Journal of Computational and Applied Mathematics 236 (2012) 3636–3646

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

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

An adaptive multiresolution method on dyadic grids: Application to transport equations

Douglas A. Castro [a,b,∗], Sônia M. Gomes [b], Jorge Stolfi [c]

[a] Universidade Federal do Tocantins, Caixa Postal 66, 77402-970, Gurupi TO, Brazil

[b] Universidade Estadual de Campinas, IMECC. Rua Sérgio Buarque de Holanda, 651. 13083-859 - Campinas SP, Brazil

[c] Universidade Estadual de Campinas, IC Caixa Postal 6176 13081-970 Campinas SP, Brazil

ARTICLE INFO

Article history:
Received 1 October 2010
Received in revised form 19 May 2011

Keywords:
Finite volumes
Multiresolution representation
Wavelets
Dyadic grids
Transport equation

ABSTRACT

We propose a modified adaptive multiresolution scheme for solving *d*-dimensional hyperbolic conservation laws which is based on cell-average discretization in dyadic grids. Adaptivity is obtained by interrupting the refinement at the locations where appropriate scale (wavelet) coefficients are sufficiently small. One important aspect of such a multiresolution representation is that we can use the same binary tree data structure for domains of any dimension. The tree structure allows us to succinctly represent the data and efficiently navigate through it. Dyadic grids also provide a more gradual refinement as compared with the traditional quad-trees (2D) or oct-trees (3D) that are commonly used for multiresolution analysis. We show some examples of adaptive binary tree representations, with significant savings in data storage when compared to quad-tree based schemes. As a test problem, we also consider this modified adaptive multiresolution method, using a dynamic binary tree data structure, applied to a transport equation in 2D domain, based on a second-order finite volume discretization.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The purpose of an adaptive grid refinement technique for partial differential equations (PDE) is to save computational resources while preserving the accuracy of the solution with respect to the uniform discretization in the finest scale level. Grid adaptation means that refined grids are used only where they are required, such as in regions where the solution exhibits localized strong features.

In this paper, we are concerned with adaptive multiresolution (MR) schemes. The principle is to use a multiresolution representation of the solution, with the MR coefficients being used as local regularity indicators. Thus, an adaptive grid can be introduced by a thresholding procedure, where only significant coefficients are retained. Discarding the small coefficients leads to coarse grids in regions where the solution is smooth, and the refinement occurs close to irregularities, where the coefficients are significant. This technique can lead to significant memory savings and accelerate considerably the simulation with respect to the discretization on the finest uniform mesh, without contaminating its accuracy. In combination with finite volume discretization and multiresolution analyses for cell averages, MR schemes have been successfully applied to conservation laws ([1–8]). For an overview on adaptive MR techniques, we refer to the books of Cohen [9] and Müller [3], and also to the review paper Schneider and Vasilyev [10], where the authors not only revisit the adaptive wavelet methodologies in fluid dynamics, but also give some perspectives for modeling and computing industrially relevant flows.

∗ Corresponding author at: Universidade Federal do Tocantins, Caixa Postal 66, 77402-970, Gurupi TO, Brazil.
E-mail addresses: douglas@ime.unicamp.br, dacastro@uft.edu.br (D.A. Castro), soniag@ime.unicamp.br (S.M. Gomes), stolfi@ic.unicamp.br (J. Stolfi).

0377-0427/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2011.05.044

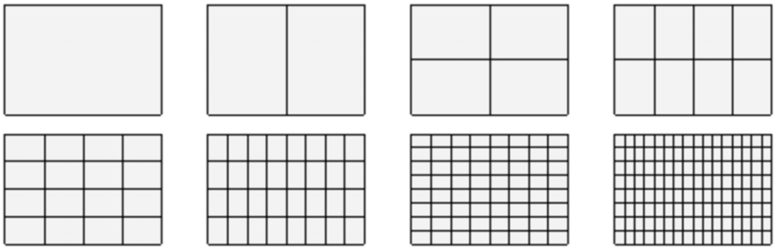

Fig. 1. The cells in the first 8 levels of a two-dimensional dyadic multigrid.

In order to support MR techniques, a memory-efficient data structure, with fast access to the stored data, is required. Usually, dynamic tree data structures or hash tables are used. Our purpose is to present an MR scheme for d-dimensional hyperbolic conservation laws, which is based on cell-average discretization in dyadic grids [11]. A dyadic grid is a hierarchy of meshes where a cell at a certain level is partitioned into two equal children at the next refined level by a hyperplane perpendicular to one of the coordinate axes, which varies cyclically from level to level. Adaptivity is obtained by interrupting the refinement at the locations where appropriate scale (wavelet) coefficients are sufficiently small. One important aspect of such an MR representation is that we can use the same binary tree data structure for domains of any dimension. The tree structure allows us to succinctly represent the data and efficiently navigate through it. Dyadic grids also provide a more gradual refinement as compared with the traditional quad-trees (2D) or oct-trees (3D) that are commonly used for multiresolution analyses.

The present text is organized as follows. In Section 2, we present a brief discussion about dyadic grids, and their representations. Section 3 is dedicated to a multiresolution representation for cell averages based on dyadic grids. Then, some examples showing the efficiency of dyadic grids in 2D, as compared to quad-grids, are discussed. In Section 4, we describe the adaptive MR finite volume scheme, giving a description of the algorithm. For the reference scheme in uniform grid, we use a finite volume discretization in space and an explicit second-order Runge–Kutta scheme in time. For the numerical flux, we use Roe's scheme with a second-order essentially non-oscillatory (ENO) interpolation, which gives a second-order accurate scheme in space. In Section 5, we present the results of the adaptive MR scheme on dyadic grids. As a test problem, we consider the transport equation in 2D domain. Our tests show significant savings in data storage and CPU time when compared to the reference scheme in uniform grid at the finest scale level.

2. Dyadic grids

In this section, we present the basic concepts related to dyadic grids. More details can be seen at Cardoso et al. [11,12].

We consider the coordinate axes of $\mathbb{R}^d$ numbered from 0 to $d-1$, so that a vector $x \in \mathbb{R}^d$ has coordinates $x_0, x_1, \dots, x_{d-1}$.

A *k-dimensional box* (or k-box for short) of $\mathbb{R}^d$ is a Cartesian product of d subsets of $\mathbb{R}$, $I_0 \times I_1 \times \cdots \times I_{d-1}$, where k of the sets I_i are bounded open intervals $(a_i, b_i) \subseteq \mathbb{R}$, and the remaining $d-k$ are singleton sets $\{x_i\}$ where $x_i \in \mathbb{R}$. Thus, for example, a zero-dimensional box is a point of $\mathbb{R}^d$; a one-dimensional box is a line segment parallel to some coordinate axis; a two-dimensional box is a rectangle with sides parallel to two axes; and so on.

The *facets* of a k-box are the $(k-1)$-boxes obtained by replacing any open interval (a_i, b_i) in the Cartesian product by a singleton set, either $\{a_i\}$ or $\{b_i\}$. Thus a k-box has $2k$ facets. The *faces* of a k-box are recursively defined as the box itself and the faces of its facets; namely 3^k boxes in total.

2.1. Toroidal d-space

Let R be a fixed d-box in $\mathbb{R}^d$, which we call the *root cell*; namely $R = (a_0, b_0) \times (a_1, b_1) \times \cdots \times (a_{d-1}, b_{d-1})$. We say that two points $p, q \in \mathbb{R}^d$ are *equivalent*, written $p \equiv q$, if and only if $p_i - q_i$ is an integer multiple of the box width $b_i - a_i$, for every coordinate index $i \in \{0, \dots, d-1\}$. We define the *d-torus*, denoted here by $\mathbb{T}^d$, as the quotient of $\mathbb{R}^d$ by the equivalence relation $\equiv$.

2.2. Dyadic grids

A *dyadic multiscale grid* (or *multigrid*) is a collection $\mathcal{G}$ of boxes which are subsets of a toroidal space $\mathbb{T}^d$. These subsets (the *grid elements*) include the root cell and all its faces, as well as every box that is obtained from them by a series of finite dyadic bisections, to be defined below. Dyadic grid elements with dimension $k = d$ are called *cells*. Elements with dimension $k = 0, 1, 2, 3$ are named, respectively, *vertices*, *edges*, *faces*, and *blocks* of grid.

Each element has a *depth*, which is the number of bisection steps needed to produce it from one of those root boxes. The *level* ℓ of $\mathcal{G}$ is the set of all elements of $\mathcal{G}$ with depth ℓ, denoted by $\mathcal{G}_\ell$.

A *dyadic bisection* consists in splitting a k-box $c \in \mathcal{G}$, with $k > 0$, into two congruent k-boxes (the *children* of c) and a $(k-1)$-box which is a facet of both. If c has depth ℓ, the split is performed by a hyperplane orthogonal to the coordinate axis number $\ell \bmod d$. That is, the orientation of the split alternates cyclically as the depth increases. See Fig. 1 (for $d = 2$) and Fig. 2 (for $d = 3$).

Download English Version:

https://daneshyari.com/en/article/6422999

Download Persian Version:

https://daneshyari.com/article/6422999

Daneshyari.com