Karlsruhe Institute of Technology

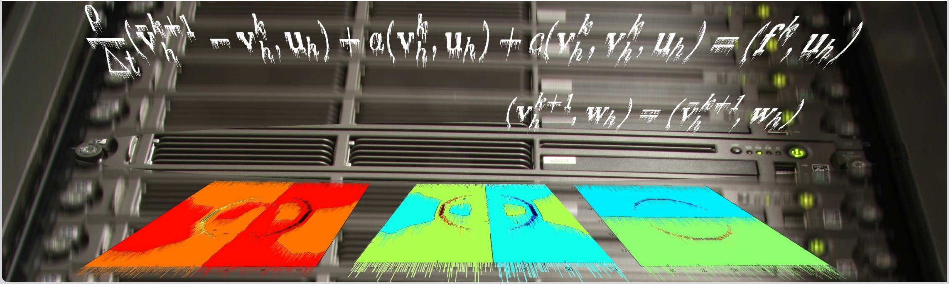
An Error Correction Solver for Linear Systems

Evaluation of Mixed Precision Implementations

Hartwig Anzt

Berkeley, June 23rd 2010

Engineering Mathematics and Computing Lab (EMCL)



Numerical Simulation, Optimization and HPC

NumHPC

EMCL

Engineering Mathematics and Computing Lab

Numerical Simulation, Optimization and HPC

Engineering Mathematics and Computing Lab

Topics

- Meteorology and Environment
- Medicine and Biotechnology
- Energy

Tools

- Mathematical Modeling
- FEM-Software
- Scientific Computing

Numerical Simulation, Optimization and HPC

Karlsruhe Institute of Technology

NumHPC

EMCL

Engineering Mathematics and Computing Lab

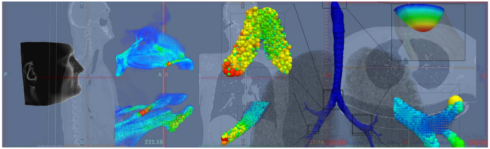
Topics

- Meteorology and Environment
- Medicine and Biotechnology
- Energy

Tools

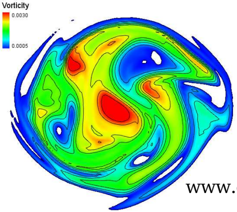
- Mathematical Modeling
- FEM-Software
- Scientific Computing

Simulation of the Human Respiratory System



www.united-airways.eu

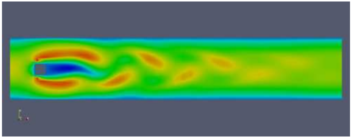
Simulation of Tropical Cyclones



www.emcl.kit.edu

Computational Fluid Dynamics

Linear Systems in Fluid Dynamics



- **Partial differential equations** model **Fluid Flow Problems**
- **Discretization methods** lead to nonlinear equations systems
- **Linearization methods** lead to linear system $Ax = b$

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

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

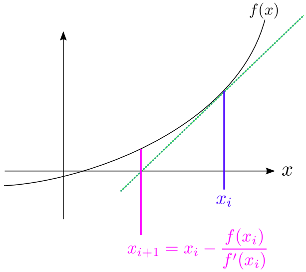
Fundamental Idea of Mixed Precision Solvers:

- 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

Newton's Method:

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



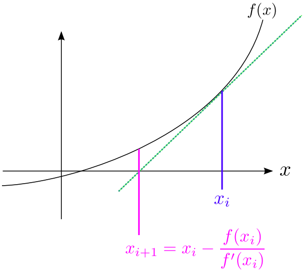
Idea: Apply Newton's Method to the function $f(x) = b - Ax$

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

Iterative Refinement Method

Newton's Method:

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



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

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

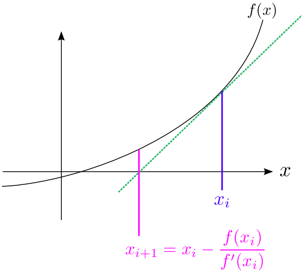
1: initial guess as starting vector: x_0
2: compute initial residual: $r_0 = b - Ax_0$
3: **while** ($\| Ax_i - b \|_2 > \varepsilon \| r_0 \|$) **do**
4: $\quad r_i = b - Ax_i$
5: $\quad$ solve: $Ac_i = r_i$
6: $\quad$ update solution: $x_{i+1} = x_i + c_i$
7: **end while**

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

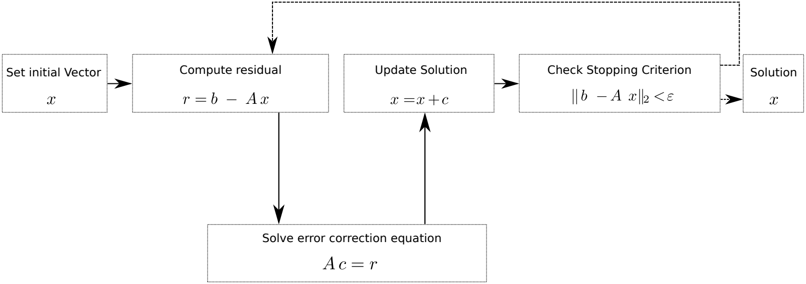
Mixed Precision Iterative Refinement

Karlsruhe Institute of Technology

```
initial guess as starting vector: x_0
compute initial residual: r_0 = b - A x_0
while (|| A x_i - b ||_2 > ε || r_0 ||) do
    r_i = b - A x_i
    solve: A c_i = r_i
    update solution: x_{i+1} = x_i + c_i
end while
```

Any error correction solver can be used

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



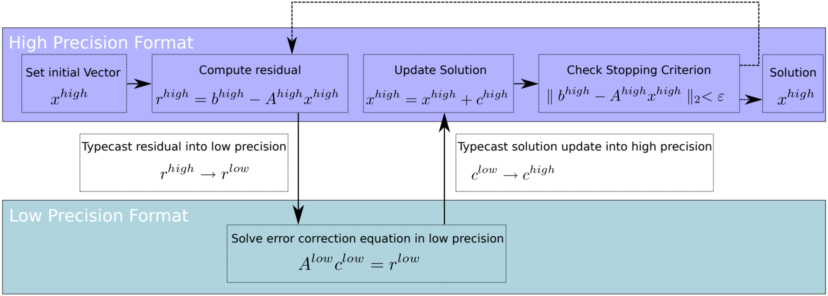
Mixed Precision Iterative Refinement

Karlsruhe Institute of Technology

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

Any error correction solver can be used

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



Numerical Experiments

Karlsruhe Institute of Technology

Implementation Issues

- Double precision GMRES-(10) on 1 core
- Mixed precision GMRES-(10) on 1 core
- Double precision GMRES-(10) on 4 cores
- Mixed precision GMRES-(10) on 4 cores
- Mixed precision GMRES-(10) on NVIDIA TESLA S1070

Test Matrices

Artificially created Test Matrices

- Increasing dimension
- Fixed Condition Number
- sparse/dense

Real Application Test Matrices

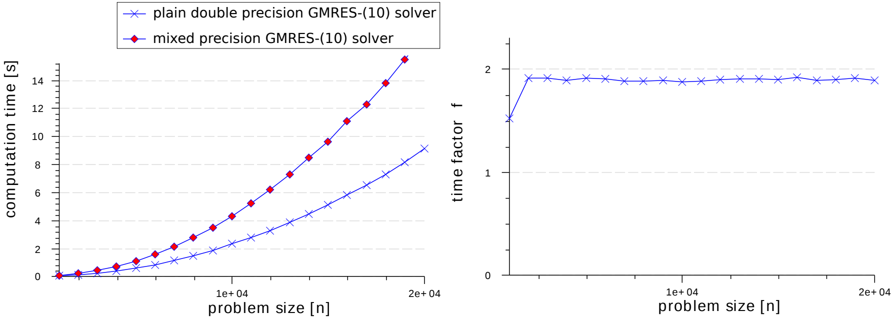
- Discretization of Fluid Flow in Venturi Nozzle
- different dimension, condition number, sparsity

Test-Case: Artificial Test Matrices

Karlsruhe Institute of Technology

M1	M2	M3
$\begin{pmatrix} 10\cdot n & * & \cdots & \cdots & * \\ * & 10\cdot n & \ddots & & \vdots \\ \vdots & \ddots & 10\cdot n & \ddots & \vdots \\ \vdots & & \ddots & \ddots & * \\ * & \cdots & \cdots & * & 10\cdot n \end{pmatrix}$	$\begin{pmatrix} W & V & * & \cdots & * \\ V & W & V & \ddots & \vdots \\ * & V & W & \ddots & * \\ \vdots & \ddots & \ddots & \ddots & V \\ * & \cdots & * & V & W \end{pmatrix}$	$\begin{pmatrix} H & -1 & 0 & \cdots & -1 & 0 & \cdots & 0 \\ -1 & H & -1 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & H & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & & 0 & -1 \\ -1 & \ddots & & & & & & \vdots \\ 0 & \ddots & \ddots & & & & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \ddots & H & -1 \\ 0 & \cdots & 0 & -1 & \cdots & 0 & -1 & H \end{pmatrix}$
problem: artificial matrix problem size: variable sparsity: $nnz = n^2$ cond. number: $\kappa < 3$ storage format: MAS	problem: artificial matrix problem size: variable sparsity: $nnz = n^2$ cond. number: $\kappa \approx 8 \cdot 10^3$ storage format: MAS	problem: artificial matrix problem size: variable sparsity: $nnz = 5n$ cond. number: $\kappa \approx 8 \cdot 10^3$ storage format: CRS

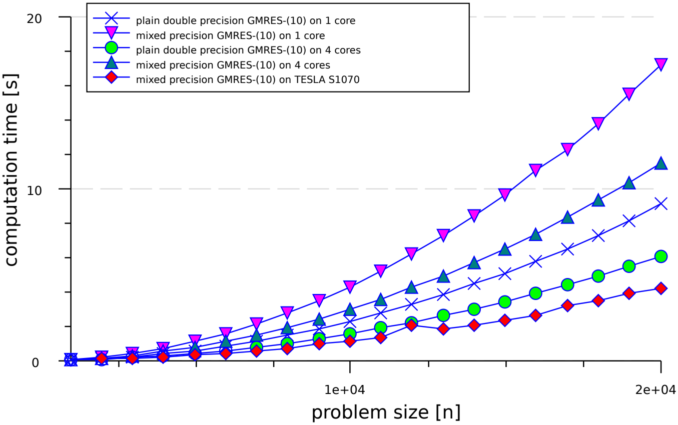
Mixed Precision Iterative Refinement GMRES



Artificial Test-Matrix M1: $\kappa < 3$, $nnz = n^2$

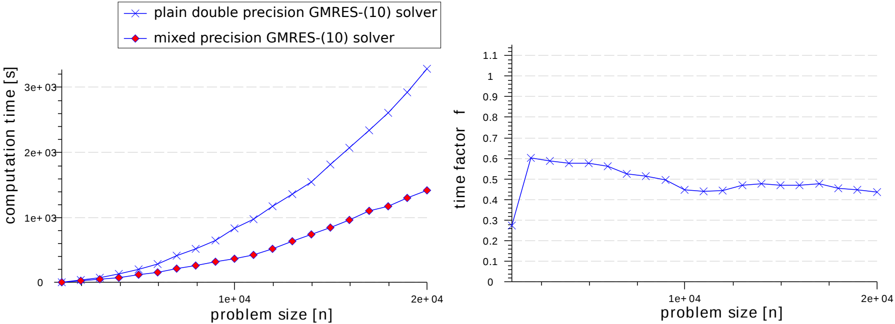
- Constant factor f
- Plain GMRES performs faster due to low condition number

Mixed Precision Iterative Refinement GMRES



Artificial Test-Matrix M1: $\kappa < 3$, $nnz = n^2$

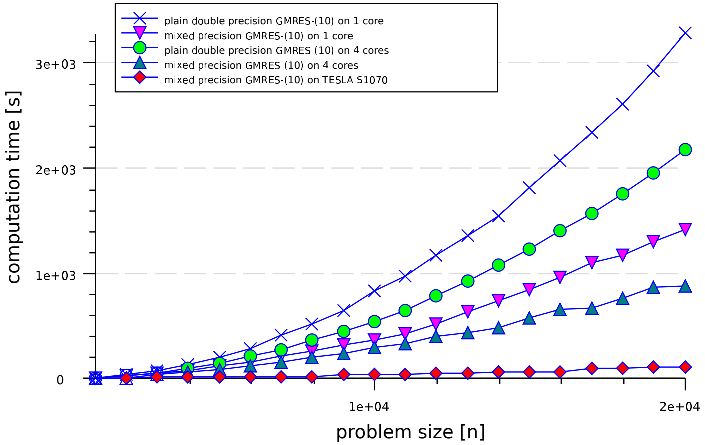
Mixed Precision Iterative Refinement GMRES



Artificial Test-Matrix M2: $\kappa \approx 8000$, $nnz = n^2$

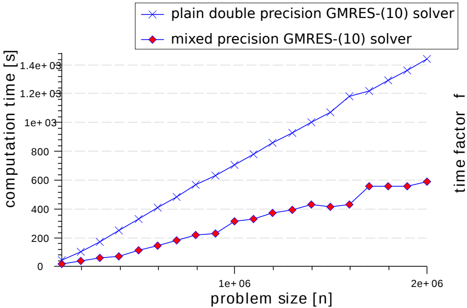
- Constant factor f
- Mixed Precision Iterative Refinement GMRES performs faster due to high condition number

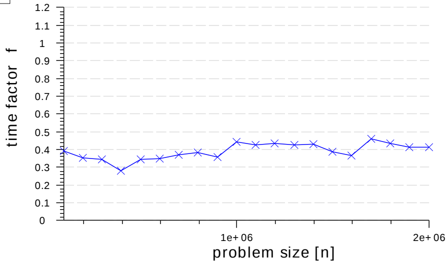
Mixed Precision Iterative Refinement GMRES



Artificial Test-Matrix M2: $\kappa \approx 8000$, $nnz = n^2$

Mixed Precision Iterative Refinement GMRES

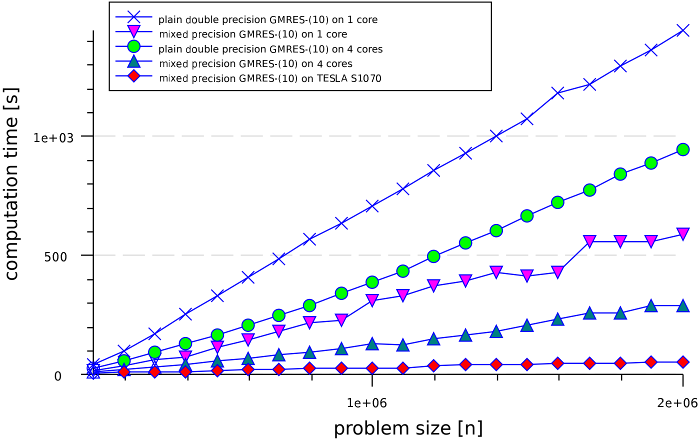




Artificial Test-Matrix M3: $\kappa \approx 8000$, $nnz = 5n$

- Constant factor f
- Mixed Precision Iterative Refinement GMRES performs faster due to high condition number

Mixed Precision Iterative Refinement GMRES

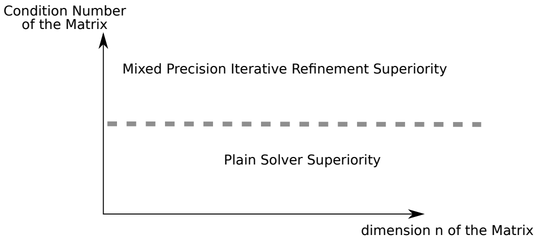


Artificial Test-Matrix M3: $\kappa \approx 8000$, $nnz = 3n$

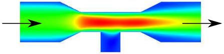
Result Interpretation

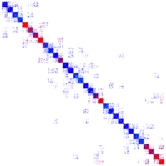
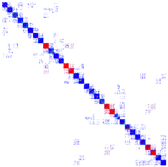
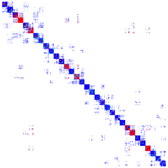
Numerical Experiments

- **Plain solver** is superior for **low condition numbers**
- **Mixed precision mode** superior for **high condition numbers**
- **Quotient** between plain- and mixed-precision- mode **independent of problem size**
- **Acceleration** by **Coprocessor Technology**

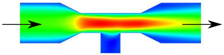


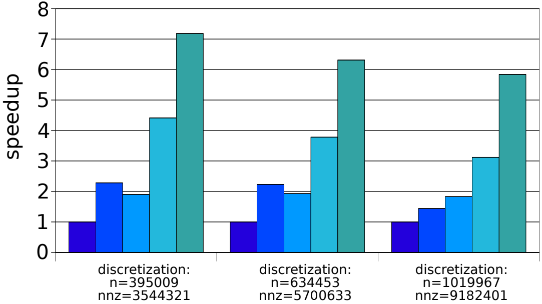
Test-Case: Fluid Flow in Venturi Nozzle



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

Test-Case: Fluid Flow in Venturi Nozzle





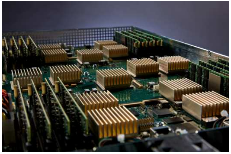
- plain double precision GMRES-(10) on 1 core
- mixed precision GMRES-(10) on 1 core
- plain double precision GMRES-(10) on 4 cores
- mixed precision GMRES-(10) on 4 cores
- mixed precision GMRES-(10) on NVIDIA TESLA S1070

Conclusions

Karlsruhe Institute of Technology

Mixed Precision Solvers

- Exploiting the excellent low precision performance of Accelerator Technology (e.g. GPU)
- High potential in case of computationally expensive problems (e.g. our CFD-Problems)
- FPGA-Technology offers free choice of Floating Point Formats



www.conveycomputer.com

TECHNIC