

On Level Scheduling for Incomplete LU Factorization Preconditioners on Accelerators

Karl Rupp, Barry Smith
rupp@mcs.anl.gov

Mathematics and Computer Science Division
Argonne National Laboratory

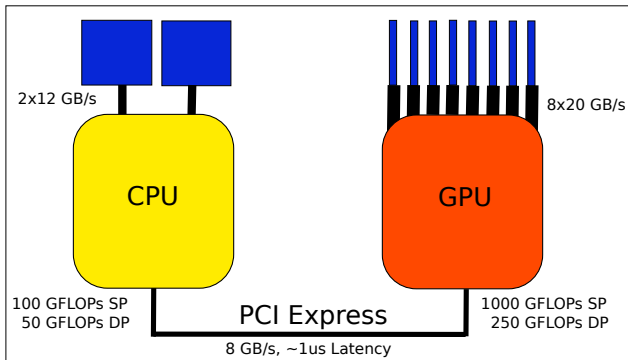
FEMTEC 2013

May 20th, 2013



Parallel Hardware Constraints

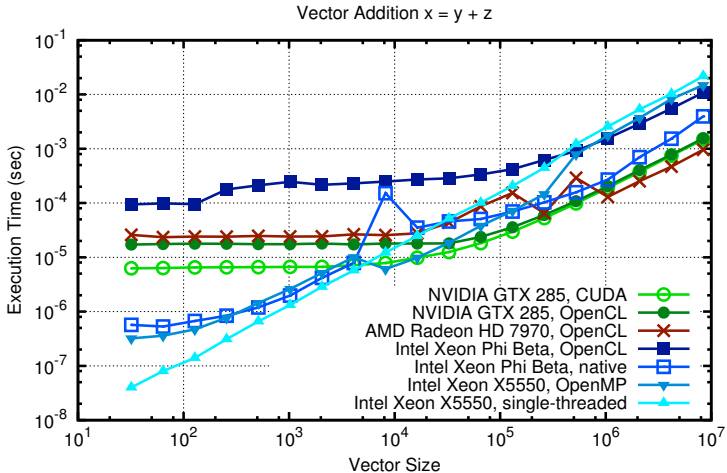
Computing Architecture Schematic



Good for large FLOP-intensive tasks, high memory bandwidth
PCI-Express can be a bottleneck

» 10-fold speedups (usually) not backed by hardware

Benchmarks



Incomplete LU Factorization Preconditioners

Level Scheduling

Build dependency graph

Substitute as many entries as possible simultaneously

Trade-off: Each step vs. multiple steps in a single kernel

$$\begin{pmatrix} 5 & \times & \times & \times & & \times & \times & & & \\ \times & 3 & \times & & & & & & & \\ \times & \times & 4 & \times & & & & & & \\ \times & & \times & 5 & \times & \times & & & & \times \\ & & & \times & 5 & \times & & \times & \times & \\ \times & & & \times & \times & 6 & \times & \times & & \\ \times & & & & & \times & 3 & & & \\ & & & & \times & \times & & 4 & \times & \\ & & & \times & \times & & & \times & 4 & \end{pmatrix}$$

Incomplete LU Factorization Preconditioners

Level Scheduling

Build dependency graph

Substitute as many entries as possible simultaneously

Trade-off: Each step vs. multiple steps in a single kernel

$$\begin{pmatrix} 5 & \times & \times & \times & & \times & \times & & & & \\ \times & 3 & \times & & & & & & & & \\ \times & \times & 4 & \times & & & & & & & \\ \times & & \times & 5 & \times & \times & & & & & \times \\ \times & & & \times & 5 & \times & & \times & \times & & \\ \times & & & \times & \times & 6 & \times & \times & & & \\ \times & & & & & \times & 3 & & & & \\ & & & & \times & \times & & 4 & \times & & \\ & & \times & \times & & & & \times & 4 & & \end{pmatrix}$$

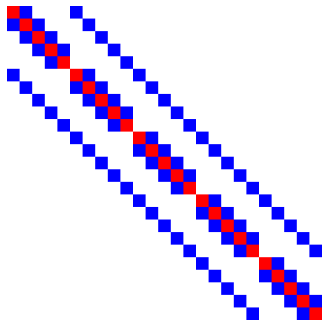
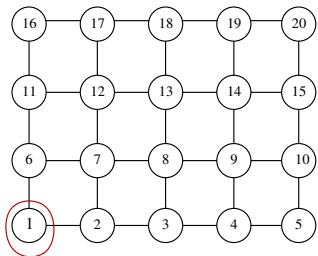
Incomplete LU Factorization Preconditioners

Interpretation on Structured Grids

2d finite-difference discretization

Substitution whenever all neighbors with smaller index computed

Works particularly well in 3d



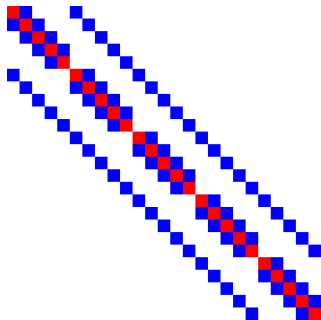
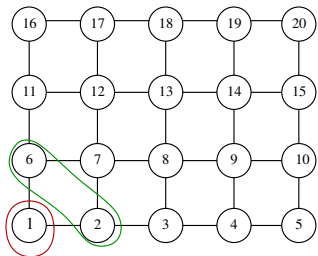
Incomplete LU Factorization Preconditioners

Interpretation on Structured Grids

2d finite-difference discretization

Substitution whenever all neighbors with smaller index computed

Works particularly well in 3d



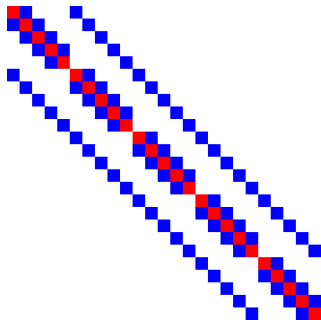
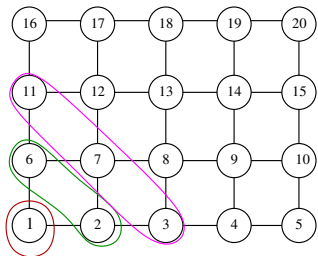
Incomplete LU Factorization Preconditioners

Interpretation on Structured Grids

2d finite-difference discretization

Substitution whenever all neighbors with smaller index computed

Works particularly well in 3d



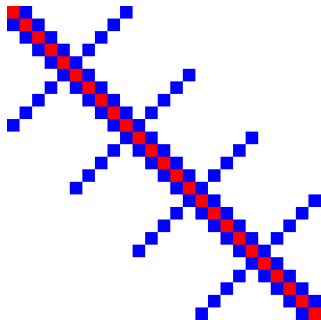
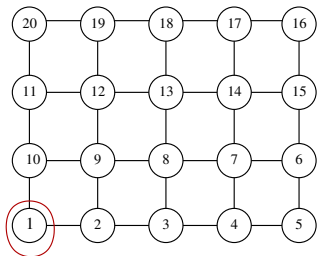
Incomplete LU Factorization Preconditioners

Interpretation on Structured Grids

2d finite-difference discretization

Substitution whenever all neighbors with smaller index computed

Works particularly well in 3d



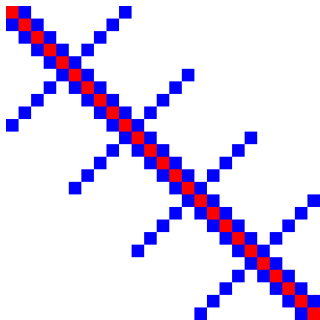
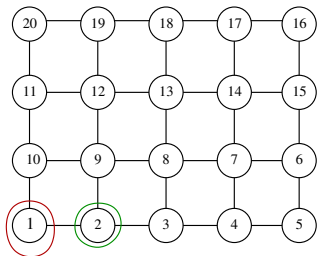
Incomplete LU Factorization Preconditioners

Interpretation on Structured Grids

2d finite-difference discretization

Substitution whenever all neighbors with smaller index computed

Works particularly well in 3d



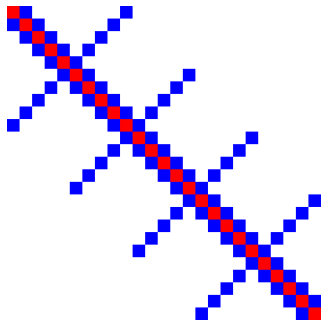
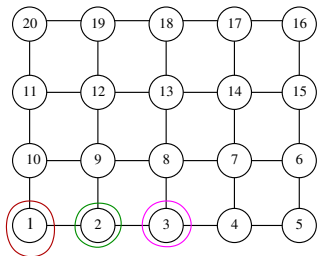
Incomplete LU Factorization Preconditioners

Interpretation on Structured Grids

2d finite-difference discretization

Substitution whenever all neighbors with smaller index computed

Works particularly well in 3d



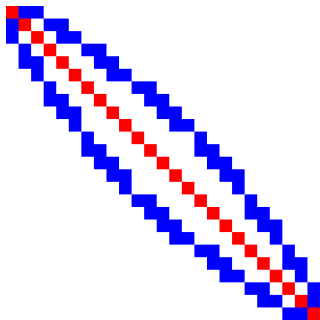
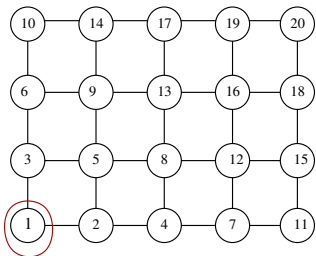
Incomplete LU Factorization Preconditioners

Interpretation on Structured Grids

2d finite-difference discretization

Substitution whenever all neighbors with smaller index computed

Works particularly well in 3d



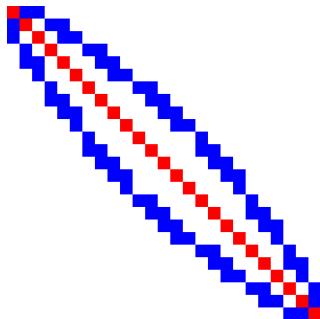
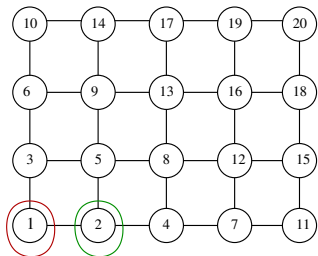
Incomplete LU Factorization Preconditioners

Interpretation on Structured Grids

2d finite-difference discretization

Substitution whenever all neighbors with smaller index computed

Works particularly well in 3d



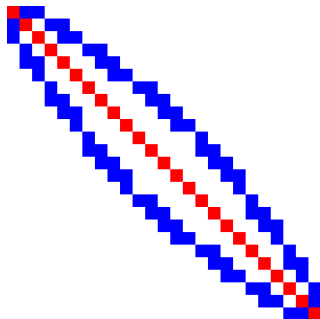
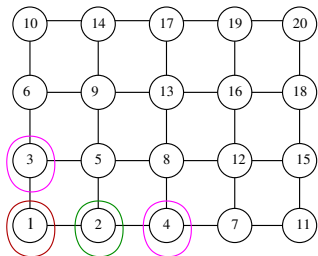
Incomplete LU Factorization Preconditioners

Interpretation on Structured Grids

2d finite-difference discretization

Substitution whenever all neighbors with smaller index computed

Works particularly well in 3d



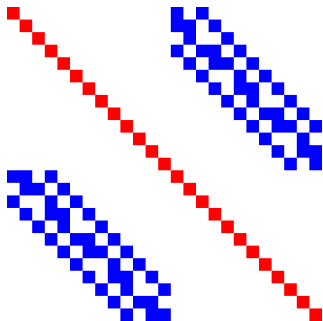
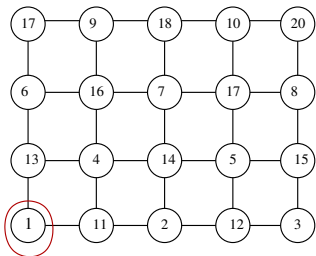
Incomplete LU Factorization Preconditioners

Interpretation on Structured Grids

2d finite-difference discretization

Substitution whenever all neighbors with smaller index computed

Works particularly well in 3d



Incomplete LU Factorization Preconditioners

Block-ILU

Apply ILU to diagonal blocks

Higher parallelism

Usually more iterations required (problem-dependent)

$$\begin{pmatrix} \boxed{\begin{matrix} 5 & \times & \times & \times \\ \times & 3 & \times & \\ \times & \times & 4 & \times \\ \times & & \times & 5 \end{matrix}} & & \times & \times & & \\ & & & & & \\ & & & & & \\ & & & & & \\ \times & & & & \times & \times & & \times \\ \times & & & & \times & & & \\ & & & & & & & \\ & & & & \times & & & \\ & & & & \boxed{\begin{matrix} 5 & \times & \times & \times \\ \times & 6 & \times & \times \\ \times & \times & 3 & \\ \times & \times & & 4 & \times \\ \times & & & \times & 4 \end{matrix}} & & & \end{pmatrix}$$

Benchmark - Setup

Hardware

NVIDIA GTX 580 (default)

AMD HD 7970 (only for final benchmark)

Intel Core2Quad 9550

Numbering

Lexicographic

Red-Black

Minimum Degree

Remarks

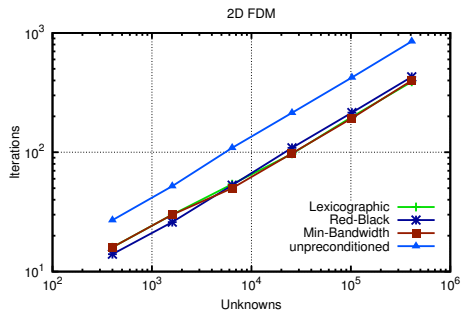
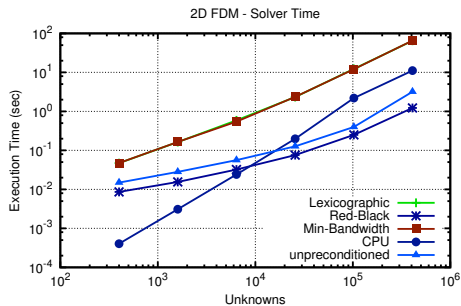
Setup purely on CPU, not included

Data transfer costs not included

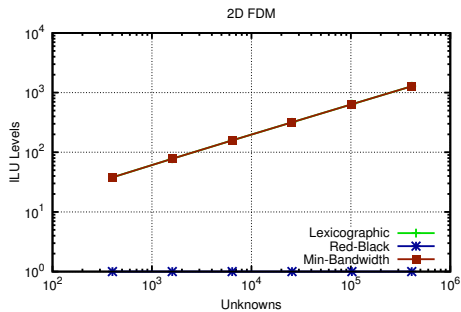
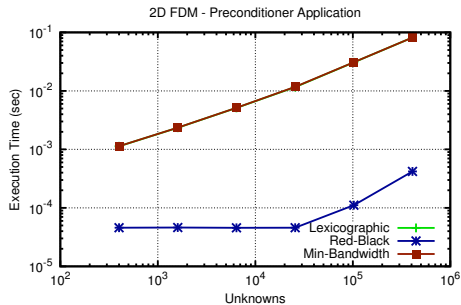
OpenCL for both GPUs

Case Study 1: 2D Poisson, Structured Grid

Benchmarks

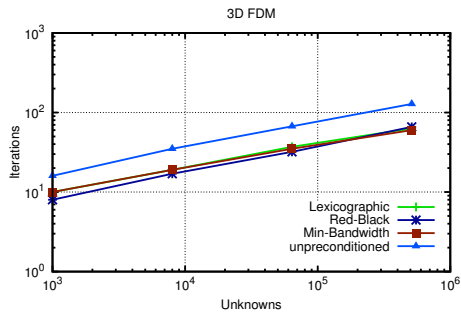
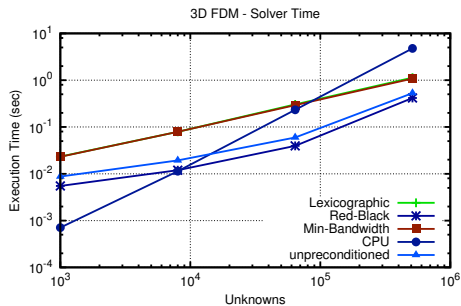


Benchmarks

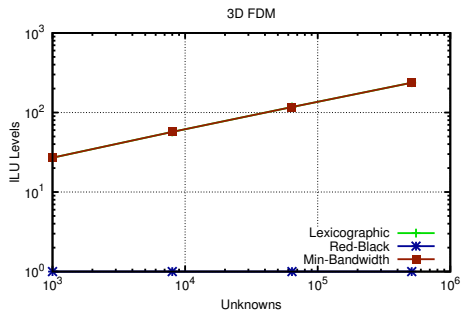
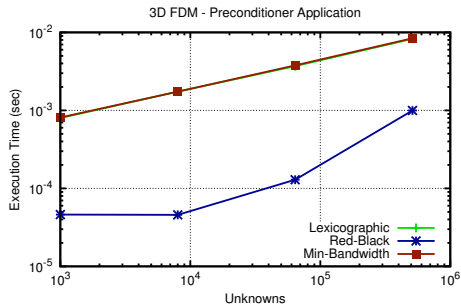


Case Study 2: 3D Poisson, Structured Grid

Benchmarks

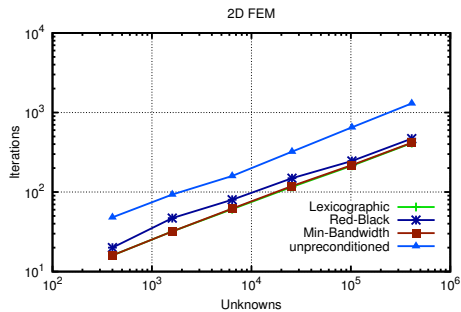
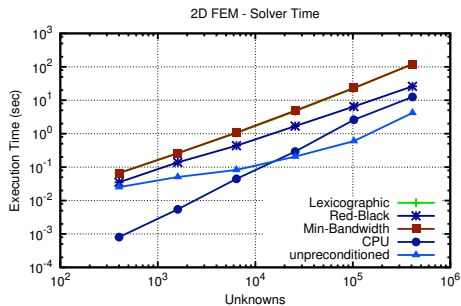


Benchmarks

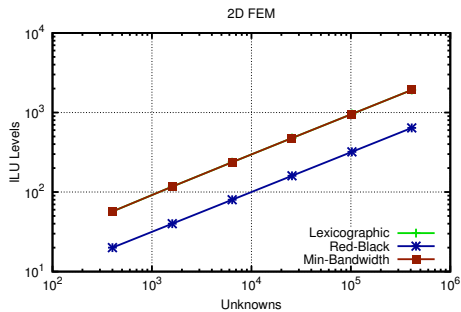
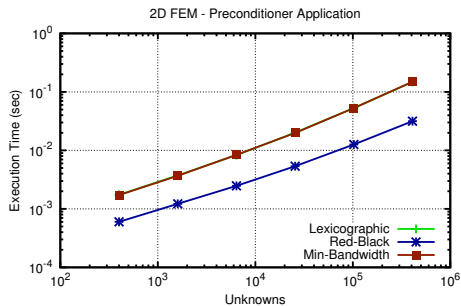


Case Study 3: 2D Poisson, Unstructured Grid

Benchmarks

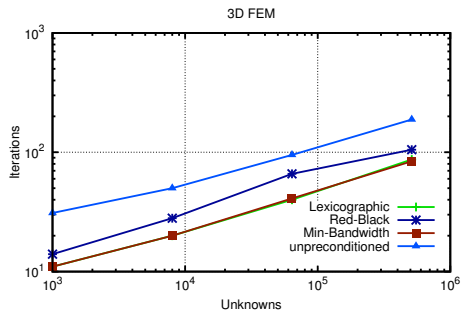
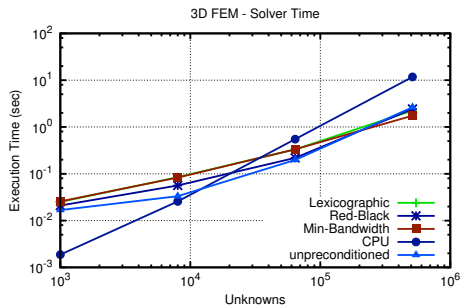


Benchmarks

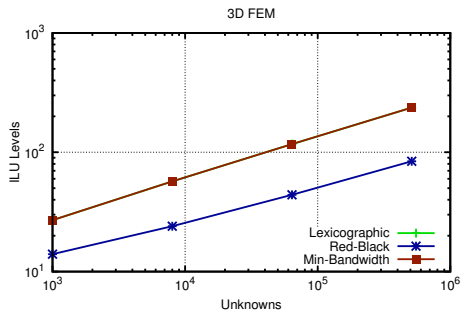
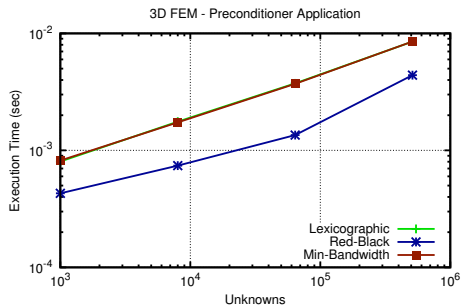


Case Study 4: 3D Poisson, Unstructured Grid

Benchmarks



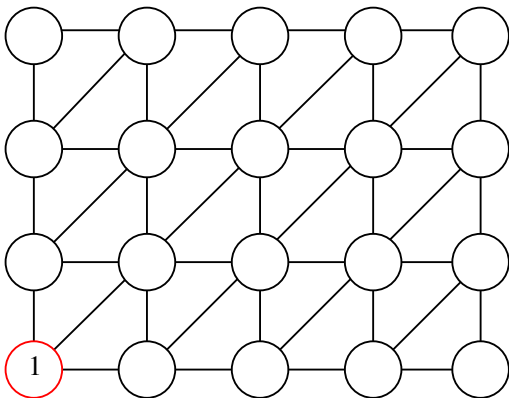
Benchmarks



Coloring

Color dependency graph

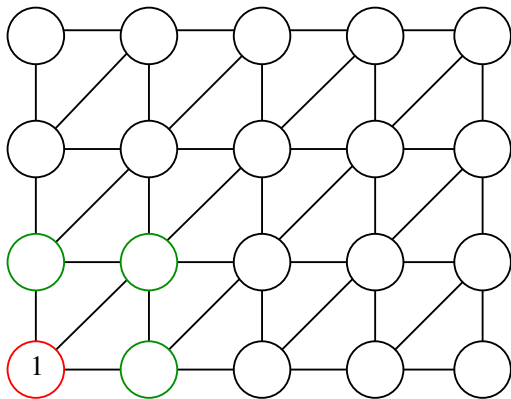
Purely algebraic



Coloring

Color dependency graph

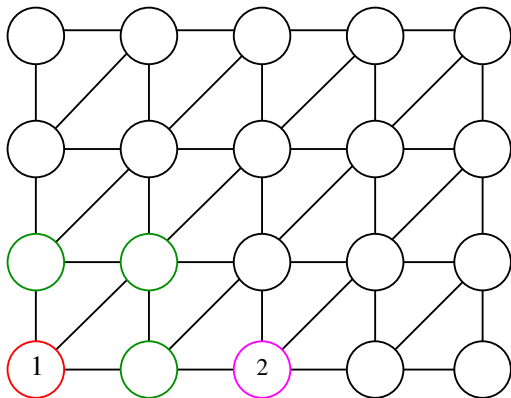
Purely algebraic



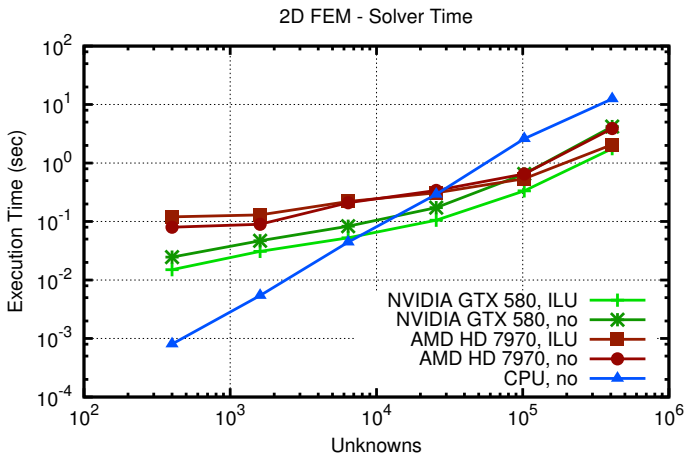
Coloring

Color dependency graph

Purely algebraic



Benchmarks



ILU Preconditioners

Fine-grained parallelism exploitable (if done right)

Higher-order discretizations less parallel

Matrix Pattern

CPU: banded for cache reuse

GPU: colored for parallelism

Availability

ViennaCL: <http://viennacl.sourceforge.net/>

(PETSc: <http://www.mcs.anl.gov/petsc/>)

ViennaCL + PETSc tutorial on Thursday afternoon!