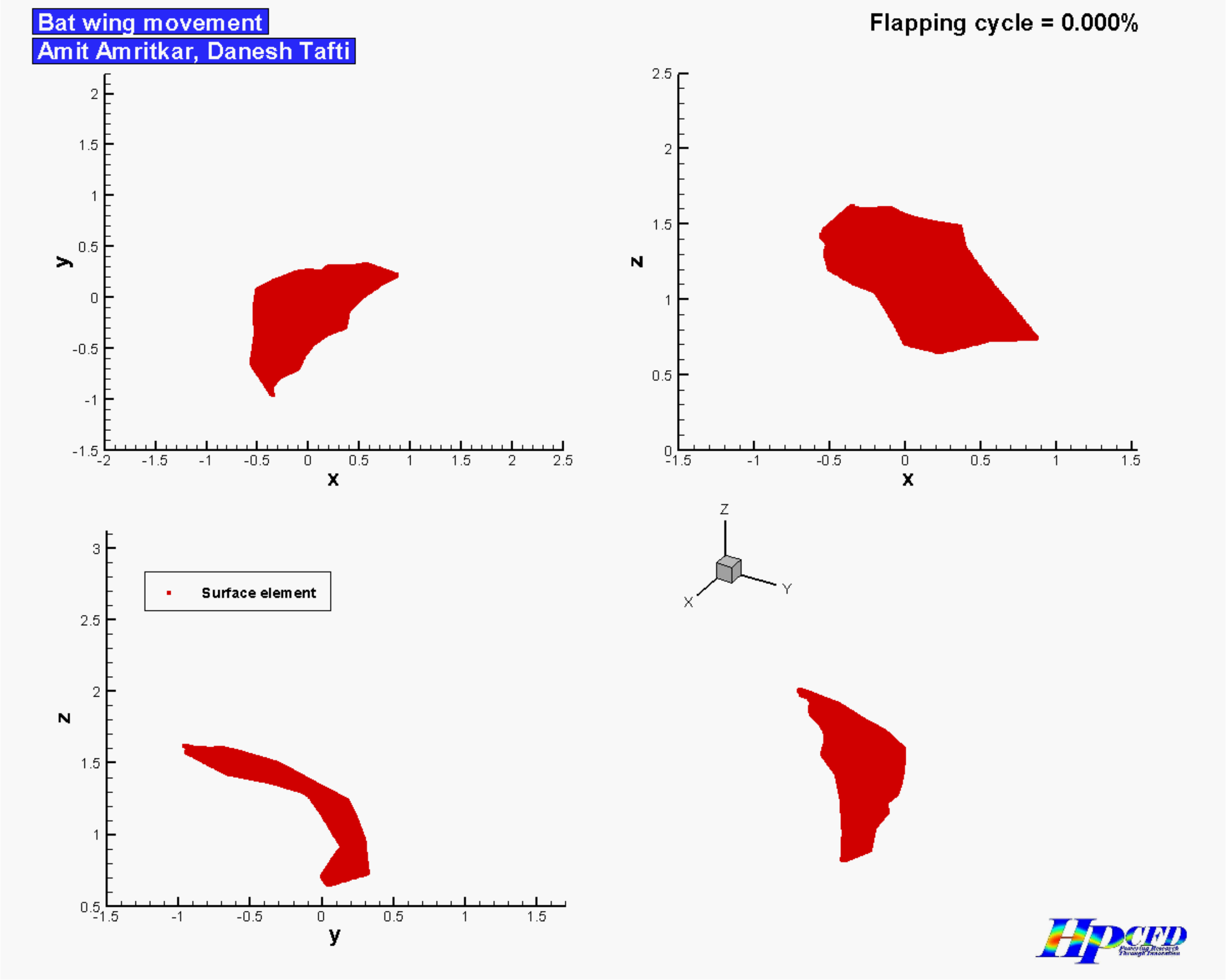
- **Cache block sizes**
  - ‘i’ direction has the most threads

Total threads in each cache block			Time (s)
32x38x40	32x40x40	32x39x40	
512	512	416	**255**
304	200	208	**263**
416	400	380	**270**
190	200	293	**293**

- **Strong scaling**
  - CPU and GPU cache block sizes are optimized

Number of CPU cores or GPUs	CPU (s)	C2050 GPU (s)	
	DP	DP	SP
2	8492	1975	1353
4	4220	1290	806
12	1426	837	492
36	511	386	255

Close up of Bat wing motion

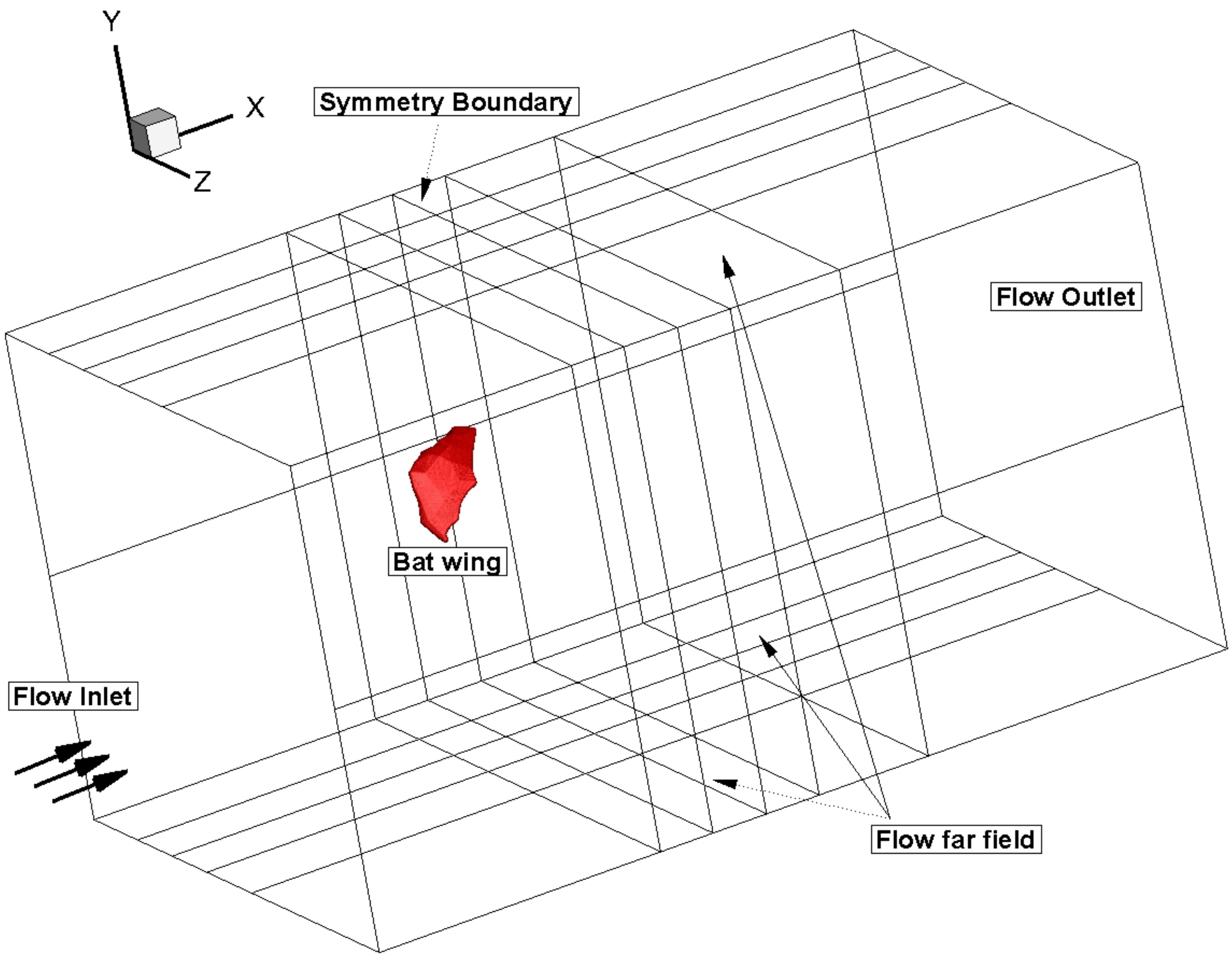


Wing flapping – simulation details

- Flapping of a bat wing using IBM
- 25 million nodes for background grid
- 9400 surface mesh elements for bat wing
- Re = 433

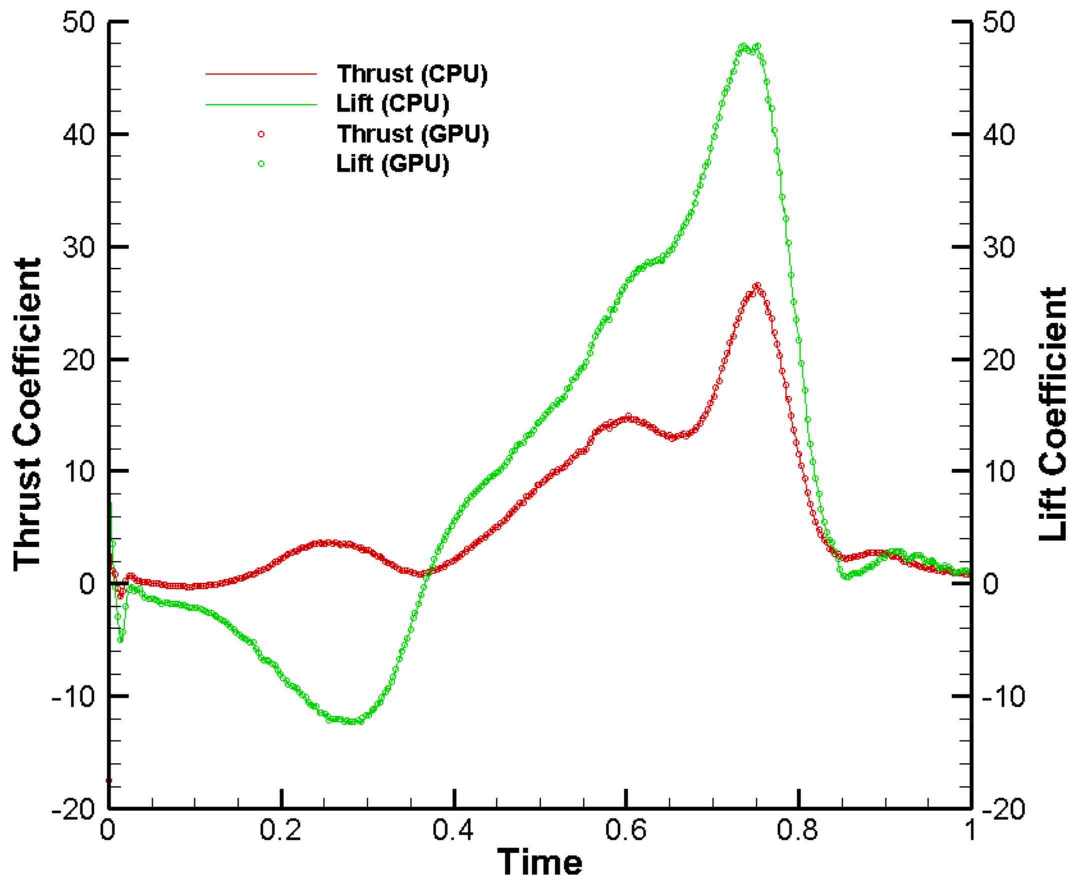
- Runs on HokieSpeed at Virginia Tech
  - 2 Nvidia M2050/C2050 GPUs
  - Dual socket Intel Xeon E5645 CPUs

Wing flapping – geometry

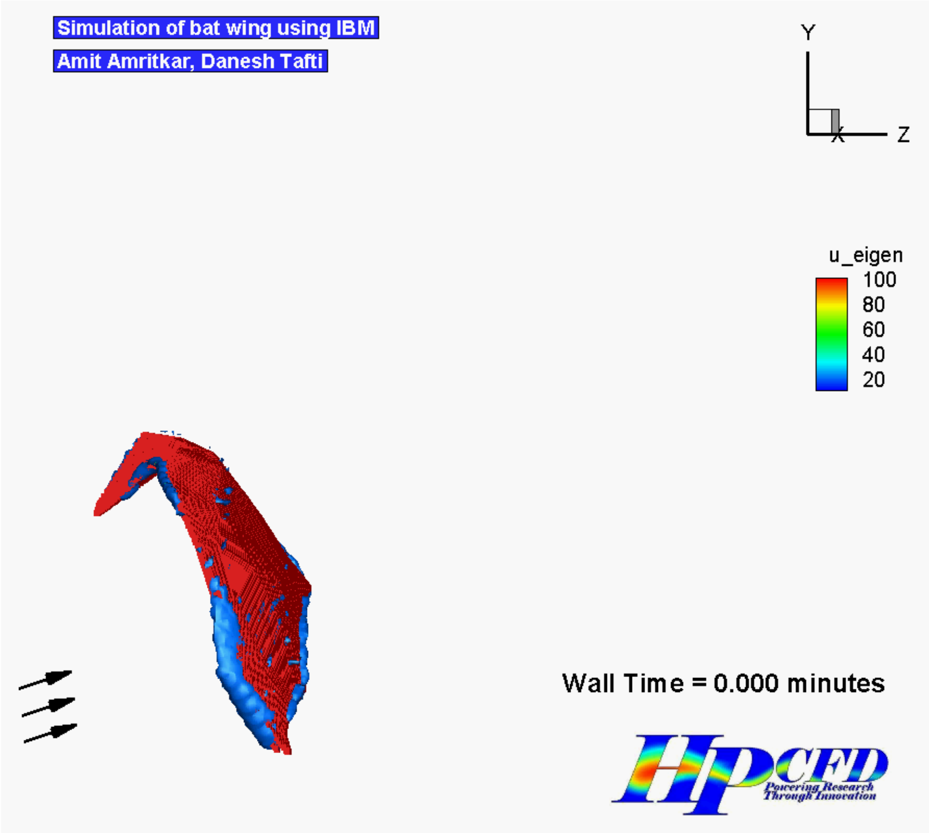


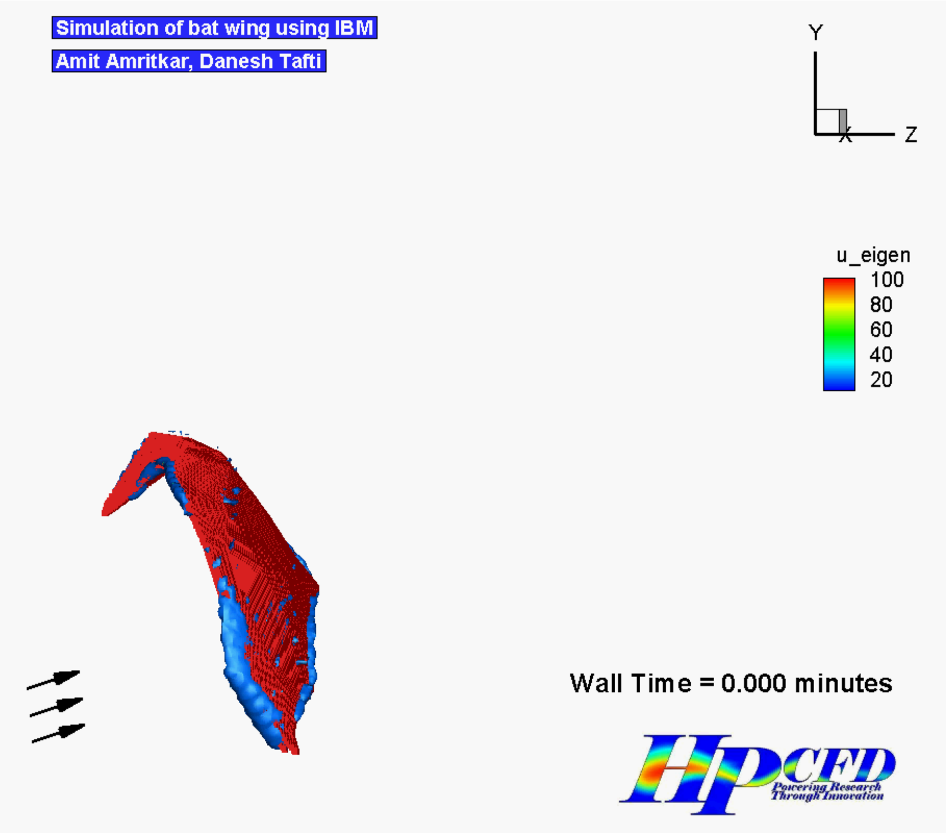
Comparison of results (GPU vs CPU)

- **Lift force over 1 cycle of bat wing flapping**



GPU (60 GPUs) CPU (60 CPU cores)





Solver co-design with Math team

- **Solution of pressure Poisson equation**
  - Most time consuming function (50 to 90 % of total time)
- **Solving linear system Ax = b**
  - 'A' remains constant from one time step to other in many CFD calculations
- **GCROT algorithm**
  - Recycling of basis vectors from one time step to the other

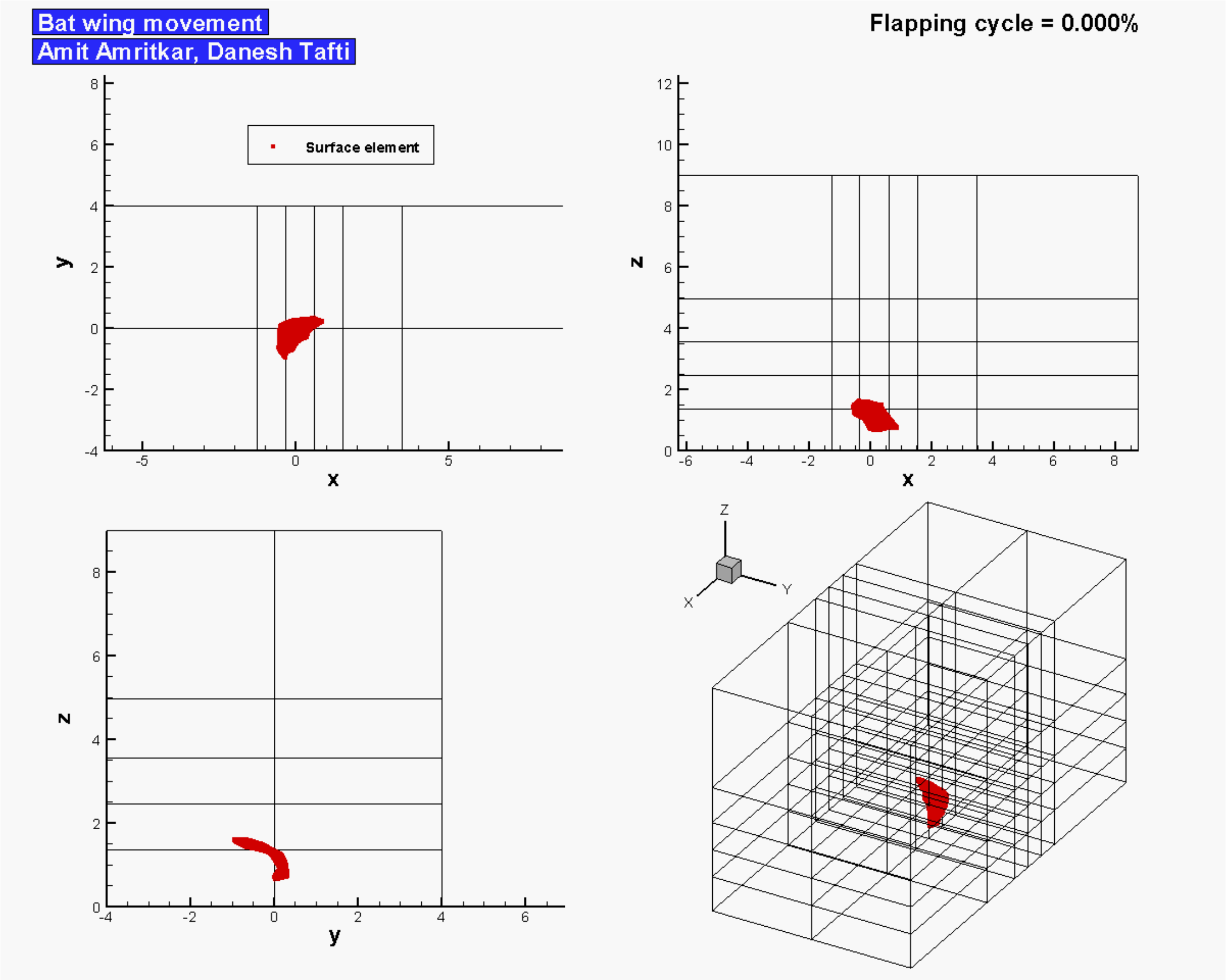
Current/Future work

- ***Use GPU aware-MPI***
  - ***GPU Direct v2 gives about 25% performance improvement***
- ***GCROT algorithm***
  - ***Optimization***
  - **Implement of GCROT on GPUs**
- **Non-orthogonal case (pipe flow)**
  - **Optimization**
- **Use of co-processing (Intel mic) for offloading**
  - **Direct offloading with pragmas**
  - **Combination of CPU and Co-processor**

Publications – *in process*

- **Accelerating Bio-Inspired MAV Computations using GPUs**
  - Amit Amritkar, Danesh Tafti, Paul Sathre, Kaixi Hou, Sriram Chivakula, Wu-Chun Feng
  - AIAA Aviation and Aeronautics Forum and Exposition 2014, 16 - 20 June 2014, Atlanta, Georgia
- **CFD computations using preconditioned Krylov solver on GPUs**
  - Amit Amritkar, Danesh Tafti
  - Proceedings of FESEM 2014, August 3-7, 2014, Chicago, Illinois, USA
- **Recycling Krylov subspaces for CFD application**
  - Katarzyna Swirydowicz, Amit Amritkar, Eric De Sturler, Danesh Tafti
  - FEDSM 2014, August 3-7, 2014, Chicago, Illinois, USA

Fruit bat wing motion

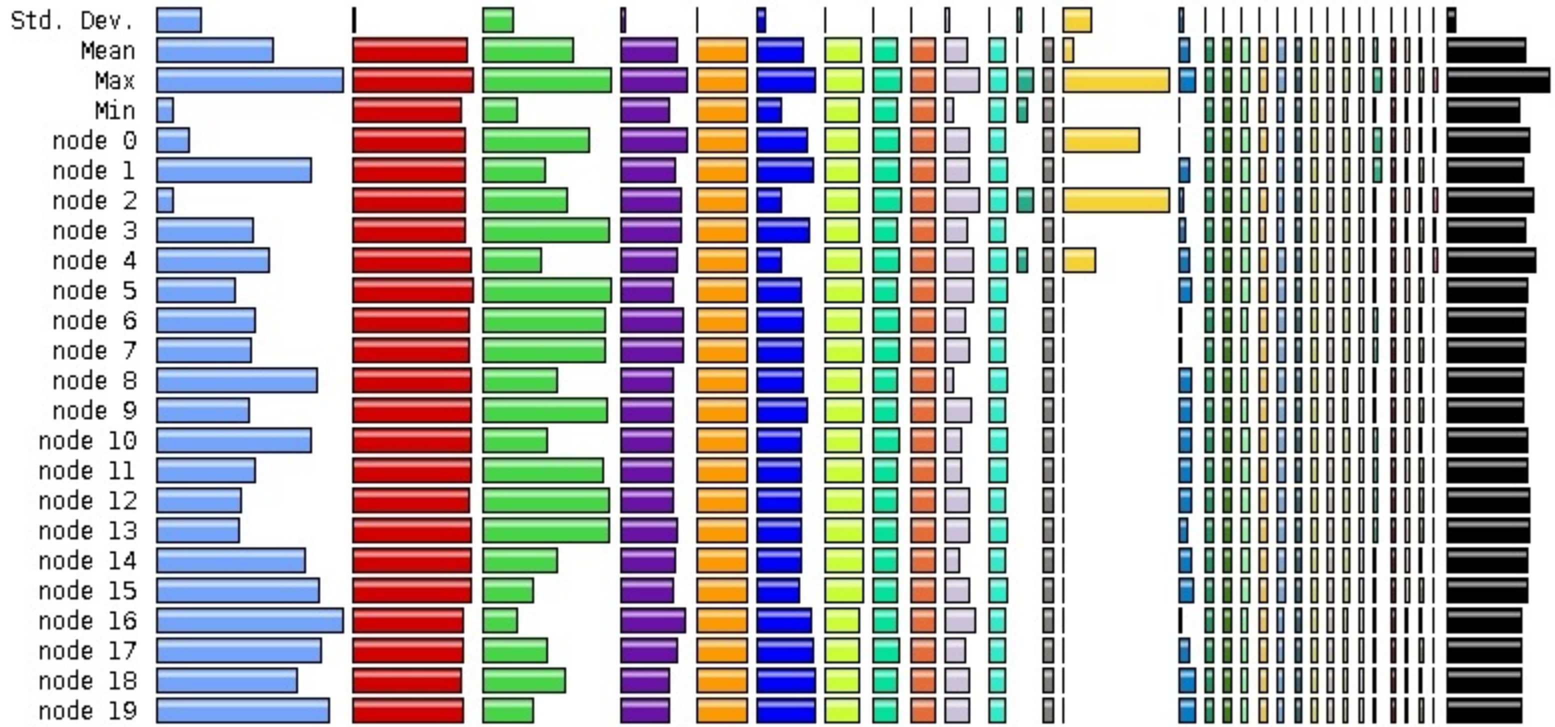


Based on work by – Viswanath, K., et al., *Straight-Line Climbing Flight Aerodynamics of a Fruit Bat.* Physics of Fluids, accepted

TAU profile data – 10 time steps

Metric: TIME
Value: Exclusive

Total = 157s, communication = 70s, initial/end (one time) = 25s



- **Mpi_allreduce**
- **Gpu_mpi_sendrecv**
- Ibm_movement
- Ibm_write_tecplot
- *Gpu_pc2*

- **Mpi_waitall**
- **Mpi_sendrecv**
- Read_grid
- *Gpu_dot_prd*
- Ibm_buffer_alloc

- **Gpu_exchange_var**
- Ibm_projection
- Alloc_field_var
- Ibm_locate_ijk