An Empirical Comparison of Graph Laplacian Solvers

Kevin Deweese\textsuperscript{1}  Erik Boman\textsuperscript{2}  John Gilbert\textsuperscript{1}

\textsuperscript{1}Department of Computer Science  
University of California, Santa Barbara

\textsuperscript{2}Scalable Algorithms Department  
Sandia National Laboratories

Algorithm Engineering and Experiments, 2016
Our focus: Solve the system of equations $Lx = b$ where $L$ is a graph Laplacian matrix.
Applications

Graphs with regular degree structure, 2D/3D meshes

- Finite element analysis
  - Electrical and thermal conductivity
  - Fluid flow modeling
- Image processing
  - Image segmentation, inpainting, regression, classification

Graphs with irregular degree, problems in network analysis

- Maximum flow problems
- Graph sparsification
- Spectral clustering
Applications

Graphs with regular degree structure, 2D/3D meshes
- Finite element analysis
  - Electrical and thermal conductivity
  - Fluid flow modeling
- Image processing
  - Image segmentation, inpainting, regression, classification

Graphs with irregular degree, problems in network analysis
- Maximum flow problems
- Graph sparsification
- Spectral clustering
Some applications use symmetric diagonally dominant (SDD) matrices
Slightly more general, allows for positive off-diagonal entries
Can be reduced to solving a Laplacian linear system
SDD Solvers With Good Asymptotic Complexity

- Linear times polylog. Spielman and Teng, 2006
- Nearly $m \log n$. Koutis, Miller, and Peng, 2011
- mostly theoretical results, few experiments
Our Goal

- Perform a comprehensive study of existing Laplacian solvers
- Select a set of test problems that are relevant/challenging
- Select performance metrics for evaluating current and future Laplacian solver performance
- Ongoing work, plan to update it with new solvers, new test problems
Test Graphs

- University of Florida Sparse Matrix Collection [Davis]
  - Irregular degree graphs
    - 10K edges to 4M edges
  - 2D/3D mesh-like graphs
    - 30K edges to 7M edges

- Block two-level Erdös Rényi (BTER) [Seshadhri et al.]
  - Designed to model web graphs with realistic degree distributions and clustering behavior

- Image segmentation graphs [Felzenszwalb and Huttenlocher]
  - Pixels=vertices, edge weights represent dissimilarity between pixel values
    - 300K edges to 7M edges
Solution Methods

- **Direct methods**
  - Solve in a finite number of operations
  - Cholesky factorization
  - Sometimes expensive in time and memory use

- **Iterative methods**
  - Form a sequence of improving approximations
  - Conjugate gradients (CG)
    - Convergence depends on matrix spectrum, bounded in terms of condition number $\kappa(A) = \frac{\lambda_n(A)}{\lambda_1(A)}$
    - Typically used with a preconditioner to improve the condition number

- **Multilevel**
  - Approximate solution on a coarser problem, occasionally correct on original
  - Form recursive hierarchy of approximations
Solution Methods all from [Trilinos]

- **Direct Solvers**
  - Cholesky factorization
    (Cholmod [Davis])
- **CG with single-level preconditioner**
  - Jacobi
  - Incomplete LU Factorization (ILU)
  - Spanning trees
- **CG with multi-level preconditioner**
  - Algebraic Multigrid (AMG)
Solution Methods all from [Trilinos]

- Direct Solvers
  - Cholesky factorization (Cholmod [Davis])
- CG with single-level preconditioner
  - Jacobi
  - Incomplete LU Factorization (ILU)
  - Spanning trees
- CG with multi-level preconditioner
  - Algebraic Multigrid (AMG)
Solution Methods all from [Trilinos]

- **Direct Solvers**
  - Cholesky factorization (Cholmod [Davis])
- **CG with single-level preconditioner**
  - Jacobi
  - Incomplete LU Factorization (ILU)
  - Spanning trees
- **CG with multi-level preconditioner**
  - Algebraic Multigrid (AMG)
Solution Methods all from [Trilinos]

- **Direct Solvers**
  - Cholesky factorization (Cholmod [Davis])

- **CG with single-level preconditioner**
  - Jacobi
  - Incomplete LU Factorization (ILU)
  - Spanning trees

- **CG with multi-level preconditioner**
  - Algebraic Multigrid (AMG)
Solution Methods all from [Trilinos]

- Direct Solvers
  - Cholesky factorization
    (Cholmod [Davis])
- CG with single-level preconditioner
  - Jacobi
  - Incomplete LU Factorization
    (ILU)
  - Spanning trees
- CG with multi-level preconditioner
  - Algebraic Multigrid (AMG)
Solution Methods all from [Trilinos]

- Direct Solvers
  - Cholesky factorization (Cholmod [Davis])
- CG with single-level preconditioner
  - Jacobi
  - Incomplete LU Factorization (ILU)
  - Spanning trees
- CG with multi-level preconditioner
  - Algebraic Multigrid (AMG)
Experimental Design

- $Lx = b$ solved on problems in 4 test sets
- $b$ randomly generated
- Solutions found to within residual tolerance of $10^{-9}$
- Mostly used default solver parameters
Performance Metrics

- Number of Iterations
- Setup Time (one time work)
- Per-solve Time (every time work)
- Total Time (Setup+Per-solve)
- Memory Usage
Setup + Per-solve (Irregular)

UF Irregular Graphs

BTER Graphs
Setup + Per-solve (Mesh-like)

UF Mesh-like Graphs

Image Segmentation Graphs

Fraction of problems within $\tau$ of best

Cholesky
Jacobi
ILU
Tree
Multilevel
Iterations (Irregular)

UF Irregular Graphs

BTER Graphs

K. Deweese, J. Gilbert, E. Boman
An Empirical Comparison of Graph Laplacian
ALENEX 2016 14 / 20
Iterations (Mesh-like)

UF Mesh-like Graphs

Image Segmentation Graphs

K. Deweese, J. Gilbert, E. Boman
An Empirical Comparison of Graph Laplacian Solvers
ALENEX 2016
Per-solve Time (Irregular)

UF Irregular Graphs

BTER Graphs

Cholesky
Jacobi
ILU
Tree
Multilevel
Per-solve Time (Mesh-like)

UF Mesh-like Graphs  Image Segmentation Graphs

K. Deweese, J. Gilbert, E. Boman  An Empirical Comparison of Graph Laplacian

ALENEX 2016  17 / 20
Summary of Results

- Relative solver performance is consistent within test sets, but very different between test sets.
- Multigrid does well on mesh-like problems; single-level preconditioners do well on irregular problems.
- BTER problems are easier for the iterative methods, more difficult for direct methods.
- The irregular problems are better conditioned, simple preconditioners like Jacobi do well.
Future Work

- Incorporate additional solvers
  - Multigrid methods designed for irregular graphs
  - Decide how to incorporate solvers outside Trilinos

- Add additional test problems

- Understand how graph structure -> condition number, solver behavior