SAND2014-18213PE

Fast Solvers for Graph Laplacians

Erik G. Boman

Sandia National Labs, NM

Sept. 19, 2014

Outline

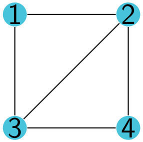
- Graph Laplacian
- Brief History: Combinatorial Solvers
- The New Simple Near-Optimal Solver by Kelner et al.
- Algebraic Interpretation
- Work in Progress

Graph Laplacian

- We consider the *combinatorial* graph Laplacian,

$$L(G) = D - A,$$

  where D is diagonal (the degrees) and A the adjacency matrix.
- Solving linear systems with the *normalized* graph Laplacian is equivalent (by symmetric scaling).



$$\begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix}$$

Solving Linear Systems

How to solve $Lx = b$?

- Sparse direct methods require too much memory.
- Iterations such as Jacobi or SGS converge but slowly.
- PCG with algebraic preconditioners is viable in many cases.
- A "piece of cake" for multigrid, right?
  - Multigrid/AMG optimal for meshes
  - Proofs need many assumptions
  - Irregular structure (scale-free, power-law) pose challenges
- Much work in CS theory over last decade.
  - Goal: Develop linear-time solver for *any* graph Laplacian

Combinatorial Preconditioners

Key idea: Use a sparser graph as a preconditioner.
We say H is a *spectral sparsifier* for G if H is sparser than G and

$$\alpha \frac{x^T L(H) x}{x^T x} \leq \frac{x^T L(G) x}{x^T x} \leq \beta \frac{x^T L(H) x}{x^T x} \quad \forall x$$

for some constants α, β (near one).

- We can use $L(H)$ as a *preconditioner* to accelerate any Krylov iterative method (eg. conjugate gradients).
- Pick H so $L(H)$ is "easy" to solve for. Example: spanning tree.
- May need to use *recursive* preconditioning via a *sequence* of sparsifiers.

Brief History

- **Vaidya ('91):** First proposed *spanning tree preconditioners*. Analysis for planar and non-planar graphs. Introduced *augmentation* (add select edges to tree). Not optimal, but good starting point.
- **Spielman & Teng ('04):** Seminal work showed *recursive graph sparsification* can give *near-linear time solver*. Very complicated. Later split into three papers, over 100 pages in total. Last one published SIMAX 2014.
- **Koutis, Miller & Peng ('13):** Simpler subgraph preconditioning using random sampling based on *stretch* to approximate *effective resistance*. Also near-linear.

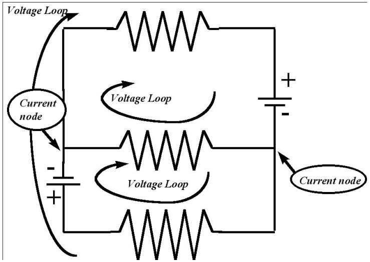
Note: Using *stretch* to measure the quality of spanning trees was first proposed by Boman (2001), not published but later used by the authors above.

Breakthrough: Kelner et al.

Kelner et al. (2013) developed a new, simpler Laplacian solver that

- converges in $O\left(m \log^2 n \log\log n \log(\epsilon^{-1})\right)$ time,
- is very simple ("proof fits on a single black-board"!),
- does not build upon the ST or KMP methods (weird),
- does not use preconditioning nor Krylov solvers,
- can potentially be practical.

Electrical Networks



The circuit is a graph, G. Find currents f on the edges and potentials (voltages, v) on the vertices. Each edge (i,j) has conductance $\frac{1}{r_{ij}}$. Ohm's Law gives:

$$L(G)v = b$$

where b are the external demands. Kirchhoff's laws useful, too.

Approach

- Solve for the current (flow), then derive the potentials (voltages).
- Any valid flow satisfies $f(e) = (v_i - v_j)/r_e$ for all edges e (Ohm).
- The potentials around a cycle in the network sum to zero (Kirchhoff). Thus,

$$\sum_{e \in C} f(e) r_e = 0$$

- Wish to satisfy condition above for every (simple) cycle.

Algorithm

- Find a cycle basis for the graph (e.g. a spanning tree) and a corresponding probability distribution over the cycles.
- Repeat until converged:
- Randomly sample a cycle C.
- If $\sum_{e \in C} f(e) r_e \neq 0$, add multiple of C to f to make it zero.

Idea: Find a violated constraint, fix it locally. Mathematically, this is a sequence of projections and corresponds to the *randomized Kaczmarz method*.

Kaczmarz's Method

Row projection method for solving $Ax = b$ (Kaczmarz, 1933). Let a_i be the ith row of A.

```
for $k = 0, 1, 2, \ldots$
    for $i = 1, 2, \ldots, n$
        $x^{k+1} = x^k + \frac{b_i - (a_i, x^k)}{\|a_i\|^2} a_i^T$
```

Convergence guaranteed but may be slow. Sensitive to ordering of A.

Strohmer & Vershynin '09: Pick rows randomly with probability proportional to $\|a_i\|$. Then

$$E\|x^k - x\|^2 \leq \left(1 - \frac{1}{\kappa(A)^2}\right)^k \|x^0 - x\|^2$$

The proof of Kelner's method uses this result.

Algebraic Interpretation

Let the voltages v be the *primal* variables and let the current flow f be the *dual* variables. Let U denote the vertex-edge incident matrix. Then the primal-dual system is

$$\begin{pmatrix} R & U^T \\ U & 0 \end{pmatrix} \begin{pmatrix} f \\ v \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

This is a *saddle-point* or *equilibrium* system, see Strang (SIREV 1988).
Usually no advantage solving a larger, indefinite system.
If we eliminate the dual variables, we get the primal system

$$Lv \equiv UR^{-1}U^T v = UR^{-1}b \equiv \hat{b},$$

where $L = UR^{-1}U^T$ is the graph Laplacian.
Hoewever, Kelner's method uses a *dual* approach.

Toledo's Dual Interpretation

For simplicity, suppose $R = I$ (no weights). We wish to solve

$$Lx = UU^T x = b$$

Given U, construct a basis N for its null space, $UN^T = 0$. Define

$$K = \begin{pmatrix} U \\ N \end{pmatrix}$$

which is square and full rank. Find a vector f such that

$$Kf = \begin{pmatrix} U \\ N \end{pmatrix} f = \begin{pmatrix} b \\ 0 \end{pmatrix}$$

Find x such that $U^T x = f$ (easy). Since $Uf = b$, we have

$$Uf = UU^T x = Lx = b$$

Kelner's method is randomized Kaczmarz on K (for certain N).

Practical Issues

To implement Kelner's method one needs:

- A cycle basis, typically given by a "nice" spanning tree.
    - Spanning tree makes it easy to find cycles. (Every non-tree edge defines a cycle.)
    - Low-stretch spanning tree used by Kelner, but tricky to compute.
    - Other spanning trees will work, but analysis need be modified.
- Data structures for fast cycle finding/updates.

Work in Progress

Collaboration with J. Gilbert and K. Deweese (UCSB).

- Have (inefficient) Matlab code. Hard to compare to other methods.
- Alternative cycle bases:
  - No fundamental reason for tree-based cycles.
  - We don't strictly need a basis: more cycles is OK.
  - Are there better cycle bases or generating sets? Empirically: Yes!
- Parallel version:
  - Can update edge-disjoint cycles simultaneously.
  - Cycles from spanning trees tend to overlap; need other basis.
  - Greedy "packing" of cycles problematic since probabilities will differ.
  - Asynchronous method possible, but changes convergence.

Conclusions

- The Kelner et al. solver is a novel and important method.
  - 34 citations in 9 months! (Google Scholar)
- Theoretically nearly-linear, but constants matter in practice . . .
- Simple enough it can be implemented (even by non-experts?).
- Lower barrier to entry into the field.
- Lots of practical issues still to be investigated.
- Opens up possibilities for follow-on work.

Acknowledgments

- Kevin Deweese (UCSB)
- John Gilbert (UCSB)
- Sivan Toledo (Tel-Aviv)