Journal of Computational and Applied Mathematics 234 (2010) 1027–1035

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Efficient geometric multigrid implementation for triangular grids

Francisco Gaspar, J.L. Gracia *, F.J. Lisbona, C. Rodrigo

Department of Applied Mathematics, University of Zaragoza, Spain

ARTICLE INFO

Article history:
Received 9 September 2008
Received in revised form 9 March 2009

MSC:
65N55
65N30

Keywords:
Geometric multigrid
Semi-structured grids
Finite element implementation
Local Fourier analysis

ABSTRACT

This paper deals with a stencil-based implementation of a geometric multigrid method on semi-structured triangular grids (triangulations obtained by regular refinement of an irregular coarse triangulation) for linear finite element methods. An efficient and elegant procedure to construct these stencils using a reference stencil associated to a canonical hexagon is proposed. Local Fourier Analysis (LFA) is applied to obtain asymptotic convergence estimates. Numerical experiments are presented to illustrate the efficiency of this geometric multigrid algorithm, which is based on a three-color smoother.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Multigrid methods [1–3] are among the most efficient numerical algorithms for solving the large algebraic linear equation systems arising from discretizations of partial differential equations. In geometric multigrids, a hierarchy of grids must be proposed. For an irregular domain, it is very common to apply a refinement process to an unstructured input grid, such as Bank's algorithm, used in the codes PLTMG [4] and KASKADE [5], obtaining a particular hierarchy of globally unstructured grids suitable for use with a geometric multigrid. A simpler approach to generating the nested grids consists in carrying out several steps of repeated regular refinement, for example by dividing each triangle into four congruent triangles [6].

An important step in the analysis of PDE problems using finite element methods (FEM) is the construction of the large sparse matrix A corresponding to the system of equations to be solved. The standard algorithm for computing matrix A is known as *assembly*: This matrix is computed by iterating over the elements of the mesh and adding from each element of the triangulation the local contribution to the global matrix A. For discretizations of problems defined on structured grids with constant coefficients, explicit assembly of the global matrix for the finite element method is not necessary, and the discrete operator can be implemented using stencil-based operations. For the previously described hierarchical grid, one stencil suffices to represent the discrete operator at nodes inside a triangle of the coarsest grid, and standard assembly process is only used on the coarsest grid. Therefore, this technique is used in this paper since it can be very efficient and is not subject to the same memory limitations as unstructured grid representation.

LFA (also called local mode analysis [7]) is a powerful tool for the quantitative analysis and design of efficient multigrid methods for general problems on rectangular grids. Recently, a generalization to structured triangular grids, which is based on an expression of the Fourier transform in new coordinate systems in space and frequency variables, has been proposed in [8]. In that paper some smoothers (Jacobi, Gauss–Seidel, three-color and block-line) have been analyzed and compared by LFA; the three-color smoother turning out to be the best choice for almost equilateral triangles.

* Corresponding author. Tel.: +34 976 762655; fax: +34 976 761886.
E-mail address: jlgracia@unizar.es (J.L. Gracia).

0377-0427/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.03.012

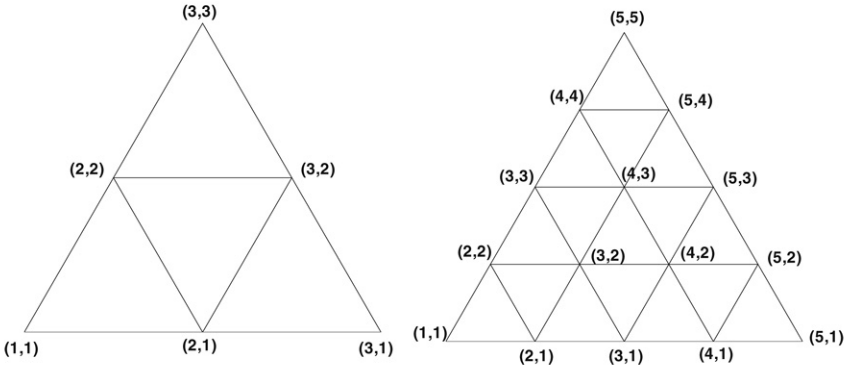

Fig. 1. Numeration of the nodes for one and two refinement levels.

In this paper an efficient implementation of geometric multigrid methods on semi-structured grids for linear finite element methods is described using a reaction–diffusion problem as a model. In Section 2, a suitable data structure is introduced; after that, we describe the discrete operator in a stencil-based form, and a procedure using a canonical stencil associated to a reference hexagon is proposed. The different components of the multigrid algorithm are also given. In Section 3, an LFA is applied to determine the efficiency of the proposed multigrid method from the convergence factors provided by the two-grid analysis. Finally, in Section 4 two numerical experiments illustrate the good performance of the method for an H-shaped domain, and it is shown that the ideas developed in this paper can be extended to systems of equations.

2. Description of the algorithm

The main features of this algorithm are described in this section. In the first place, we will consider a particular triangulation of the domain consisting in a semi-structured grid obtained by local regular refinement of an input unstructured grid. The semi-structured character of the grid allows use of low cost memory storage of the discrete operator based on stencil form. Such storage permits simpler implementation of the geometric multigrid method. The different multigrid components are described in the last subsection paying special attention to the relaxation process.

2.1. Semi-structured grids

Let $\mathcal{T}_0$ be a coarse triangulation of a bounded open polygonal domain Ω of $\mathbf{R}^2$, satisfying the usual admissibility assumption, i.e. the intersection of two different elements is either empty, a vertex, or a whole edge. This triangulation is assumed to be rough enough in order to fit the geometry of the domain. Once the coarse triangulation is given, each triangle is divided into four congruent triangles connecting the midpoints of their edges, and this is repeated until a mesh $\mathcal{T}_l$ is obtained with the desired fine scale to approximate the solution of the problem. This strategy generates a hierarchy of conforming meshes, $\mathcal{T}_0 \subset \mathcal{T}_1 \subset \cdots \subset \mathcal{T}_l$, where transfer operators between two consecutive grids can be defined geometrically.

As the number of neighbors of the vertices of the coarsest grid $\mathcal{T}_0$ is not fixed, the corresponding unknowns must be treated as unstructured data. Thus, two different types of data structure must be used, one of them totally unstructured, whereas the other is a hierarchical structure. For a refinement level i of a triangle of the coarsest grid, a local numeration with double index (n, m), $n = 1, \ldots, 2^i + 1$, $m = 1, \ldots, n$, is used in such a way that the indices of its vertices are $(1, 1)$, $(2^i + 1, 1)$, $(2^i + 1, 2^i + 1)$, as we can observe in Fig. 1 for one and two refinement levels. This way of numbering nodes is very convenient for identifying the neighboring nodes, which is crucial in performing the geometric multigrid method.

Due to the fact that the multigrid method uses a blockwise structure, there are several points in the algorithm, such as relaxation and residual calculation, where information from neighboring triangles must be transferred. To facilitate this communication, each triangle of the coarsest grid is augmented by an overlap-layer of so-called ghost nodes that surround it. To be more precise, each triangle receives the data corresponding to its own overlap region from its neighboring triangles of the coarsest grid (see Fig. 2b). The width of this overlap region is mainly determined by the extent of the stencil operators involved; in this case we use an overlap of one grid point (see Fig. 2a).

2.2. A stencil-based finite element implementation

Let us consider the model problem

$$-\Delta u + u = f, \quad \text{in } \Omega, u = 0, \text{ on } \Gamma, \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain with boundary Γ and, for simplicity of presentation, homogeneous Dirichlet boundary conditions are imposed. Let $\mathcal{T}_h$ be a triangulation in the hierarchy of conforming meshes $\mathcal{T}_0 \subset \mathcal{T}_1 \subset \cdots \subset \mathcal{T}_l$, defined in the

Download English Version:

https://daneshyari.com/en/article/4640483

Download Persian Version:

https://daneshyari.com/article/4640483

Daneshyari.com