Parallel Rotor Simulation Tools

C. B. Allen

Department of Aerospace Engineering,

University of Bristol.

Rotor Simulation Problem

- Rotor flow simulation extremely expensive

  - need much finer meshes away from surface to capture vortical wake than fixed-wing

  - long numerical integration times for wake to develop.

Rotor Simulation - Talk

- Long term: want to simulate complete vehicle
  - Main rotor + fuselage + tail rotor
  - → Wake capture over large distances

  → Parallelisation essential.

- Talk:
  - Rotor simulations performed on HPCx
  - Tools developed towards long term goal.

Rotor Simulation Tools - Talk

- Large simulation requirements:

  - Efficient parallel flow-solver
  - Fast high quality grid generator
    - *with low user-input requirements*
  - Efficient parallel mesh motion scheme
  - Post-processing/visualisation tools.

Grid Generator

- Grid generation often the bottle-neck for CFD use.

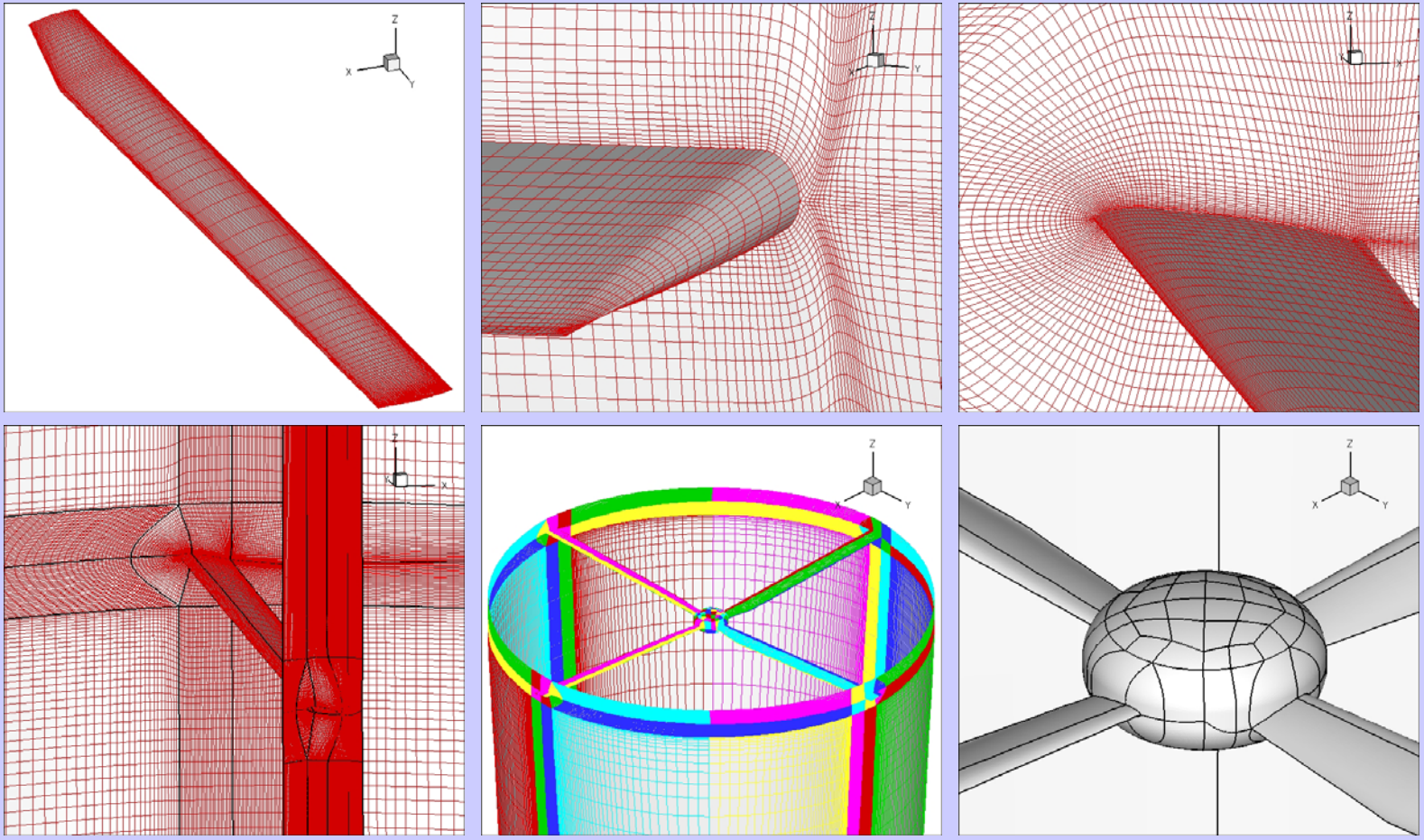
- Structured multiblock generator for fixed- and rotary-wing flows developed. (Used at Agusta-Westlands and QinetiQ)

- Fully automatic
  - only NPOINTS needs to be input
  - optimum grid parameters (point distribution, spacing, blocking) computed automatically.

- Very efficient, so can be run serial
  - 64 million point, 500 block mesh
  < 30 minutes on P4
  < 2Gbytes RAM
  With no user-input.

Grid Generator

7A Rotor. Hover and forward flight examples.



Flow-Solver

- Implicit multiblock, multigrid, unsteady inviscid solver for fixed-wing and rotors
  - Parallelised using MPI.

- Decomposition linked to G.G. so can be run prior to generation stage to optimise block sizes/balancing.

- All array allocation and pointers set on each CPU separately (1D arrays)
  → No global data storage,
  → Cost: ~ 0.5Gbytes/1M points (5MG levels).

Outline of Talk

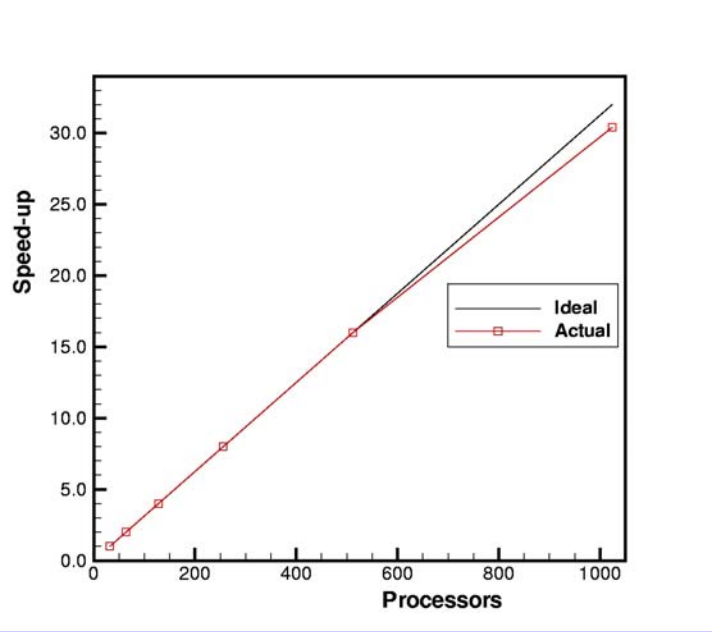
- Outline of problem.
- Description of grid generator.
- Description of parallel flow-solver.
- **Parallel performance.**

  (Code previously validated for hover cases – not shown here)

Performance on HPCx

- Benchmarks performed on code by Andy Sunderland of Terascaling Team
- 20 Million point hover case

  - 960x speed-up on 1024 CPU's
  - Awarded `Gold Star'.

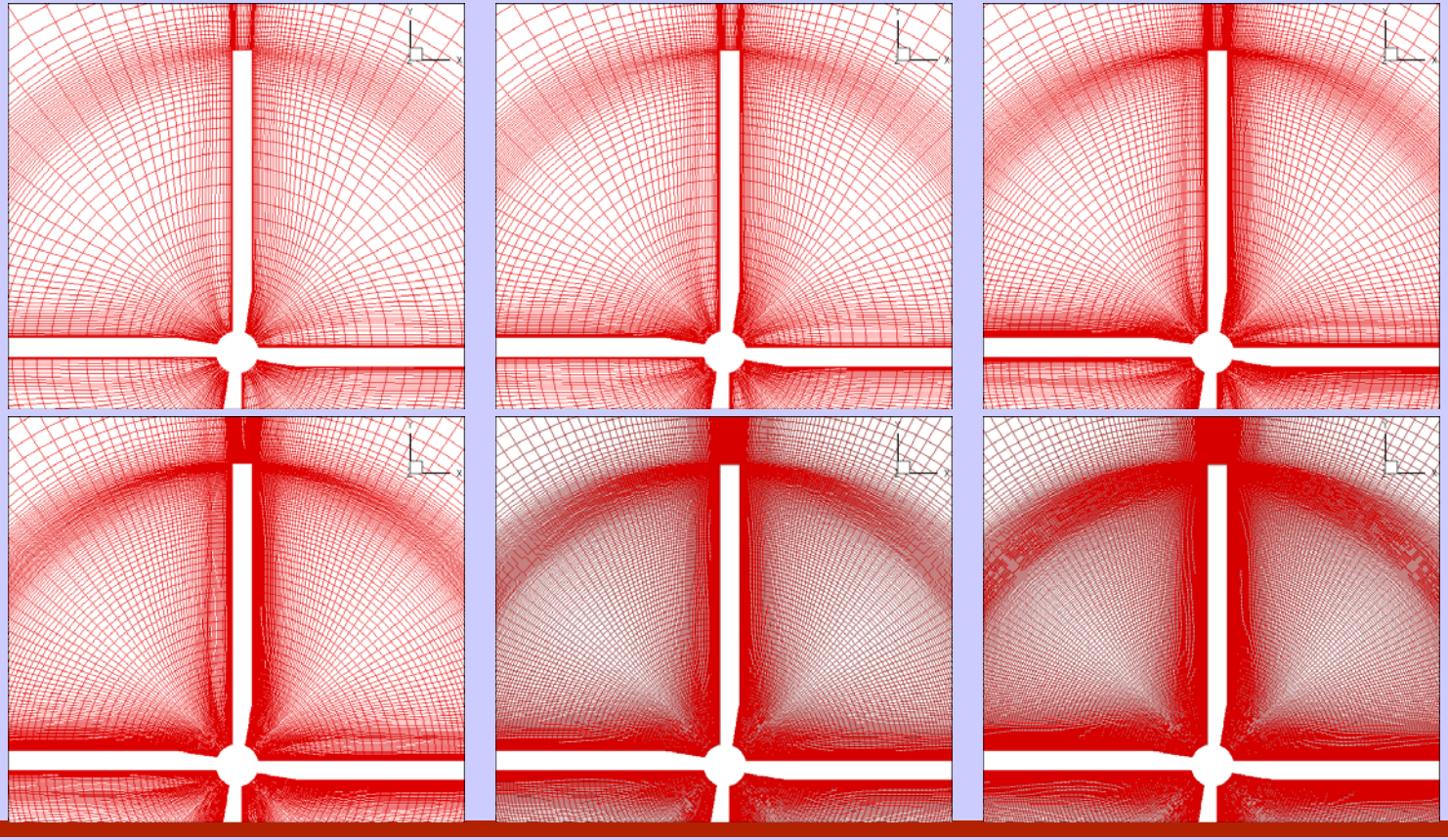


Outline of Talk

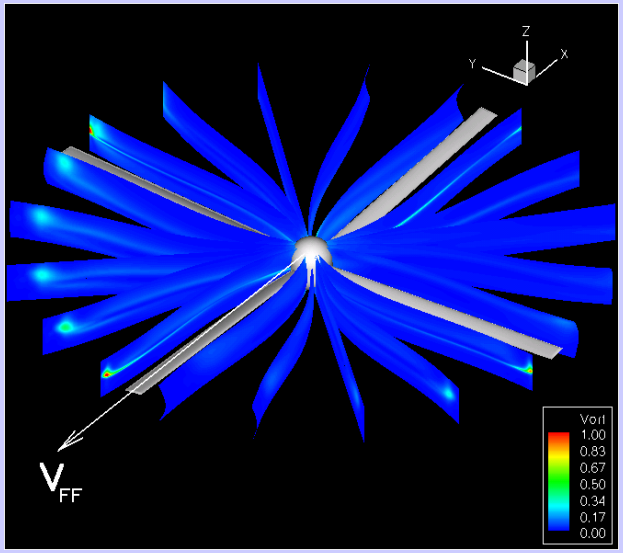
- Outline of problem.
- Description of grid generator.
- Description of parallel flow-solver.
- Parallel performance.
- **Results of forward flight simulations.**

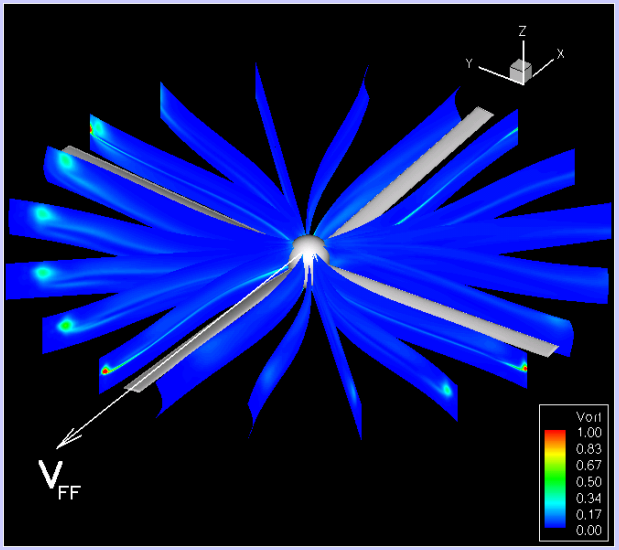
Forward Flight Case

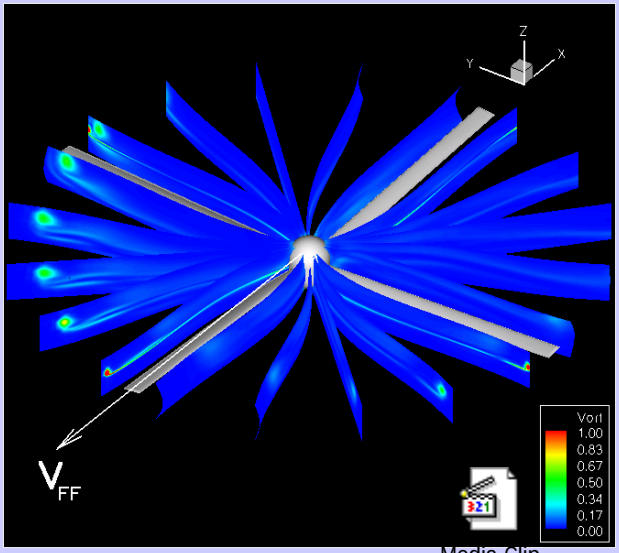
- 7A 4-bladed rotor. Mtip = 0.618, mu=0.214 shaft -3.72 degrees.
- Run on HPCx using upto 512 CPUs.
- Rotor disk grid - 1M, 2M, 4M, 8M, 16M, 32M points.



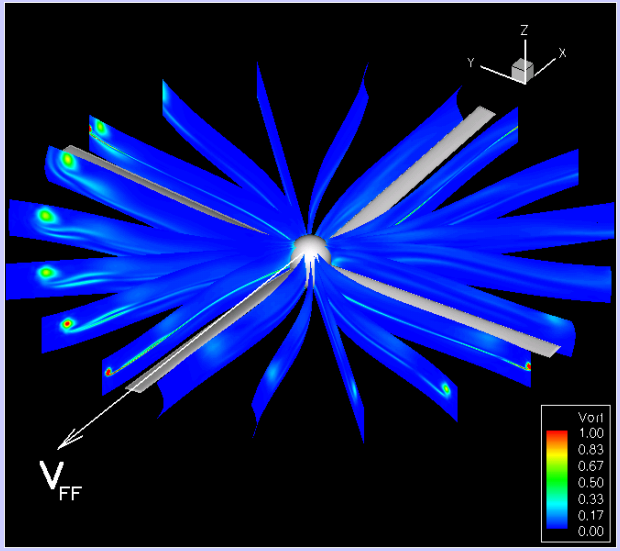
Vorticity Shading (0.0-1.0) 1,2,4,8,16, 32M.

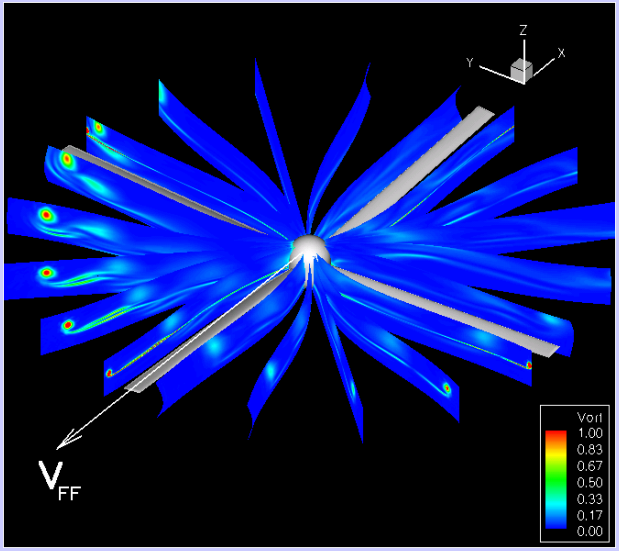


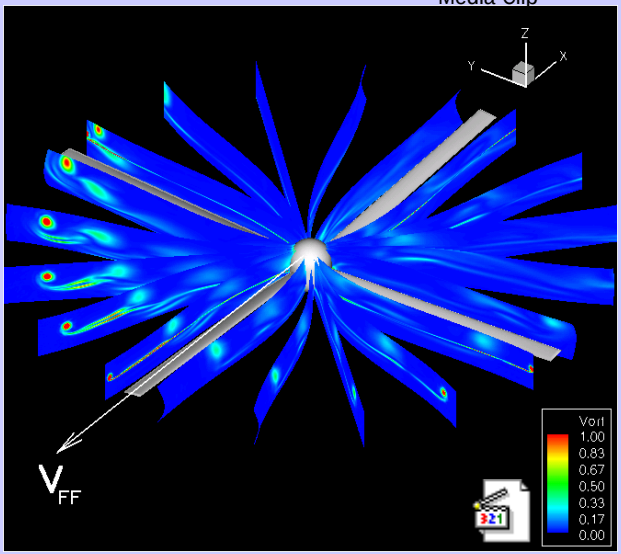




Media Clip







Media Clip

Outline of Talk

- Outline of problem.
- Description of grid generator.
- Description of parallel flow-solver.
- Parallel performance.
- Results of forward flight simulations.
- Unsteady flows →**Parallel mesh motion.**

Parallel Mesh Motion

- Any simulation with moving surfaces requires grid motion
  - rotors have cyclic pitch motion to balance load.

- Computation of moving or deformed mesh every time-step can be very expensive

- Seek efficient and parallel mesh movement scheme.

Parallel Mesh Motion

- Want method that:

  1) Works for structured and unstructured.

  2) Requires NO connectivity, i.e. motion of any point depends solely on its coordinate → this satisfies 1). Also, means no communications required.

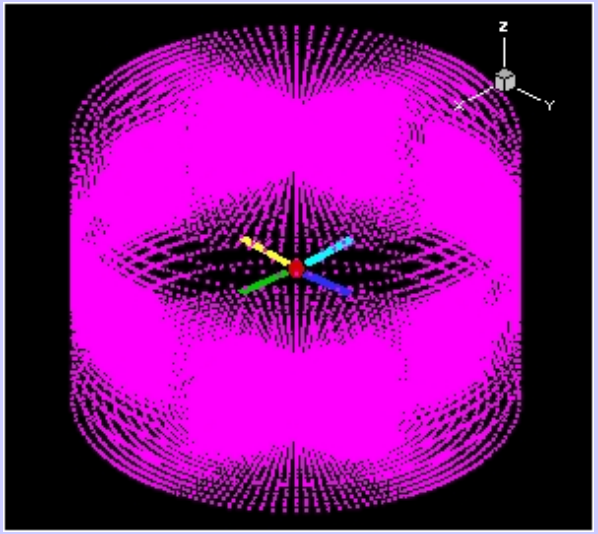
  3) Maintains grid quality (orthogonality) by accounting for effective rotation (essential !).

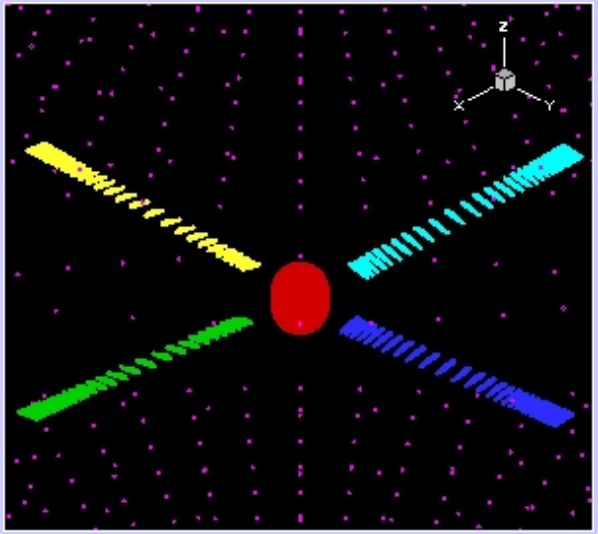
  4) Needs no computation of parameters during simulation, and is algebraic, i.e.

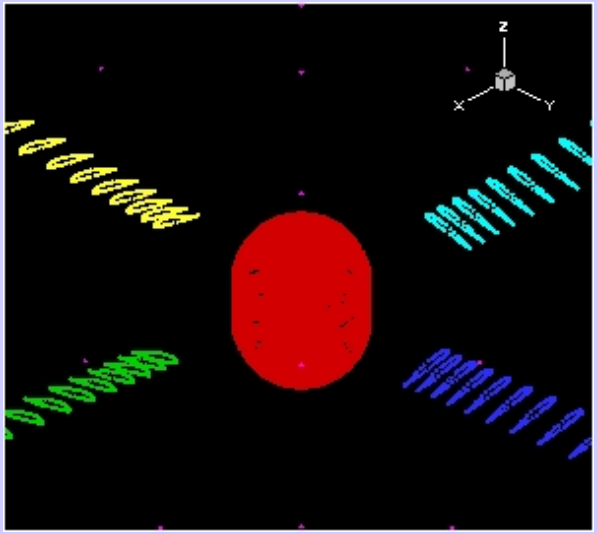
$$\mathbf{X}(t) = \mathbf{f}(\mathbf{X}(t=0), \Delta\mathbf{X}\text{surface}(t))$$

Parallel Mesh Motion

- Pre-solution stage, mesh blocks scanned and surface data extracted, as unstructured data set
  → NSURF clouds of points.

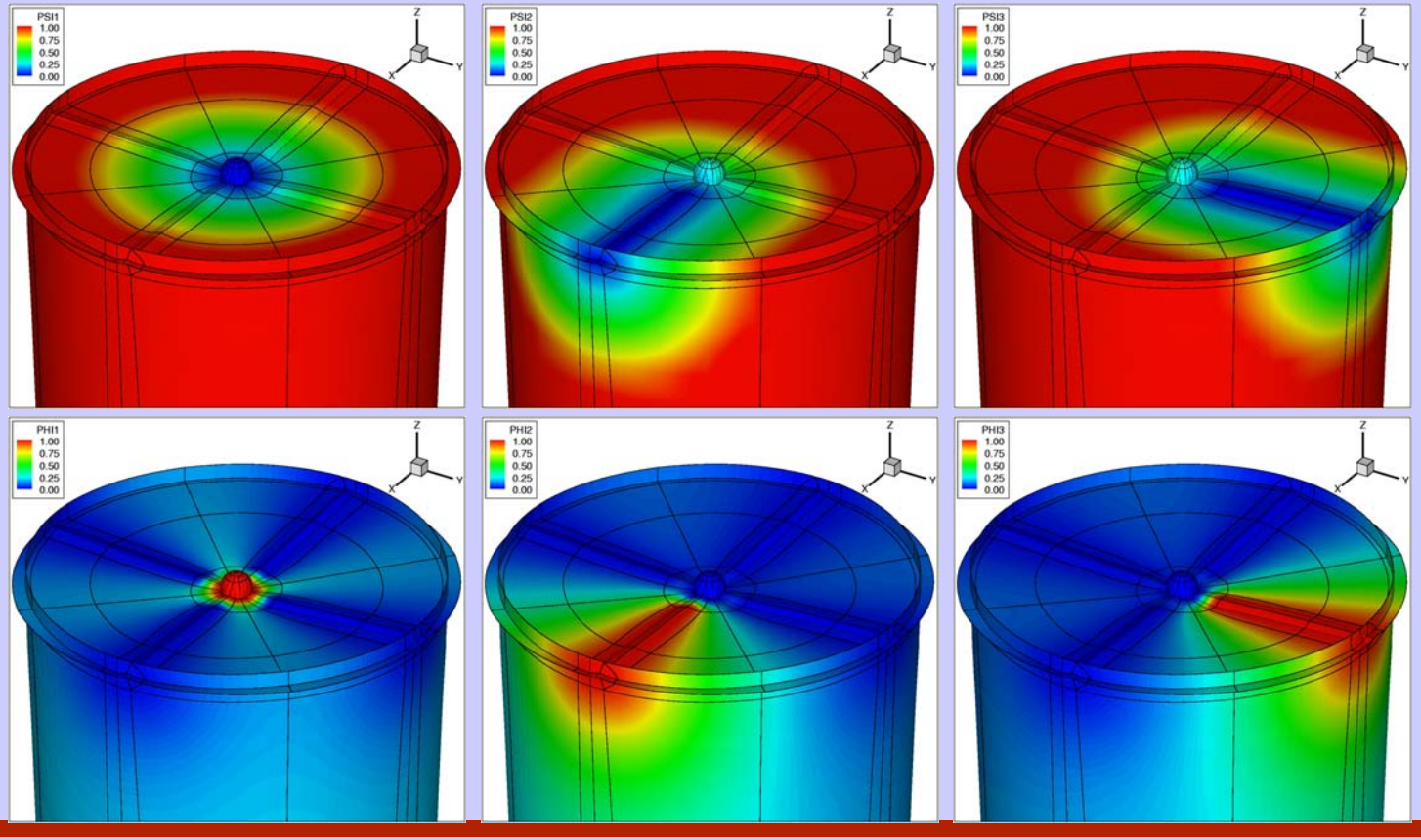






Parallel Mesh Motion

- Search algorithm then defines weightings
  - parametric functions and surface functions
  - these MUST be smooth.



Parallel Mesh Motion

- Mesh at any time then algebraic functions of surface displacement and rotations.
- NO connectivity/communication – ‘perfectly parallel’, each CPU only updates its blocks. Boundaries match by default.
- Example: cyclic pitch

$$\theta = \theta s \sin(\psi) + \theta c \cos(\psi)$$

$$\psi = \text{azimuth angle.}$$

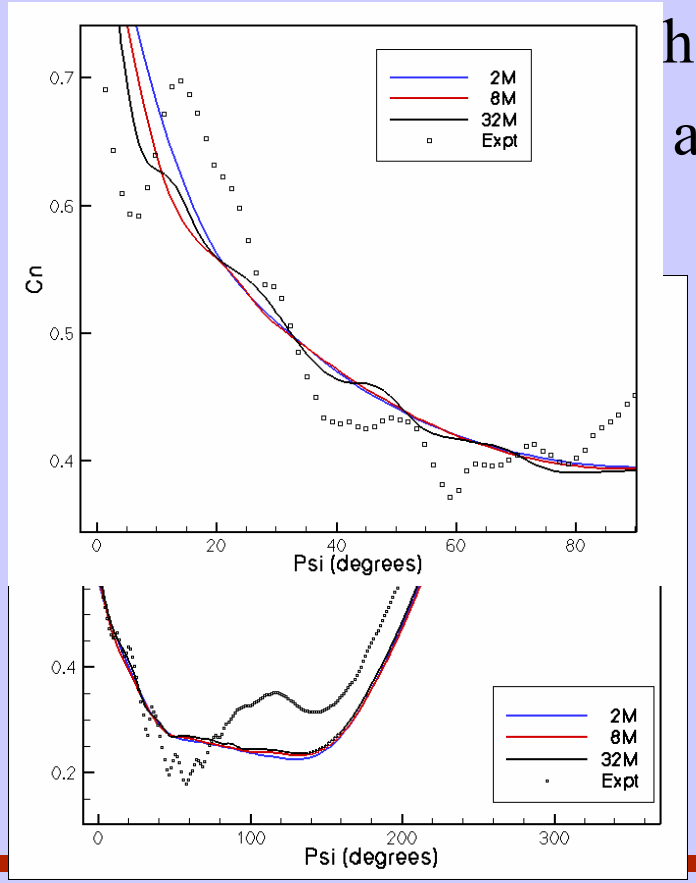
- Assume

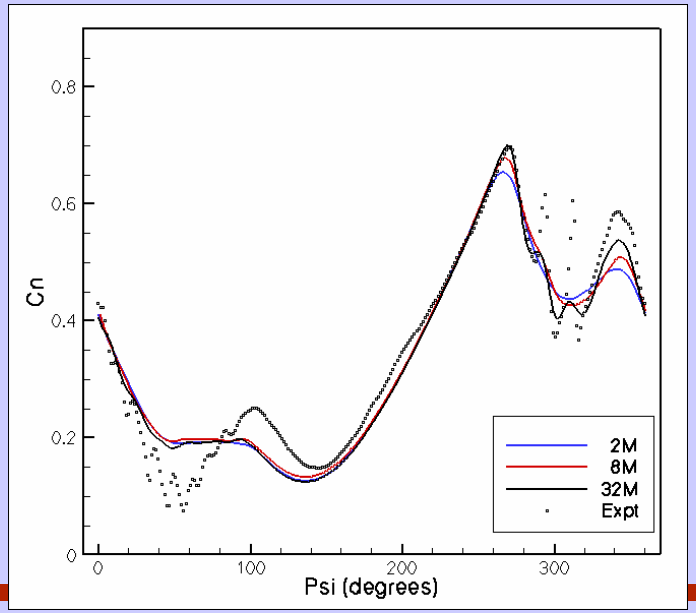
$$\theta s = -\theta c = 45 \deg$$

Media Clip Media Clip Media Clip Media Clip

Forward Flight with Cyclic Pitch

- Same test case: 7A 4-bladed, 2M, 8M, 32M points
  - $\theta = \theta s \sin(\psi) + \theta c \cos(\psi)$
  - $\theta s = 2.66\,\text{deg} \qquad \theta c = -2.44\,\text{deg}$
- h motion < 1% of time-step cost.
- at r/Rtip = 0.82, 0.92



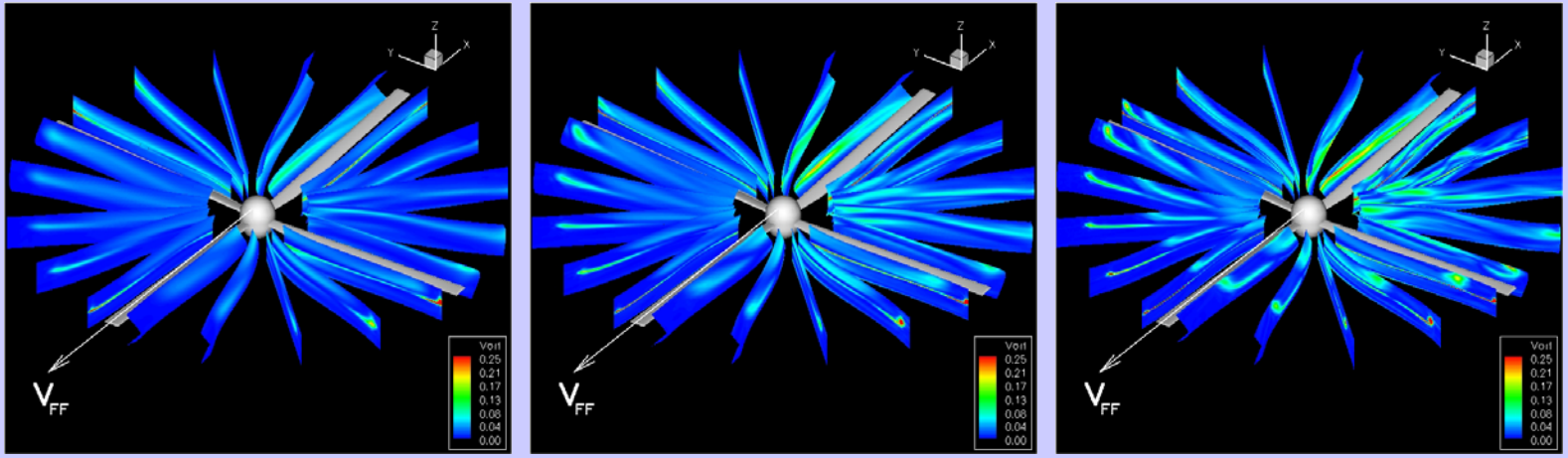


Outline of Talk

- Outline of problem.
- Description of grid generator.
- Description of parallel flow-solver.
- Parallel performance.
- Results of forward flight simulations.
- Parallel mesh motion.
- **Visualisation options.**
- Conclusions and ongoing work.

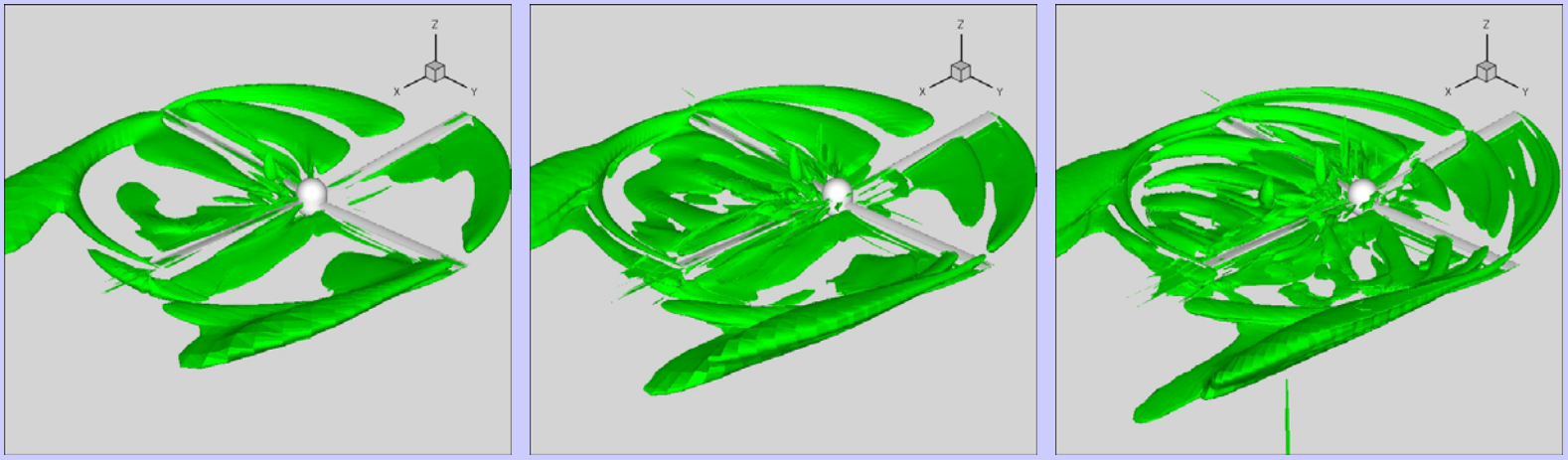
Visualisation Options

- Many post-processing codes developed - vorticity often used on planes



Media Clip

- Lambda2 quantity more useful for field data



Conclusions

- Large rotor simulations run on HPCx
  → Presented tools developed for these simulations

  - Unsteady **flow-solver**: almost linear scaling upto 1024 CPU's – awarded Gold Star for scaling performance,

  - Fast, efficient, fully automatic **grid generation** software generates very fine meshes with *no user input*,

  - `Perfectly parallel' algebraic **mesh motion** scheme requires *no* connectivity data or communication. Costs < 1% of flow-solution,

  - Lambda2 quantity more useful then vorticity for flow field **visualisation**.

→ Moving towards simulation of complete vehicle. (UKAAC2)

Ongoing Work

- Incorporation of sliding grid plane option, to allow rotor + fuselage simulation.
- Inclusion of structural model, aeroelastic simulation.
- Aerodynamic optimisation → aeroelastic optimisation.

Acknowledgements

- EPSRC for funding CPU time on HPCx for UK Applied Aerodynamics Consortium. Grant GR/S91123/1.
- Andy Sunderland for performing scaling test.
- Roderick Johnstone for Lambda2 work.