Journal of Computational Mathematics
Vol.33, No.2, 2015, 209–226.
http://www.global-sci.org/jcm
doi:10.4208/jcm.1412-m2014-0041

A CASCADIC MULTIGRID ALGORITHM FOR COMPUTING THE FIEDLER VECTOR OF GRAPH LAPLACIANS*

John C. Urschel and Jinchao Xu
Department of Mathematics, Penn State University, Pennsylvania, USA
Email: jcurschel@gmail.com, xu@math.psu.edu
Xiaozhe Hu
Department of Mathematics, Tufts University, Medford, MA 02155
Email: xiaozhe.hu@tufts.edu
Ludmil T. Zikatanov
Department of Mathematics, Penn State University, Pennsylvania, USA
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria
Email: ludmil@psu.edu

Abstract

In this paper, we develop a cascadic multigrid algorithm for fast computation of the Fiedler vector of a graph Laplacian, namely, the eigenvector corresponding to the second smallest eigenvalue. This vector has been found to have applications in fields such as graph partitioning and graph drawing. The algorithm is a purely algebraic approach based on a heavy edge coarsening scheme and pointwise smoothing for refinement. To gain theoretical insight, we also consider the related cascadic multigrid method in the geometric setting for elliptic eigenvalue problems and show its uniform convergence under certain assumptions. Numerical tests are presented for computing the Fiedler vector of several practical graphs, and numerical results show the efficiency and optimality of our proposed cascadic multigrid algorithm.

Mathematics subject classification: 65N55, 65N25.
Key words: Graph Laplacian; Cascadic Multigrid; Fiedler vector; Elliptic eigenvalue problems.

1. Introduction

Computation of the Fiedler vector of graph Laplacians has proven to be a relevant topic, and has found applications in areas such as graph partitioning and graph drawing [1]. There have been a number of techniques implemented for computation of the Fiedler vector, most notably by Barnard and Simon [2]. They implemented a multilevel coarsening strategy, using maximal independent sets and created a matching from them. For the refinement procedure, Rayleigh quotient iteration was used. We note that the term refinement refers to the smoothing process that occurs, and has a different meaning in the multigrid literature. Although at the time this was significantly faster than the standard recursive spectral bisection, it leaves room for improvement. The majority of the improvement has been in the form of coarsening algorithms. Better coarsening techniques, such as heavy edge matching (HEM), have been used more frequently, and have exhibited much shorter run times [3, 4].

* Received August 22, 2014 / Revised version received December 11, 2014 / Accepted December 22, 2014 / Published online March 13, 2015 /

For more general eigenproblems of symmetric positive definite matrices, techniques such as Jacobi-Davison [5] and the Locally Optimal Preconditioned Conjugate Gradient Method [6] (see also [7]) have been used and shown to give good approximations to eigenvalues and eigenvectors. These techniques can easily be extended to computing a Fiedler vector. Other eigensolvers are provided by setting an Algebraic MultiGrid (AMG) tuned specifically for graph Laplacians (see, e.g. Lean AMG [8]) as a preconditioner in the LOPCG Method.

In this paper, we introduce a new and fast coarsening algorithm, based on the conecpt of heavy edge matching, with a more aggressive coarsening procedure. For refinement, we implement a form of power iteration. For both our coarsening and refinement procedures we have created algorithms that are straightforward to implement. While heavy edge matching is complicated and tough to implement in high level programming languages, since it involves selecting an edge with heaviest weight between two unmatched vertices, heavy edge coarsening is significantly easier because we do not need to worry about whether a vertex has been aggregated or not. For the refinement procedure, power iteration does not require the inversion of a matrix, making its use much more straightforward than for Rayleigh quotient iteration, which requires some technique to approximately invert the matrix.

Based on these two improved components, we propose a cascadic multigrid (CMG) method to compute the Fiedler vector. The CMG method has been treated in the literature, most notably by Bornemann and Deuflhard [9,10], Braess, Deuflhard, and Lipnikov [11], and Shaidurov [12–14]. However, little has been done with respect to the elliptic eigenvalue problem. Our technique is a purely algebraic approach which only uses the given graph. Moreover, although the purely algebraic approach is technically difficult to analyze, we consider the CMG method for the elliptic eigenvalue problem in the geometric setting. Based on the standard smoothing property and approximation property, we show that the geometric CMG method converges uniformly for the model problem, which indirectly provides theoretical justification of the efficiency of the CMG method. This also shows the potential of our CMG method for solving other eigenvalue problems from different applications.

The remainder of the paper is organized as follows. In Section 2, we briefly review the Fiedler vector and introduce our cascadic multigrid method for computing the Fiedler vector of a graph Laplacian. The cascadic multigrid method for elliptic eigenvalue problems is proposed in Section 3 and its convergence analysis is also provided. Section 4 presents numerical experiments to support the theoretical results of CMG method for elliptic eigenvalue problems and demonstrate its efficiency for computing the Fiedler vector of some graph Laplacian problems from real applications. We conclude the paper in Section 5 by some general remarks on this work and proposed future work.

2. Cascadic MG Method for Computing the Fiedler Vector

We begin by formally introducing the concept of a graph Laplacian and Fiedler vector. We start with the concept of a graph. A weighted graph $G = (V, E, w)$ is said to be undirected if the edges have no orientation. A graph is a multigraph if $(i, i) \notin E$ for all $1 \leq i \leq |V|$ ($|V|$ is the number of vertices). For the remainder of this paper, we assume that all graphs are undirected and multigraphs.

We consider the task of representing a graph in matrix form. One of the most natural representations is through its Laplacian. The Laplacian of a graph is defined as follows:

Definition 2.1. *Let $G = (V, E, w)$ be a weighted graph. We define the Laplacian matrix of G,*