AeroFluidX: A Next Generation GPU-Based CFD Solver for Engineering Applications

Dr. Bjoern Landmann

Dr. Kerstin Wieczorek

Stefan Bachschuster

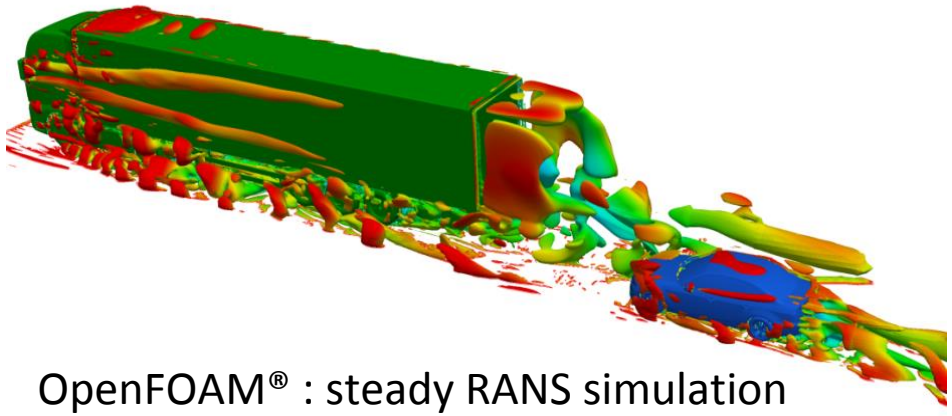
Content

- CPU vs. GPU computing for CFD
- Short summary of **Culises** – hybrid GPU-CPU approach
  - Approach for partially accelerated CFD applications
  - Industrial problem set and achievable speedups
- **aeroFluidX** – fully ported flow solver on GPU
  - Technical approach
  - Problem set and achievable speedups
  - Multi-GPU scaling
- Conclusions and future roadmap for aeroFluidX

Potential of GPU-computing for CFD

FluiDyna®

Example from automotive industry: Car-truck interference



OpenFOAM® : steady RANS simulation
120M computing cells

Simulation time:

Medium CPU cluster:
22 dual-CPU socket blades:
→ 44 CPUs of type Sandy Bridge Xeon E5-2650, 8-core
→ runtime ≈ 2 days

Computing unit	Cost Q1/2015 [€]	Theoretical peak DP performance [Gflops]	Memory	Max. memory bandwidth [GB/s]	Max. power consumption [W]
Intel E5-2650 V3 2.6 GHz, 10 cores	1400	416	Compute node dependent	**68**	105
Nvidia K40	3500 (2.3x)	1430 (3.4x) 1660 (4.0x)	12 GB	**288 (4.2x)**	235 (2.2x)
Nvidia K80	4800	Aggregate of 2 K40 GPUs			300

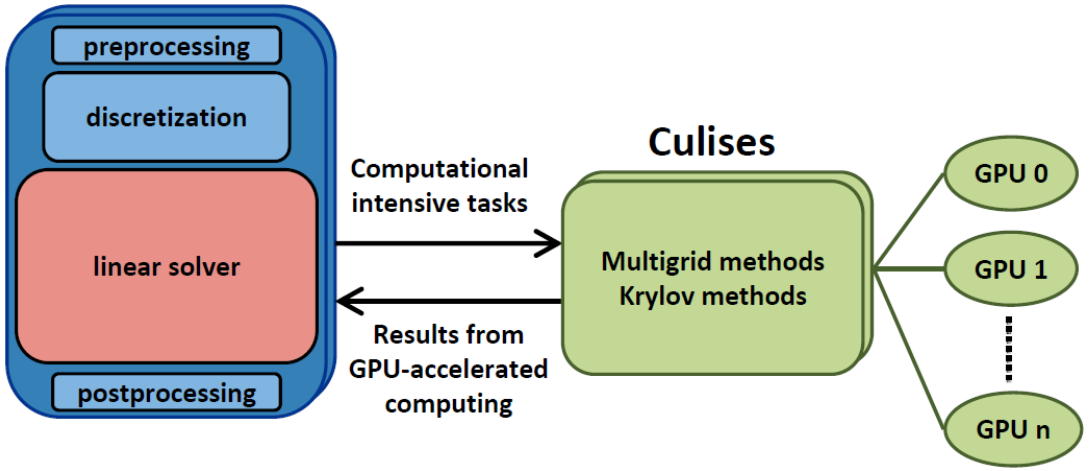
Classical (unstructured) CFD codes are **memory bandwith limited**

Computing platform	Theoretical peak performance
22 blades equipped with 2 x Intel Xeon E5-2650 V3 **62k € for CPUs only** +blade hardware: mainboard, memory, power supply … +air-conditioned server room required	**18.3 TFLOPS** **(4620 Watt)**
Hybrid CPU-GPU Two blades: single socket, 2x Xeon E5-2650 **+ 6 Nvidia Tesla K80** **31k €**	832 Gflops **+19920 Gflops** **20.7 TFLOPS** **(2010 Watt)**

Conclusion: Hardware and energy costs more than halved!

Library Culises – concept and features

Simulation tool
e.g. OpenFOAM®



Culises = **Cu**da **Li**brary for **S**olving Linear **E**quation **S**ystems

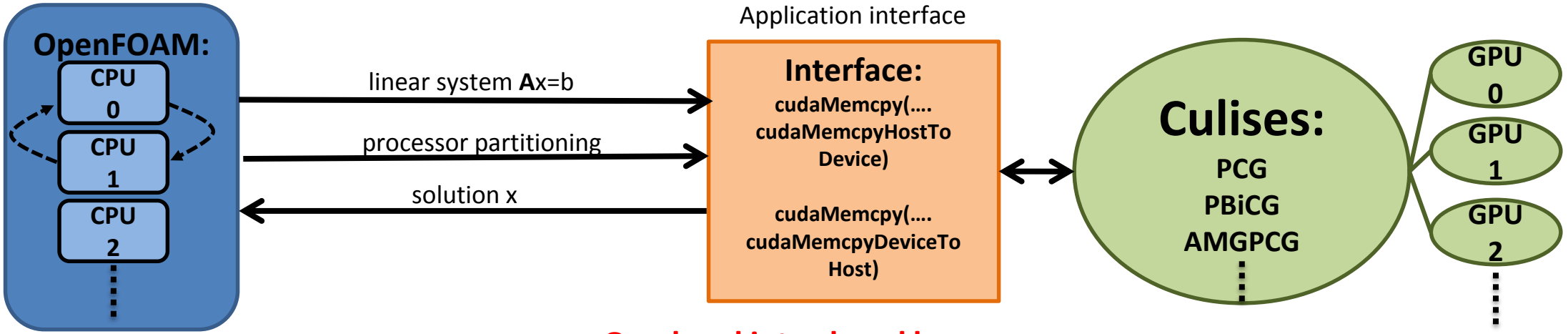
See also www.culises.com

- State-of-the-art solvers for solution of linear systems (AmgX library of Nvidia)
  - Multi-GPU and multi-node capable
  - Single precision or double precision available
- Krylov subspace methods
  - CG, BiCGStab, GMRES for symmetric /non-symmetric matrices
  - Preconditioning options
    - Jacobi (Diagonal)
    - Incomplete Cholesky (IC)
    - Incomplete LU (ILU)
    - Algebraic Multigrid (AMG), see below
- Stand-alone multigrid method
  - Algebraic aggregation and classical coarsening
  - Multitude of smoothers (Jacobi, Jacobi L1, multi-color Gauss-Seidel, ILU etc.)
- Flexible interfaces for arbitrary applications e.g.: established coupling with OpenFOAM®

Hybrid CPU-GPU scenario (MPI+Cuda)

OpenFOAM®
MPI-parallelized CPU implementation
based on domain decomposition

Culises dynamic library: Solves linear system(s) on multiple GPUs



MPI-parallel assembly (discretization etc.) of system matrices remains on CPUs

Overhead introduced by memory transfer and matrix format conversion

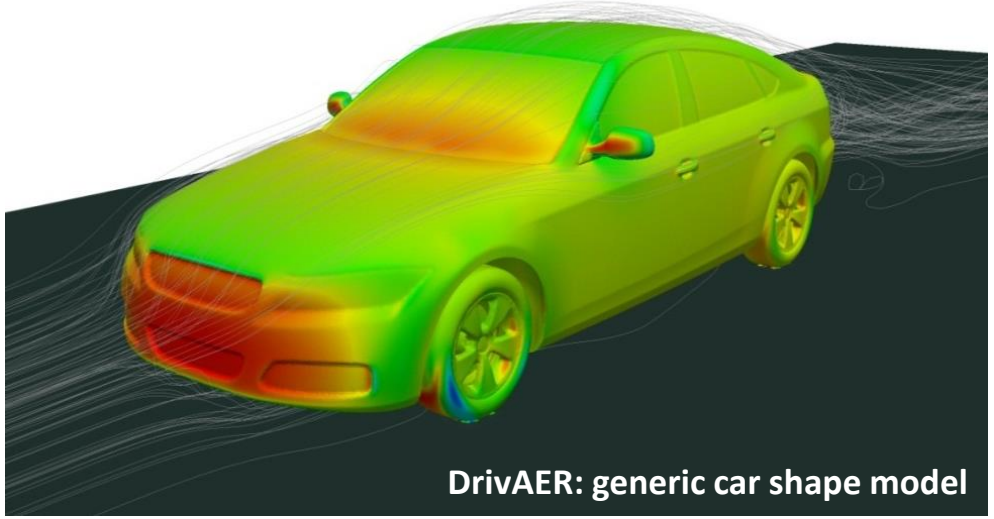
Benchmark case

FluiDyna®

Automotive setup:

- simpleFoam solver from OpenFoam®
  - Steady-state (SIMPLE[1]) method
  - k-ω SST model with wall functions
- Linear solver setup
    - Only linear system for pressure correction accelerated by Culises on GPU
    - All other linear systems solved on CPU
      - Momentum x, y, z
      - Kinetic energy k
      - Specific rate of dissipation omega

    CPU<->GPU overhead outbalances GPU-acceleration


DrivAER: generic car shape model

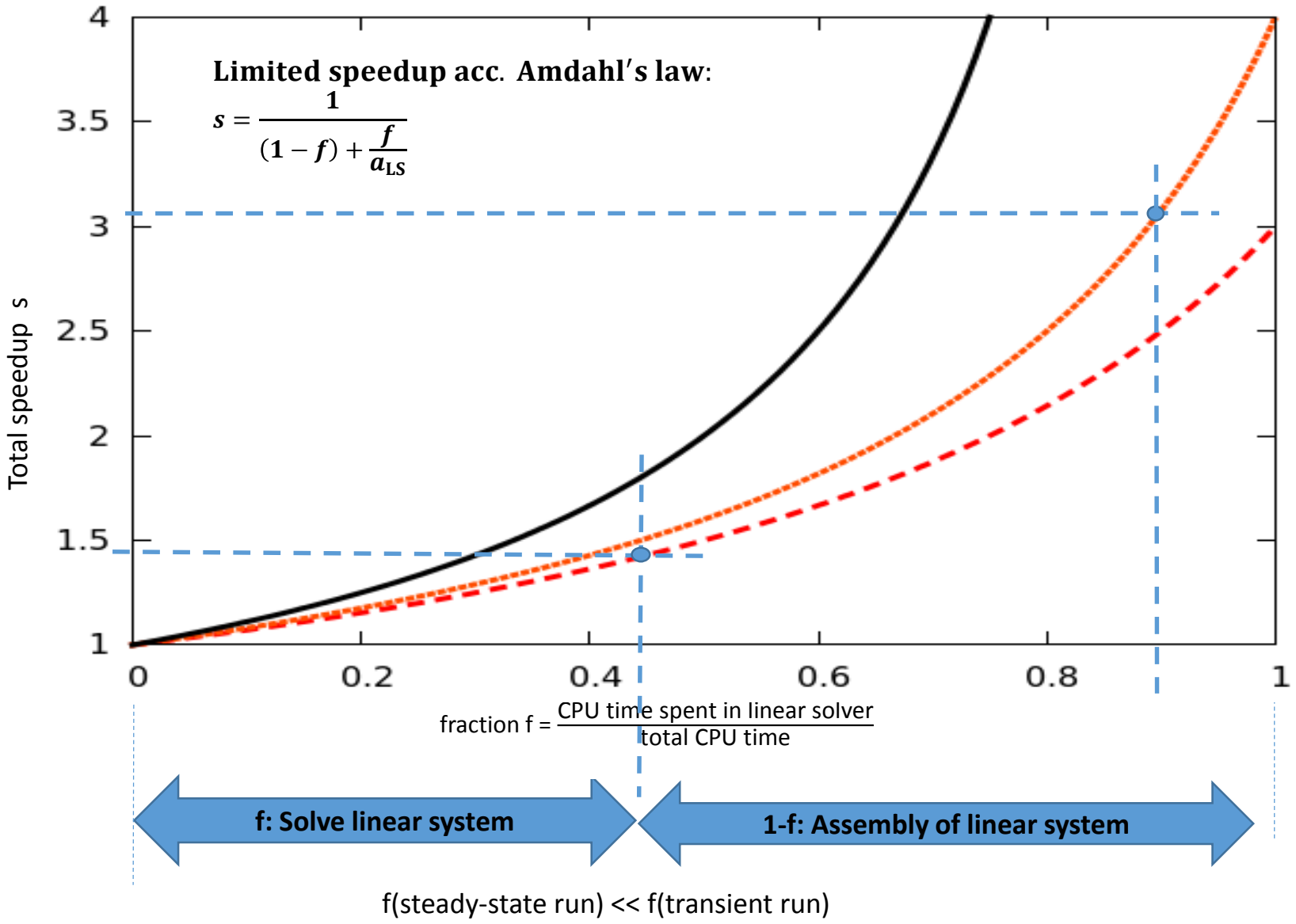
1: SIMPLE=**S**emi-**I**mplicit **M**ethod for **P**ressure-**L**inked **E**quations

Benchmark results

- **Automotive industrial setup (Japanese OEM)**
- Same solver applied as with DrivAER case
- CPU linear solver for pressure: geometric-algebraic multigrid (GAMG) of OpenFoam (V2.3.1)
  GPU linear solver for pressure: AMG preconditioned CG (AMGPCG) of Culises (V1.1)
- 200 SIMPLE iterations

Grid Cells	CPU cores Intel E5-2650	GPUs Nvidia K40	Linear solve time [s]	Total simulation time [s]	Speedup linear solver	Speedup total simulation
18M	8	1	1779	8407	3.83	1.60
18M	16	2	943	4409	3.17	1.44
62M	16	2	3662	16124	3.00	1.44
62M	32	4	1811	7519	2.60	1.36

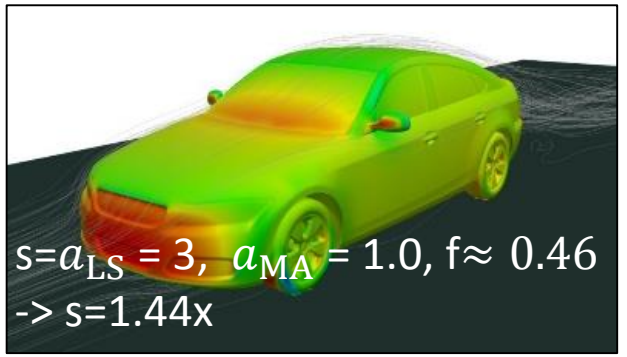
Potential speedup for hybrid approach

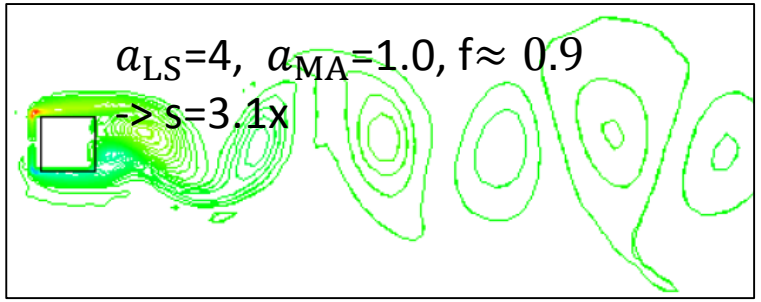


acceleration of linear solver on GPU:

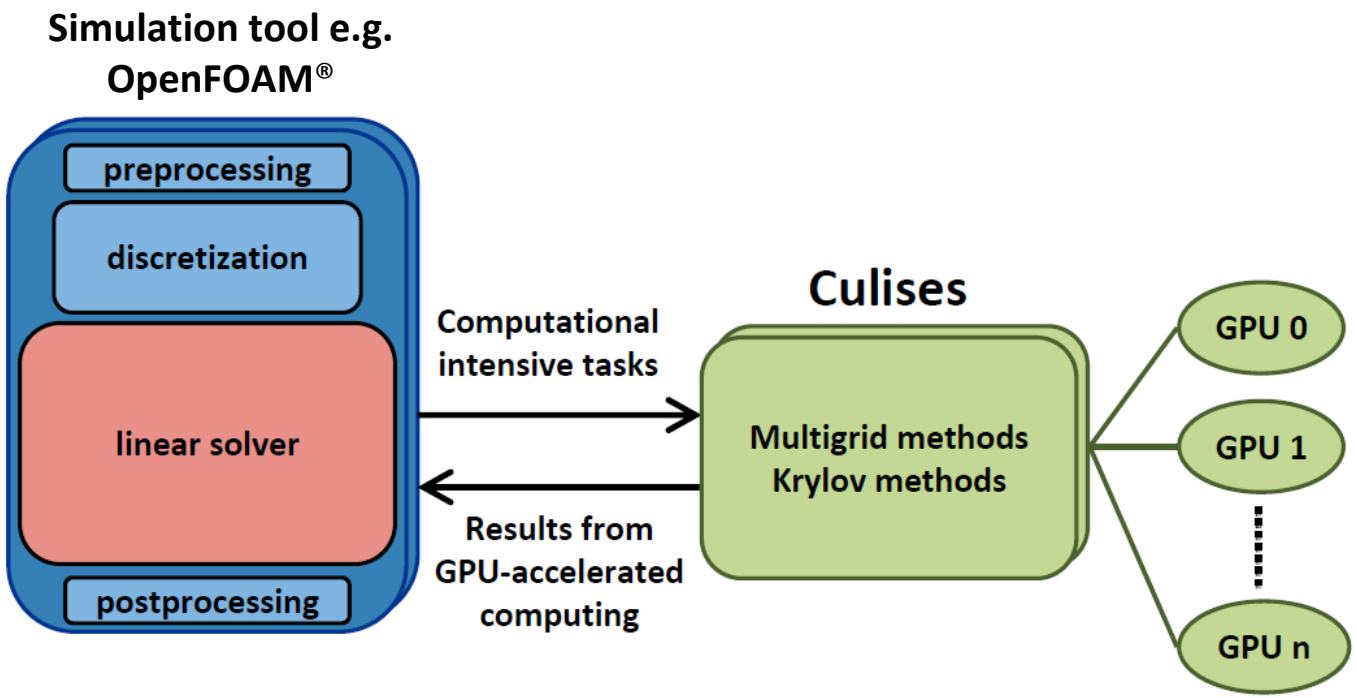
- a_{LS} = 3.0, a_{MA} = 1.0
- a_{LS} = 4.0, a_{MA} = 1.0
- $a_{LS} \rightarrow \infty$, a_{MA} = 1.0

a_{LS} :Speedup linear solver
a_{MA}: Speedup matrix assembly (=discretization, etc.)





Summary hybrid approach



Advantages:

+ Universally applicable (coupled to simulation tool of choice)

+ Full availability of existing flow models

+ No validation needed

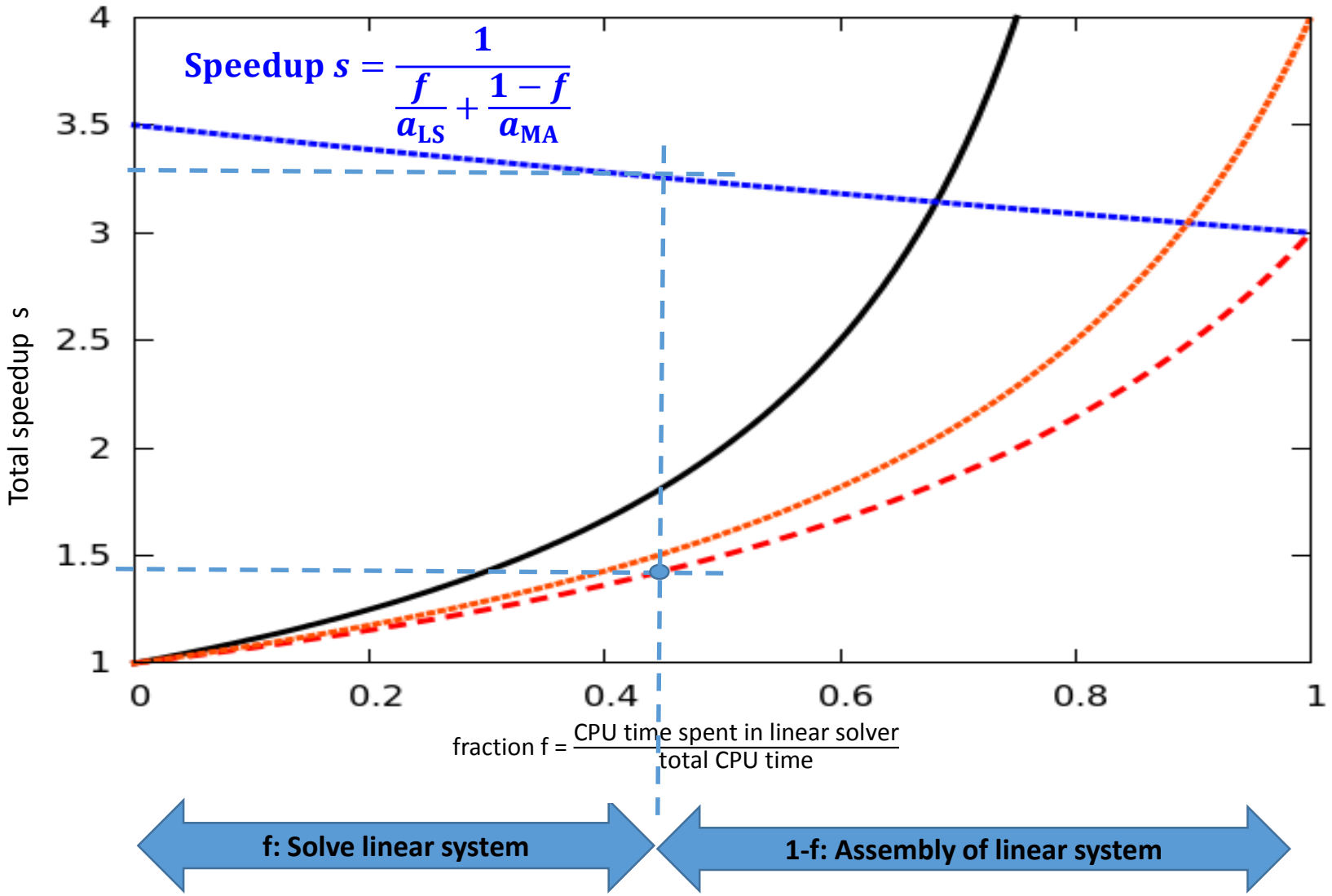
Disadvantages:

- Hybrid CPU-GPU produces overhead

- In case that solution of linear system not dominant (f<0.5)
  → **Application speedup can be limited**

Potential speedup for full GPU approach

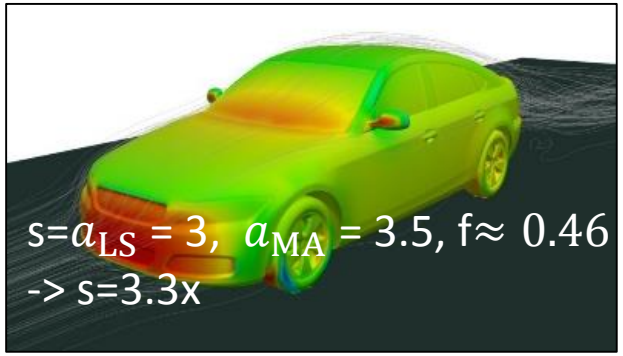
FluiDyna®



acceleration of linear solver on GPU:

- a_{LS} = 3.0, a_{MA} = 1.0
- a_{LS} = 4.0, a_{MA} = 1.0
- $a_{LS} \rightarrow \infty$, a_{MA} = 1.0
- a_{LS} = 3.0, a_{MA} = 3.5

a_{LS} :Speedup linear solver
a_{MA}: Speedup matrix assembly (=discretization, etc.)



aeroFluidX – a full GPU approach

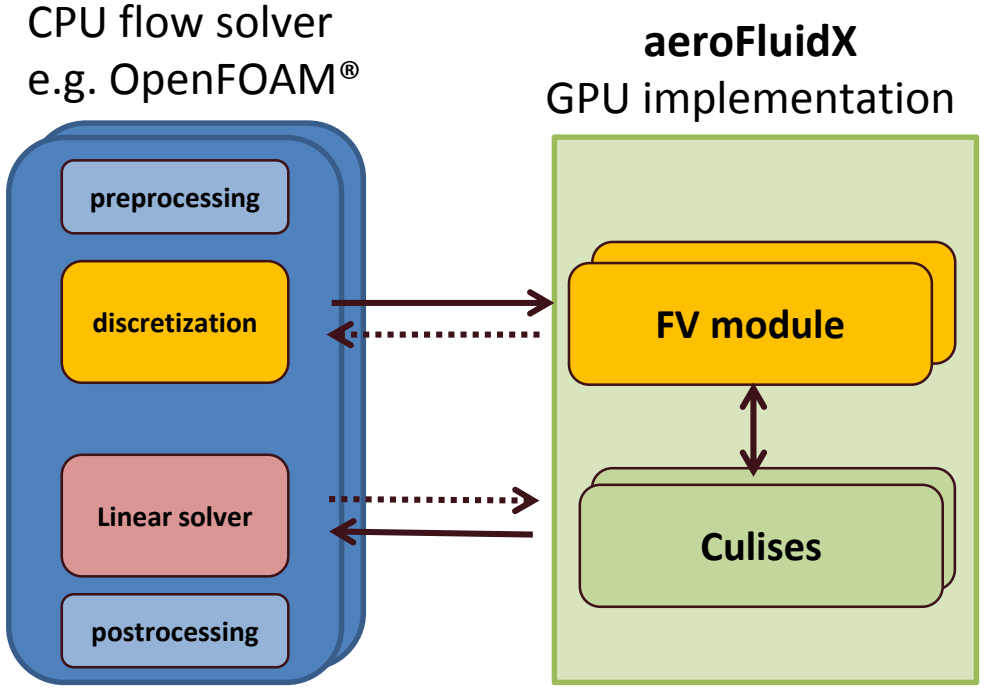
- Targeted physical flow model:
  - Incompressible Navier-Stokes equations
  - Single-phase flow
- Numerical discretization method
  - Finite Volume (FV) method
    - Using **unstructured** mesh
    - Classical choice for
      - Flux (upwind)
      - Gradient evaluation,
      - Interpolation method, etc.
  - Pressure-velocity coupling using classical segregated approach to save GPU memory:
    - SIMPLE method for steady-state flow
    - PISO method for transient flow

+ RANS with wall model
+ DES

Profiling shows pre- & post-processing are negligible
→ mainly 2 parts dominate solution process:

(1)Assembly of linear systems
(momentum and pressure correction)
(2)Solution of linear systems (Culises)

aeroFluidX – a full GPU approach



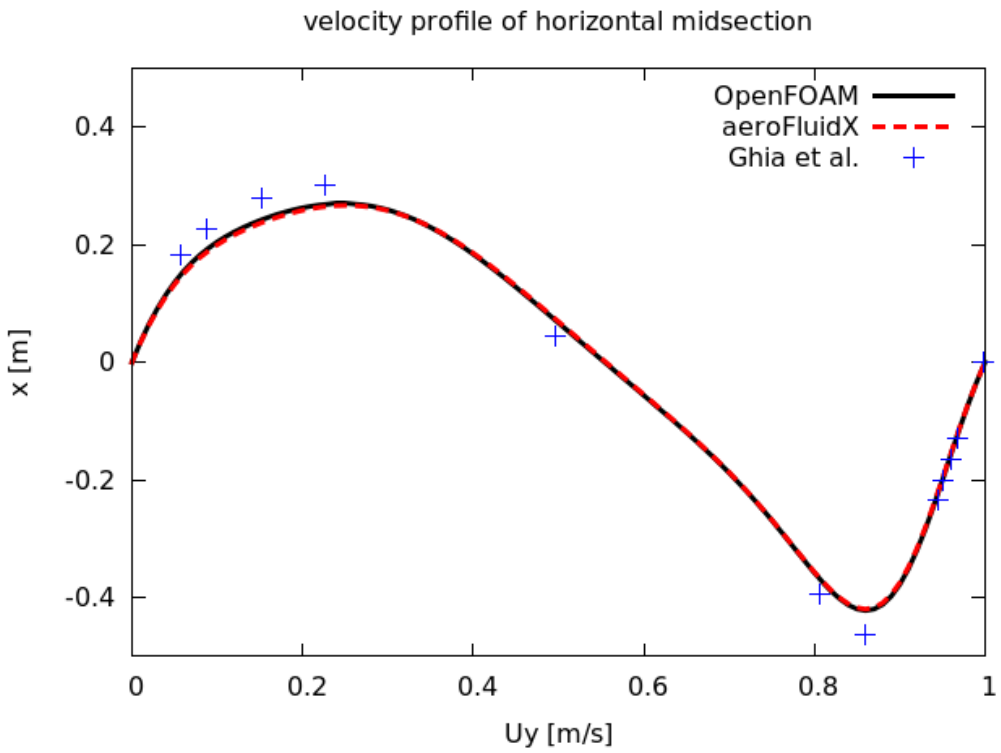
- Porting discretization of equations to GPU
  - ➔ discretization module (Finite Volume) running on GPU
  - ➔ Possibility of direct coupling to Culises/AMGX
    - ➔ Zero overhead from CPU-GPU-CPU memory transfer and matrix format conversion
    - ➔ Solution of momentum equations and turbulence also beneficial
- OpenFOAM® environment supported
  - ➔ Enables plug-in solution for OpenFOAM® customers
  - ➔ But communication with other input/output file formats possible

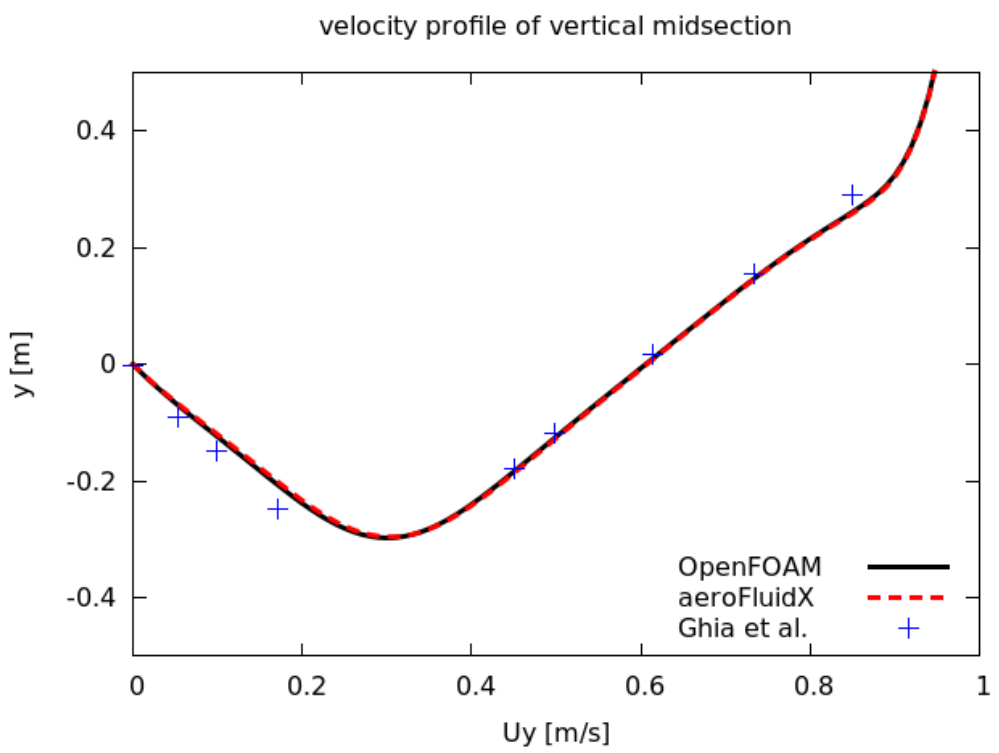
aeroFluidX – reasonable benchmarking

- CFD: simpleFoam solver (OpenFOAM® V2.3.1)
- GPU: aeroFluidX
- Fair comparison between OpenFOAM® and aeroFluidX
  - Linear solver:
    - **Convergence criterion:**
      satisfy same tolerance for norm of residual
    - Solver choice:
      select best available solver on CPU
      vs.
      best available linear solver on GPU
  - Discretization approach:
    use same methods for flux, gradient, interpolation, etc.

aeroFluidX – cavity flow test case

Validation: Re=400 (laminar), grid 250x250

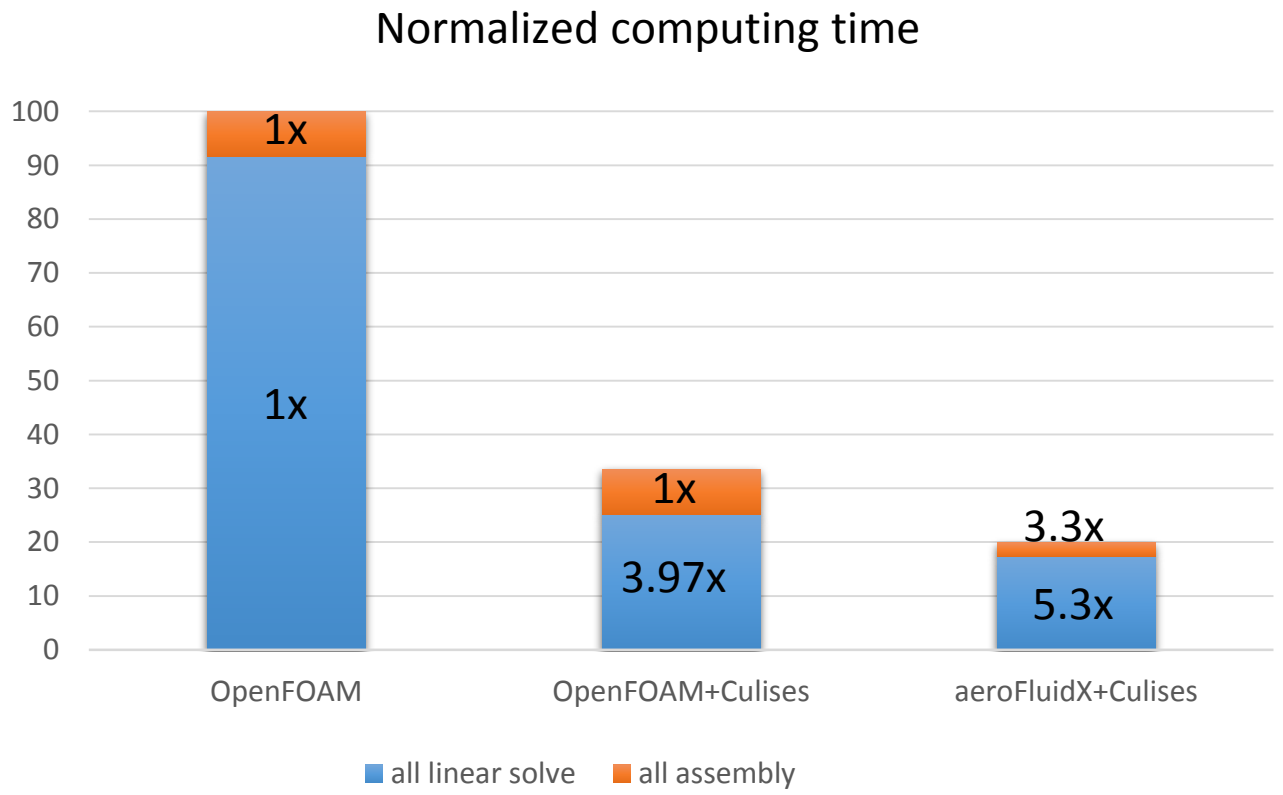




Ghia et al: High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method (1982)

aeroFluidX – cavity flow test case

- CPU: Intel E5-2650 (all 8 cores)
  GPU: Nvidia K40
- 4M grid cells (unstructured)
- Running 100 SIMPLE steps with:
  - OpenFOAM® (OF)
    - pressure: GAMG
    - Velocitiy: Gauss-Seidel
  - OpenFOAM® (OFC)
    - Pressure: Culises AMGPCG (2.4x)
    - Velocity: Gauss-Seidel
  - aeroFluidX (AFXC)
    - Pressure: Culises AMGPCG
    - Velocity: Culises Jacobi
- Total speedup:
  - OF (1x)
  - OFC 2.98x
  - AFXC 5.0x

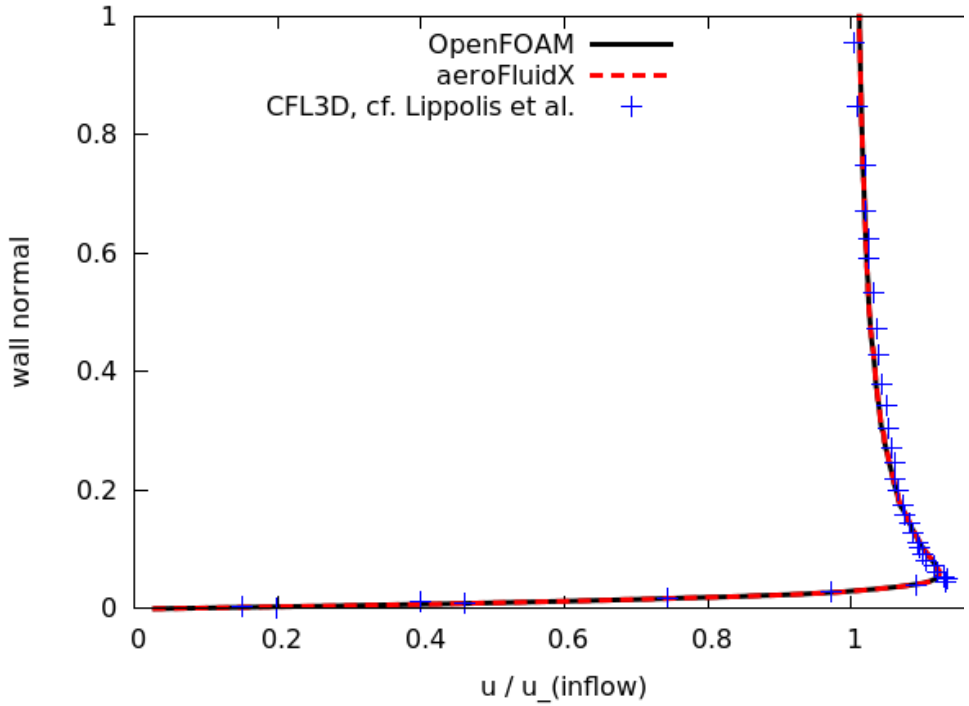


all assembly = assembly of all linear systems (pressure and velocity)
all linear solve = solution of all linear systems (pressure and velocity)

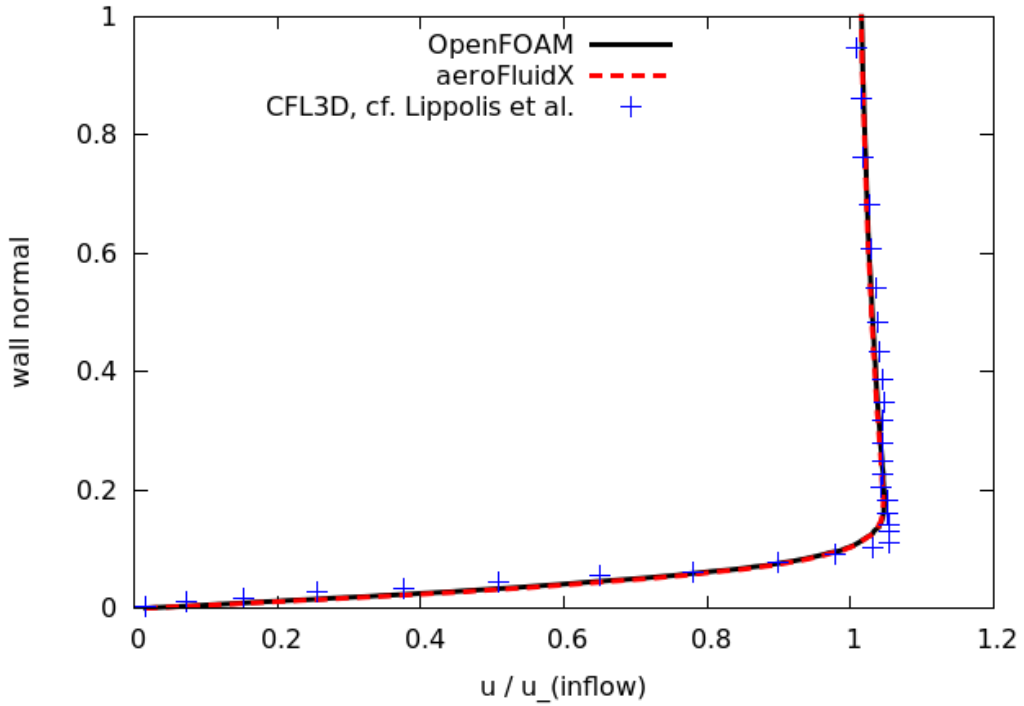
aeroFluidX – airfoil flow test case

Validation: Re=2000 (laminar); angle of attack 0°; 40K grid cells

Velocity profile at 20% chord length



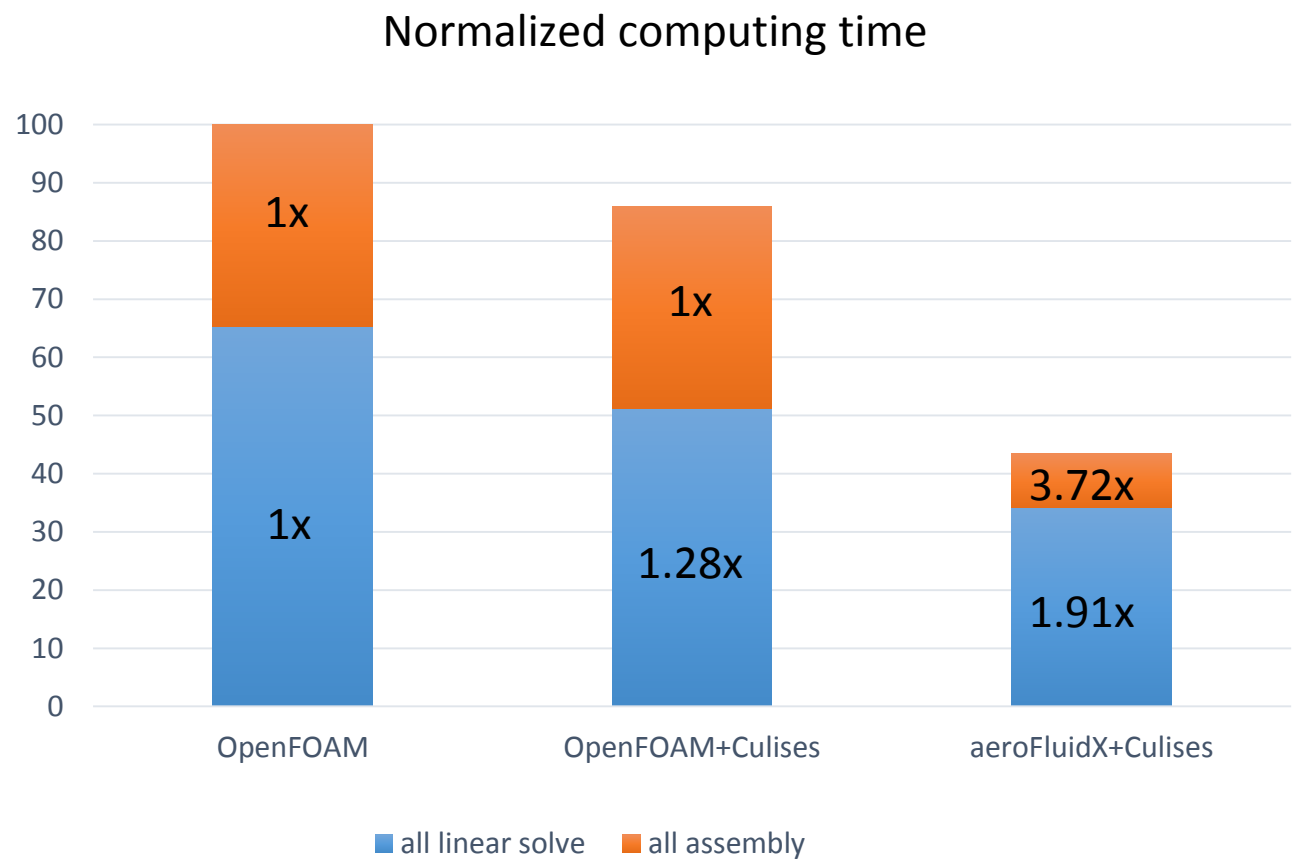
Velocity profile at 80% chord length



Lippolis et al: Incompressible Navier-Stokes Solutions on Unstructured Grids Using a Co-Volume Technique (1993)

aeroFluidX – airfoil flow test case

- CPU: Intel E5-2650 (all 8 cores)
  GPU: Nvidia K40
- 8M grid cells (unstructured)
- Running 100 SIMPLE steps with:
  - OpenFOAM® (OF)
    - pressure: GAMG
    - Velocitiy: Gauss-Seidel
  - OpenFOAM® (OFC)
    - Pressure: Culises AMGPCG (1.8x)
    - Velocity: Gauss-Seidel
  - aeroFluidX (AFXC)
    - Pressure: Culises AMGPCG
    - Velocity: Culises Jacobi
- Total speedup:
  - OF (1x)
  - OFC 1.16x
  - AFXC 2.3x

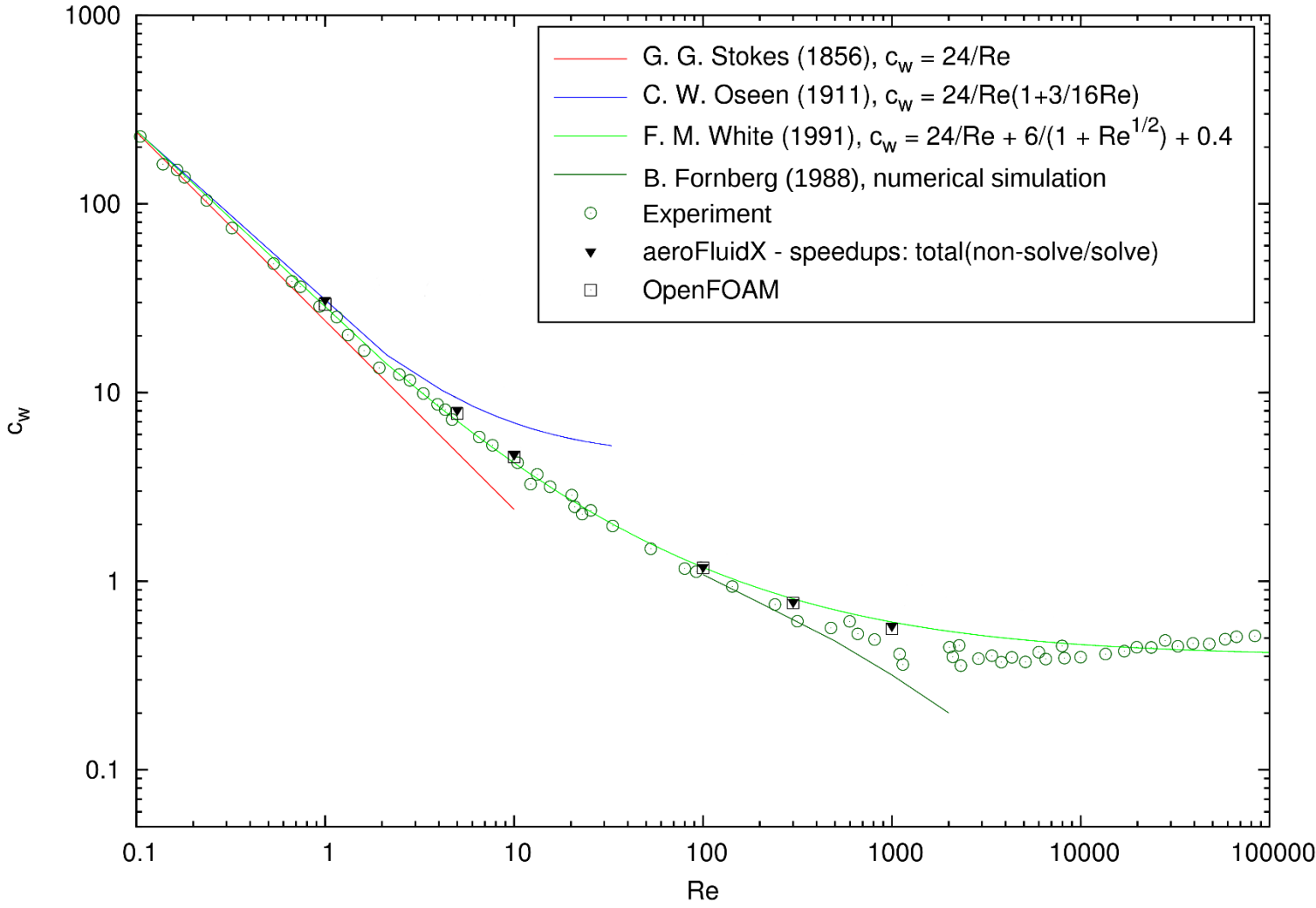


all assembly = assembly of all linear systems (pressure and velocity)
all linear solve = solution of all linear systems (pressure and velocity)

aeroFluidX – 3D sphere test case (laminar)

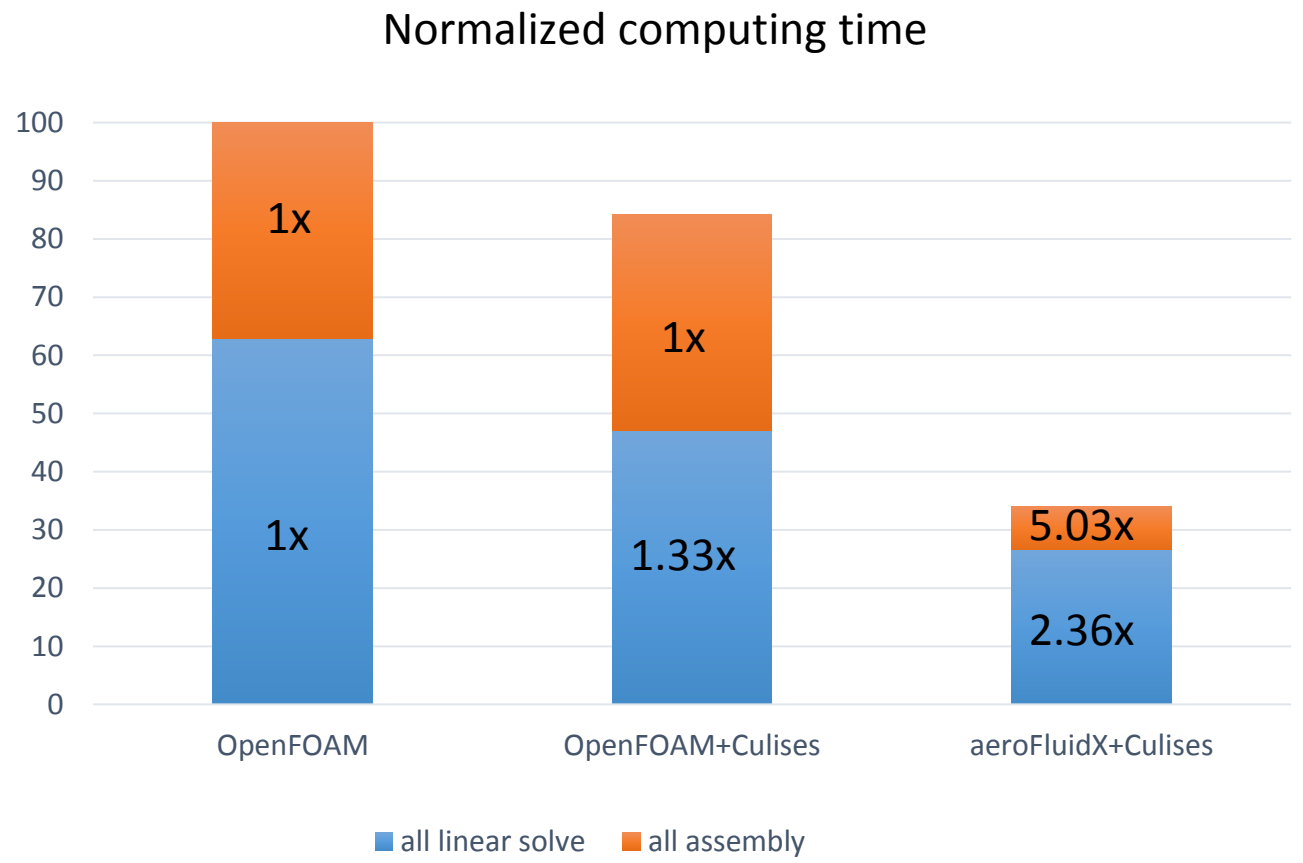
FluiDyna®

- CPU: Intel E5-2650 2 GHz; 8-core Sandy Bridge
  GPU: K40
- Comparison: OpenFOAM vs. aeroFluidX
- 2.4M grid cells (unstructured)
- Running 1000 SIMPLE steps with:
  - OpenFOAM® (OF)
    - Pressure: GAMG
    - Velocitiy: Gauss-Seidel
  - aeroFluidX (AFXC)
    - Pressure: Culises AMGPCG
    - Velocity: Culises Jacobi
- Agreement of OpenFOAM and aeroFluidX results; both fit to experimental results



aeroFluidX – sphere flow test case

- CPU: Intel E5-2650 (all 8 cores)
  GPU: Nvidia K40
- 8M grid cells (unstructured)
- Running 1000 SIMPLE steps with:
  - OpenFOAM® (OF)
    - pressure: GAMG
    - Velocitiy: Gauss-Seidel
  - OpenFOAM® (OFC)
    - Pressure: Culises AMGPCG (1.8x)
    - Velocity: Gauss-Seidel
  - aeroFluidX (AFXC)
    - Pressure: Culises AMGPCG
    - Velocity: Culises Jacobi
- Total speedup:
  - OF (1x)
  - OFC 1.18x
  - AFXC 2.9x



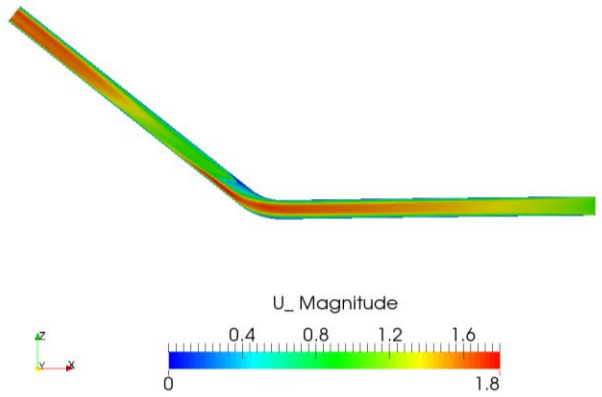
all assembly = assembly of all linear systems (pressure and velocity)
all linear solve = solution of all linear systems (pressure and velocity)

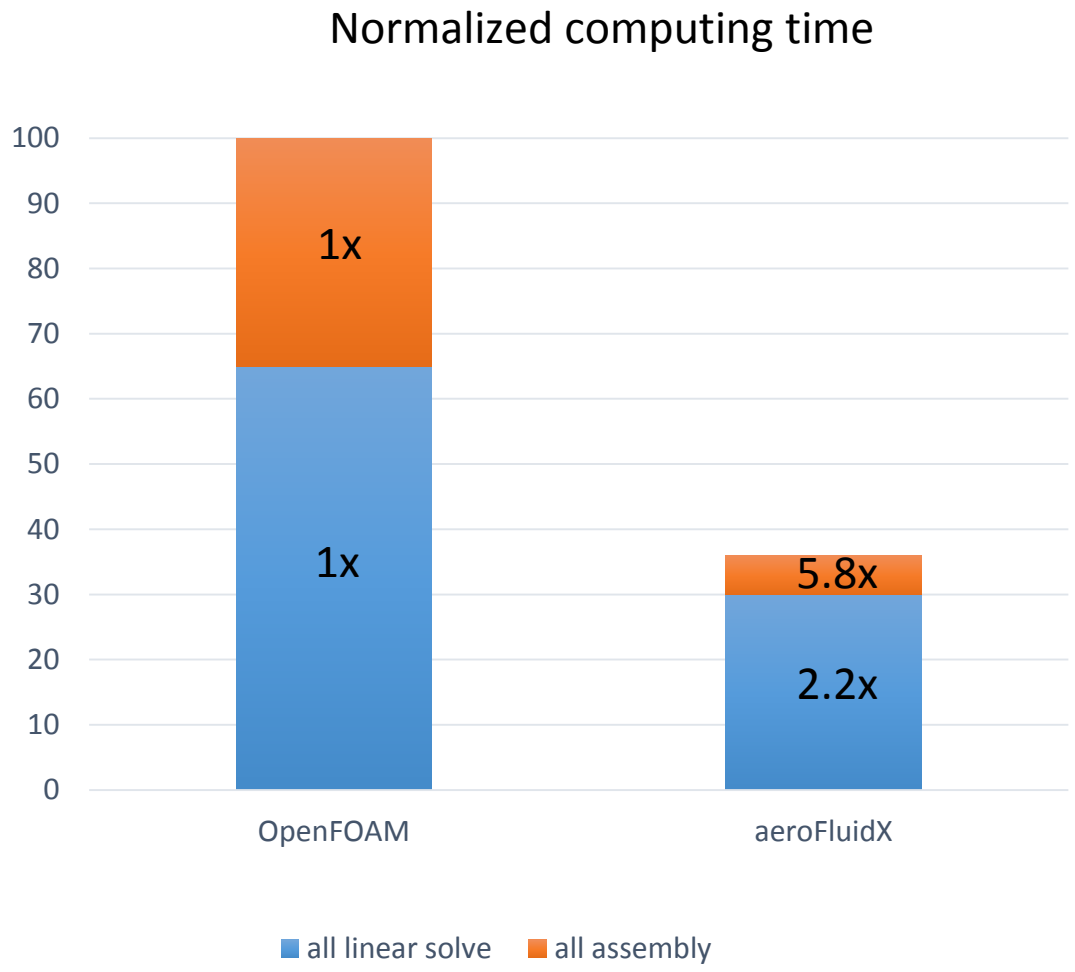
aeroFluidX – 3D elbow pipe flow (laminar)

- Comparison:
  OpenFOAM® vs. aeroFluidX
- 8M grid cells (unstructured)
- 6000 SIMPLE steps
- Speedup with optimized assembly code
  total simulation: 2.8x

Part of JAMA benchmarking test suite

JAMA=Japan Automobile Manufacturers Association

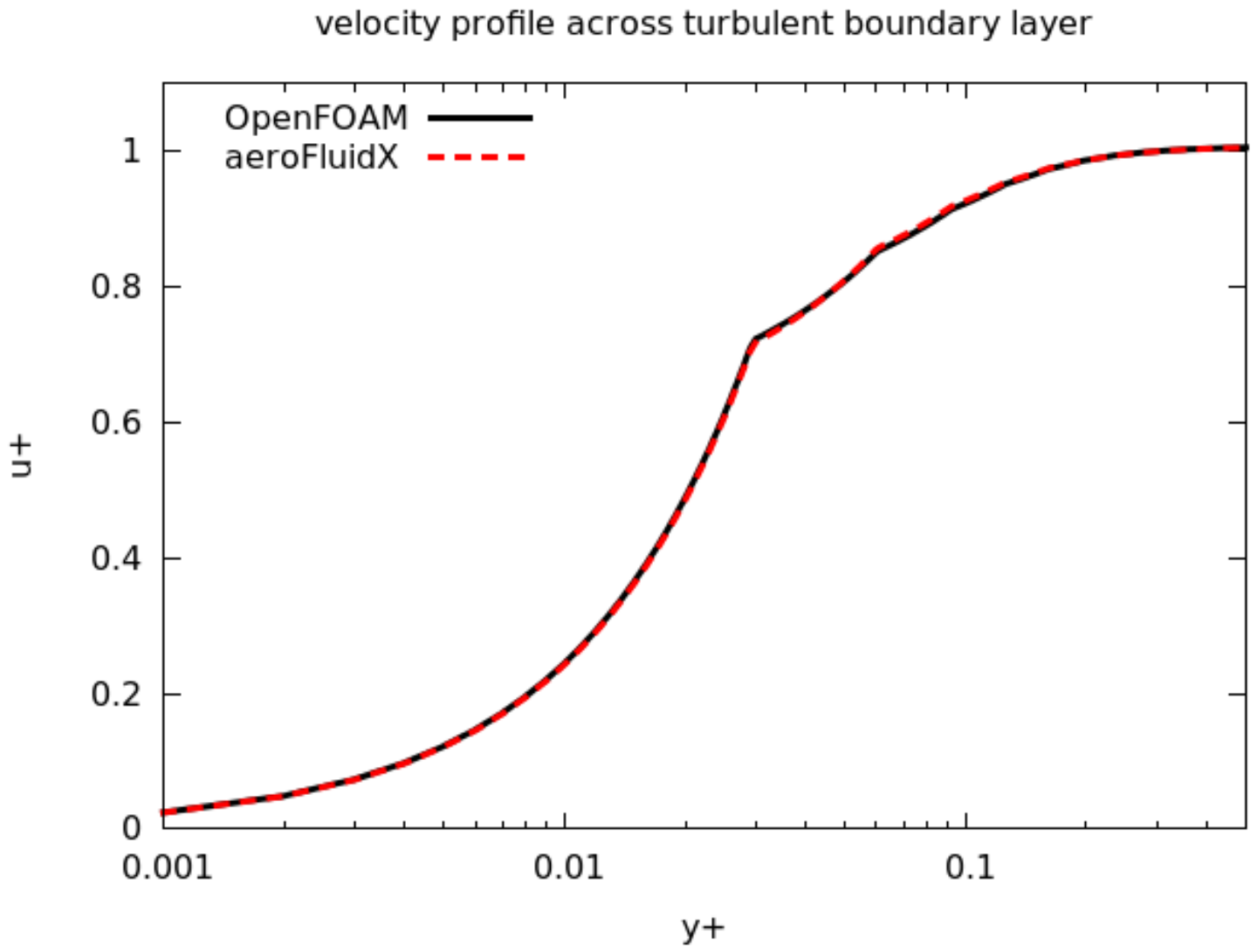




aeroFluidX – turbulent flow validation

- Comparison: OpenFOAM vs. aeroFluidX
- 10K grid cells (structured)
- Re=180 000
- k-omega SST model
  with standard wall functions

→ Good agreement of OpenFOAM® and aeroFluidX



aeroFluidX – multi-GPU weak scaling

FluiDyna®

- Cavity flow, laminar
- Structured mesh 8M, 16M, 24M, 32M grid cells; **8M is maximum for Tesla K40**
- 100 SIMPLE iterations
- GPU exchange via host memory (no use of CUDA-aware MPI **yet**)

GPUs Nvidia K40	Parallel efficiency system assembly	Parallel efficiency (AMGX)			Parallel efficiency total
		Velocity systems	Pressure system	Pressure system Iterations (# levels)	
1	100	100	100	42 (6)	**100**
2	91	67	80	44 (6)	**79**
3	81	65	64	47 (6)	**65**
4	79	62	62	55 (7)	**63**

aeroFluidX – multi-GPU strong scaling

FluiDyna®

- Airfoil flow, laminar
- Unstructured mesh 8M cells; **8M is maximum for Tesla K40**
- 100 SIMPLE iterations
- **Strong** scaling up to 4 GPUs;
  GPU exchange via host memory
- Scaling improved by merging coarse level matrices of each GPU to one single GPU -> called coarse-level consolidation

```
AMG grid for pressure system
    Number of Levels: 6
     LVL      ROWS          NNZ    SPRSTY      Mem (GB)
    ---------------------------------------------------------
      0(D)    8023002      32086418  4.98e-07      0.557
      1(D)    1251522       7966120  5.09e-06      0.227
      2(D)    178726        1238808  3.88e-05      0.0347
      3(D)    25179         175283  0.000276      0.00491
      4(D)    3529          24377   0.00196       0.000684
      5(D)    494           3346    0.0137        8.95e-05
```

GPUs Nvidia K40	Speedup system assembly	Speedup linear system solver (AMGX)		Speedup total simulation vs. 1 GPU
		Velocity systems	Pressure system	
1	1	1	1	**1**
2	1.95	1.64	1.72	**1.76**
3	2.94	2.48	2.36	**2.49**
4	3.60	3.20	2.8	**2.99**

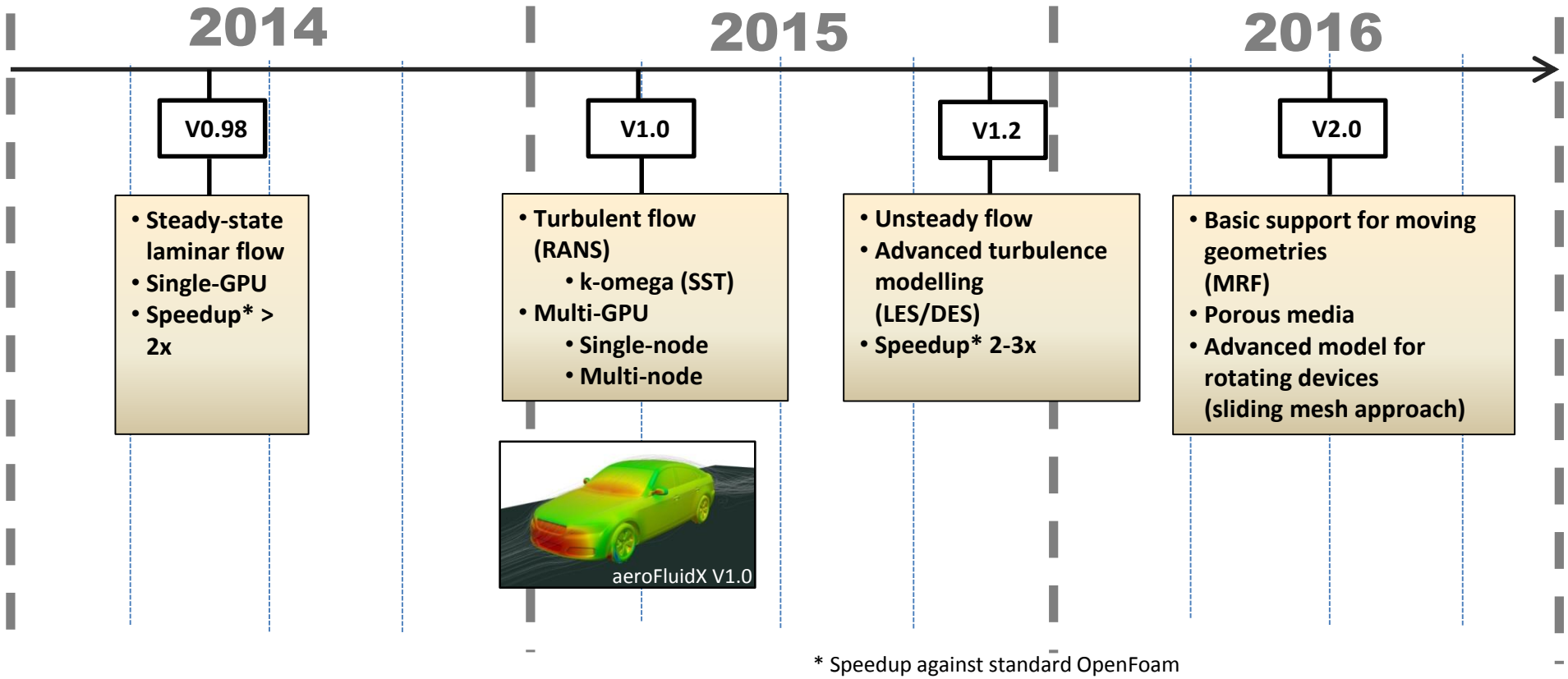
- AeroFluidX assembly phase scales better than linear solver phase
- Linear solver for pressure scales worse than velocity
  as expected due to multi-level vs. one-level method scaling

Conclusions

- **Culises** - hybrid approach for accelerated CFD applications (OpenFOAM®)
  - **General applicability** for industrial cases including various existing flow models
  - Significant speedup of linear solver employing GPUs
  - **Moderate speedup** of total simulation; better acceleration for **transient flow problems**
  - Multi-GPU scaling of multigrid-based methods improvable

- **aeroFluidX** - fully ported flow solver on GPU to harvest full GPU computing power
  - General applicability requires **rewrite of large portion of existing code;** validation necessary
  - Steady-state, incompressible, **multi-GPU**, **unstructured multigrid** flow solver established & validated
  - Significant speedup of matrix assembly
  - **Enhanced speedup** of total simulation

Release planning



* Speedup against standard OpenFoam

Contact information

- Licensing & testing

  www.culises.com
  Culises V1.1 released: Commercial and academic licensing available
  Free testing & benchmarking opportunities at FluiDyna GPU-servers

  www.aerofluidx.com

- Questions & Feedback

  Email to:

  ax@FluiDyna.com

  aerofluidx@FluiDyna.com

  Bjoern.Landmann@FluiDyna.de

- Acknowledgement:

  AmgX team of Nvidia (Joe Eaton, Maxim Naumov, Andrei Schaffer, Marat Arsaev, Alexandre Fender)