Solvers and Preconditioning

Lesson 4

Nathan Bell

NVIDIA

Iterative Solvers

- Stationary Methods
  - Richardson
  - Gauss-Seidel relaxation
  - Jacobi relaxation
- Krylov Methods
  - Conjugate Gradient (CG)
  - Generalized Minimum Residual (GMRES)
  - Biconjugate Gradient Stabilized (BiCG-stab)
  - Etc.

Stationary Methods

- **Matrix Splitting**

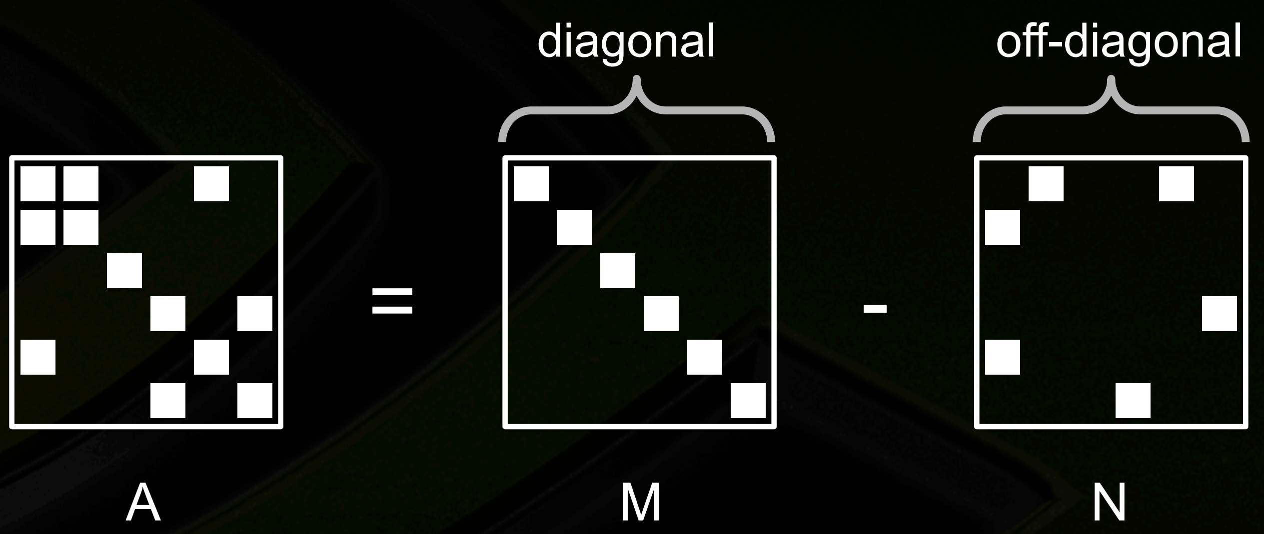
$$A = M - N$$

- **Update solution**

$$M\, x_{k+1} = N\, x_k + b$$
$$x_{k+1} = M^{-1} (N\, x_k + b)$$

Jacobi Relaxation

- **Matrix Splitting**



Jacobi Relaxation

- **Update solution**

$$M = D$$
$$N = D - A$$
$$x_{k+1} = M^{-1} (N x_k + b)$$

- **Simplified Form**

$$x_{k+1} = x_k + D^{-1} (b - A x_k)$$

Krylov Methods

- **Krylov subspace**

$$K_N(A, v) = \text{span}\,\{v, Av, A^2v, A^3v, \ldots A^{N-1}v\}$$

- **Select "best" solution in Krylov subspace**

$r_0 = b - A^*x_0$
Compute "best" u_k in $K_N(A, r_0)$
$x_k = x_0 + u_k$

GMRES

- **Generalized *Minimum Residual* Method**
  - **"Best" = smallest residual**

Compute
    u_k in $K_N(A, r_0)$
such that
    $|| b - A (x0 + u_k) ||$
is minimized

- **Cost grows rapidly with N**
  - **Pick small N and restart (e.g. N = 10)**

CG

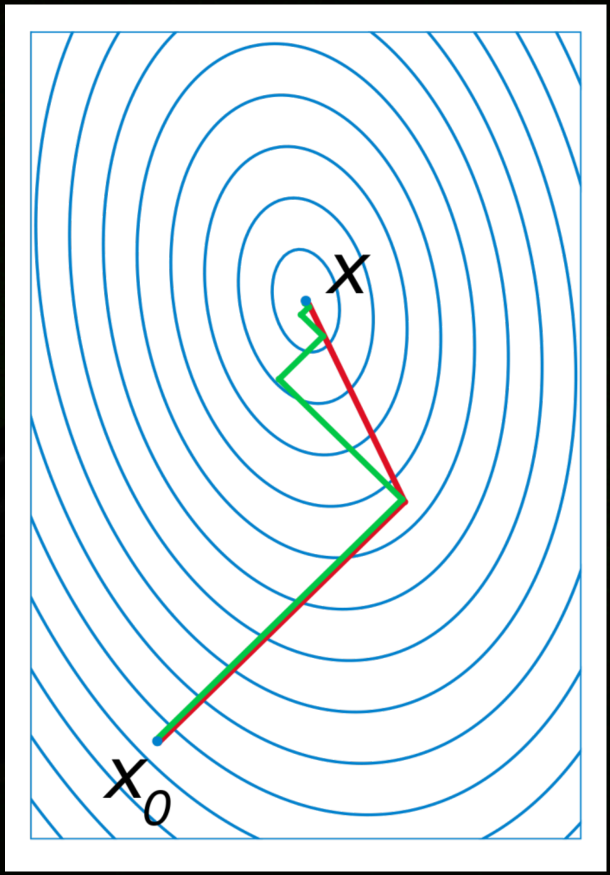
- **Conjugate Gradient Method**
  - **"Best" = smallest residual in the A-norm**

Compute
u_k in $K_N(A, r_0)$
such that
$|| b - A (x0 + u_k) ||_A$
is minimized, where
$|| v ||_A = v^T A v$

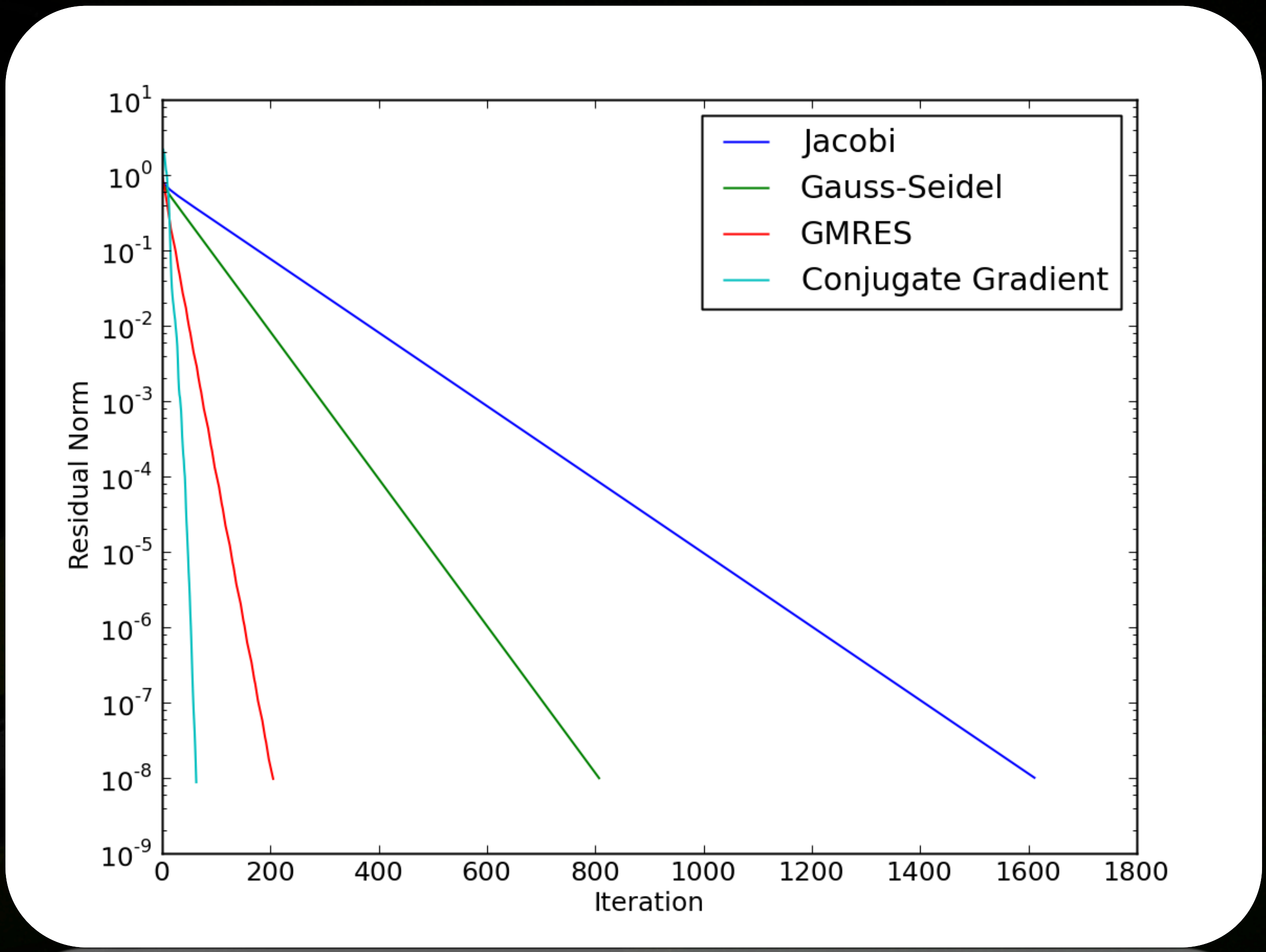
- **Matrix must be S.P.D.**
  - **Symmetric and Positive-Definite**

CG

- Search directions are conjugate
  - $w^T * A * v = 0$
- Finite termination
  - In exact arithmetic



Iterative Solver Performance



What is a Preconditioner?

- **Approximate inverse**
  - **Helps steer solver in the right direction**
  - **Better approximation → Faster Convergence**

$$M \approx A^{-1}$$

Diagonal Preconditioner

- **Simplest preconditioner**
  - **Inverse of matrix diagonal**
  - **Always cheap, sometimes effective**

$$\begin{bmatrix} 1 & 7 & 0 & 0 \\ 0 & 2 & 8 & 0 \\ 5 & 0 & 3 & 9 \\ 0 & 6 & 0 & 4 \end{bmatrix}$$

A

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix}$$

M

Diagonal Preconditioner

```
#include <cusp/krylov/cg.h>
#include <cusp/precond/smoothed_aggregation.h>

...

// set stopping criteria
//  iteration_limit = 100
//  relative_tolerance = 1e-6
cusp::default_monitor<float> monitor(b, 100, 1e-6);

// setup preconditioner
cusp::precond::diagonal<float, cusp::device_memory> M(A);

// solve A * x = b to default tolerance with preconditioned CG
cusp::krylov::cg(A, x, b, monitor, M);
```

Diagonal Preconditioner

- **Always cheap, sometimes effective**
  - **Example: poorly scaled matrix**

```
Solving with no preconditioner
Solver will continue until residual norm 5e-06
  Iteration Number  | Residual Norm
                0        5.000000e+00
                1        1.195035e+01
                2        1.147768e+01
                3        1.544747e+01
                4        1.735989e+01
                5        1.160256e+01
                6        1.550610e+01
                7        2.534706e+01
               …
               89        4.568771e-05
               90        3.513509e-05
               91        3.462404e-05
               92        3.977090e-05
               93        2.056327e-06
Successfully converged after 93 iterations.
```

```
Solving with diagonal preconditioner (M = D^-1)
Solver will continue until residual norm 5e-06
  Iteration Number  | Residual Norm
                0        5.000000e+00
                1        5.148771e+01
                2        4.341677e+01
                3        3.299794e+01
                4        1.074329e+01
                5        2.807501e+00
                6        1.739602e+00
                7        1.538450e+00
                8        4.079266e-01
                9        7.029972e-02
               10        2.436168e-02
               11        6.656054e-03
               12        1.284295e-03
               13        1.432453e-06
Successfully converged after 13 iterations.
```

Identity Preconditioner

- **Equivalent to no preconditioner**

1	7	0	0
0	2	8	0
5	0	2	9
0	6	0	4

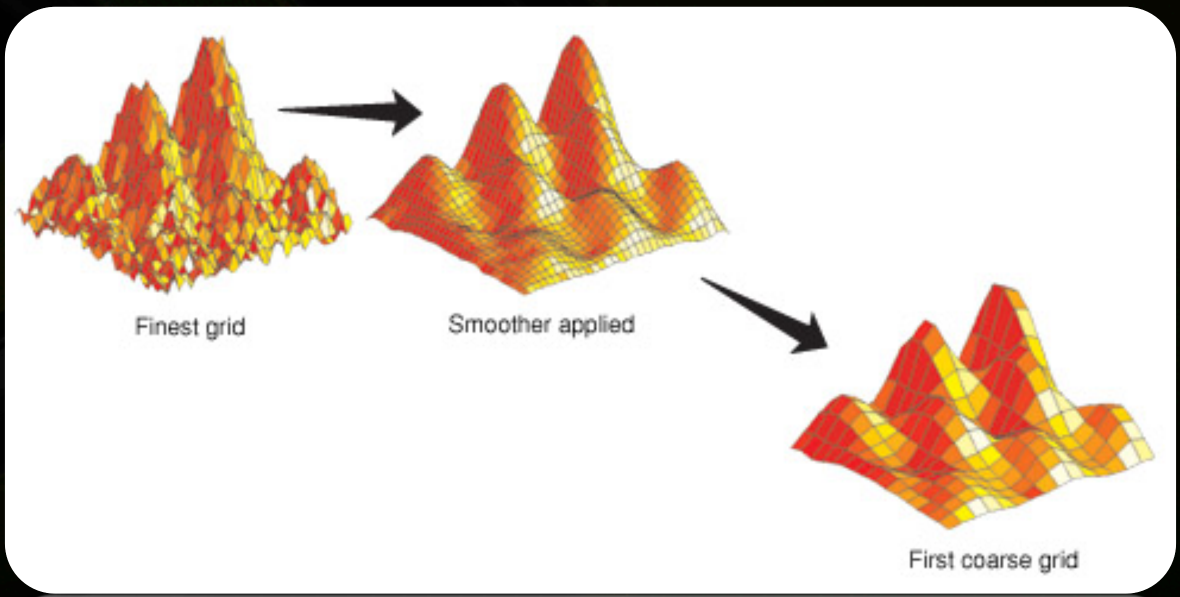
A

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

M

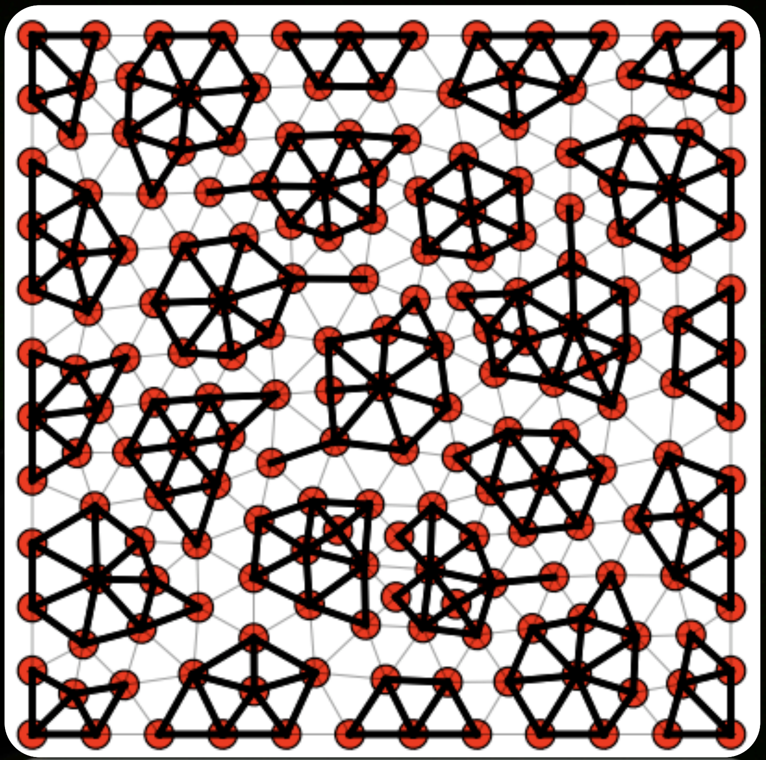
Multigrid

- Hierarchy of grids and transfer operators
  - Smooth error can be represented on coarse grid



Algebraic Multigrid

- Construct grids from matrix
  - No geometry
- Wide variety of methods
  - Classical AMG
  - Aggregation-based AMG



Algebraic Multigrid

```
#include <cusp/krylov/cg.h>
#include <cusp/precond/smoothed_aggregation.h>

...

// set stopping criteria
//  iteration_limit = 100
//  relative_tolerance = 1e-6
cusp::default_monitor<float> monitor(b, 100, 1e-6);

// setup preconditioner
cusp::precond::smoothed_aggregation<int, float, cusp::device_memory> M(A);

// solve A * x = b to default tolerance with preconditioned CG
cusp::krylov::cg(A, x, b, monitor, M);
```

Algebraic Multigrid

- Can be extremely effective

```
Solving with no preconditioner...
Iteration Number  | Residual Norm
               0        2.560000e+02
               1        2.039984e+03
               2        2.023926e+03
               3        2.056031e+03
               4        2.368466e+03
               5        2.289376e+03
               6        2.265133e+03
               7        2.325146e+03
        ...
             512        3.064404e-04
             513        2.943765e-04
             514        2.796355e-04
             515        2.690365e-04
             516        2.472412e-04
Successfully converged after 516 iterations.
```

```
Solving with smoothed aggregation preconditioner...
Iteration Number  | Residual Norm
               0        2.560000e+02
               1        4.373383e+03
               2        2.359818e+03
               3        1.211452e+03
               4        6.210880e+02
               5        3.169577e+02
               6        1.569394e+02
               7        8.138102e+01
         ...
              20        6.011106e-03
              21        2.555795e-03
              22        1.083367e-03
              23        4.799830e-04
              24        2.213949e-04
Successfully converged after 24 iterations.
```

Parallel Preconditioners

- Diagonal
- Multigrid
- Polynomial
- Approximate-Inverse
- Ongoing development in Cusp

References

Iterative Methods for Sparse Linear Systems
Yousef Saad
http://www-users.cs.umn.edu/~saad/IterMethBook_2ndEd.pdf (online)

A Multigrid Tutorial
William L. Briggs
https://computation.llnl.gov/casc/people/henson/mgtut/ps/mgtut.pdf