EMCON Technologies℠

DANSIS 2007
New Trends in CFD

OpenFOAM and STAR-CD

Integration, Interoperability and Symbiosis

Dr. Mark Olesen

Mark.Olesen@emconTechnologies.com

Overview

- Choosing a CFD code
  - Motivation
  - Costs
  - Concerns
- Phase-In
  - Requirements: solver, workflow
  - Interoperability
- Test Cases
  - with/without porosity
- Summary
- STAR-CD application example (Time permitting)
  - DPF, Vaporizer

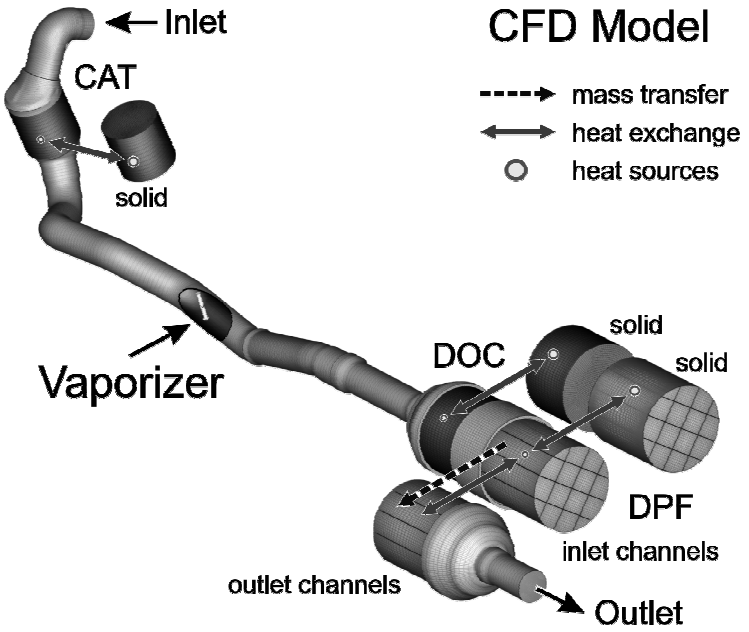
Company Information

- OEM emission technology – light and commercial vehicles
  - $3 billion business, 19 countries, 7,500 employees
  - privately owned – One Equity Partners (JPMorgan Chase & Co)

- Simulation in Augsburg (Europe/Asia Headquarters)
  - Acoustics, CFD, FEA
  - 40-60 cpu Linux cluster – Grid Engine
  - abaqus, GT-Power, NASTRAN, OpenFOAM, RadTherm, STAR-CD
  - HyperMesh, ICEM, pro-STAR

Our Motivation for trying OpenFOAM

- Geometry Optimization
  - Potentially *many* geometries (> 500 per study)
  - Commercial licenses too prohibitive
- Reduce (or limit) long-term license costs
  - additional calculation capacity
- Alternative
  - Capabilities
  - Supplier



Choosing a CFD code

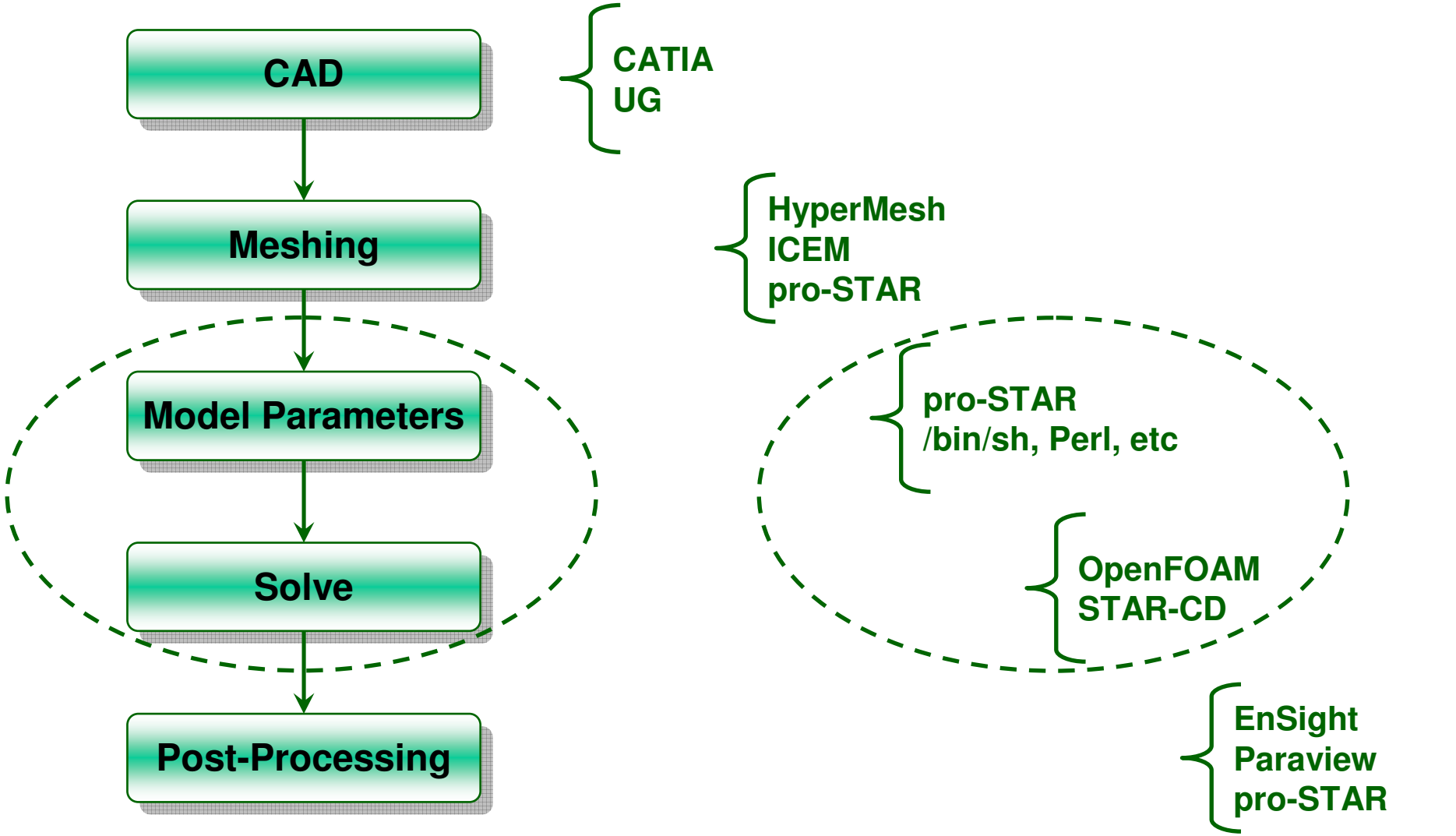
- Cost
  - Licenses, support, infrastructure
- Capability
  - chemistry, sprays, moving mesh, turbulence models, etc.
- Flexibility
  - Bending the code to do what you need
  - Access to fields, operators, data structures, etc.
  - Avoiding vendor lock-in
- Usability
  - Robustness, friendliness, performance

Costs

- CFX, Fluent, STAR-CD, etc.
  - yearly license costs
    - advanced budget planning
    - licenses > 2–4 hardware costs †
- OpenFOAM
  - support only
    - unlimited usage
  - better utilization of cluster capacity
- Increase capacity
- Use all cpu cores

† ignoring amortization, discounts, etc.

Replace my *entire* CFD program?



Quick Checklist (1)

- OpenFOAM
  - support directly from developers
  - fast problem resolution
  - can customize to suit requirements
  - can (must) change the source code

- STAR-CD
  - existing knowledge base, customer acceptance
  - many models are ready to go (and should likely work)
  - GUI for most settings
  - **pro-STAR** – mesh manipulation

Quick Checklist (2) – Pre/Post-Processing

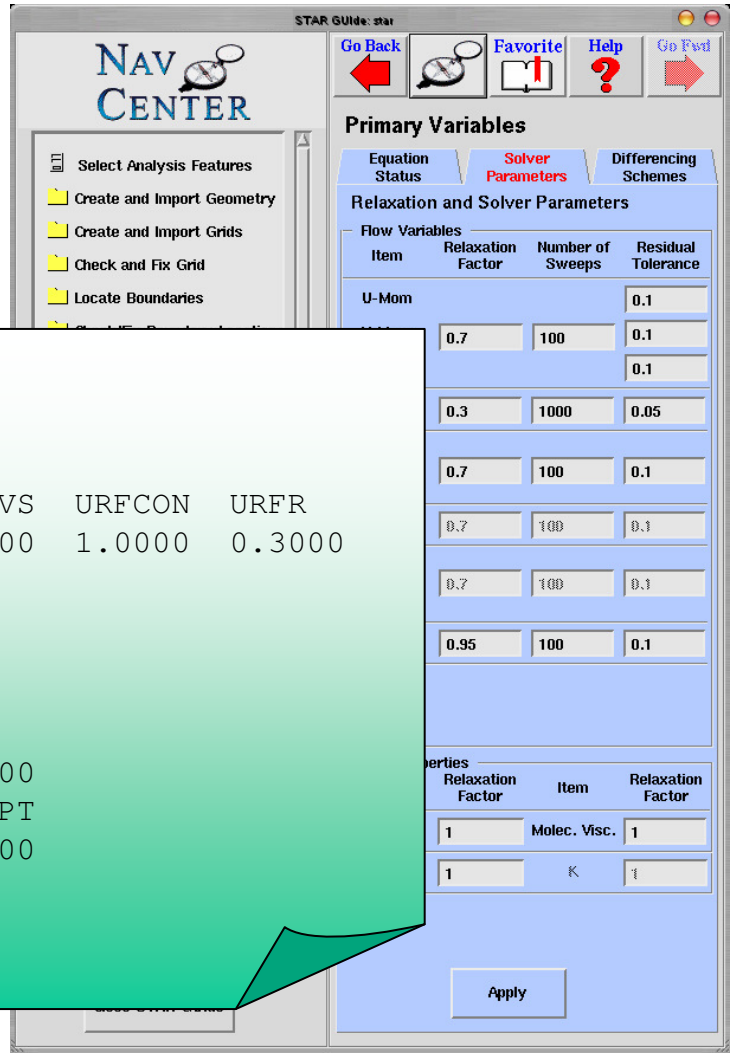
- OpenFOAM
  - mesh manipulation
    - command-line
  - boundary identification
    - autoPatch (command-line), patchTool (GUI)
  - solver settings
    - FoamX: Java + CORBA → mostly useless
    - **text editor (or scripting)**
  - Post-Processing
    - EnSight, paraview, etc.

weak

- STAR-CD – **pro-STAR** for all the above
  - use it for OpenFOAM as well

STAR-CD – GUI and Manual

- relax,0.7,0.3,0.7,0.95,1,1,1, , , , ,
- switch 50 on → CASENAME.ctrl




```
 ECHO   BOECHO   ITEST   IRESI   NDUMP   NFSAVE
  F      F        T       F       10       10
 DT             MAXCOR    RESOC      URFPCR
0.500000E-02       20    0.25000    1.00000
 URFUVW    URFP     URFTE    URFT     URFTVS    URFDEN  URFLVS   URFCON   URFR
  0.6000   0.2000   0.7000   0.7000   1.0000   0.8000   1.0000   1.0000   0.3000
 SORMAX           IJKMON
0.00001000         16060
 GGUVW     GGKE     GGT      GGDEN    GGSCA
  0.0000   0.0000   0.0000   0.0100   0.0000
  SORU     SORV     SORW     SORP     SORK     SORE     SORT
  0.1000   0.1000   0.1000   0.1000   0.1000   0.1000   0.0100
   NSWPU    NSWPV    NSWPW   NSWPP     NSWPK    NSWPE    NSWPT
     100      100      100     100      100      100      100
 NCRPR     NFRRE
   0         500
```

- $ cp CASENAME.ctrl CASENAME_0001/

OpenFOAM – Manual

- system/fvSolution

```
relaxationFactors
{
    p               0.3;
    U               0.7;
    k               0.9;
    epsilon         0.9;
    h               0.9;
}
```

- constant/turbulenceProperties
  - w/o ‘include’ directive

```
turbulenceModel kEpsilon;
turbulence      on;

kEpsilonCoeffs
{
    Cmu             Cmu [0 0 0 0 0 0 0] 0.09;
    C1              C1 [0 0 0 0 0 0 0] 1.44;
    C2              C2 [0 0 0 0 0 0 0] 1.92;
    C3              C3 [0 0 0 0 0 0 0] -0.33;
    alphah          alphah [0 0 0 0 0 0 0] 1;
    alphak          alphak [0 0 0 0 0 0 0] 1;
    alphaEps        alphaEps [0 0 0 0 0 0 0] 0.76923;
}
```

- All registry objects → readIfModified()
- Same setup:
  - /bin/sh, Perl, CVS, etc

OpenFOAM – Manual *is* sometimes better

- **constant/**
  - **polyMesh/**
  - thermophysicalProperties
  - turbulenceProperties
- **system/**
  - controlDict
  - fvSchemes
  - fvSolution
- Initial and boundary conditions:
  - **0/**
    - T, U, epsilon, k, p
- Results:
  - **1.25e-5/** , **100/** , etc
    - T, U, epsilon, k, p, rho, Ma, mut, yPlus, etc.

OpenFOAM – Concept

- C++ toolkit for building CFD solvers
- Modular, Object-Oriented
  - define a solver for a particular task
  - cf. monolithic with many if's and switches
- Abstract
  - mathematical operators:
    - eg, div(), grad(), laplacian()
- Open, Extensible
- Not just for freaks
  - Define a particular solver *once* and keep reusing it

OpenFOAM – Phase-In (1)

- Introduce OpenFOAM alongside commercial code
  - Free download
  - Learning by doing (no time limit)
- Mix and match
  - Find synergies
  - Pick the best (favourite) features from each
- New capabilities
- New flexibility
  - See where it goes

OpenFOAM – Phase-In (2)

- Short-Term
  - OpenFOAM for standard CCC calculations
  - Integrate in standard workflow

- Middle-Term
  - Geometry Optimization
  - More Complex Phenomena
    - Reacting Flow, Spray, DPF, etc

- Long-Term
  - Open-ended
  - General toolkit for miscellanea

Workflow Requirements

1. Import of STAR-CD mesh files

2. Export of OpenFOAM mesh files

3. Export of OpenFOAM results
   - EnSight
   - pro-STAR (optional)

Solver Requirements (1)

- Standard Solver
  - U, p, T, k/ε
  - rho(p,T), Ma < 1.3
  - steady-state (SIMPLE)
  - possibly transient SIMPLE *open*

- Anisotropic Porosity Model
  - local coordinate system
  - Darcy / Forchheimer
  - cell zone specific

$$-\frac{\Delta P}{\Delta L} = D\,\mu|V| + F\,\frac{\rho}{2}\,V^2$$

- Support costs
  - ca. 36 hours

Solver Requirements (2)

- Boundary Conditions
  - inlets – integral
    - constant massflow, normal to inlet could contain swirl
    - turbulent intensity and length scale
  - outlets – pressure

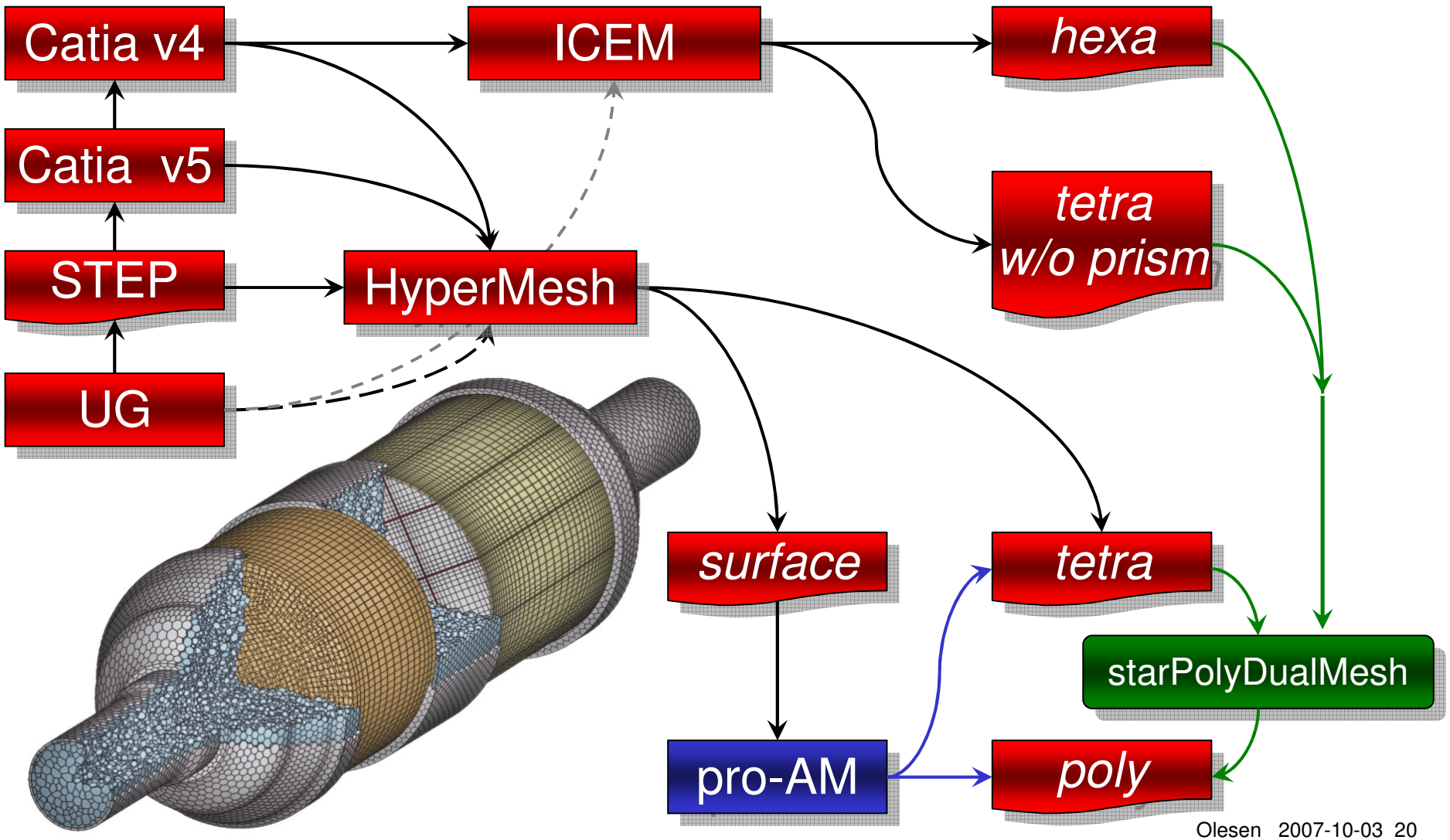
- Extra STAR-CD features
  - Baffles
    - as slip/no-slip walls
    - as porous flow resistances *open*

- Integral boundary conditions and baffles
    - implemented w/o support

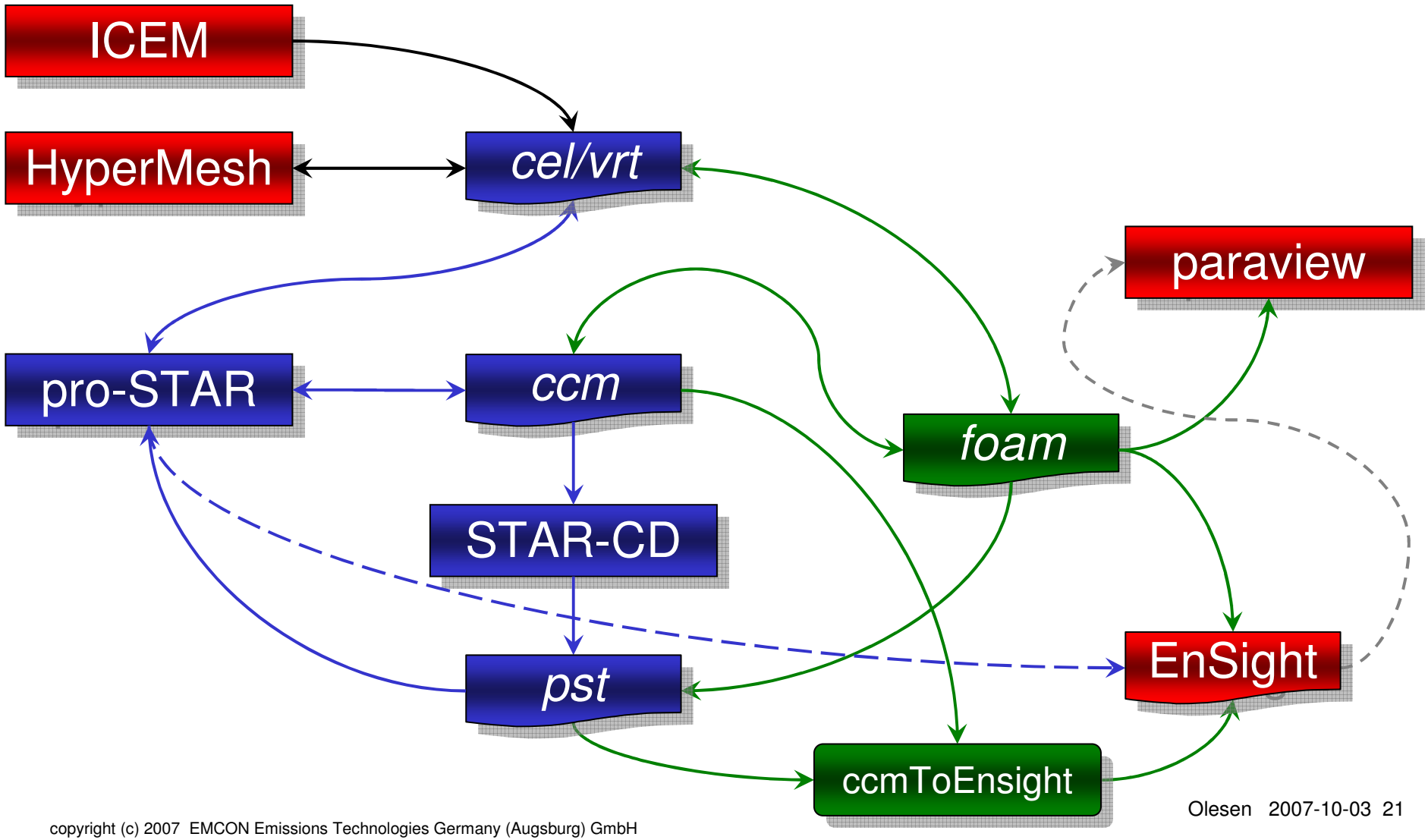
Interoperability – Library / Utilities

- Library Ingredients
  - ccmReader, ccmWriter
    - polyMesh ←→ CCM file
  - starMeshReader, starMeshWriter
    - polyMesh ←→ .cel/.vrt/.bnd files
  - ensightFile, ensightParts, etc.
    - polyMesh → EnSight files
  - polyDualMesh
    - dualize polyMesh→ polyMesh
- Utilities
  - ccmToFoam, foamToCcm
  - star4ToFoam, foamMeshToStar
  - foam(Zone)ToEnsight
  - ccmToEnsight
    CCM file
    → EnSight files
  - starPolyDualMesh
    .cel/.vrt/.bnd files
    → polyMesh
    → dualize polyMesh
    → .cel/.vrt/.bnd files

Interoperability – Mesh Sources



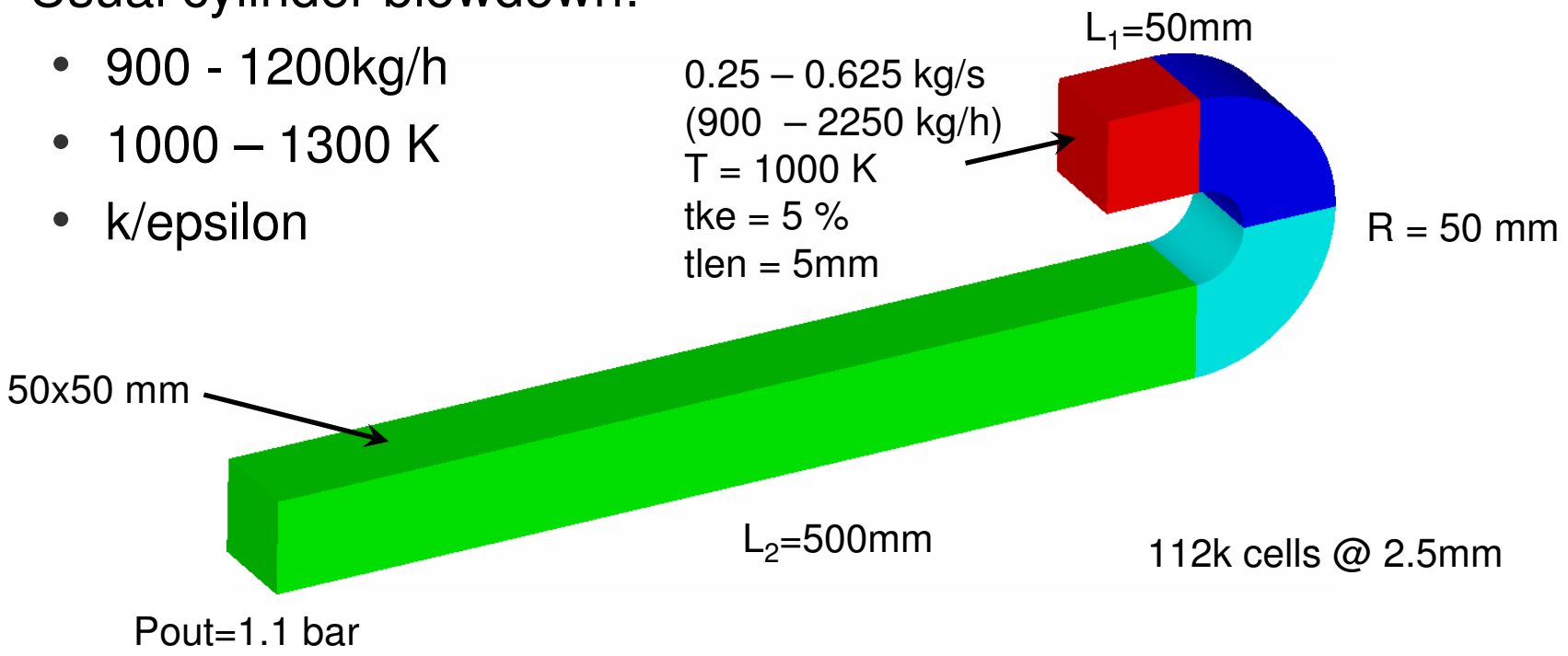
Interoperability – Data Formats



Simple “Hello World” test case

- Anonymous geometry
  - blockMesh for simple (compact) description

- Usual cylinder blowdown:
  - 900 - 1200kg/h
  - 1000 – 1300 K
  - k/epsilon

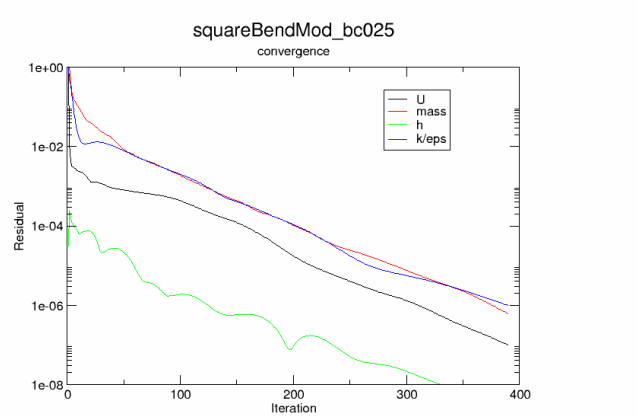


Solver parameters

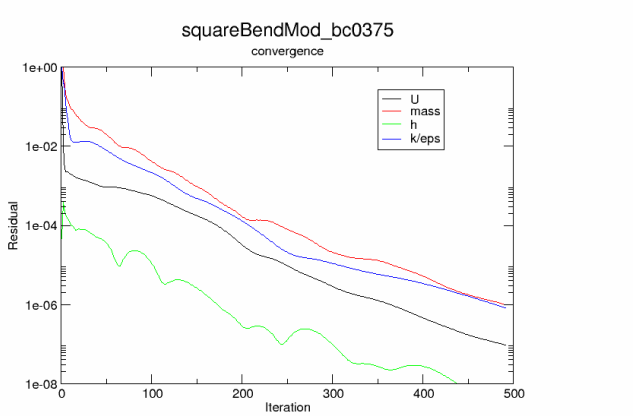
- STAR-CD
  - SIMPLE, UD
  - k/epsilon – std
  - AMG, 1e-6, double
  - relax
    - U=0.7, p=0.3
    - k/eps=0.7, h=0.95

- OpenFOAM
  - SIMPLEC, UD
  - k/epsilon – std
  - GAMG, 1e-7, double
  - relax
    - U=0.9, p=1
    - k/eps=0.9, h=0.95
  - rhoSimplecFoam
    - details (ask Henry Weller)

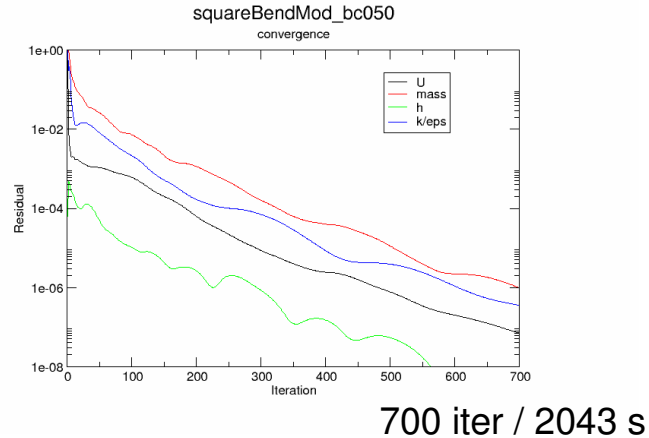
Convergence – STAR-CD



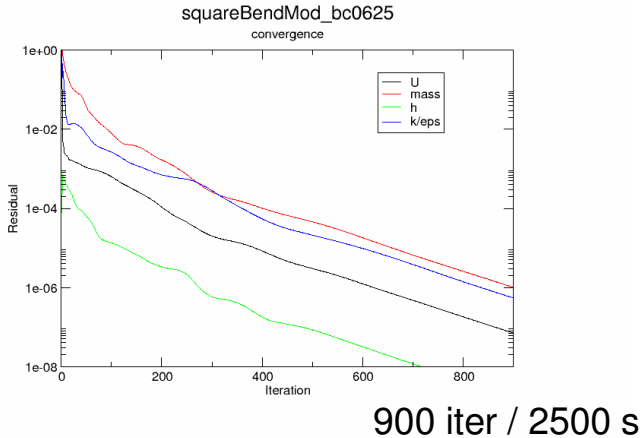
390 iter / 1114 s



490 iter / 1434 s

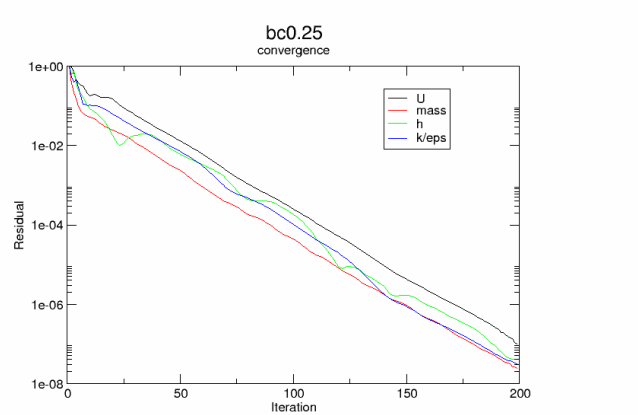


700 iter / 2043 s

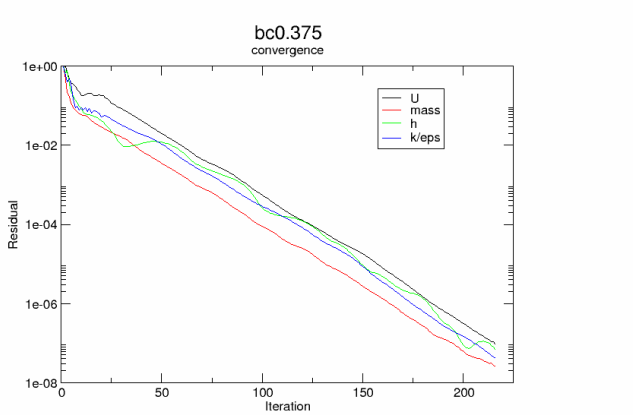


900 iter / 2500 s

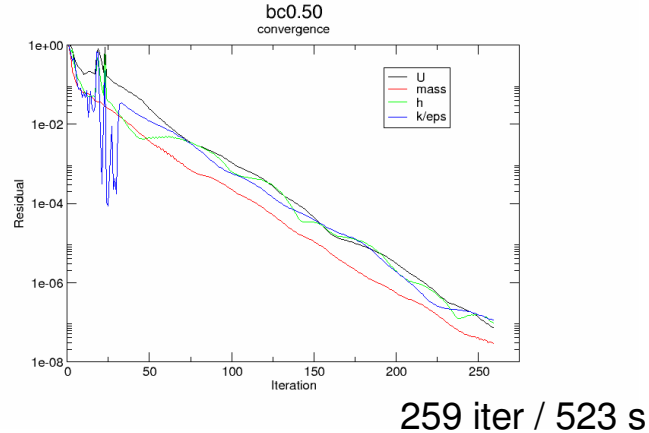
Convergence – OpenFOAM



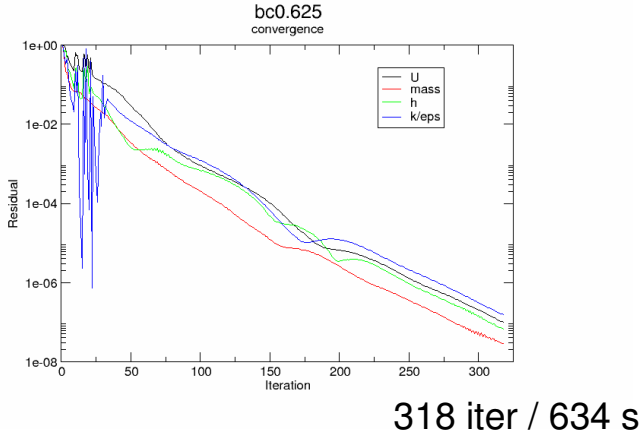
199 iter / 600 s



216 iter / 637 s



259 iter / 523 s



318 iter / 634 s

Similar results, but …

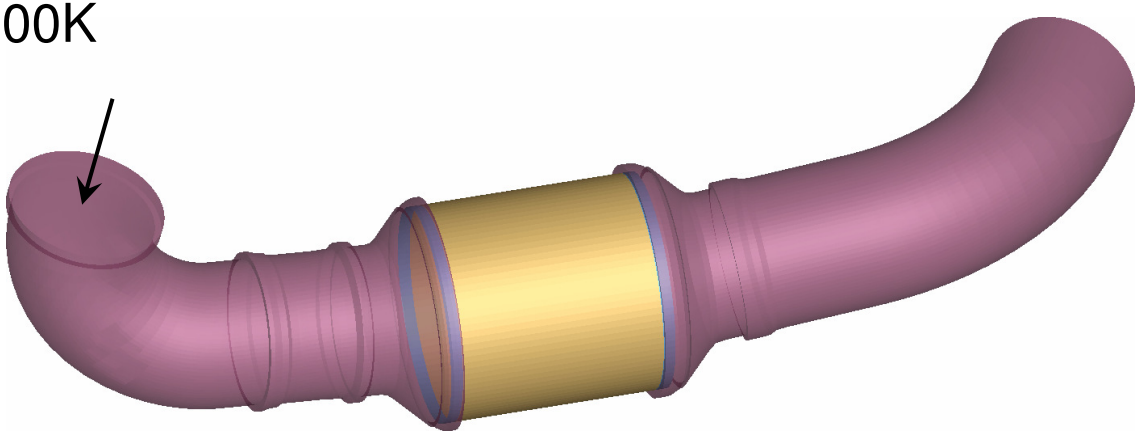
(mid-plane)	STAR-CD		OpenFOAM	
U [m/s]	*max*	*min*	*max*	*min*
0.25	383	130	387	88
0.375	530	189	541	138
0.50	617	231	643	181
0.625	716	250	685	205
Ma				
0.25	0.62	0.20	0.62	0.14
0.375	0.88	0.30	0.91	0.22
0.50	1.06	0.36	1.12	0.30
0.625	1.16	0.37	1.21	0.33

(mid-plane)	STAR-CD		OpenFOAM	
T [K]	*max*	*min*	*max*	*min*
0.25	1017	948	1013	939
0.375	1029	894	1023	876
0.50	1034	851	1028	817
0.625	**1200**	946	1030	785
Ptotal [mbar]	*in*	*dP*	*in*	*dP*
0.25	1323	92	1327	98
0.375	1602	214	1600	219
0.50	1987	397	1964	388
0.625	2248	**323**	2395	592

simpleCCC Test Case

- 165 k cells
- ICEM/Hexa mesh

  - *inlet*
    - 540-900 kg/h, 1200K
    - 10% / 5mm turb.
  - *outlet*
    - 1.45bar
  - 400/4 ceramic
    - Darcy = 3.7e+7 1/m^2
    - Forchheimer = 20 1/m



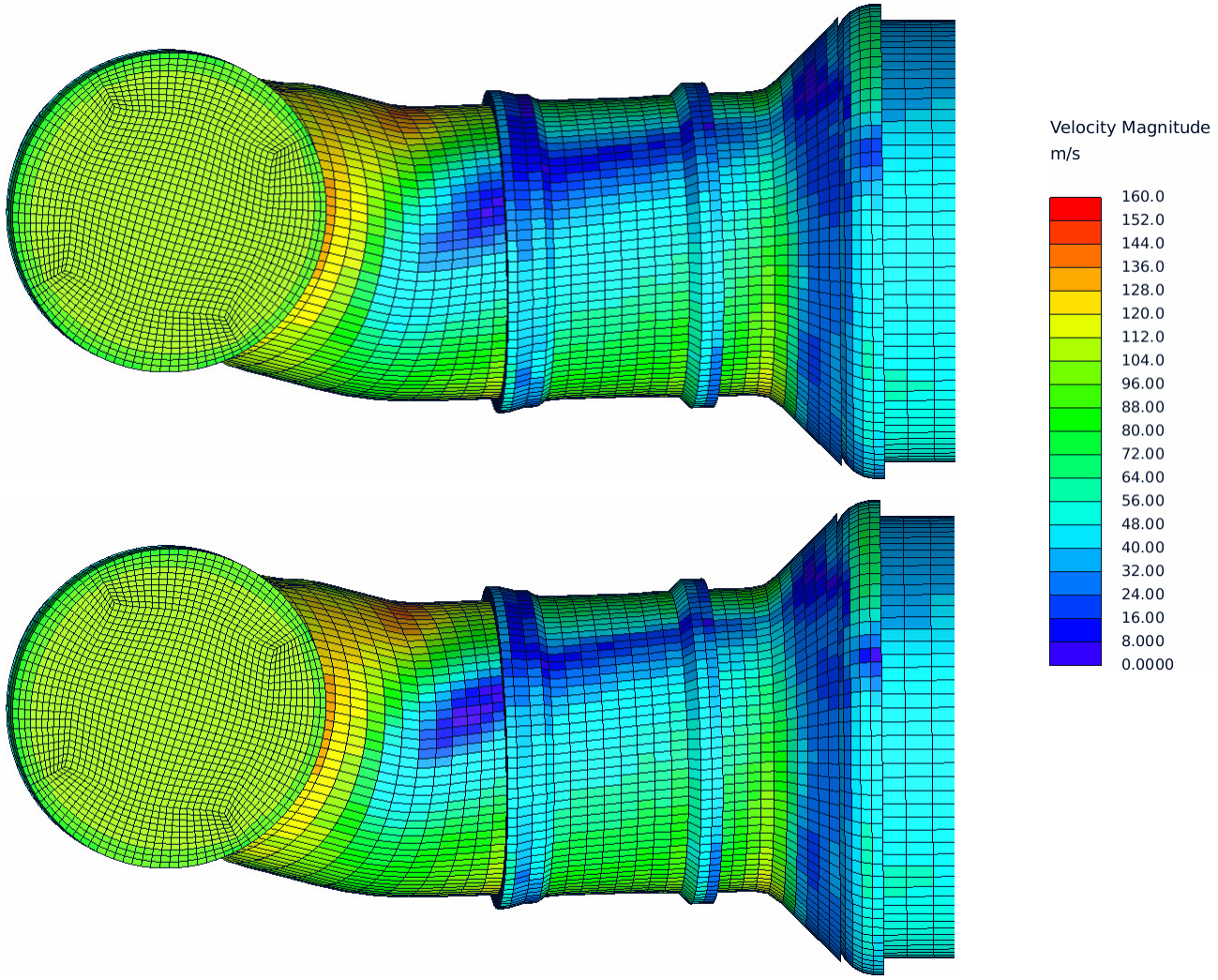
Solver parameters

- STAR-CD
  - SIMPLE, UD
  - k/epsilon – std & RNG
  - AMG, double
  - convergence
    - 1e-4
  - relax
    - U=0.7, p=0.3
    - k/eps=0.7, h=0.95
  - porosity via user Fortran
- OpenFOAM
  - SIMPLE, UD
  - k/epsilon – std & RNG
  - GAMG, double
  - convergence
    - 1e-4
  - relax
    - U=n/a, p=0.3
    - k/eps=0.7, h=0.95
  - rhoImplicitPorousSimpleFoam

STAR-CD – 0.15 kg/s

k/eps

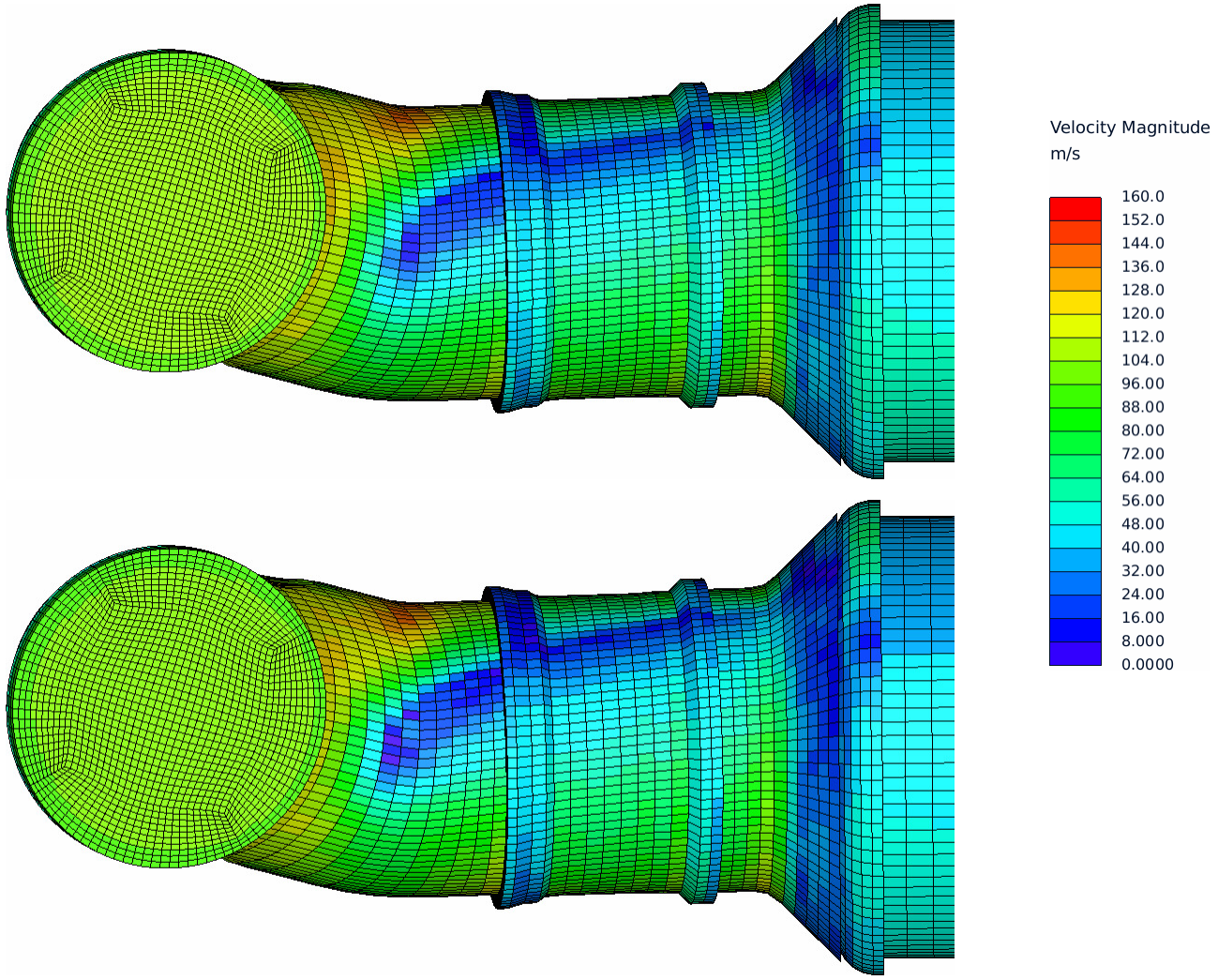
k/eps RNG



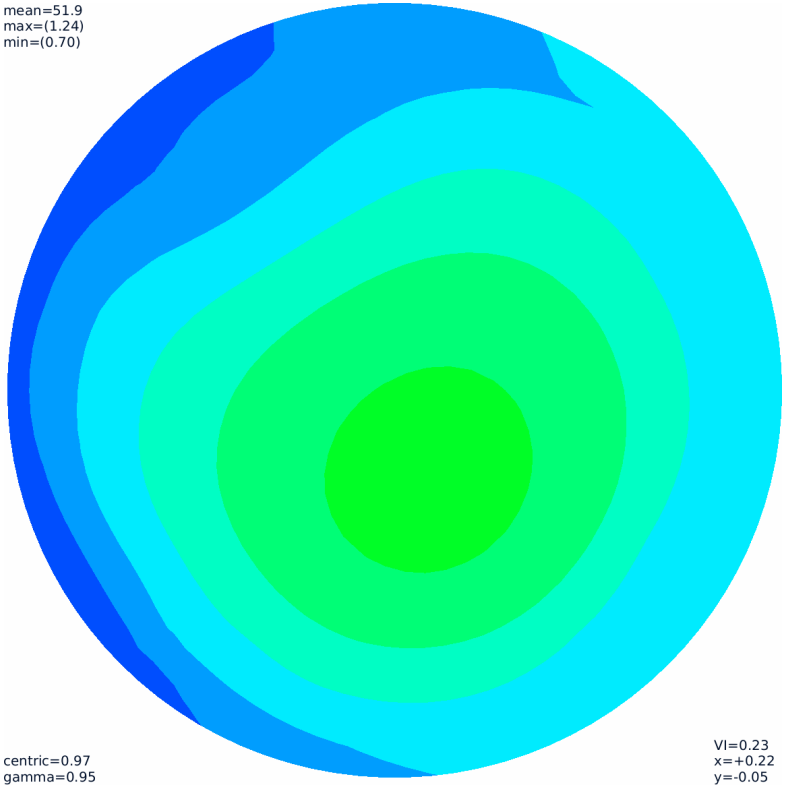
OpenFOAM – 0.15 kg/s

k/eps

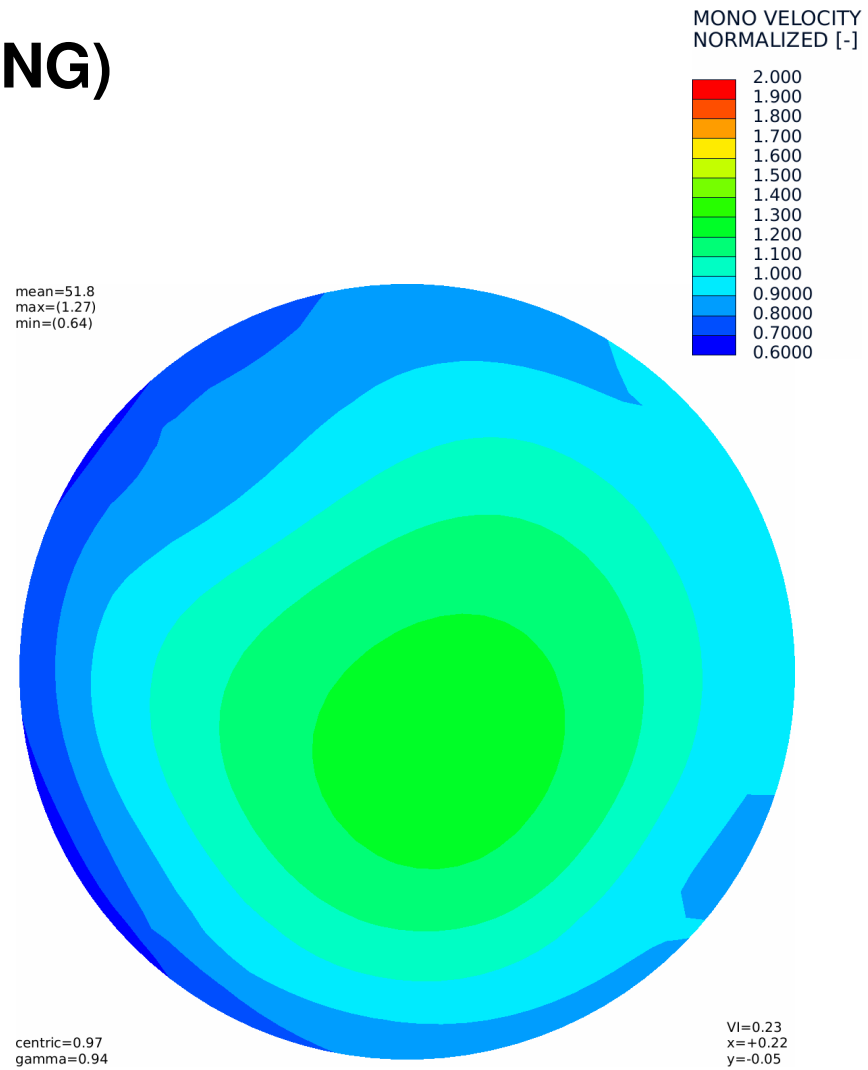
k/eps RNG



Uniformity – 0.15 kg/s (k/eps RNG)



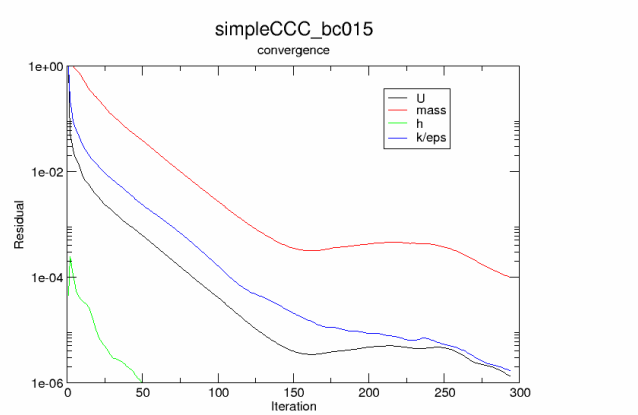
STAR-CD



OpenFOAM

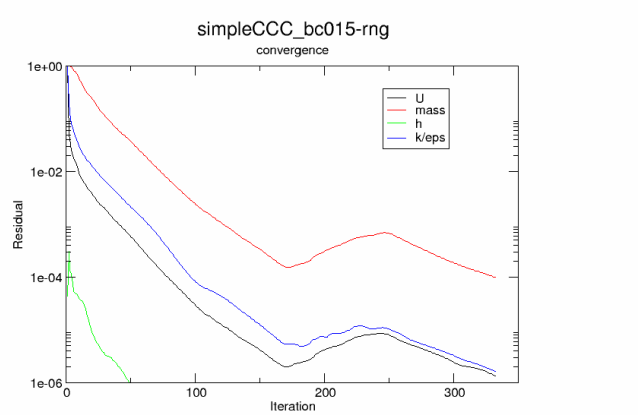
Convergence – STAR-CD (8 cpu)

standard

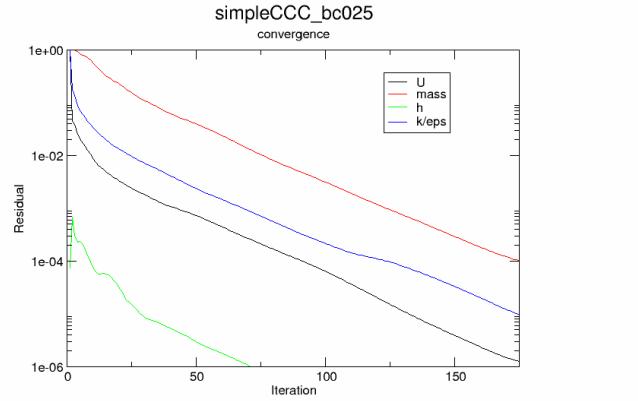


294 iter / 308 s

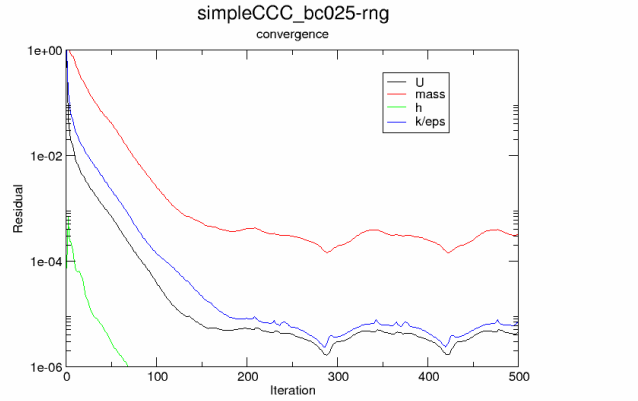
RNG



358 iter / 332 s



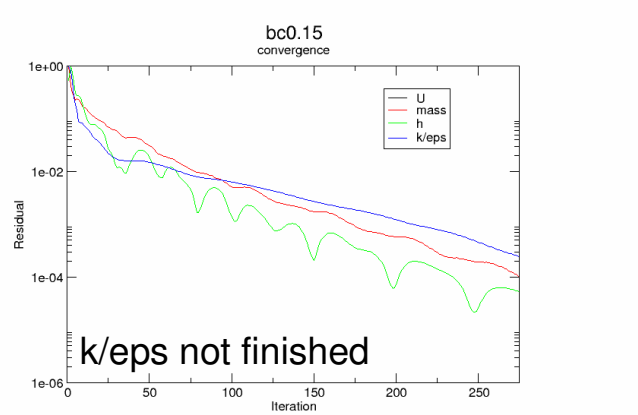
176 iter / 191 s



no convergence

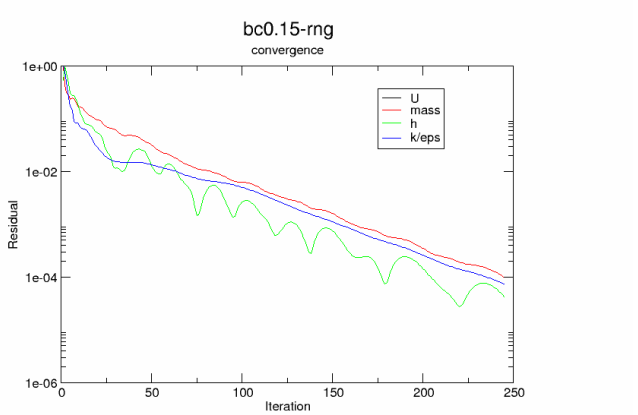
Convergence – OpenFOAM (8 cpu)

standard

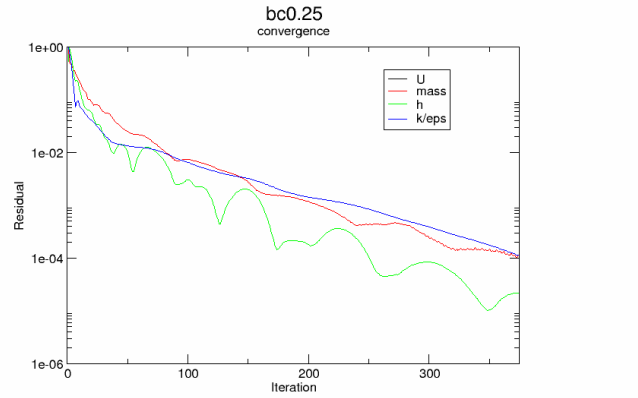


277 iter / 262 s

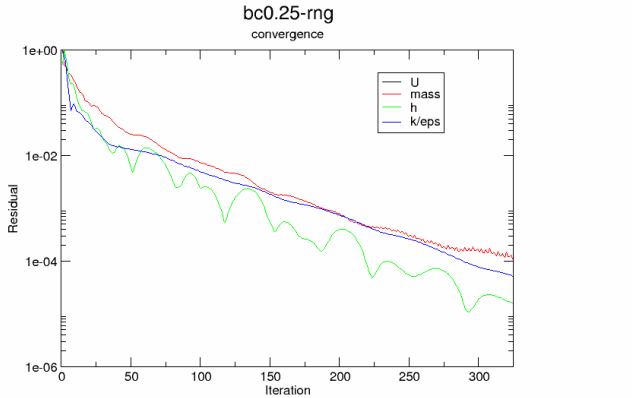
RNG



244 iter / 217 s



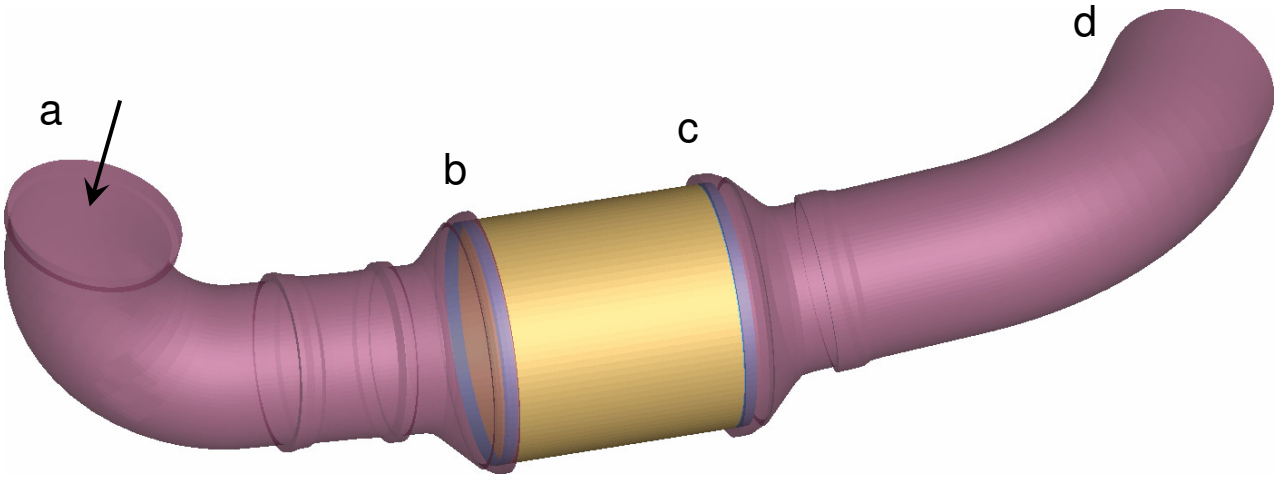
376 iter / 326 s



327 iter / 312 s

Back-Pressure – 0.15 kg/s RNG

- STAR-CD
  - a-b = 32 mbar
  - b-c = 96 mbar
  - c-d = 13 mbar
  - Total = 141 mbar
- OpenFOAM
  - a-b = 35 mbar
  - b-c = 97 mbar
  - c-d = 14 mbar
  - Total = 146 mbar



Summary (1)

- OpenFOAM and STAR-CD Integration
  - mesh I/O
  - results I/O
- Solvers
  - ‘similar’ speed and results
- OpenFOAM Libraries
  - Extensive
  - Open
  - Readable

Summary (2)

- OpenFOAM at your company?
  - interoperability
  - solver capabilities
  - OpenFOAM *and* commercial
- OpenFOAM freedom
  - use if/when desired
  - alone or parallel to existing code
  - ‘on-demand’ computing
  - no lock-in
  - freedom in the future (GNU General Public License)
- May the best code(s) win!