Karlsruhe Institute of Technology

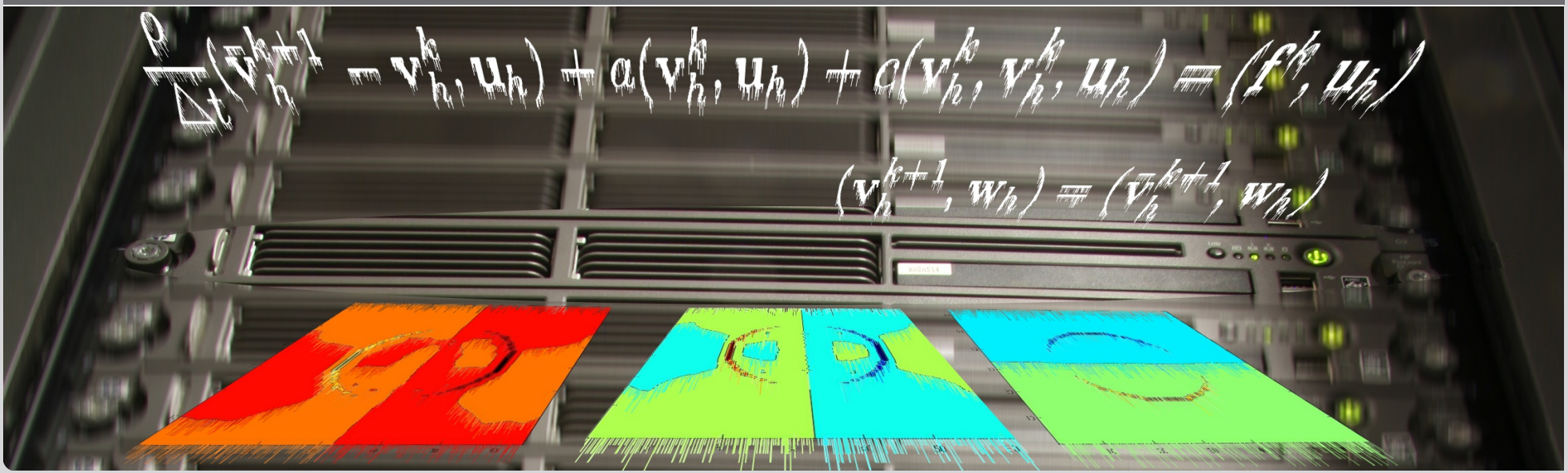
Mixed Precision Iterative Refinement Methods

Energy Efficiency on Hybrid Hardware Platforms

Björn Rocker

Hamburg, June 17th 2010

Engineering Mathematics and Computing Lab (EMCL)



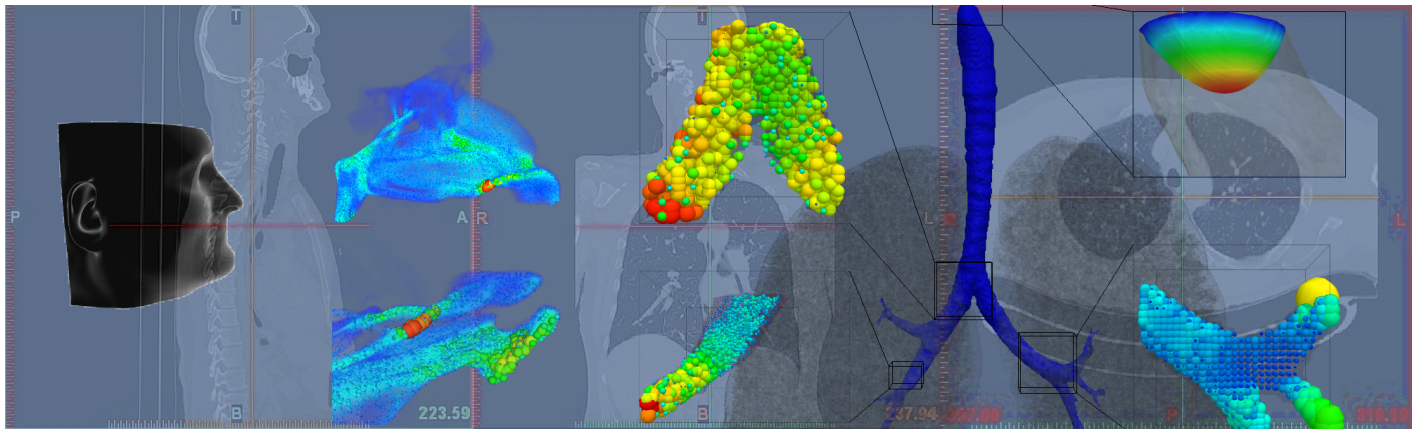
Computational Fluid Dynamics

Karlsruhe Institute of Technology

Linear Systems in Computational Fluid Dynamics

- **Fluid flow problems** can be modeled by **partial differential equations**
- Often **Finite Element Methods** are used to find solutions
- **Linearization methods** lead to linear system $Ax = b$

Solving the linear system is typically the most time-consuming step in the simulation process.



www.united-airways.eu

Acceleration Scenarios

Karlsruhe Institute of Technology

Computational Effort

- Characteristics of the linear problem
- Characteristics of the applied solver
- Used floating point format

Acceleration Concepts

- Use of different precision formats
- Use of coprocessor technology
- Parallelization of the linear solvers

Fundamental Idea of Mixed Precision Solvers:

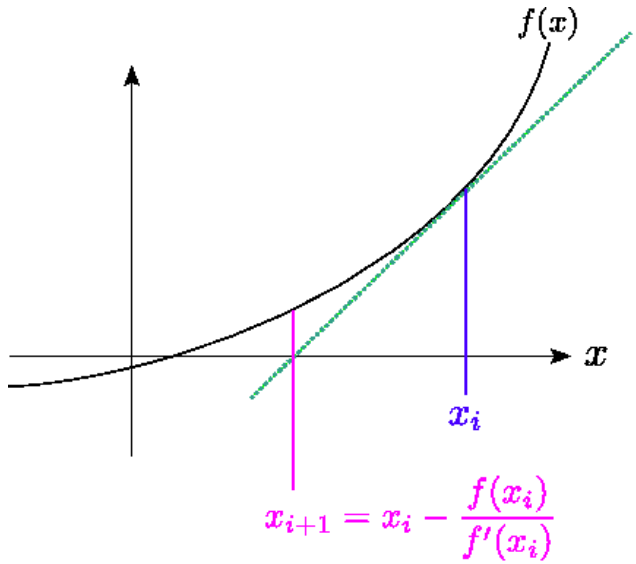
- Acceleration without loss of accuracy
- High precision only in relevant parts
- Low precision for most of the algorithm
- Outsource low precision computations on parallel low precision coprocessor
- Use different precision formats within the Iterative Refinement Method

Iterative Refinement Method

Karlsruhe Institute of Technology

Newton's Method:

$$x_{i+1} = x_i - (\nabla f(x_i))^{-1} f(x_i)$$



Idea: Apply Newton's Method to the function $f(x) = b - Ax$

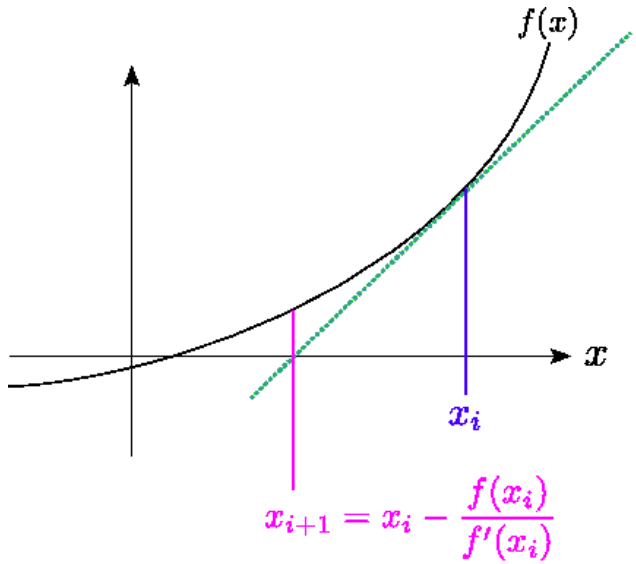
$$\begin{aligned} x_{i+1} &= x_i - (\nabla f(x_i))^{-1} f(x_i) \\ &= x_i - (-A^{-1})(b - Ax_i) \\ &= x_i + A^{-1}(b - Ax_i) \\ &= x_i + \underbrace{A^{-1} r_i}_{=:c_i} \end{aligned}$$

Iterative Refinement Method

Karlsruhe Institute of Technology

Newton's Method:

$$x_{i+1} = x_i - (\nabla f(x_i))^{-1} f(x_i)$$



Idea: Apply Newton's Method to the function $f(x) = b - Ax$

$$\begin{aligned} x_{i+1} &= x_i - (\nabla f(x_i))^{-1} f(x_i) \\ &= x_i - (-A^{-1})(b - Ax_i) \\ &= x_i + A^{-1}(b - Ax_i) \\ &= x_i + \underbrace{A^{-1} r_i}_{=:c_i} \end{aligned}$$

1: initial guess as starting vector: x_0
2: compute initial residual: $r_0 = b - Ax_0$
3: **while** ($\| Ax_i - b \|_2 > \varepsilon \| r_0 \|$) **do**
4: $\quad r_i = b - Ax_i$
5: $\quad$ solve: $Ac_i = r_i$
6: $\quad$ update solution: $x_{i+1} = x_i + c_i$
7: **end while**

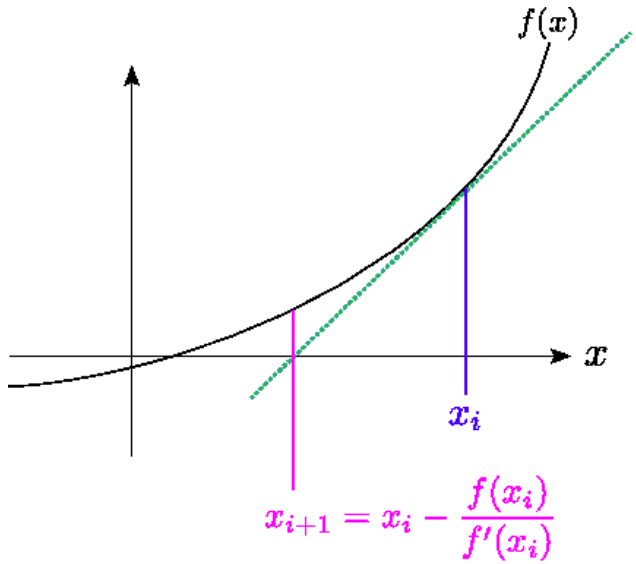
EMCL Engineering Mathematics and Computing Lab

Iterative Refinement Method

Karlsruhe Institute of Technology

Newton's Method:

$$x_{i+1} = x_i - (\nabla f(x_i))^{-1} f(x_i)$$



Idea: Apply Newton's Method to the function $f(x) = b - Ax$

$$\begin{aligned} x_{i+1} &= x_i - (\nabla f(x_i))^{-1} f(x_i) \\ &= x_i - (-A^{-1})(b - Ax_i) \\ &= x_i + A^{-1}(b - Ax_i) \\ &= x_i + \underbrace{A^{-1} r_i}_{=:c_i} \end{aligned}$$

In-Exact Newton Method

- Apply iterative method to solve $Ac_i = r_i$
- Residual stopping criterion $\varepsilon_{inner} >> \varepsilon$
- E.g. Krylov Subspace Solver

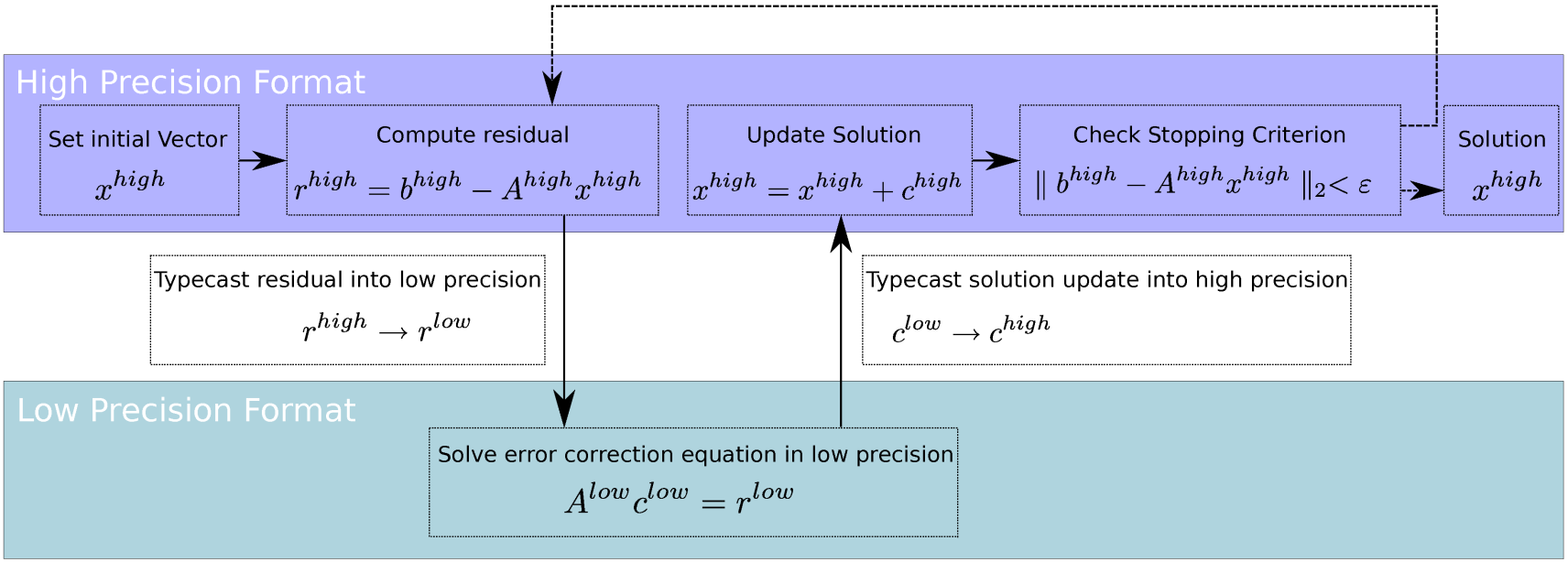
EMCL Engineering Mathematics and Computing Lab

Mixed Precision Iterative Refinement

1: initial guess as starting vector: x_0
2: compute initial residual: $r_0 = b - Ax_0$
3: **while** ($\| Ax_i - b \|_2 > \varepsilon \| r_0 \|$) **do**
4: $r_i = b - Ax_i$
5: solve: $Ac_i = r_i$
6: update solution: $x_{i+1} = x_i + c_i$
7: **end while**

Any error correction solver can be used

i.e. Krylov Subspace Methods (CG, GMRES ...)



Hardware Platform

Karlsruhe Institute of Technology

	HC3	Tesla	IC1
Processor type	Xeon 5540	Xeon 5450	Xeon 5355
Accelerator type	-	Tesla T10	-
Processors per node	2	2 CPUs / 1 GPU	2
Cores per processor	4	8 / 240	4
Theoretical comp. rate / core	10.1 GFlop/s	12 / 3.9 GFlop/s	10.7 GFlop/s
Theoretical comp. rate / node	81 GFlop/s	96 / 933 GFlop/s	85.3 GFlop/s
L2-cache per processor	8 MB	8 / - MB	8 MB
Nodes	278 / 32 / 12	-	200
Memory per node	24 / 48 / 144 GB	32 GB	16 GB
Memory full machine	10.3 TB	-	32 TB
Theoretical comp. rate full machine	27 TFlop/s	1.0 TFlop/s	17.6 TFlop/s
Power consumption load	80.8 kW	539 / 187,8 W[a]	103 kW

[a] http://www.nvidia.com

Numerical Experiments

Karlsruhe Institute of Technology

Implementation Issues

Iterative Refinement

- Double precision GMRES
- Mixed precision GMRES
- MKL-Based on CPU
- CUDA-Based on GPU

Test Matrices

Fluid Flow Problem

- Fluid Flow in Venturi Nozzle
- multiple dimensions
- different condition number
- different sparsity

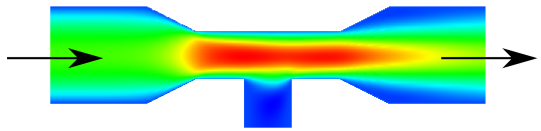
*

* Kansas City Star, 1936

EMCL Engineering Mathematics and Computing Lab

Test-Case CFD

Simulation of a Newtonian Fluid through a Venturi Nozzle.



Fluid can be described by the (incompressible) Navier-Stokes equations:

$$\rho \frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0, \tag{1}$$

Where $\mathbf{u}$ is the fluid velocity field, p the pressure field, ρ the constant fluid density, μ its molecular viscosity constant and $\mathbf{f}$ combines external forces acting on the fluid. The operator D depicts the non-linear material derivative.

Test-Cases CFD

Karlsruhe Institute of Technology

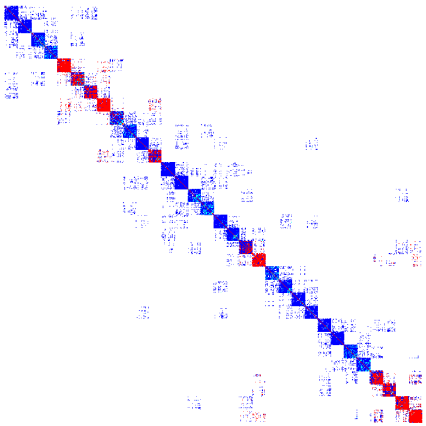
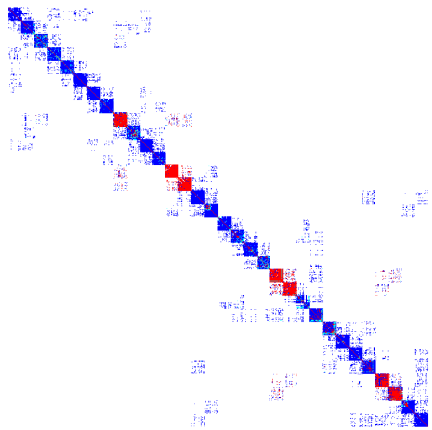
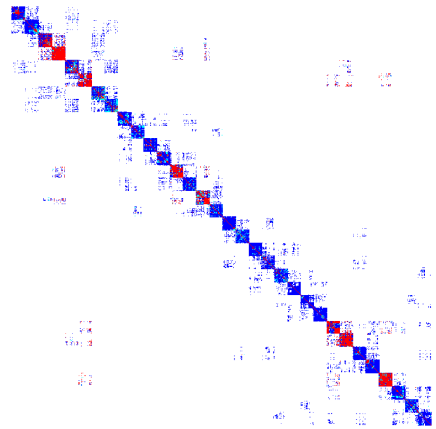
CFD1	CFD2	CFD3
		
problem: 2D fluid flow dimension: $n = 395009$ sparsity: $nnz = 3544321$ storage format: CRS	problem: 2D fluid flow dimension: $n = 634453$ sparsity: $nnz = 5700633$ storage format: CRS	problem: 2D fluid flow dimension: $n = 1019967$ sparsity: $nnz = 9182401$ storage format: CRS

Table: Sparsity plots and properties of the CFD test-matrices.

Computation time CFD 1

Karlsruhe Institute of Technology

Table: Computation time in s for problem CFD 1 based on a GMRES-(10) as inner solver for the error correction method.

		CPU-Cores			GPU
		1	4	8	
Computation time [s]	HC3 double	2267.47	1245.12	776.09	
	HC3 mixed	886.46	567.51	309.61	
	IC1 double	3146.61	1656.53	1627.77	
	IC1 mixed	1378.56	712.83	659.80	
	Tesla mixed				438.13

Computation time CFD 2

Karlsruhe Institute of Technology

Table: Computation time in s for problem CFD 2 based on a GMRES-(10) as inner solver for the error correction method.

		CPU-Cores			GPU
		1	4	8	
Computation time [s]	HC3 double	10765.30	4528.09	3363.44	
	HC3 mixed	4827.98	2177.19	1648.27	
	IC1 double	13204.70	6843.66	6673.07	
	IC1 mixed	5924.32	3495.09	3681.28	
	Tesla mixed				2092.84

Computation time CFD 3

Karlsruhe Institute of Technology

Table: Computation time in s for problem CFD 3 based on a GMRES-(10) as inner solver for the error correction method.

		CPU-Cores			GPU
		1	4	8	
Computation time [s]	HC3 double	62210.70	19954.50	16541.90	
	HC3 mixed	42919.80	9860.26	8828.28	
	IC1 double	60214.50	32875.10	32576.50	
	IC1 mixed	41927.40	19317.00	19836.80	
	Tesla mixed				10316.70

Energy efficiency

Assumptions:

$$E = P \cdot t$$

	Power Consumption (W)	Remarks
IC1	514*	no variation in consumption
HC3	244*	225 - 244* *Wdependingonload* CPU-frequency "ondemand", SMT off
Tesla	726*	node: 539* *W*, *GPU* : 187** *W*

* : measurements

** : manufacturer information

Energy Consumption CFD 1

Karlsruhe Institute of Technology

Table: Energy consumption in Wh for problem CFD 1 based on a GMRES-(10) as inner solver for the error correction method.

		CPU-Cores			GPU
		1	4	8	
Energy consumption in Wh	HC3 double	153.37	84.22	52.49	
	HC3 mixed	59.96	38.39	20.94	
	IC1 double	449.27	236.52	232.41	
	IC1 mixed	196.83	101.78	94.2	
	Tesla mixed				87.36

Energy Consumption CFD 1

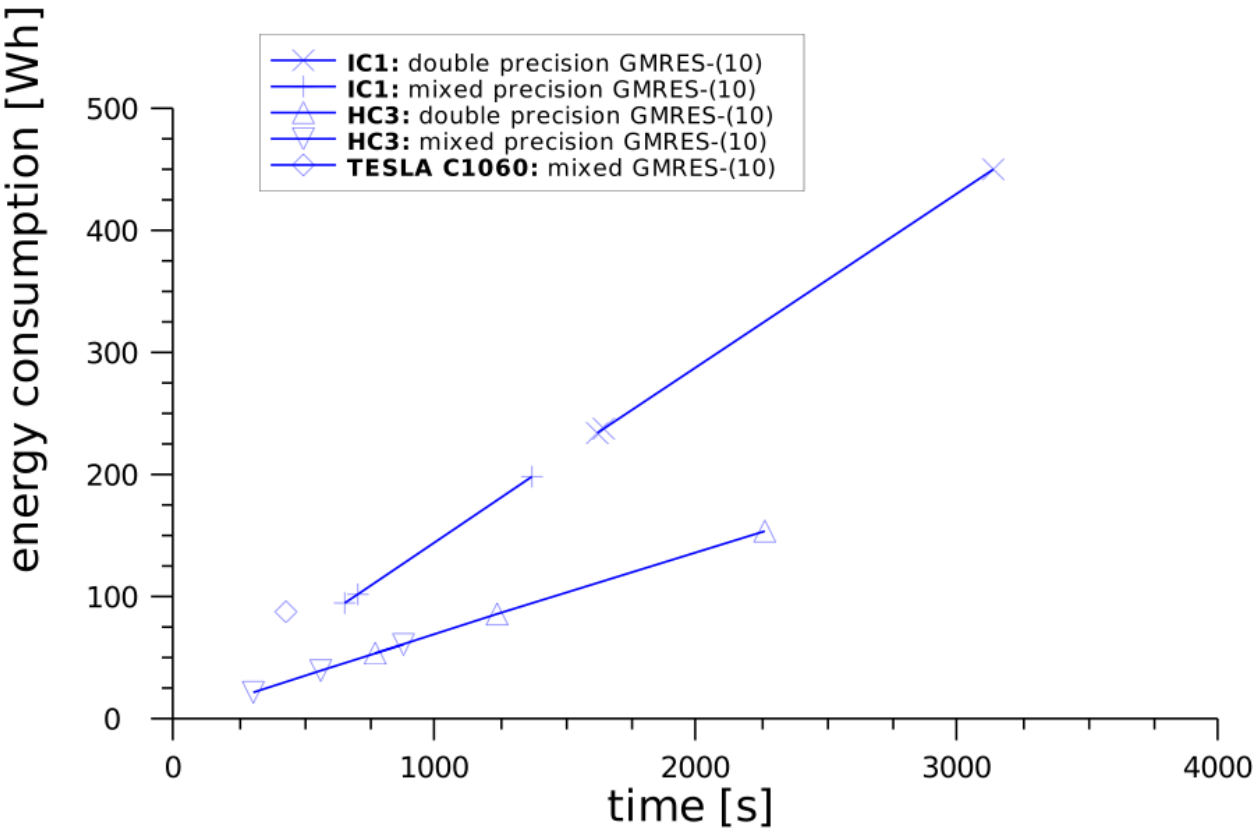


Figure: Energy consumption as a function of time for solving the CFD1 test-case on HC3, IC1 and Tesla. The inner solver is a GMRES-(10).

Energy Consumption CFD 2

Karlsruhe Institute of Technology

Table: Energy consumption in Wh for problem CFD 2 based on a GMRES-(10) as inner solver for the error correction method.

		CPU-Cores			GPU
		1	4	8	
Energy consumption in Wh	HC3 double	728.15	306.27	227.50	
	HC3 mixed	326.56	147.26	111.49	
	IC1 double	1885.34	977.12	952.77	
	IC1 mixed	845.86	499.02	525.60	
	Tesla mixed				417.29

Energy Consumption CFD 2

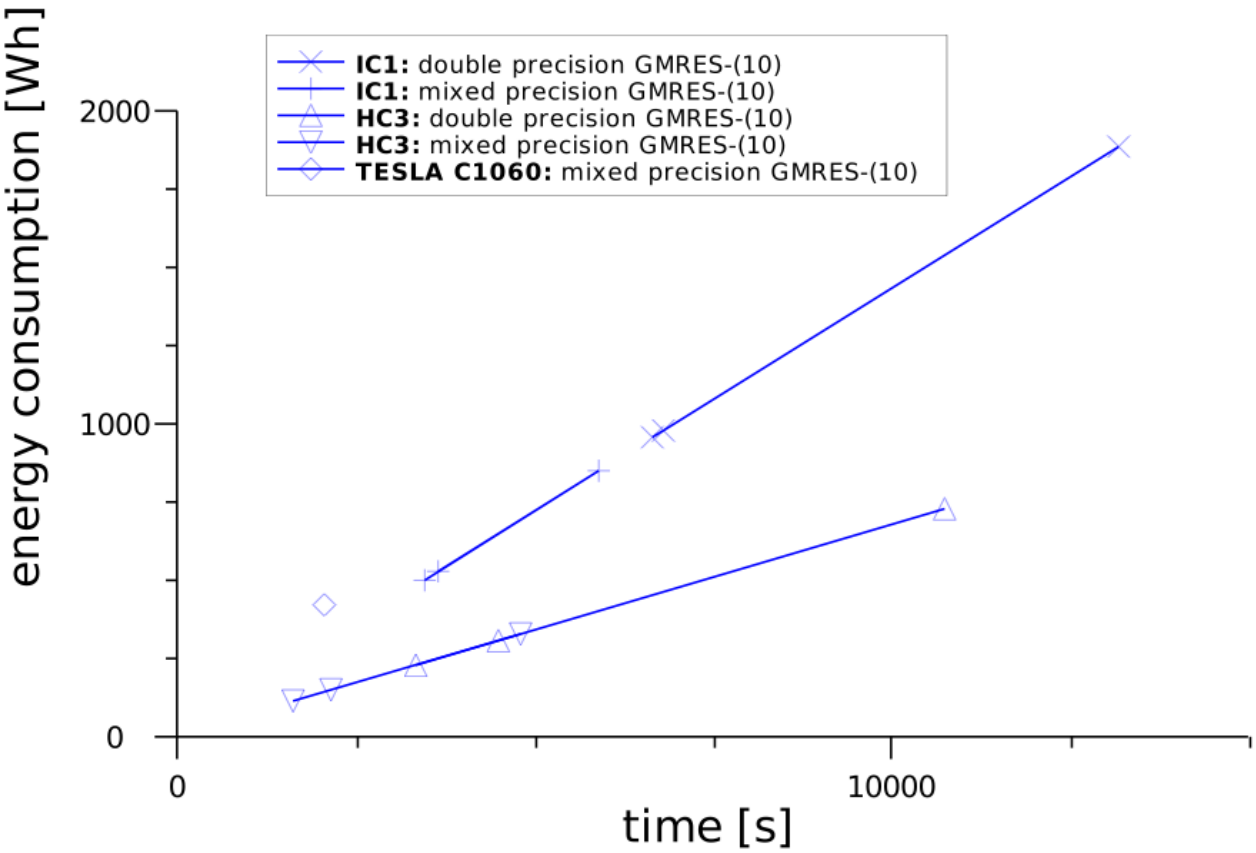


Figure: Energy consumption as a function of time for solving the CFD2 test-case on HC3, IC1 and Tesla. The inner solver is a GMRES-(10).

Energy Consumption CFD 3

Karlsruhe Institute of Technology

Table: Energy consumption in Wh for problem CFD 3 based on a GMRES-(10) as inner solver for the error correction method.

		CPU-Cores			GPU
		1	4	8	
Energy consumption in Wh	HC3 double	4207.86	1349.70	1118.88	
	HC3 mixed	2903.05	666.94	597.14	
	IC1 double	8597.29	4693.83	4651.20	
	IC1 mixed	5986.30	2758.04	2832.25	
	Tesla mixed				2057.04

Energy Consumption CFD 3

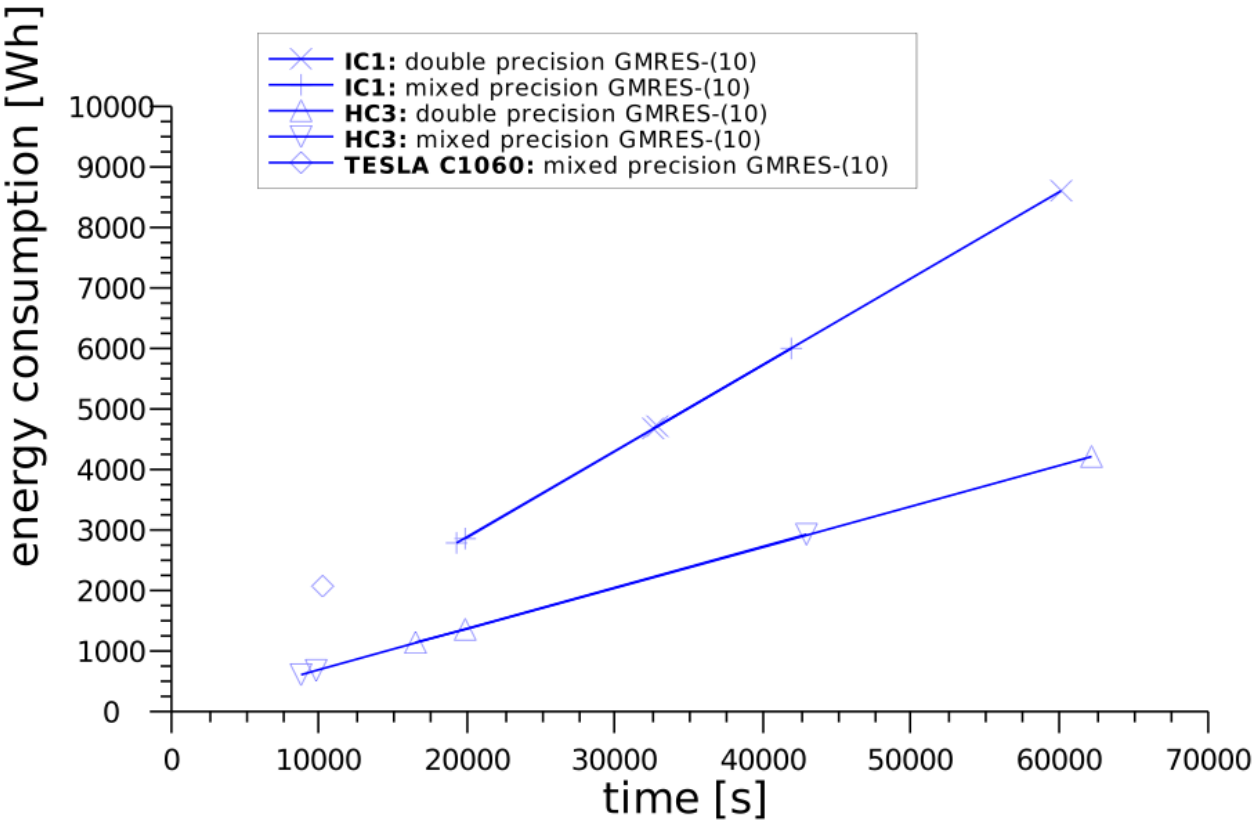


Figure: Energy consumption as a function of time for solving the CFD3 test-case on HC3, IC1 and Tesla. The inner solver is a GMRES-(10).

Evaluation of GPU-based linear solvers

Karlsruhe Institute of Technology

Name	Tesla C2050	Tesla C1060	GTX 480	GTX 280a
Chip	T20	T10	GF100	GT200
Transistors	ca. 3 Mrd.	ca. 1,4 Mrd.	ca. 3 Mrd.	ca. 1,4 Mrd.
Core frequency	1.15 GHz	1.3 GHz	1.4 GHz	1.3 GHz
Shaders (MADD)	448	240	480	240
GFLOPs (single)	1030	933	1.345	933
GFLOPs (double)	515	78	168	78
Memory	3 GB GDDR5	4 GB GDDR3	1.5 GB GDDR5	1 GB GDDR3
Memory Frequency	1.5 GHz	0.8 GHz	1.8 GHz	1.1 GHz
Memory Bandwidth	144 GB/s	102 GB/s	177 GB/s	141 GB/s
ECC Memory	yes	no	no	no
Power Consumption	247 W	187 W	250 W	236 W
IEEE double/single	yes/yes	yes/partial	yes/yes	yes/partial

Table: Key system characteristics of the four GPUs used for the tests. Computation rate and memory bandwidth are peak respectively theoretical values.

Performance

Experiment setup		Computation Time (s)					
problem	solver type	C2050	C1060	GTX 480	GTX 280	HC3	IC1
CFD 1	double	164.84	252.74	145.23	183.37	230.31	482.90
	mixed	80.48	129.19	60.98	98.46	91.71	195.59
CFD 2	double	473.38	778.75	456.17	518.49	819.46	1626.00
	mixed	273.99	510.38	256.43	301.41	401.57	896.94
CFD 3	double	993.63	1921.64	1145.08	1046.49	2493.33	4909.04
	mixed	554.28	1555.36	669.57	697.12	1330.70	2990.09

Table: Computation time (s) for problem CFD 1, CFD 2 and CFD 3 based on a GMRES-(30). IC1 and HC3 results on 8 Cores

Rankings

Performance

1. GTX 480 (256 s)
2. C2050 (273 s)
3. GTX 280 (301 s)
4. HC3 (401 s)
5. C1060 (510 s)
6. IC1 (896 s)

Energy consumption

1. GTX 480 (17,7 Wh)
2. C2050 (18,7 Wh)
3. GTX 280 (19,7 Wh)
4. HC3 (27,2 Wh)
5. C1060 (26,5 Wh)
6. IC1 (127,93 Wh)

Performance and energy ranking for problem CFD 2 based on a GMRES-(30). IC1 and HC3 results on 8 Cores. Energy efficiency for GPUs computed without energy consumption for the host.

Node for the host for GTX 480 has to consume less then 133 W to be more energy efficient than the HC3!

Conclusion

Karlsruhe Institute of Technology

Conclusion

- Iterative refinement shows high potential in case of computationally expensive problems (e.g. our CFD-Problems)
- Iterative refinement is able to exploit the excellent low precision performance of accelerator technologies (e.g. GPU)
- Energy efficiency and performance can be improved by accelerate older computing nodes by GPUs

Future Work

Karlsruhe Institute of Technology

Future Work

- Numerical analysis to optimize choice of floating point format
- Evaluate more hardware constellations
- Define more testcases and optimize solvers
- More accurate power consumption measurements
- FPGA-Technology offers free choice of Floating Point Formats

Engineering Mathematics and Computing Lab