Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computational Physics 212 (2006) 454–472

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Spectral (finite) volume method for conservation laws on unstructured grids V: Extension to three-dimensional systems

Yen Liu [a,*], Marcel Vinokur [b], Z.J. Wang [c]

[a] *NASA Ames Research Center, Moffett Field, CA 94035, United States*
[b] *Eloret Corp., Sunnyvale, CA 94087, United States*
[c] *Department of Aerospace Engineering, Iowa State University, Ames, IA 50011, United States*

Received 16 November 2004; received in revised form 23 June 2005; accepted 24 June 2005
Available online 24 August 2005

Abstract

In this paper, the fifth in a series, the high-order spectral finite-volume, or spectral volume (SV) method for unstructured grids is extended to three dimensions. Limitations of conventional structured and unstructured methods are first reviewed. The spectral finite-volume method for generalized conservation laws is then described. It is shown that if all grid cells are partitioned into structured sub-cells in a similar manner, the discretizations become universal, and are reduced to the same weighted sum of unknowns involving just a few simple adds and multiplies. Important aspects of the data structure and its effects on communication and the optimum use of cache memory are discussed. Previously defined one-parameter partitions of the SV in 2D are extended to multiple parameters and then used to construct 3D partitions. Numerical solutions of plane electromagnetic waves incident on perfectly conducting circular cylinders and spheres are presented and compared with the exact solution to demonstrate the capability of the method. Excellent agreement has been found. Computation timings show that the new method is more efficient than conventional structured and unstructured methods.

MRI: 65M60; 65M70; 35L65

Keywords: High-order; Unstructured grid; Spectral volume; 3D Systems of conservation laws; Maxwell equations

DOI of original article: 10.1016/j.jcp.2003.09.012.

[*] Corresponding author. Tel.: +1 650 604 6667.
E-mail address: yen.liu@nasa.gov (Y. Liu).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.06.024

1. Introduction

We continue the development of the spectral finite-volume, or spectral volume (SV) method for hyperbolic conservation laws on unstructured grids following the basic formulation [23], development for two-dimensional scalar conservation laws [24], one-dimensional systems and partition optimization [25], and two-dimensional systems [26]. The ultimate goal of this research is to pursue a numerical method for conservation laws which has all of the following properties: (a) conservative, (b) high-order accurate, (c) geometrically flexible, (d) computationally efficient, and (e) simply formulated. Such a method is sought after in many "real world" problems with complex geometries in computational aeroacoustics (CAA), large-eddy-simulation (LES) and direct numerical simulation (DNS) of turbulence, and computational electromagnetics (CEM), to name just a few. Preliminary demonstrations have indeed shown that the SV method has many advantages comparing with the current state-of-the-art numerical methods such as the high-order *k*-exact finite volume (FV) method [2,8,13], essentially non-oscillatory (ENO) and weighted ENO (WENO) methods [1,11,12,21,22], and the discontinuous Galerkin (DG) method [4–7]. For example, the SV method is more efficient than the *k*-exact and WENO method (with the same number of unknowns) because the reconstruction is universal for all cells. The SV method has higher resolution for discontinuities than the DG method because of the sub-element resolution in the SV method enables limiters to be designed for the sub-cells (or control volumes).

Ultimately, the SV method is a Godunov-type finite-volume method [9], which has been under development for several decades, and has become the-state-of-the-art for the numerical solution of hyperbolic conservation laws. For a more detailed review of the literature on the Godunov-type method, refer to [23], and the references therein. What distinguishes the SV method from the *k*-exact FV method is the data reconstruction. Instead of using a large stencil of neighboring cells to perform a high-order polynomial reconstruction, the stencil is constructed by partitioning each grid cell, called a spectral volume (SV), into a set of "structured" sub-cells, called control volumes (CVs). The unknowns are the cell averages over the CVs. One can show that if all the SV cells are partitioned into polygonal or polyhedral CV sub-cells in a geometrically similar manner, the reconstructions for all the SVs become *universal*, irrespective of their shapes, sizes, orientations, or locations. It follows that the discretization is reduced to a weighted sum of unknowns involving just a few simple adds and multiplies, and those weights are universal and can be pre-determined once and for all. Numerical tests with scalar and system conservation laws in both one dimension and two dimensions have verified that the SV method is indeed highly accurate, conservative, efficient, geometrically flexible, and simple [24].

In this paper, we further extend the SV method to three-dimensional systems for unstructured grids. In Section 2, we first summarize the limitations of conventional finite-difference and finite-volume methods, for both structured and unstructured grids. We then describe the basic formulation of the spectral finite-volume method in Section 3. The details of an efficient code to solve three-dimensional problems for any order of accuracy are then presented. Important aspects of the data structure and its effects on communication and the optimum use of the cache memory are discussed. The most critical part of the SV method is the partitioning of the SV into CVs. In Section 4, we present the partitioning of triangular SVs into polygonal CVs up to quartic reconstructions and tetrahedral SVs into polyhedral CVs up to cubic reconstructions with one or more free parameters. The Lebesgue constant of the reconstruction matrix is used as an indicator to measure the quality of the partition. Numerical examples for a wave propagation problem in two and three dimensions are tested and compared with the exact solution in Section 5. In addition, computation timings of the SV method are compared with those of conventional structured and unstructured FV methods. Finally, some concluding remarks and recommendations for future study are summarized in Section 6.

2. Limitations of conventional methods

2.1. Finite-difference methods

The most widely used method is the finite difference method employing a body-fitted curvilinear coordinate system. The limitations for high order of accuracy implementation are:

a. The spatial differencing is essentially one-dimensional, carried out along coordinate directions. Thus a large number of data points near the unknown to be updated are ignored. The large stencil has to be modified near boundaries, where one-sided formulas are often necessary.
b. The metric terms are evaluated by numerically differencing the grid point coordinates. Since the methods used in numerical grid generation are mostly only second-order accurate, the overall accuracy of the solution can be severely degraded if the grid is not sufficiently smooth. This is particularly true in highly stretched areas, or near corners or boundaries with very high curvature.
c. The unknowns are values at grid points. While the differencing can be performed in a numerically "conservative" manner, the true integral conservation laws can only be satisfied to second-order accuracy.
d. A single, structured grid is not feasible for complex geometries. Calculations must be carried out over a set of patched or overlapping grids. At interface boundaries between patches, or in the overlap regions, the high accuracy is generally degraded, unless one uses sophisticated interface algorithms.

2.2. Finite-volume methods for structured grids

Finite-volume methods are often employed to overcome limitations b and c above. The unknowns are cell averages over quadrilaterals (2D) or hexahedra (3D). A high-order reconstruction in terms of neighboring unknowns is used to obtain values at cell boundaries, which may be modified by appropriate limiters where necessary. These are used to calculate the flux, using (approximate) Riemann solvers. In practice, the method is subject to the same limitations as the finite difference method.

a. The reconstruction is still done one-dimensionally along coordinate directions.
b. The surface area vectors can be exactly calculated in terms of the cell vertices. But the flux integral is approximated by a one-point quadrature at the computational face center, which is equal to the face centroid only to second order. For a non-planar face in 3D, a face centroid which lies on the face may not even exist.
c. While the cell volume can be precisely calculated, in order to obtain values at cell boundaries, the unknowns are implicitly assigned to a computational cell center, which is equal to the cell centroid only to second order.
d. If the grid is very unsmooth, and highly curved, even the formal second-order accuracy is reduced to first order.

2.3. Finite-volume methods for unstructured grids

The unknowns are normally cell averages over triangles (2D) or tetrahedra (3D), but can generally be over polygons (2D) or polyhedra (3D). A reconstruction of any desired order of accuracy is obtained in terms of unknowns at an appropriate number of neighboring cells in all directions. The flux integral for each face is evaluated using a quadrature approximation of the same order of accuracy as the reconstruction.

When applied to polyhedra, a non-planar face must first be subdivided into triangular facets. The flux at each quadrature point is obtained using the reconstructed solution for the two cells sharing that face. In principle, one can in this manner obtain a numerical solution of any desired order of accuracy. In practice, this method has severe computational limitations.

a. When carrying out the reconstruction, it is often difficult to obtain a non-singular stencil. In general, one is faced with an overdetermined problem, which requires a least-squares inversion. For high orders of accuracy, the number of cells, and thus the size of the matrix to be inverted, becomes prohibitively large in three dimensions.
b. Each cell requires a different reconstruction stencil. If the inversion coefficients are stored, the memory requirements become prohibitive for 3D. On the other hand, repeating the inversion for every cell at each time step would require impractically large CPU times.
c. Due to the unstructured nature of the data in physical space, the data from neighboring cells involved in the computation can be far apart in memory. This would hamper the efficiency of the code due to data gathering and scattering.

3. The spectral finite-volume method

3.1. General description

The main motivation behind the spectral finite-volume method is to find a simple way to obtain a single non-singular stencil that can be applied to all the cells in an unstructured grid. We start with a relatively coarse unstructured grid of cells, triangles in 2D and tetrahedra in 3D, called spectral volumes (SVs). Each SV is then partitioned into a number of "structured" sub-cells, called control volumes (CVs), that support a polynomial expansion of a desired degree of precision. The unknowns are now the cell averages over the CVs. The partition has a high degree of structure, making use of all the symmetries of the simplex geometry. The CVs can be polygons or polyhedra. For 3D, they can have non-planar faces, which must be subdivided into planar facets in order to perform the required integrations. All the SVs are partitioned in a geometrically similar manner. We thus obtain a single, universal reconstruction for all SVs. Due to the symmetry of the partition, only a few distinct coefficients appear in the expansion in terms of the CV unknowns.

A CV face that lies on an SV boundary will have a discontinuity on its two sides. A Riemann solver is then necessary to compute the flux on that face. If the flux is a linear function of the unknowns, the flux integration can be performed analytically without invoking quadratures [13], and the result expressed as a weighted sum of all the CV unknowns in the two SVs. The weights for each type of CV face are universal numbers that are pre-calculated once and read in as input to the program. If the flux is a non-linear function of the unknowns, a quadrature approximation of the appropriate degree of precision is required [15]. The conservative variable on one side of a quadrature point can again be expressed as a weighted sum of the CV unknowns in the SV on that side. Since the quadrature points belong to just a few symmetry groups, the total number of distinct weights that need to be stored is relatively small. In order to suppress spurious numerical oscillations, it may be necessary to modify the reconstructed solution using appropriate limiters [20] or filters.

The reconstruction within each spectral volume is continuous. Therefore, a linear flux over a CV face that lies in the interior of a SV can be evaluated directly, and the weights for each type of facet can be stored. For a non-linear flux, a similar procedure can be carried out for each quadrature point. Again a modification involving limiters or filters may be required in certain regions.

3.2. Details of the spectral finite-volume method

We present further details of the formulation for a generalized conservation law. We employ a vector notation for brevity. The most general form of a conservation law can be written as

$$\frac{\partial u}{\partial t} + \nabla * F = 0, \tag{1}$$

where the conservative variable u can be a scalar or a vector, and the generalized flux F can be a scalar, vector, or tensor. The term $\nabla{*}F$ represents the divergence, curl, or gradient of F, depending on the physical definition of u. Integrating (1) over each CV, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_{j,i}} u \,\mathrm{d}V + \sum_{k=1}^{K} \int_{S_{k,j,i}} \mathrm{d}\mathbf{S} * F = 0, \tag{2}$$

where $V_{j,i}$ is the volume of the jth CV in the ith SV, and $S_{k,j,i}$ is the area of planar facet k bounding $V_{j,i}$. (In 2D, each facet is actually a line segment.) The unknowns are the volume averages of u, defined as

$$\bar{u}_{j,i} = \frac{1}{V_{j,i}} \int_{V_{j,i}} u \,\mathrm{d}V. \tag{3}$$

The partitioning of each SV into CVs depends on the choice of basis functions for the reconstruction. For a complete polynomial basis, a reconstruction of degree of precision n requires a partition into (at least) N CVs, where

$$N = \begin{cases} (n+1) & \text{1D}, \\ (n+1)(n+2)/2 & \text{2D}, \\ (n+1)(n+2)(n+3)/6 & \text{3D}. \end{cases} \tag{4}$$

Note that the order of accuracy of the SV method is one order higher than the reconstruction degree of precision. In the present work, we partition the SV into N CVs, so that the reconstruction involves only the inversion of a square matrix. The partitions may involve several parameters. The optimum choice of parameters for each degree of precision can be determined by minimizing the Lebesgue constant [3] of the reconstruction matrix or similar convergence criteria. This requires further research and shall be reported in a future publication. In this paper, we just use the Lebesgue constant as an indicator to measure the quality of the partition.

In order to reconstruct u within each SV, we introduce a set of complete polynomials of nth degree of precision $\phi_l(\mathbf{r})$ and expand $u_i(\mathbf{r})$ in the ith SV as

$$u_i(\mathbf{r}) = \sum_{l=1}^{N} c_{l,i}\phi_l(\mathbf{r}). \tag{5}$$

Eq. (3) can then be written as

$$\bar{u}_{j,i} = \sum_{l=1}^{N} R_{jl,i} c_{l,i}, \tag{6}$$

where

$$R_{jl,i} = \frac{1}{V_{j,i}} \int_{V_{j,i}} \phi_l(\mathbf{r}) \,\mathrm{d}V. \tag{7}$$

The exact integrations of polynomials over arbitrary polygons or polyhedra in terms of the coordinates of vertices can be found in [15]. The elimination of the coefficients $c_{l,i}$ in Eqs. (5) and (6) can be shown succinctly using matrix algebra. Let c_i, $\bar{\mathsf{u}}_i$, and $\phi(r)$ be algebraic vectors with components $c_{l,i}$, $\bar{u}_{j,i}$, and $\phi_l(\mathbf{r})$, respectively, and let R_i be the matrix with elements $R_{jl,i}$. Eqs. (5) and (6) can then be written as

$$u_i(\mathbf{r}) = \phi^{\mathrm{T}}(\mathbf{r})\mathsf{c}_i \tag{8}$$

and

$$\bar{\mathsf{u}}_i = \mathsf{R}_i\mathsf{c}_i. \tag{9}$$

Eliminating c_i, we obtain the cardinal form

$$u_i(\mathbf{r}) = \mathsf{L}_i(\mathbf{r})\bar{\mathsf{u}}_i, \tag{10}$$

where

$$\mathsf{L}_i(\mathbf{r}) = \phi^{\mathrm{T}}(\mathbf{r})\mathsf{R}_i^{-1}. \tag{11}$$

Although the matrix R can be ill-conditioned, we use *Mathematica* [27] to obtain its inverse analytically. The functions $L_{j,i}(r)$, which are the components of the algebraic row vector $\mathsf{L}_i(r)$, are known as shape functions or cardinal basis functions. In expanded form, Eq. (10) can be written as

$$u_i(\mathbf{r}) = \sum_{j=1}^{N} L_{j,i}(\mathbf{r})\bar{u}_{j,i}. \tag{12}$$

For a facet k on an interior CV face, if F is a linear function of u, the flux integral in (2) can be evaluated by simply integrating u over that facet. Substituting expression (12), we can write the result as a weighted sum of the CV unknowns

$$\int_{S_{k,i}} u\,\mathrm{d}S = S_{k,i}\sum_{j=1}^{N} m_{k,j}\bar{u}_{j,i}. \tag{13}$$

Here

$$m_{k,j} = \frac{1}{S_{k,i}}\int_{S_{k,i}} L_{j,i}(r)\,\mathrm{d}S, \tag{14}$$

the surface integrals of the shape functions per unit area, are *universal*, irrespective of the SVs. There are only a few of these coefficients for each partition, which can be calculated exactly and stored in advance. For non-linear flux functions, the flux integral is evaluated by an nth degree of precision quadrature approximation of the form

$$\int_{S_{k,i}} \mathrm{d}\mathbf{S} * F = S_{k,i}\sum_{q} w_q\mathbf{n} * F(u_i(\mathbf{r}_q)), \tag{15}$$

where the w_q are known quadrature weights. Using (12), we can evaluate $u_i(\mathbf{r}_q)$ as a weighted sum of the CV unknowns

$$u_i(\mathbf{r}_q) = \sum_{j} l_{q,j}\bar{u}_{j,i}. \tag{16}$$

These weights

$$l_{q,j} = L_{j,i}(\mathbf{r}_q) \tag{17}$$

are the functional values of the shape functions at the quadrature point, which are also *universal*, irrespective of the SVs. There are also only a few of these coefficients, which can be calculated exactly and stored in advance.

For a CV face on an SV boundary, since u may be discontinuous, the flux is replaced by a Riemann flux of the form [13,16–18]

$$\mathbf{n} * F(u(\mathbf{r})) = F_{\text{Riem}}(u_L(\mathbf{r}), u_R(\mathbf{r}), \mathbf{n}). \tag{18}$$

For linear flux functions, using Eq. (13), the surface integral of Eq. (18) can be expressed as a weighted sum of the $\bar{u}$ in both SVs sharing the face. For non-linear flux functions, $\mathbf{n} * F$ in Eq. (15) is replaced by Eq. (18) with u_L and u_R evaluated at quadrature points using Eq. (16).

3.3. Data structure

There are several aspects of the data structure which can lead to a very efficient parallelizable code. The global grid data consists of face numberings, vertex numberings and locations, and cell numberings. The topology is specified by listing for each face its vertex numbers, in an order indicating its orientation, and the two adjacent cell numbers. In order to make use of the universal nature of the partitioning, all global cells are mapped into a *single* standard SV. Each global face can have three possible orientations in the standard SV for 2D, and 12 for 3D. For each SV partition, the local CV connectivities are predetermined. This information and the corresponding weights $m_{k,j}$ or $l_{q,j}$ are read in as input to the program. We thus can write a single code valid for 2D or 3D, and any desired order of accuracy.

There is an aspect inherent in the spectral finite-volume method that permits an optimum use of cache memory, resulting in great computational efficiency on modern computers. Since all unknowns in a single SV cell are packed together, when performing calculations for a given cell, all the data required from the cell is found contiguous in memory. Since data from at most two SVs is involved in any single computation, data communication between the CPU and memory is minimized. Furthermore, the weights never change, and always stay in *L1* cache during the flux computations. All other needed data may also fit into *L1* cache. This results in great reductions in memory access.

4. Partitions of the spectral volume

The most critical part of the SV method is the partitioning of the SV into CVs. Based on the approximation theory, it can be established that the Lebesgue constant [3]

$$L = \max_{\mathbf{r} \in SV_i} \sum_j |L_{j,i}(\mathbf{r})| \tag{19}$$

can be used to decide the quality of the partitions based on the uniform norm. It was shown in [24] that a smaller Lebesgue constant may give a better interpolation polynomial. Therefore, the problem becomes finding the partition with the smallest Lebesgue constant. Partitions for the 2D case with one parameter up to degree of precision 3 were presented in [24]. For the quadratic partition, a practical minimum value for the Lebesgue constant does not exist. We therefore consider more general cases of multiple-parameter partitions. We require that all the CVs are symmetric with respective to all symmetries of the triangle or tetrahedron. The partitions must also result in a non-singular matrix R_i as defined by Eq. (7).

4.1. Partitionings of a triangular SV

Since the partitioning of tetrahedra makes use of the partitioning of the triangular faces, we first discuss the partitioning of a triangular SV. For the linear partition there is no free parameter involved. In [24], we presented two possible partitions. Here we use the one, shown in Fig. 1A, with the smaller Lebesgue constant. Due to symmetry, one only needs to precompute and store weights for one boundary face and one

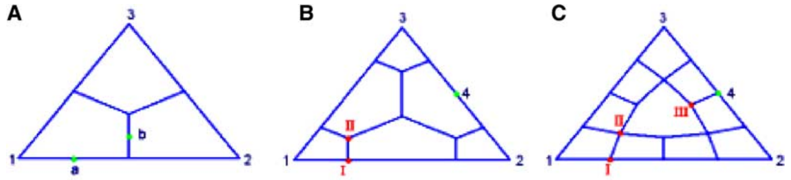

Fig. 1. Two-dimensional SV partitions: (A) linear partition, (B) quadratic partition, (C) cubic partition.

interior face, a total of six coefficients. Shown in Fig. 1A, points a and b are the centroids for faces a and b, respectively. Since u is linear, weight coefficients in Eqs. (6) and (8) have the same numerical values. We can also show that

$$u_a = \frac{1}{S_a}\int_{S_a} u \, \mathrm{d}S = \frac{20}{15}\bar{u}_1 + \frac{2}{15}\bar{u}_2 - \frac{7}{15}\bar{u}_3 \tag{20a}$$

and

$$u_b = \frac{1}{S_b}\int_{S_b} u \, \mathrm{d}S = \frac{8}{15}\bar{u}_1 + \frac{8}{15}\bar{u}_2 - \frac{1}{15}\bar{u}_3. \tag{20b}$$

Here $\bar{u}_i$ denotes the volume average for the CV whose vertices include the SV vertex i. The weight coefficients for other faces or quadrature points are simply the permutations of the constants in the above equations.

For the quadratic partition, we now consider a two-parameter partitioning. As shown in Fig. 1B, the parameter α is defined as the fraction of the distance of vertex I from vertices 1 to 2, while the parameter β is the fraction of the distance of vertex II from vertex 1 to point 4, the mid-point of edge 2–3. The special case $\beta = 2/3$ gives the one-parameter partition of [24], while for $\beta = \alpha$ the corner CVs reduces to triangles. The choice $\alpha = 1/4$ for this latter case gives a Lebesque constant of 3.6, compared to the value of 8 for the same α in the one-parameter partition. We will use the values $\alpha = \beta = 1/4$ in our calculations. The number of weight coefficients $m_{k,j}$ and $l_{q,j}$ needed to be stored are 24 and 42, corresponding to two boundary and two interior faces, and seven quadrature points, respectively.

For the cubic partition, shown in Fig. 1C, a third parameter γ is defined as the fraction of the distance of vertex III from vertices 4 to 1. The special case $\beta = 2\alpha$ and $\gamma = \alpha$ gives the one-parameter partition used in [24], while for $\gamma = \beta/2$ the central CV reduces to a triangle. For the one-parameter partition, it was found in [24] that the Lebesgue constant is near the minimum when $\alpha = 1/15$. Varying β and γ from their one-parameter values reduces the Lebesgue constant only by a few percent. In our calculations, we will therefore use the one-parameter partition. The number of weight coefficients $m_{k,j}$ and $l_{q,j}$ needed to be stored are 50 and 100, corresponding to 2 boundary and 3 interior faces, and 10 quadrature points, respectively.

4.2. Partitionings of a tetrahedral SV

For partitions with degrees of precision up to three, all the CVs have at least one face on the SV boundary. There are no CVs in the interior of the SV. If no additional interior parameters are introduced, all the CVs then consist of the vertices of the 2D CVs for each face of the SV connected to the SV centroid with straight edges. From Eq. (4) it follows that a linear partition consists of four CVs. Since a tetrahedron has four vertices and four faces, two types of partitions are possible, one type associated with the vertices and the other associated with the faces. These are shown in Figs. 2A and B. There is no free parameter involved in these partitions. The vertex-based CVs in partition 2A are hexahedra obtaining by connecting each face centroid to its three mid-edges with straight lines and connecting the SV centroid to the four face centroids

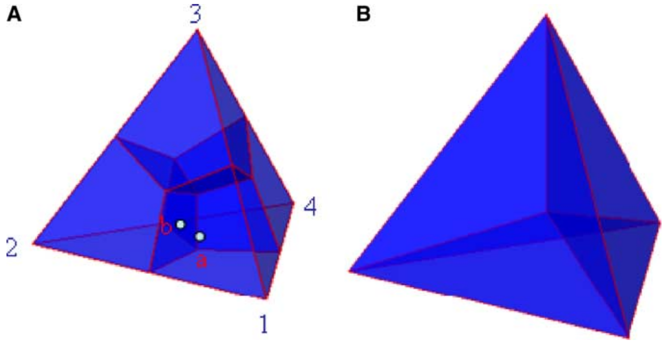

Fig. 2. Three-dimensional linear partitions: (A) vertex-based partition, (B) face-based partition.

with straight linnes. The faces of the hexahedra are all planar quadrilaterals. There are total of 12 boundary CV faces and 6 interior CV faces. The face-based CVs in partition 2B are tetrahedral obtained by connecting the three vertices of each face to the SV centroid with straight lines. There are total of four boundary CV faces and six interior CV faces. The Lebesgue constants are 95/26 and 13/2 for these linear partitions 2A and 2B, respectively. While the vertex-based partition has a smaller Lebesgue constant, it requires three times as many Riemann solver calls on each SV boundary face as the face-based partition. For higher order accuracy, partitions involving a single CV on each SV face may offer advantages, since they would involve far fewer Riemann solver calls than the vertex-based partitions. This is a subject of future research. In the remainder of this paper, we shall not consider this type of partition. For the linear partition, due to symmetry, there are only two sets of weights coefficients needed to be stored. Each CV shown in Fig. 2A is numbered corresponding to the SV vertex i. Shown also in Fig. 2A, points a and b are the centroids for the boundary face a and the interior face b, respectively. Since u is linear, one thus obtains

$$u_a = \frac{1}{S_a}\int_{S_a} u\,\mathrm{d}S = \frac{65}{52}\bar{u}_1 + \frac{5}{52}\bar{u}_2 + \frac{5}{52}\bar{u}_3 - \frac{23}{52}\bar{u}_4 \tag{21a}$$

and

$$u_b = \frac{1}{S_b}\int_{S_b} u\,\mathrm{d}S = \frac{29}{52}\bar{u}_1 + \frac{29}{52}\bar{u}_2 - \frac{3}{52}\bar{u}_3 - \frac{3}{52}\bar{u}_4. \tag{21b}$$

Weight coefficients $m_{k,j}$ for other CV faces or $l_{q,j}$ at other quadrature points are simply permutations of the coefficients appeared in Eqs. (21).

The quadratic partition comprises 10 CVs in two symmetry groups. One consists of the four vertex-based hexahedra at the corners of the SV, and the other group consists of the six mid-edge polyhedra. For the one-parameter partition, each vertex-based CV has three exterior and three interior quadrilateral faces, while each edge-based polyhedron has two exterior triangular faces and the two interior quadrilateral faces it shares with the vertex-based CVs. Note that the interior quadrilateral faces are no longer planar in general. A general one-parameter quadratic partition is shown in Fig. 3(a). As in the 2D case, a practical minimum value of the Lebesgue constant does not exist. The constant asymptotes to a value of 11. We again consider a multiple-parameter partition. In addition to the 2D parameters α and β, we need to introduce a third parameter δ to describe the locations of the interior CV vertices. It is defined as the fractional distance of the interior CV vertex from the SV vertex along the cell median (the line from a SV vertex to the center of the opposite SV face). In general, each edge-based CV is an octahedron, with two pentagonal faces on the SV boundaries and six interior quadrilateral faces. There are total of 10 CVs, 24 boundary faces, and 24 interior faces in each SV. A general three-parameter quadratic partition is shown in Fig. 3(b). The choice $\beta = \frac{2}{3}$ and $\delta = \frac{3}{4}$ reduces the three-parameter partition to the one-parameter partition. One can show that if

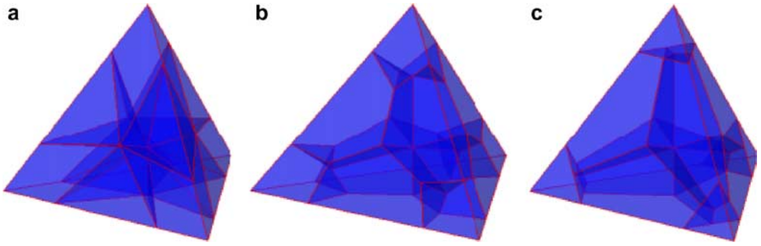

Fig. 3. Three-dimensional quadratic partitions: (a) one-parameter partition, (b) three-parameter partition, (c) $\alpha = \beta = \delta = 1/4$.

$$\delta = \frac{3\alpha\beta}{4\alpha - \beta}, \tag{22}$$

then all the interior quadrilateral faces become planar. The choice $\beta = \alpha$ further reduces the vertex-based CVs to tetrahedra with planar faces. In our numerical tests, based on the two-dimensional results, we use the values $\alpha = \beta = \delta = 1/4$. This partition is shown in Fig. 3(c). The Lebesgue constant is 5.2688, compared to the value of 16 for the same α in the one-parameter partition.

From (4) it follows that a cubic partition consists of 20 CVs. It consists of 4 vertex-based CVs, 4 faced-based CVs, and 12 edged-based CVs located symmetrically on each of the six edges. The four vertex-based CVs are the hexahedra determined as before by the three parameters defined by the quadratic partition. Each of the edge-based CVs is a hexahedron, four of whose vertices it share with the vertex-based CV. Two more vertices are on the SV faces that share the edge. Their location is determined by the 2D parameter γ, defined previously as their fractional distance from the SV edge midpoint along the SV face median. Another CV vertex is an interior vertex whose location is determined by the parameter ε, defined as its fractional distance from the SV edge midpoint along the cell bimedian (the line joining the midpoints of opposite SV edges). The final CV vertex is the edge midpoint. The edged-based CV is then defined by connecting the eight vertices by straight lines. The four face-based CVs are decahedra, defined by connecting the vertex-based CVs interior vertex to the cell centroid. The 10 faces of the face-based CV are a hexagonal face on the SV boundary, 6 interior quadrilateral faces it shares with the edge-based CVs, and 3 interior quadrilateral faces it shares with the other face-based CVs. A general cubic partition is shown in Fig. 4(a). If one chooses

$$\varepsilon = \frac{2\gamma\delta(2 - 3\beta)}{3\beta(1 - 3\gamma) + 4\delta(2\gamma - \beta)}, \tag{23}$$

then the interior quadrilateral faces shared by the edge-based and face-based CVs are planar. There are again special cases involving a single parameter that result in partitions with simpler geometries. The choice

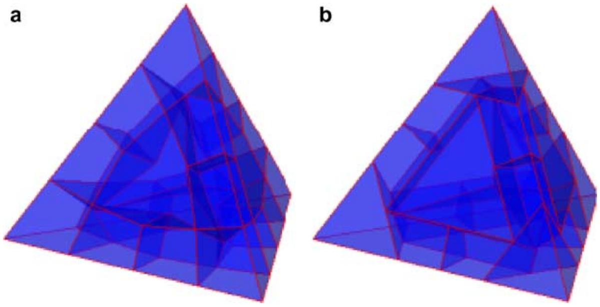

Fig. 4. Three-dimensional cubic partitions: (a) six-parameter partition, (b) $\beta = \gamma = \delta = \alpha$.

$\beta = \gamma = \delta = \alpha$ reduces the vertex-based CVs to tetrahedra, the edge-based CVs to hexahedra with planar quadrilateral faces, and the face-based CVs to septahedra with four triangular faces and three plane quadrilateral faces. This partition is shown in Fig. 4(b). The optimum choice of the five parameters in this cubic partition is a subject of further research. Our numerical tests presented in the next section are based on the linear and quadratic partitions.

5. Numerical results

5.1. Numerical solutions for the Maxwell equations

In order to demonstrate the high accuracy of the method, it was decided in this initial phase to choose problems for which there exist exact solutions. To this end we solve the Maxwell equations

$$\frac{\partial \mathbf{D}}{\partial t} - \nabla \times \mathbf{H} = 0, \tag{24a}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0. \tag{24b}$$

The electric and magnetic flux density vectors $(\mathbf{D}, \mathbf{B})$ and the intensity vectors $(\mathbf{E}, \mathbf{H})$ are related through the constitutive relations

$$\mathbf{D} = \varepsilon \mathbf{E}, \tag{25a}$$

$$\mathbf{B} = \mu \mathbf{H}, \tag{25b}$$

where ε is the permittivity and μ is the permeability of the material. Exact solutions exist for plane waves incident on simple bodies in two and three dimensions. Except for the data reconstruction, the algorithm and the boundary conditions used here are the same as those previously developed in [13]. In the old procedure, for every cell, one must first constructs a large stencil of neighboring cells, and then performs a least-squares reconstruction. The procedure is repeated at every time step, or one must save the weight coefficients for all faces. Since the weight coefficients are universal for the current method, the new procedure becomes extremely simple, without these repeated calculations or data communications.

In the absence of electric currents, the Maxwell equations are non-dissipative. But in many numerical schemes in which all the field components are co-located, some artificial dissipation is necessary to smooth undesirable numerical oscillations. In addition, special procedures are necessary at physical and computational boundaries. Riemann solvers have long been utilized in computational fluid dynamics calculations for capturing discontinuities. This technique was first applied to electromagnetic computations by Shankar et al. [18] and later modified by Liu [13,14], though their derivations were different. Since the Maxwell equations are linear, an exact characteristic formulation is possible. All numerical procedures for outer boundary, material interface, zonal interface, and interior SV face can be unified with a single characteristic formulation. The load balancing in a massive parallel computing environment is therefore easier to achieve. Here we consider the one-dimensional Maxwell equations in the direction $\mathbf{n}$ (unit normal) along the coordinate ζ

$$\frac{\partial \mathbf{D}}{\partial t} - \mathbf{n} \times \frac{\partial \mathbf{H}}{\partial \zeta} = 0, \tag{26a}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \mathbf{n} \times \frac{\partial \mathbf{E}}{\partial \zeta} = 0. \tag{26b}$$

Taking the cross products $\mathbf{n} \times (\mathbf{n} \times (26\text{a}))$ and $\mathbf{n} \times (26\text{b})$ and using the identity $\mathbf{n} \times (\mathbf{n} \times (\mathbf{n} \times \mathbf{H})) = -\mathbf{n} \times \mathbf{H}$, one obtains two decoupled characteristic equations

$$\frac{\partial w^+}{\partial t} + c\frac{\partial w^+}{\partial \zeta} = 0, \tag{27a}$$

$$\frac{\partial w^-}{\partial t} - c\frac{\partial w^-}{\partial \zeta} = 0, \tag{27b}$$

in terms of Riemann invariants

$$w^+ = \mathbf{n} \times (\mathbf{H} + \varepsilon c\mathbf{n} \times \mathbf{E}), \tag{28a}$$

$$w^- = \mathbf{n} \times (\mathbf{H} - \varepsilon c\mathbf{n} \times \mathbf{E}). \tag{28b}$$

At an interface, let the superscript * denote the interface state, and the subscripts L and R the states on the two sides. Since the tangential components of both the electric and magnetic fields must be continuous at the interface, the Riemann invariants thus yield

$$\mathbf{n} \times (\mathbf{H}^* + (\varepsilon c)_L\mathbf{n} \times \mathbf{E}^*) = \mathbf{n} \times (\mathbf{H}_L + (\varepsilon c)_L\mathbf{n} \times \mathbf{E}_L), \tag{29a}$$

$$\mathbf{n} \times (\mathbf{H}^* - (\varepsilon c)_R\mathbf{n} \times \mathbf{E}^*) = \mathbf{n} \times (\mathbf{H}_R - (\varepsilon c)_R\mathbf{n} \times \mathbf{E}_R). \tag{29b}$$

One can simplify the above equations and obtain

$$\mathbf{n} \times \mathbf{H}^* = \frac{1}{(\mu c)_R + (\mu c)_L}\mathbf{n} \times [(\mu c)_R\mathbf{H}_R + (\mu c)_L\mathbf{H}_L - \mathbf{n} \times (\mathbf{E}_R - \mathbf{E}_L)], \tag{30a}$$

$$\mathbf{n} \times \mathbf{E}^* = \frac{1}{(\varepsilon c)_R + (\varepsilon c)_L}\mathbf{n} \times [(\varepsilon c)_R\mathbf{E}_R + (\varepsilon c)_L\mathbf{E}_L + \mathbf{n} \times (\mathbf{H}_R - \mathbf{H}_L)]. \tag{30b}$$

The first two terms on the right-hand side of the above equations form a non-dissipative centered scheme, while the last term adds numerical dissipation which changes the scheme to upwind. These equations can be used directly at material interfaces and numerical outer boundaries. In the interior on the boundary between two SVs, solutions are discontinuous, and therefore some sort of smoothing or artificial dissipation is required to stabilize the solutions. Eqs. (30) can also be used there to form a fully upwind scheme. However, since the discontinuity is normally small there, within the truncation error of the scheme, the amount of artificial dissipation given by Eqs. (30) is too excessive. In practice, only a fraction of that dissipation is sufficient. We therefore modify Eqs. (30) by adding a parameter ω to adjust the strength of the dissipation, resulting in a unified formulation

$$\mathbf{n} \times \mathbf{H}^* = \frac{1}{(\mu c)_R + (\mu c)_L}\mathbf{n} \times [(\mu c)_R\mathbf{H}_R + (\mu c)_L\mathbf{H}_L - \omega\mathbf{n} \times (\mathbf{E}_R - \mathbf{E}_L)], \tag{31a}$$

$$\mathbf{n} \times \mathbf{E}^* = \frac{1}{(\varepsilon c)_R + (\varepsilon c)_L}\mathbf{n} \times [(\varepsilon c)_R\mathbf{E}_R + (\varepsilon c)_L\mathbf{E}_L + \omega\mathbf{n} \times (\mathbf{H}_R - \mathbf{H}_L)]. \tag{31b}$$

Note that the above Riemann solvers are linear with respect to the **E** and **H** fields, and thus the surface integrals for each planar CV facet can be expressed as

$$\int \mathbf{n} \times \mathbf{H}^* \,\mathrm{d}S = \frac{1}{(\mu c)_R + (\mu c)_L}\mathbf{n} \times \left[(\mu c)_R\widehat{\mathbf{H}}_R + (\mu c)_L\widehat{\mathbf{H}}_L - \omega\mathbf{n} \times (\widehat{\mathbf{E}}_R - \widehat{\mathbf{E}}_L)\right], \tag{32a}$$

$$\int \mathbf{n} \times \mathbf{E}^* \,\mathrm{d}S = \frac{1}{(\varepsilon c)_R + (\varepsilon c)_L}\mathbf{n} \times \left[(\varepsilon c)_R\widehat{\mathbf{E}}_R + (\varepsilon c)_L\widehat{\mathbf{E}}_L + \omega\mathbf{n} \times (\widehat{\mathbf{H}}_R - \widehat{\mathbf{H}}_L)\right], \tag{32b}$$

where $\widehat{\mathbf{E}} \equiv \int \mathbf{E}\,\mathrm{d}S$ and $\widehat{\mathbf{H}} \equiv \int \mathbf{H}\,\mathrm{d}S$. The surface integrals $\widehat{\mathbf{E}}$ and $\widehat{\mathbf{H}}$ are evaluated using the universal interpolations equations (13), in terms of the CV volume averaged unknowns $\bar{\mathbf{D}}$ and $\bar{\mathbf{B}}$. However, at a numerical outer boundary, the surface integration on the exterior side must be performed with the prescribed incident field. At a perfect conductor surface, the known boundary field $\mathbf{n} \times \mathbf{E} = 0$ is used, and thus the surface integral $\int \mathbf{n} \times \mathbf{E}^* \,\mathrm{d}S = 0$ on the surface. The other surface integral $\int \mathbf{n} \times \mathbf{H}^* \,\mathrm{d}S$ is obtained via interpolation using

Eq. (13). Inside each SV, the fields are continuous ($\mathbf{E}_R = \mathbf{E}_L$ and $\mathbf{H}_R = \mathbf{H}_L$), and consequently no dissipation is added.

Eqs. (24) can be integrated with any stable time integration scheme, e.g., a Runge–Kutta scheme [19]. It was shown in [13] that a low-order leapfrog scheme can generate a more accurate solution than a high-order Runge–Kutta scheme for wave propagation problems, since there is a spatial and temporal error cancellation. The conventional leapfrog scheme is unconditionally unstable for solving the Maxwell equations with a spatial discretization that is dissipative. We therefore use the staggered leapfrog scheme. Omitting the SV subscript i and the CV subscript j, it can be written as

$$\bar{\mathbf{D}}^{n+1} = \bar{\mathbf{D}}^{n} + \frac{\Delta t}{V}\sum_{k}\frac{1}{(\mu c)_R + (\mu c)_L}\mathbf{n}\times\left[(\mu c)_R\overset{\frown}{\mathbf{H}}{}^{n+\frac{1}{2}}_{k,R} + (\mu c)_L\overset{\frown}{\mathbf{H}}{}^{n+\frac{1}{2}}_{k,L} - \omega\mathbf{n}\times\left(\overset{\frown}{\mathbf{E}}{}^{n}_{k,R} - \overset{\frown}{\mathbf{E}}{}^{n}_{k,L}\right)\right], \tag{33a}$$

$$\bar{\mathbf{B}}^{n+\frac{1}{2}} = \bar{\mathbf{B}}^{n-\frac{1}{2}} + \frac{\Delta t}{V}\sum_{k}\frac{1}{(\varepsilon c)_R + (\varepsilon c)_L}\mathbf{n}\times\left[(\varepsilon c)_R\overset{\frown}{\mathbf{E}}{}^{n}_{k,R} + (\varepsilon c)_L\overset{\frown}{\mathbf{E}}{}^{n}_{k,L} + \omega\mathbf{n}\times\left(\overset{\frown}{\mathbf{H}}{}^{n-\frac{1}{2}}_{k,R} - \overset{\frown}{\mathbf{H}}{}^{n-\frac{1}{2}}_{k,L}\right)\right], \tag{33b}$$

where the superscript n denote the time level, and Δt is the time step. Note that in order to maintain the explicit nature of the scheme, old time level fields are used for the dissipation terms.

It follows from Eq. (24b) that $\nabla\cdot\mathbf{B} = 0$ if this condition is true initially. From Eq. (24a), a similar statement can be made about $\nabla\cdot\mathbf{D}$. Therefore, these two conditions are implicitly satisfied by the Maxwell equations (24a) and (24b). This is in contrast to incompressible flows, in which the velocity $\mathbf{v}$ must satisfy $\nabla\cdot\mathbf{v} = 0$ as a separate equation. In some numerical methods, such as the Yee scheme [28], the divergence conditions can be satisfied automatically within round-off errors. However, in FV methods [18,13], as well as the present scheme, these conditions are satisfied within truncation errors.

In the following we illustrate the stability of the present scheme applied to the simplified model equations

$$\frac{\bar{\mathbf{E}}^{n+1} - \bar{\mathbf{E}}^{n}}{\Delta t} = \lambda_1\bar{\mathbf{E}}^{n} + \mathrm{i}\lambda_2\bar{\mathbf{H}}^{n+\frac{1}{2}}, \tag{34a}$$

$$\frac{\bar{\mathbf{H}}^{n+\frac{1}{2}} - \bar{\mathbf{H}}^{n-\frac{1}{2}}}{\Delta t} = \mathrm{i}\lambda_2\bar{\mathbf{E}}^{n} + \lambda_1\bar{\mathbf{H}}^{n-\frac{1}{2}}. \tag{34b}$$

Here λ_1 (a pure negative real number, proportional to ω) represents the eigenvalue from the discretization of the dissipation terms, and $\mathrm{i}\lambda_2$ (a pure imaginary number) the eigenvalue from the discretization of the non-dissipative centered terms. Fig. 5 shows the stable region on the $\lambda_1\Delta t - \lambda_2\Delta t$ complex plane. It is seen that the maximum Δt (or the Courant number) for the scheme to be stable is decreased as we add more and more dissipation (i.e., increasing ω). In all our numerical tests, we used $\omega = 1$ for the outer boundary and material interface, and a value in the range 0.2–0.3 for all other interior SV faces. Inside each SV, ω is essentially equal to zero, since no dissipation is added.

In order to illustrate the high accuracy of the method and the effect of the value of ω, we first consider a plane wave incident on a perfectly conducting circular cylinder. Calculations were carried out over a grid, shown in Fig. 6, consisting of 2024 SV cells. The average size of the SV cell is about one-fifth of the cylinder radius. Outer boundaries are two radii away from the body surface, with no PML (perfectly matched layer). This gives approximately $30/ka$ SV cells per wavelength, or approximately $30\sqrt{p(p+1)}/ka$ CV cells per wavelength. Here k is he wave number, a is the radius of the cylinder, and $p = n + 1$ is the order of the accuracy of the method. The incident wave with $ka = 5$ is propagating from left to right. The time step used is $\Delta t = \frac{2\pi a}{350c}$. This corresponds to a Courant number $\frac{2\pi}{70}$ based on the average size of the SV cell. Contour plots of H_z at $t = 6\pi\frac{a}{c}$ for a TE wave for the second-order SV method are shown in Fig. 7(a) for $\omega = 1.0$ and Fig. 7(b) for $\omega = 0.2$. Analogous plots for the third-order method with $\Delta t = \frac{2\pi a}{600c}$ are shown in Figs. 7(c) and (d). We have also plotted the exact solution [10, p. 232] (red lines) on the same figures for easy comparisons. The plotting package we used only carries out linear interpolation within triangles, and thus is

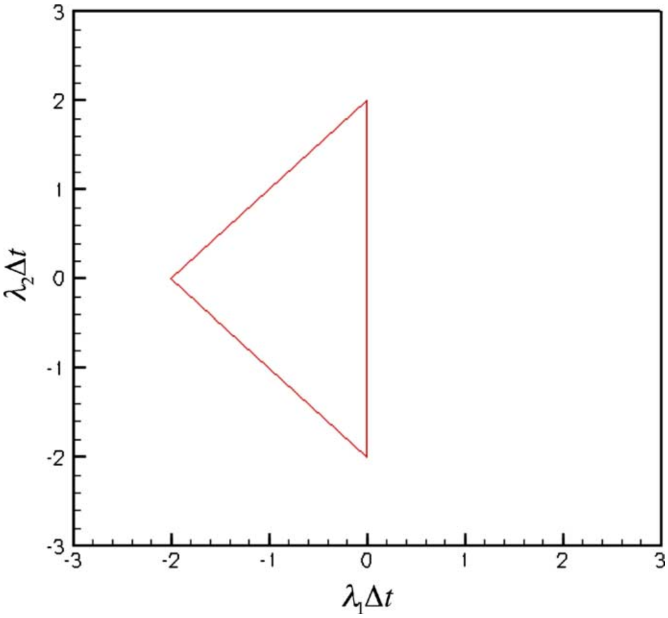

Fig. 5. Stable region for the staggered leapfrog scheme.

unable to show features of high-order solutions within the SV cells. We therefore choose to plot every local representation, Eq. (10), within each SV by subdividing the SV into many triangular sub-cells. In our method, solutions across the SV boundaries are discontinuous. This is more pronounced when the grid resolution or the order of accuracy is low, as shown in Figs. 7(a) and (b) for both cases of the second-order method, in which the grid resolution is approximately 15 CV cells per wavelength. However, reducing the

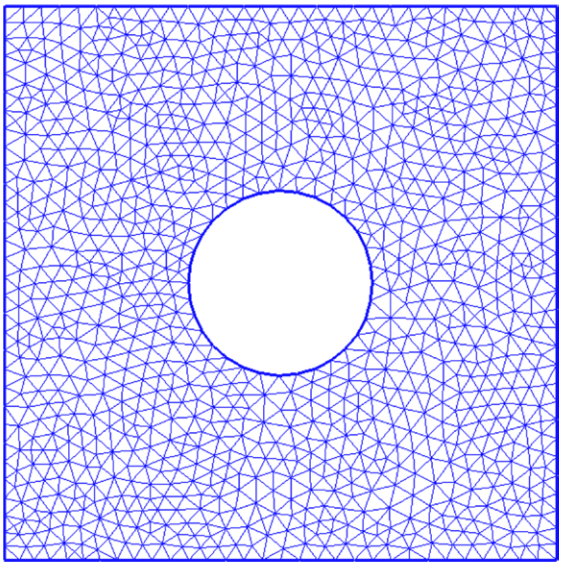

Fig. 6. Grid for a region exterior to a circular cylinder.

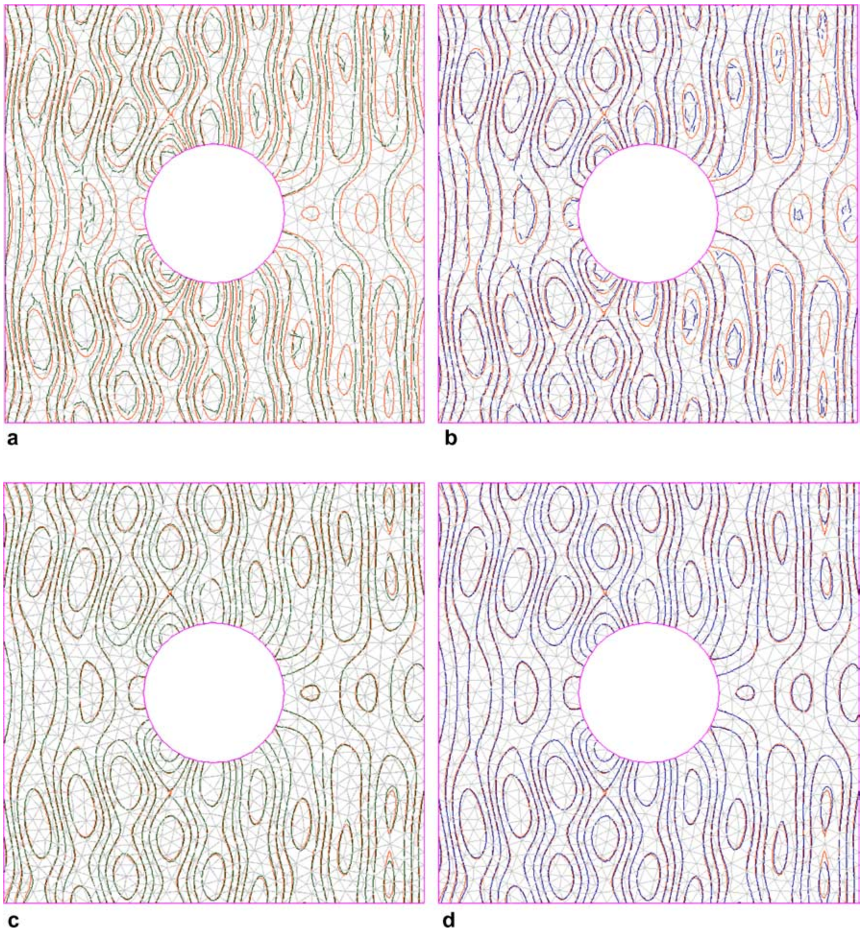

Fig. 7. Contour plot of H_z for a plane wave incident on a perfectly conducting circular cylinder: (a) second order, $\omega = 1.0$, (b) second order, $\omega = 0.2$, (c) third order, $\omega = 1.0$, (d) third order, $\omega = 0.2$.

value of ω from 1.0 to 0.2 gives a great improvement in accuracy. As the grid resolution or the order of accuracy increases, the difference between the solutions across the SV boundaries decreases. For the third-order method with approximately 21 CV cells per wavelength, as depicted in Figs. 7(c) and (d), the discontinuity is hardly visible from the plots. Here, both the numerical solutions are almost identical to the exact solution, and the effect of ω is not noticeable. By reducing the grid resolution below 10 CV cells per wavelength for the third-order method, the effect of ω starts to show up. Of equal importance is that the maximum allowable time step to achieve a stable solution is twice as large for $\omega = 0.2$ as for $\omega = 1.0$. Similar results were also obtained for a TM wave.

For the three-dimensional test case, we considered a plane wave incident on a perfectly conducting sphere. Calculations were carried out over two grids, shown in Figs. 8(a) and (b). Outer boundaries are two radii away from the body surface, also with no PML. Grid 1 consists of 62,657 tetrahedral SV cells, and grid 2 113,352 SV cells. This gives approximately $15/ka$ SV cells or $15\sqrt[3]{p(p+1)(p+2)}/ka$ CV cells per wavelength for grid 1 (at the outer boundaries), and twice the number of SV or CV cells per wavelength for grid 2. The calculations were carried out over the two grids with the second- and third-order accuracy, using $\Delta t = \frac{2\pi a}{600c}$ for the former and $\Delta t = \frac{2\pi\ a}{1500c}$ for the latter. The incident wave, propagating in the positive z

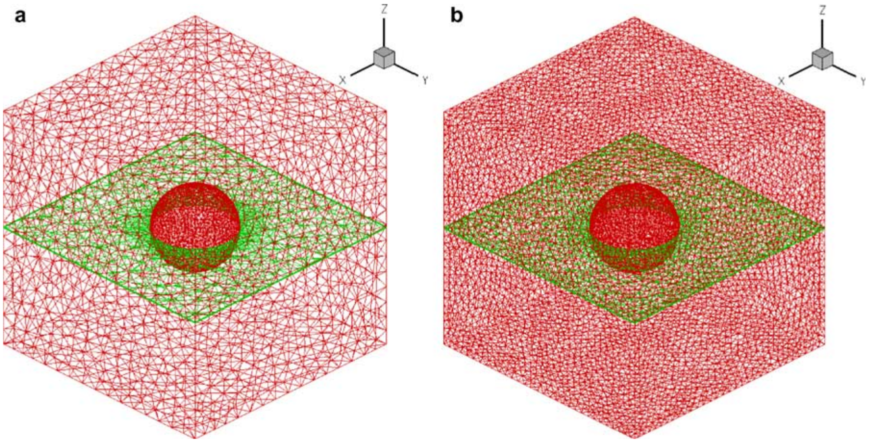

Fig. 8. Grid for a region exterior to a sphere: (a) grid 1: 62,576 SV cells, (b) grid 2: 113,352 SV cells.

direction, has a x component for the **E** field and a y component for the **H** field, with $ka = 3$. The parameter ω is set to 0.2. Figs. 9(a)–(d) show the color contours of E_x plotted on the sphere, plane of symmetry, and the outer boundaries. The exact solution [10, p. 292] is also plotted on the same figures with solid contour lines for easy comparisons. Excellent agreement was obtained, even at the outer boundaries, attesting to the accuracy of the numerical non-reflecting boundary procedure. Inspecting the contour lines, one can see the improvements by each grid refinement and order advancement. Note that the plotting package we used is unable to plot arbitrary unstructured polyhedral CV data. Due to storage limitations, we could not subdivide each SV into many tetrahedral sub-cells for plotting, as was done for the two-dimensional cases. We therefore first interpolated the numerical solution at the SV vertices from the CV averaged values. If there were multiple values, a simple average was then used. The exact solution was also calculated at the SV vertices. This accounts for the jagged nature of the contour lines in our plots.

5.2. Timing comparison

It is interesting to compare the computational cost of the SV method and some conventional methods. In Table 1, we present the CPU times per cell, per time step for the three-dimensional sphere problem using the second- and third-order SV method and the second- and third-order unstructured FV method of [13]. The numerical procedures for the two methods are identical, except that we save all the weight coefficients in Eq. (13) for the latter to study the effect of the repeated process in fetching this large data set from memory. We also include an upwind structured FV method with the Lax-Wendroff scheme. The control volumes for the second-order SV method are hexahedra. We therefore use a hexahedral grid for both the conventional structured and unstructured FV methods for the comparison. The table also includes the number of boundary faces and interior faces per cell. The number of boundary faces determines the number of Riemann solvers. For the unstructured FV method, triangular facets were counted instead of the non-planar faces. The CPU times were measured on a 600 MHz MIPS R14000 SGI Octane2 workstation. It is shown that the SV method is 5.5 times as fast as a conventional unstructured method for the second order, and 8.8 times for the third order. This reduction in CPU times is primarily due to fewer Riemann flux calculations at boundary faces and a large savings in memory access time. The table also shows that the second-order SV method is 4.3 times as fast as a conventional structured method. Note that the number of boundary faces per cell decreases as the order of accuracy of the SV method increases. The CPU time for the third-order SV method is just slightly higher than that for the second order. For the 2D case, the CPU time actually decreases slightly as the order of accuracy increases.

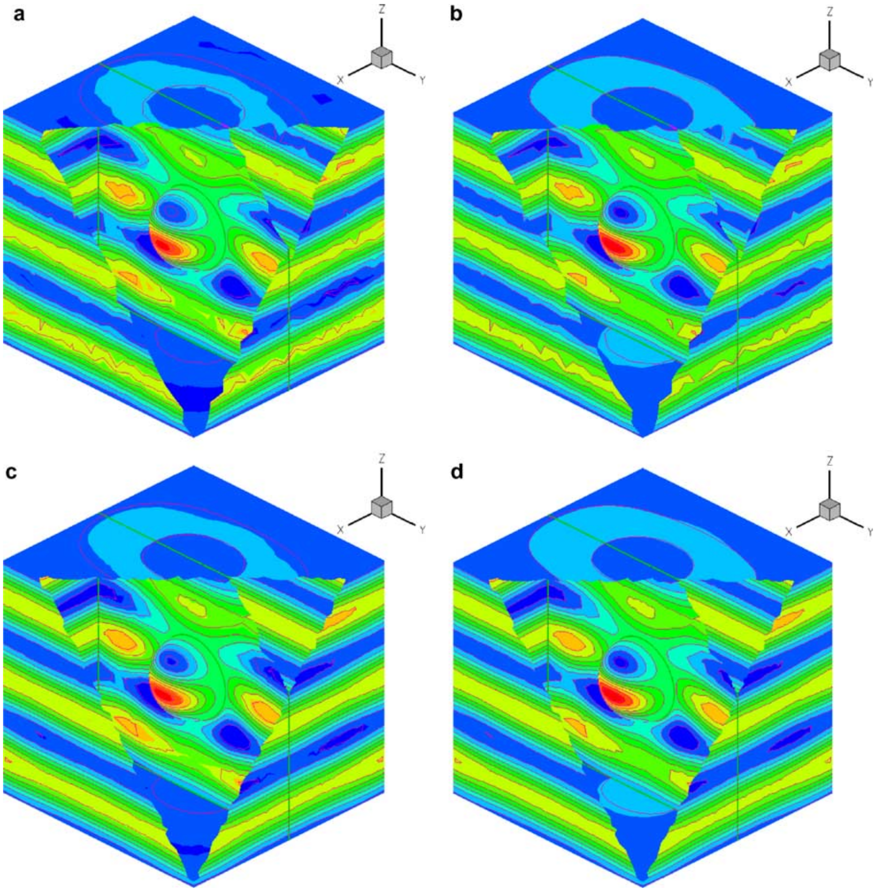

Fig. 9. Contour plot of E_x for a plane wave incident on a perfectly conducting sphere color contours: numerical solutions, solid lines: exact solutions. (a) Grid 1, second order, (b) grid 1, third order, (c) grid 2, second order, (d) grid 2, third order.

Table 1
Comparison of CPU times for the SV method and conventional structured and unstructured methods

Method	SV second order	SV third order	Unstructured second order	Unstructured third order	Structured second order
CPU time/cell/step	2.89 μs	3.08 μs	15.95 μs	27.32 μs	12.41 μs
Boundary faces/cell	1.5	1.2	6	6	3
Interior face/cell	1.5	2.4	0	0	0

6. Concluding remarks

In this paper, we have extended the high-order spectral finite-volume method for unstructured grids to three dimensions. We first showed how the SV method overcomes some of the limitations of conventional structured and unstructured methods. By partitioning all grid cells into sub-cells in a similar manner, the reconstruction becomes universal as a weighted sum of unknowns involving just a few operations. In addition, the resulting data structure allows much more efficient use of cache memory. New multiple parameter partitions of the SV in 2D were derived, and utilized in constructing 3D partitions. Numerical solutions of a wave propagation problem in two and three dimensions showed excellent agreement with the exact solutions. Improvements were seen with each grid refinement and increasing order of accuracy. Finally, we presented computational timings to show quantitatively the greater computational efficiency of the SV method. Future work will include extensions to higher orders, non-linear conservation laws such as the Euler and Navier–Stokes equations, and curved boundaries.

Acknowledgments

The third author gratefully acknowledges a start-up fund provided by the Department of Aerospace Engineering, College of Engineering of Iowa State University. The work is also partially funded by the Department of Energy under Contract DE-FG02-05ER25677.

References

[1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.

[2] T.J. Barth, P.O. Frederickson, High-order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper No. 90-0013, 1990.

[3] Q. Chen, I. Babuska, Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle, Comput. Methods Appl. Mech. Eng. 128 (1995) 405–417.

[4] B. Cockburn, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput. 52 (1989) 411–435.

[5] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84 (1989) 90–113.

[6] B. Cockburn, S. Hou, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Math. Comput. 54 (1990) 545–581.

[7] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Garlerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.

[8] M. Delanaye, Yen Liu, Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids, AIAA Paper No. 99-3259-CP, 1999.

[9] S.K. Godunov, A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271.

[10] R.F. Harrington, Time-Harmonic Electromagnetic Fields, MaGraw-Hill, New York, 1961.

[11] Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes III, J. Comput. Phys. 71 (1987) 231.

[12] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.

[13] Y. Liu, A generalized finite volume algorithm for solving Maxwell's equations on arbitrary grids, in: Proceedings of 10th Annual Review of Progress in Applied Computational Electromagnetics, 1994.

[14] Y. Liu, Fourier analysis of numerical algorithms for the Maxwell equations, J. Comput. Phys. 120 (1996) 396–416.

[15] Y. Liu, M. Vinokur, Exact integration of polynomials and symmetric quadrature formulas over arbitrary polyhedral grids, J. Comput. Phys. 140 (1998) 122–147.

[16] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.

[17] V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 267–279.

[18] V. Shankar, W.F. Hall, A.H. Mohammadian, A CFD-based finite-volume procedure for computational electromagnetics – interdisciplinary application for CFD methods, AIAA paper 89-1987, 1989.

[19] C.-W. Shu, Total-variation-diminishing time discretizations, SIAM J. Sci. Stat. Comput. 9 (1988) 1073–1084.

[20] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, Math. Comput. 49 (1987) 105–121.

[21] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor (Eds.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, in: A. Quarteroni (Ed.), Lecture Notes in Mathematics, vol. 1697, Springer, Berlin, 1998, pp. 325–432.

[22] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II, J. Comput. Phys. 83 (1989) 32.

[23] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210.

[24] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation, J. Comput. Phys. 179 (2002) 665–697.

[25] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization, J. Sci. Comput. 20 (2004) 137–157.

[26] Z.J. Wang, Y. Liu, L. Zhang, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional systems, J. Comput. Phys. 194 (2004) 716–741.

[27] S. Wolfram, The Mathematica Book, fourth ed., Wolfram Media and Cambridge University Press, New York, 1999.

[28] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE Trans. Antennas Propagat. AP-14 (1966) 302–307.