# Third-order sparse grid generalized spectral elements on hexagonal cells for uniform-speed advection in a plane

J. Steppeler, J. Li, F. Fang & I. M. Navon

**Meteorology and Atmospheric Physics**

ISSN 0177-7971

Meteorol Atmos Phys
DOI 10.1007/s00703-019-00718-0

ONLINE FIRST

Springer

Your article is protected by copyright and all rights are held exclusively by Springer-Verlag GmbH Austria, part of Springer Nature. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com”.

Springer

Author's personal copy

Meteorology and Atmospheric Physics
https://doi.org/10.1007/s00703-019-00718-0

ORIGINAL PAPER

# Third-order sparse grid generalized spectral elements on hexagonal cells for uniform-speed advection in a plane

J. Steppeler[2] · J. Li[1,3] · F. Fang[4] · I. M. Navon[5]

Received: 31 January 2019 / Accepted: 4 December 2019
© Springer-Verlag GmbH Austria, part of Springer Nature 2019

**Abstract**
This paper investigates sparse grids on a hexagonal cell structure using a Local-Galerkin method (LGM) or generalized spectral element method (SEM). Such methods allow sparse grids to be used, known as serendipity grids in square cells. This means that not all points of the full grid are used. Using a high-order polynomial, some points of each cell are eliminated in the discretization, and thus saving Central Processing Unit (CPU) time. Here a sparse SEM scheme is proposed for hexagonal cells. It uses a representation of fields by second-order polynomials and achieves third-order accuracy. As SEM, LGM is strictly local for explicit time integration. This makes LGM more suitable for multiprocessing computers compared with classical Galerkin methods. The computer time depends on the possible timestep and program implementation. Assuming that these do not change when going to a sparse grid, the potential saving of computer time due to sparseness is 1:2. The projected CPU saving in 3-D from sparseness is by a factor of 3:8. A new spectral procedure is used in this paper, called the implied spectral equation (ISE). This procedure allows for some collocation points to use any finite difference scheme of high order and the time derivatives of other spectral coefficients are implied.

## 1 Introduction

A new generation of global atmospheric models is constructed by dividing the sphere into patches of nearly regular grids (Taylor et al. 1997; Giraldo 2001; Williamson 2007; Satoh et al. 2008; Skamarock and Klemp 2008; Skamarock et al. 2012; Staniforth and Thuburn 2012; Zängl et al. 2015). The basic principle of such icosahedral or cubed-sphere models has been known for a long time (Williamson 1968; Sadourny 1972). It was about the year 2000 until polygonal methods in the form of icosahedral and cubed-sphere models were considered fit for use in realistic atmospheric models (Rancic et al. 1996; Steppeler and Prohl 1996; Ringler et al. 2000). In the mean-time, spectral models or models with latitude–longitude grids became the method of choice for global atmospheric models. At present, most new developments of realistic models use polygonal methods (Taylor et al. 1997; Giraldo 2001; Skamarock and Klemp 2008; Zängl et al. 2015).

With hindsight, there are two problems which prevented the immediate application of the polygonal grids (Williamson 1968; Sadourny 1972). One problem is the mapping of a plane to the sphere, which some models employ for the discretization. If a global coordinate is used, this mapping of the plane to the sphere will lead to singularities. This implies similar problems as encountered with global latitude–longitude grids, even though polygonal grids are nearly regular. This problem was solved in Steppeler and Prohl (1996) by abandoning a global coordinate system and rather using a local coordinate at each grid point. In Steppeler et al. (2008), a coordinate-free method was used, which can also be considered as using the earth centered orthogonal coordinate system together with a local basis for

Responsible Editor: C. Simmer.

✉ J. Li
ljx2311@mail.iap.ac.cn

[1] International Center for Climate and Environment Sciences, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029, China

[2] Climate Service Center Germany (GERICS), Fischertwiete 1, 20095 Hamburg, Germany

[3] Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, England, UK

[4] Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London SW7 2AZ, UK

[5] Department of Scientific Computing, Florida State University, Tallahassee, FL 32306-4120, USA

Published online: 23 December 2019

Author's personal copy

vectors at each grid point. Another problem is the occurrence of low-order approximations at poles and edges of the icosahedron. This will lead to the icosahedral grid to be seen distorting the solutions, which is called grid imprinting (Peixoto and Barros 2013; Weller et al. 2012). An example of problems caused by non-uniform approximation order at the poles was given by Tomita et al. (2001). Solutions without grid imprinting were developed in Baumgardner and Frederickson (1985) for second-order approximation and for a non-meteorological problem. In Steppeler et al. (2008), a third-order icosahedral global model without grid imprinting was proposed. Baumgardner and Frederickson (1985) and Steppeler et al. (2008) avoided grid imprinting using approximations of uniform order two or three. Uniform order $p$ means that there is no grid point where the order of approximation is below $p$. The grid imprinting reported in Tomita et al. (2001) was caused by approximations going below second-order at poles and icosahedral edges. Baumgardner and Frederickson (1985) obtained successful approximations on the sphere for a non-meteorological problem. Steppeler and Prohl (1996) obtained second-order solutions of the shallow water equations on the sphere and proposed to transfer the approximation principles in Baumgardner and Frederickson (1985). However, the schemes proposed by Baumgardner and Frederickson (1985) and Steppeler et al. (2008) were not mass conserving.

To combine high-order approximations on the sphere with mass conservation, finite element method (FEM) or other Local-Galerkin methods (LGMs) can be used (Marras et al. 2016). These methods are naturally conserving and provide a uniform order of approximation. FEMs divide the computational area into cells and use a polynomial representation in each cell to describe the fields. These polynomials can be defined using several gridpoint values or collocation points for each cell (see Steppeler (1987) for a review). The collocation points are used for the field representation by interpolations in Galerkin methods. The number of collocation points must be sufficient (e.g., order 2 or order 3) to define a polynomial of the requested order. In this study, we use continuous Galerkin (CG) methods where polynomials fit together continuously at cell boundaries (Taylor et al. 1997; Li et al. 2018). Differentiation of this field representation, to compute the right-hand side of dynamic equations will generally lead out of this continuous function system and create discontinuities at cell boundaries. A Galerkin method is used to map this again to the original function system. Standard time stepping procedures (e.g., Runge–Kutta time integration) can then be applied. The original Galerkin procedure involves the solution of a linear equation system, the mass matrix equation. Even though the mass matrix is often of band diagonal form (Cote et al. 1983), this solution may cause some problems on modern computers. Widely used methods to solve the matrix equation are recursive and the solution may have global impact in one time-step, even though the equation to be solved is local. On massively parallel computers this causes considerable problems, as typically a lot of communication is involved. Approximations to the original Galerkin procedure have been introduced, which make the method local and thus suitable for massively parallel computing. Such approximated or alternative Galerkin procedures are called Local-Galerkin methods. For a general outline of the LGM see Steppeler and Klemp (2017). Discontinuous field representations (DG methods) are used with a variety of LGMs (Cockburn and Shu 2001), but only CG methods are currently near practical use in realistic models. Spectral element methods (SEMs), using Gauss–Lobatto points to define the fields (Giraldo 2001) and the third-degree method (Steppeler 1976) are examples for LGMs. Considerable progress has been achieved in this way to make high-resolution atmospheric modeling possible. CG methods for weather models nearly exclusively use the SEM as LGM (Taylor et al. 1997), being based on the quadrature approximation, since alternative LGMs so far are used with toy models only. In this paper we use a special LGM called Implied Spectral Equations (ISE). With ISE, the definition of some spectral coefficients is implied by the principle of conservation (see details in Sect. 3).

Advantages of SEMs are that they combine an approximation order higher than two with conservation and are of uniform approximation order. The latter means that there exist no grid points such as poles in a global model, where the approximation order drops below a target value. The uniform-order models mentioned above run on the original unsmoothed icosahedral grid on the sphere. SEM models on the sphere, such as Giraldo (2001), combine an approximation order above three with mass conservation. The SEM allows an approximation order higher than two in combination with conservation. Older high-order models, such as that of Kalnay et al. (1977) and Navon and Alperson (1978) were non-conserving. There are alternative possibilities with CG methods to obtain LGMs. Such a method was proposed by Steppeler (1976) which also employed the grid sparsity coming from serendipity elements (Ahlberg et al. 1967). The FEMs with higher than first-degree polynomials use sparse grids to represent the fields in each cell. This means that some points of the full grid are omitted in each cell. Such elements with sparse grids are called serendipity elements. They are quite easy to implement with rectangular cells. Their implementation in hexagonal cells is less trivial, but the sparseness factors are larger. The sparse grids can be used in the same way with LGMs as with the original Galerkin procedure (Steppeler 1976).

In an ordinary grid system, we call the regular grid the full grid where all the values of fields on the grid points are forecasted. With the sparse grid system, not all grid point values are forecasted, but rather a subset. The non-forecasted

Author's personal copy

points can be computed diagnostically for plotting and other purposes, such as the interface to non-atmospheric models or physical parameterizations. The sparse grids offer a potential saving of computer time. The forecasted grid points in the sparse grid systems form the dynamic grid. These are the points where the dynamic equations are solved. In the following sections, the fields in the hexagonal grid are stored on the full grid, where some points with indices are unused. On computers with many processors this is not optimal, as this may imply message passing for unused points. Therefore, it is desirable to number the points in such a way that no storage is provided for unused points which is called the compact grid representation. Appendix 1 defines a compact grid for hexagonal grids.

In this study, we investigate the LGM o2o3 method with hexagonal cells. This method uses second-degree polynomials to represent the field in the cells and the numerical procedure is designed to become third-order by defining the fluxes as piecewise third-order splines. The LGM o3o3 investigated in Steppeler et al. (2019) could also have been used and the definition of o3o3 on hexagons is discussed in the appendix. The o3o3 method would have more grid points on each hexagonal cell than o2o3. Therefore, o2o3 is easier to program which is the reason why o2o3 was used for the first attempt in this paper. The application of o3o3 in the hexagon is discussed but not tested and implemented in the appendix. We first illustrate the basic hexagonal grid and field representations in Sect. 2. Section 3 outlines the governing equations and corresponding discrete FDM equations. Section 4 presents a series of idealized advection test cases with sparse grids. Finally, Sect. 5 presents a discussion and concludes this study.

## 2 Definition of grids and field representation

Figure 1a illustrates the grid distribution to cover the computational domain which is divided by the hexagonal cells. The *m, n* grid is useful to define a Cartesian grid for the definition of the initial state and plotting. The assignment of the *m, n* to the hexagonal points can be seen from Fig. 1a. This is referred to as the Cartesian representation. Let $s$ be the edge length of the hexagons so that the grid lengths are $dx = \frac{1}{2}s$ and $dy = \frac{\sqrt{3}}{4}s$. The Cartesian coordinates $x_{m,n}$ and $y_{m,n}$ of the hexagonal points are associated with each point *m, n*: $m=0, \ldots, M$; $n=0, \ldots, N$.

$$x_{m,n} = \begin{cases} mdx, & \text{for } n \text{ even} \\ mdx + \frac{1}{2}dx, & \text{for } n \text{ odd} \end{cases}, \tag{1}$$

$$y_{m,n} = ndy. \tag{2}$$

In the following it is assumed $dx = 1$ and $dy = \frac{\sqrt{3}}{2}$. As seen from Fig. 1a for the full grids there is a one-to-one correspondence between hexagonal points and the *m*, *n* grid. However, though the computational domain can be separated by the hexagons, the gird points on the edges of the hexagons are shared by two, even three hexagons such that it is hardly to determine which hexagon the grid belongs to.

To assign every grid point to only one computational element, we introduce a new index system and the corresponding non-intersecting element $(k, i, j)$ (the blue rectangles in Fig. 1) where the structured indices are $i=0, \ldots,$ *i.e.,* $j=0, \ldots, j_e$ and $k$ $(=0, \ldots, k_e)$ is the index pointing to a particular point. If $k_e=23$, then the grid is regarded as the full grid system, including all the points numbering from 0 to 23 in one blue rectangular in Fig. 1b. If $k_e=11$, then the grid is regarded as the sparse grid system, including all black points numbering from 0 to 11 in one blue rectangular in Fig. 1b. For the sparse grid, the points numbering from 12 to 23 are unused diagnostic. Therefore, the grid points within non-intersecting element $(k, i, j)$ are designed for achieving the one-to-one correspondence and coding. The transformation between the indices $(k, i, j)$ and *m, n* is seen from Fig. 1a, b. Comparing the new index system $(k, i, j)$ (the light blue index) and the *m, n* grid system (the black index) in Fig. 1a, every non-intersecting element contains six *m*-points in the horizontal while four *n*-points in the vertical.

To define a numerical scheme, it is necessary to have unique indices for each point which is shown in Fig. 1b for the computer implementation. This can be naturally done using an unstructured gridding. Alternatively, any 3-D model using unstructured indexing can be adapted to the hexagonal scheme. In this study, we use a sparse structured programming indexed by $(k, i, j)$ in 2-D. Therefore, the compact grid defines the sparse grid $(k_e=11)$ without assigning memory storage to unused points which implies that a 3-D realization could be done using any structured 3-D model as a basis. When using the Cartesian grid for the representation of the sparse grid, though half of the points *m, n* are unused, these grids still occupy a considerable proportion of the storage and message capacity with multiprocessing. It is a waste of computational resources and only acceptable for research models.

To achieve a numerical scheme, it is important to design the field representation. In every element $(k, i, j)$, there are two centers ($k=0$ and 11) of the hexagons such that we discuss the field representations in two different, but similar hexagons with center points numbering $k=0$ and 11 shown in Fig. 1c. In each hexagon, there are 19 points for potential use to define amplitudes for the field representation by second-degree polynomials. Solid points with indices $k=1$, 4,

Author's personal copy

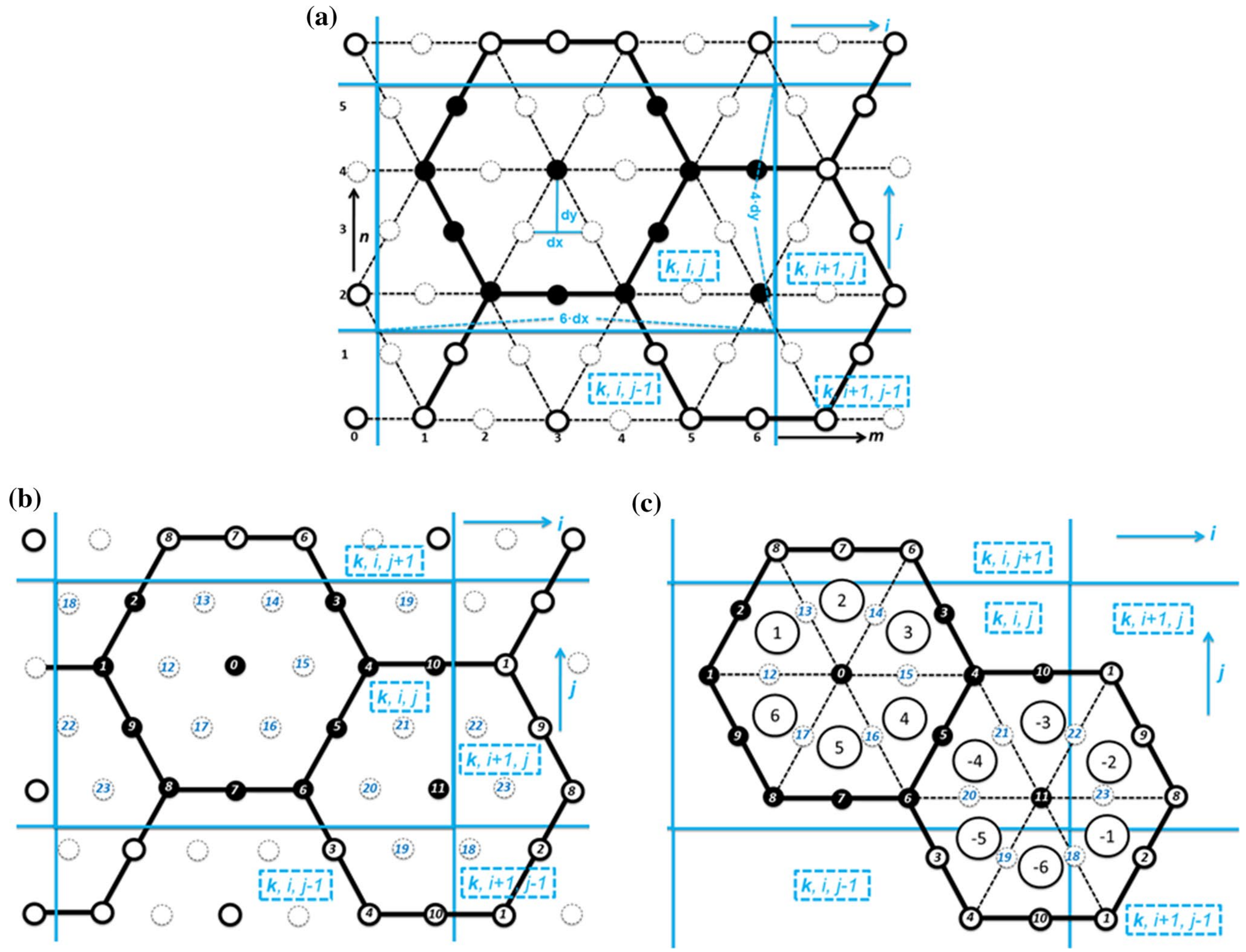

**Fig. 1** The hexagon grid distribution in the computational domain and the index system for the grid point. **a** The covering of the computational area by hexagonal cells and the non-intersecting elements $(k, i, j)$ indicating that every grid point is assigned uniquely to one element only: the solid black and white points are dynamic points and the dashed points are the unused diagnostic points. **b** The index system for every non-intersecting element $(k, i, j)$ with all dynamic (solid) and diagnostic (dashed) points. **c** The definitions of the grid points in every hexagon with collocation points and cell-related indices: the indices $-6, -5, \ldots, -1, 1, 2, \ldots, 6$ in the circles with bigger size are used for numbering the six triangles in each hexagon

6, 8 in element $(k,\ i,\ j)$, $k = 6$, 8 in element $(k,\ i,\ j+1)$, $k = 1$, 8 in element $(k,\ i+1,\ j)$, $k = 1$ in element $(k,\ i+1,\ j-1)$ and $k = 4$ in element $(k,\ i+1,\ j-1)$ are the corner points and points with indices $k = 2$, 3, 5, 7, 9, 10 in element $(k,\ i,\ j)$, $k = 7$ in element $(k,\ i,\ j+1)$, $k = 9$ in element $(k,\ i+1,\ j)$, $k = 2$ in element $(k,\ i+1,\ j-1)$ and $k = 3$, 10 in element $(k,\ i,\ j-1)$ are the edge points on the boundaries. Points with indices $k = 12$, 13, 14, 15, 16, 17, 20, 21 in element $(k,\ i,\ j)$, $k = 18$ in element $(k,\ i+1,\ j-1)$, $k = 22$, 23 in element $(k,\ i+1,\ j)$ and $k = 19$ in element $(k,\ i,\ j-1)$ is the interior points which can be used for the second-order full grid systems, not for sparse grid. Except for the center point $k = 0$, 11 in element $(k,\ i,\ j)$, all dynamic points are located on cell boundaries. To obtain the field representation by polynomials, interior points will be defined diagnostically using dynamic amplitudes.

Since the corner and edge points on boundaries of hexagons belong to more than one hexagon, the weights of center, edge, corner and interior points shared in each hexagon are 1, $\frac{1}{2}$, $\frac{1}{3}$, 1 in 2-D hexagonal grid such that the number of independent amplitudes in each hexagon is $1 \times 1 + \frac{1}{2} \times 6 + \frac{1}{3} \times 6 + 1 \times 6 = 12$ (Fig. 2a). In this study, as a sparse grid is used, we will not use the dashed interior points as dynamic points. These are shown in Fig. 2a and not shown in Fig. 2b. Dynamic points for this sparse grid system are solid points. When going to a sparse grid system, the potential saving of computer time due to sparseness is 1:2 in 2-D (Fig. 2a) under the assumption of the same

Author's personal copy

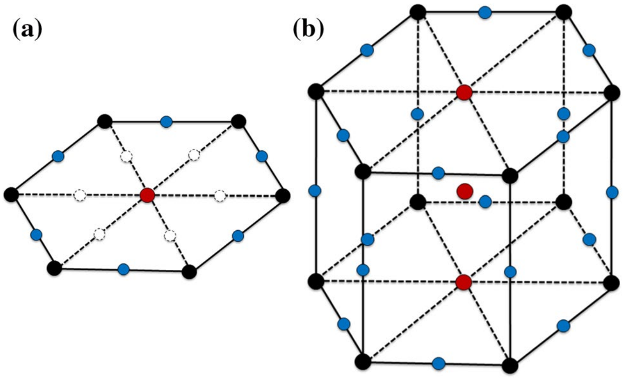

**Fig. 2** 2-D (**a**) and 3-D (**b**) hexagonal grid box where the red, black and blue points are center, corner and edge points. The white dashed points in (**a**) are the interior (unused) points which do not show in (**b**)

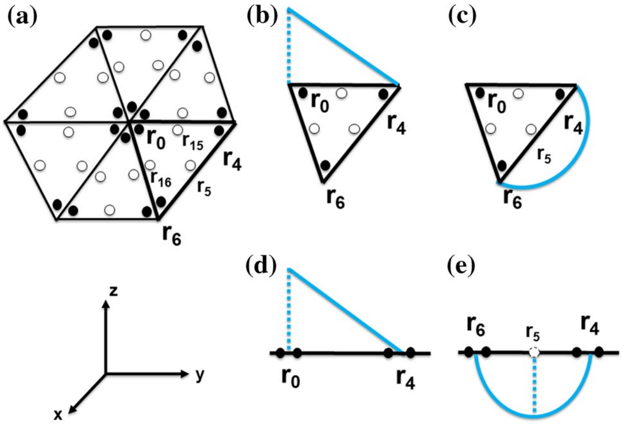

**Fig. 3** The first-order and second-order basis functions defined in one triangle within a hexagon centered at point $\mathbf{r}_{0,i,j}$ for field representation

timestep and the CPU time per dynamic point for both cases. In 3-D hexagonal box (Fig. 2b), there are two layers (the bottom and middle layers) containing different numbers of dynamic point. The bottom layer contains six corner, one center and six horizontal edge points with the weights of $\frac{1}{3}, 1, \frac{1}{2}$, while the middle layer contains six vertical edges, each shared by three boxes and the center point. Therefore, the middle layer contains $\frac{1}{3} \times 6 + 1 = 3$ dynamic points and the full grid points are for the middle layer the same as for the bottom layer which means the 3-D sparseness factor is 9:24 (3:8).

In this study, basis functions used for the field representation can be first- and second-order in each triangle, respectively. In the following, we set one triangle in the element $(k, i, j)$ as an example with numbered corners $\mathbf{r}_{0,i,j}$, $\mathbf{r}_{4,i,j}$, $\mathbf{r}_{6,i,j}$. The distribution of triangles and corner and edge points between the hexagonal cells can be seen in Fig. 3. Note that the amplitude of the field for each point belongs to more than one triangle and one hexagon: each corner amplitude belongs to one hexagonal cell if it locates at a center point $\mathbf{r}_{0,i,j}$ and to three hexagonal cells if it locates at hexagonal corner point $\mathbf{r}_{4,i,j}$, $\mathbf{r}_{6,i,j}$. An edge amplitude belongs to two hexagons and two triangles when it locates at edge points $\mathbf{r}_{5,i,j}$. As seen from Fig. 3, there exist edge points $\mathbf{r}_{15,i,j}$, $\mathbf{r}_{16,i,j}$ for two triangles, which are interior points of a hexagon, meaning that they are triangular edge points, but not hexagonal edge points. From Fig. 3, these relations become clear.

The triangular amplitudes are defined at corners: $\mathbf{r}_{0,i,j}$, $\mathbf{r}_{4,i,j}$, $\mathbf{r}_{6,i,j}$ and edges $\mathbf{r}_{5,i,j}, \mathbf{r}_{15,i,j}, \mathbf{r}_{16,i,j}$, where $\mathbf{r}_{5,i,j} = \frac{1}{2}\left(\mathbf{r}_{4,i,j} + \mathbf{r}_{6,i,j}\right), \mathbf{r}_{15,i,j} = \frac{1}{2}\left(\mathbf{r}_{0,i,j} + \mathbf{r}_{4,i,j}\right), \mathbf{r}_{16,i,j} = \frac{1}{2}\left(\mathbf{r}_{0,i,j} + \mathbf{r}_{6,i,j}\right)$ The corner points determine the linear basis functions $e(\mathbf{r})$ (Fig. 3b, d). We provide the basis functions for the case of irregular triangles, even though we use only the regular case in this paper. Without loss of generality, we give the basis function belonging to corner point $\mathbf{r}_{0,i,j}$ and edge point $\mathbf{r}^{\perp}_{0,i,j}$ defined at $\mathbf{r}^{\perp}_{0,i,j} = \frac{1}{2}\left(\mathbf{r}_{4,i,j} + \mathbf{r}_{6,i,j}\right)$. The unit vector of the mid-line combining corner point $\mathbf{r}_{0,i,j}$ and edge point $\mathbf{r}^{\perp}_{0,i,j}$ is defined as $\mathbf{n}^{m}_{0,i,j} = \frac{\mathbf{r}^{\perp}_{0,i,j} - \mathbf{r}_{0,i,j}}{\left|\mathbf{r}^{\perp}_{0,i,j} - \mathbf{r}_{0,i,j}\right|}$ and the unit perpendicular vector is defined as $\mathbf{n}^{s}_{0,i,j} = \frac{\mathbf{r}_{6,i,j} - \mathbf{r}_{4,i,j}}{\left|\mathbf{r}_{6,i,j} - \mathbf{r}_{4,i,j}\right|}$. We introduce the local coordinates $(\lambda, \mu)$ for each triangular corner as a non-orthogonal coordinate:

$$\mathbf{r} = \lambda \mathbf{n}^{m}_{0,i,j} + \mu \mathbf{n}^{s}_{0,i,j} \tag{3}$$

and define the linear corner basis function $e_0$ at corner point $\mathbf{r}_{0.i,j}$ as

$$e_0(\mathbf{r}) = e_0(\lambda, \mu) = 1 - \frac{2}{\sqrt{3}}\lambda \tag{4}$$

Therefore, each corner point defines a linear basis function $e(\mathbf{r})$ inside the triangle while outside the triangle, these linear basis functions are defined to be zero. This linear basis function, belonging to corner point $\mathbf{r}_{0,i,j}$ is the linear function being 1 at $\mathbf{r}_{0,i,j}$ and 0 at $\mathbf{r}_{4,i,j}$, $\mathbf{r}_{6,i,j}$.

Three second-order basis functions are associated with the edge points $\mathbf{r}_{5,i,j}, \mathbf{r}_{15,i,j}, \mathbf{r}_{16,i,j}$ in the triangle. Without loss of generality, we assume that the center of coordinates is at $\mathbf{r}_{0,i,j}$, meaning $\mathbf{r}_{0,i,j} = 0$, for the definition of this basis function. We define one of the second-order basis functions as $b^2_{p,g}(\mathbf{r})$, which is associated with the edge point $\mathbf{r}_{4,6,i,j} = \frac{1}{2}\left(\mathbf{r}_{4,i,j} + \mathbf{r}_{6,i,j}\right) = \mathbf{r}_{5,i,j}$ (Fig. 3c, e). Then we define $\mathbf{r}^{\perp}_{4,i,j}$ and $\mathbf{r}^{\perp}_{6,i,j}$ to be a set of vectors orthogonal to $\mathbf{r}_{4,i,j}$ and $\mathbf{r}_{6,i,j}$. Therefore, the second-order basis function in this triangle is set as:

$$b^{2}_{p,g}\left(\mathbf{r}\right) = \frac{b'^{2}_{p,g}\left(\mathbf{r}\right)}{\left(\mathbf{r}_{4,i,j} \bullet \mathbf{r}\right)\left(\mathbf{r}_{6,i,j} \bullet \mathbf{r}\right)} \tag{5}$$

Author's personal copy

where

$$b'^2_{p,g}(\mathbf{r}) = \begin{cases} \left(\mathbf{r}^{\perp}_{4,i,j} \bullet \mathbf{r}\right)\left(\mathbf{r}^{\perp}_{6,i,j} \bullet \mathbf{r}\right), & \Delta \text{ with corner points 4,6 in hexagon centered at point 0} \\ 0, & \text{outside } \Delta \text{ with corner points 4,6} \end{cases},$$

the dot between two vectors indicates the dot product and $b^2_{p,g}\left(\mathbf{r}_{4,6,i,j}\right) = b^2_{p,g}\left(\mathbf{r}_{5,i,j}\right) = 1$. Note that all basis functions defined above are discontinuous. If, however, all basis functions belonging to the same corner or edge point are summed, the result will be a continuous basis function. From this follows, that using all corner and edge amplitudes and summing over all basis functions will result in a continuous field.

Now let the field $h(\mathbf{r})$ be given at all corner and edge points of the hexagonal grids. We give the representation formula for triangular amplitudes. For a hexagonal grid, let the corner point amplitudes be $\mathbf{r}_{p,i,j}$ and the edge amplitudes be $\mathbf{r}_{p,q,i,j}$. Note that edge points and edge amplitudes are defined only if $\mathbf{r}_{p,i,j}$ and $\mathbf{r}_{q,i,j}$ are immediate neighbours. Then we have the general form of the field representation for $h(\mathbf{r})$ in the hexagonal grid with center point 0:

$$h(\mathbf{r}) = \sum_{\text{six } \Delta\text{s in one hexagon}} \left[ \sum_{p} h_p e_p(\mathbf{r}) + \sum_{p,q \text{ are neighbours}} h_{p,q} b^2_{p,g}(\mathbf{r}) \right], \tag{6}$$

where $h_p$ is the corner amplitude and $h_{p,q}$ is the edge amplitude. Equation (6) means that the field is determined by corner amplitudes $h_p$ and edge amplitudes $h_{p,q}$, where the corner and edge amplitudes are all dynamic points and the six interior amplitudes are determined by the surrounding amplitudes on the boundaries and the center of the hexagon. Alternatively, there is another intuitive representation of the field $h(\mathbf{r})$ which is defined as:

$$h(\mathbf{r}) = \sum_{\text{six } \Delta\text{s in one hexagon}} \left[ \sum_{p=0}^{9} h_p e^c_{(p,i,j),\Delta}(\mathbf{r}) + \sum_{p=12}^{17} h_p e^c_{(p,i,j),\Delta}(\mathbf{r}) + \sum_{p=6}^{8} h_p e^c_{(p,i,j+1),\Delta}(\mathbf{r}) \right], \tag{7}$$

where $e^c_{(p,i,j),\Delta}(\mathbf{r}) = \begin{cases} 1, & \mathbf{r} \in \Delta \& \left[(p,i,j),\Delta\right] \text{ see in Tab.1} \\ 0, & \text{else} \end{cases}$ are the characteristic basis functions for each point and each triangle in the hexagonal grid with center point 0 (see $(p, \Delta)$ in Table 1). This is hardly used for program implement while is convenient and intuitive to understand how to represent the field. Once the triangles and grid points in one hexagon meet the conditions of the combination in Table 1, the value of the basis function $e^c_{(p,i,j),\Delta}(\mathbf{r})$ can be determined.

Note that the amplitudes of interior points are diagnostic which means that the interior points in the hexagon are unused in the sparse grid scheme. However, unused points can be interpolated to achieve a full grid, which can be used for plotting, if one does not want to go for this purpose to the Cartesian representation. The interpolation of unused points from dynamic points results in smooth plots of fields. If the unused points are filled with the value of zero, the grid structure can be seen in Fig. 4 which shows a plot of a part of the field on the used points, when the field is defined zero on the unused points. The hexagonal structure becomes visible, with the corners, edges and centers showing local maxima, providing a good indication of the hexagonal cell boundaries. According to Fig. 4, the small hexagons inside the large ones are those where the points are unused for dynamics and put to values of zero.

**Table 1** List of triangle numbers $\Delta$ belonging to a grid point $p$ meaning that basis function $e^c_{(p,i,j),\Delta}(\mathbf{r})$ is equal to one being different from zero

| Point $p$ | Triangle number $\Delta$ | Point $p$ | Triangle number $\Delta$ | Point $p$ | Triangle number $\Delta$ |
|---|---|---|---|---|---|
| 0, $i,j$ | (0, $i$, $j$), 1 and (0, $i$, $j$),2 and (0,$i$,$j$),3 and (0, $i$ ,$j$),4 and (0, $i$ ,$j$), 5 and (0, $i$, $j$), 6 | 1, $i$, $j$ | (1, $i$, $j$), 1 and (1, $i$, $j$), 6 | 2, $i$, $j$ | (2, $i$, $j$), 1 |
| 3, $i$, $j$ | (3, $i$, $j$), 3 | 4, $i$,$j$ | (4, $i$,$j$), 3 and (4, $i$, $j$), 4 | 5, $i$, $j$ | (5, $i$, $j$), 4 |
| 6, $i$, $j$ | (6, $i$, $j$), 4 and (6, $i$, $j$), 5 | 7, $i$, $j$ | (7, $i$, $j$), 5 | 8, $i$, $j$ | (8, $i$, $j$), 5 and (8, $i$, $j$), 6 |
| 9, $i$, $j$ | (9, $i$, $j$), 6 | 12, $i$, $j$ | (12, $i$, $j$), 1 and (12, $i$, $j$), 6 | 13, $i$, $j$ | (13, $i$, $j$), 1 and (13, $i$, $j$),2 |
| 14, $i$, $j$ | (14, $i$, $j$), 2 and (14, $i$, $j$),3 | 15, $i$, $j$ | (15, $i$, $j$), 3 and (15, $i$, $j$), 4 | 16, $i$, $j$ | (16, $i$, $j$), 4 and (16, $i$, $j$),5 |
| 17, $i$, $j$ | (17, $i$, $j$), 5 and (17, $i$, $j$),6 | 6, $i$, $j+1$ | (6, $i$, $j+1$), 2 and (6, $i$, $j+1$), 3 | 7, $i$, $j+1$ | (7, $i$, $j+1$), 2 |
| 8, $i$, $j+1$ | (8, $i$, $j+1$), 1 and (8, $i$, $j+1$),2 | | | | |

Author's personal copy

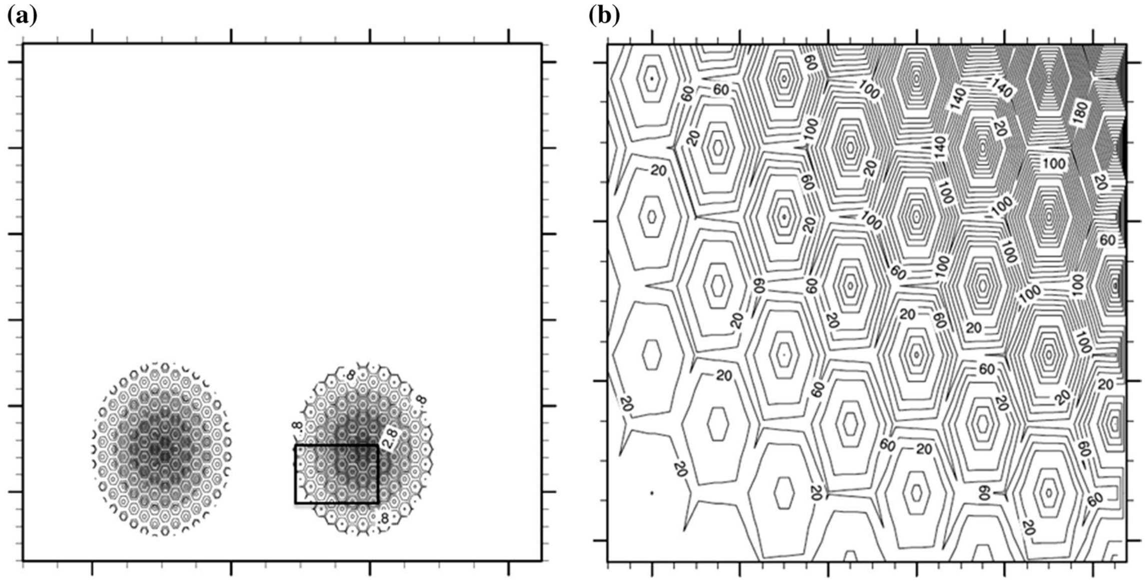

**Fig. 4** A schematics of the field using unconventional plotting. **a** The field and the field shifted to the right (only the used points are shifted to the right). The amplitudes at the unused points remain at their original position. This way of plotting shows fields which add up to a smooth function, if the right part of (**a**) is shifted to the left again. The right and left parts of (**a**) are like the positive and negative of a picture. **b** The details of a part of the field in (**a**) marked with a black rectangle. As the amplitude of unused points is set to zero, the hexagonal grid structure becomes visible

## 3 The construction of finite difference formula

The principle of Galerkin method is to represent fields piecewise as polynomials. Using a number of collocation grid points within each element, the o2o3 discretization on hexagonal cells can be written as a FDM scheme based on the structured index $(i,j)$ as indicated in Fig. 1. For the sparse grid scheme, only half of points (the solid black grid points shown in Fig. 1c) are used, which demonstrates that a compact data structure is possible which may turn out to be handy for realistic applications. Different from the ordinary FDM schemes where the same form of schemes is used in different elements, these schemes use different FDM equations for the collocation points indexed by $k$ in each element $(k, i, j)$ and the formula for the different $k$ have to be programmed each differently. However, for the same $k$ and different $(i, j)$ in Fig. 1c, the FDM schemes are analogous.

In the present paper, a new LGM, implied spectral equation (ISE), is used in which mass conservation in a center point is used to define the corresponding spectral amplitudes, while other amplitudes on the edge and corner points are provided by any high-order FDM scheme with the target order. If all amplitudes of all points in a hexagon except the center point are computed, the balance equation for the amplitude at the center point can be written as an equation for the time derivative of the center amplitude. This is analogous to the procedure used with the o3o3 scheme (Steppeler et al. 2019). This mass balancing ISE method, known from SEM described by Steppeler (1987) for a large range of LGMs will be used to achieve mass conservation when using the quadrature approximation. This means for our method of hexagons that mass conservation needs to be a consideration only for the center point whilst other points can be treated by differentiating along straight lines. It will be shown in the following that how the principle of mass conservation allows to compute the time derivative of the spectral amplitudes at center point which is then transformed to gridpoint space to obtain a conserving scheme. The FDM schemes used may not refer to the spectral representation and will normally use points on a straight line for efficiency. A minimum of one implied spectral amplitude is sufficient to obtain an overall mass-conserving scheme.

In details, we use the advection equation

$$\partial_t h = -u \cdot \partial_x h - v \cdot \partial_y h, \tag{8}$$

as a test where $u, v$ are assumed to be constant and $\partial_t h, \partial_x h, \partial_y h$ indicates the differentiation of variable $h$ with respective to $t$ (time), $x$ and $y$ (location).

Most derivatives used to compute the divergence are obtained by differencing along lines and taking directional derivatives. Let $\mathbf{n}_l = \left(n_l^x, n_l^y\right)$ be any unit vector along a line $l$. The $l$-directional derivative of a function $h\left(\mathbf{r} + l \cdot \mathbf{n}_l\right)$ is defined as:

$$\begin{aligned}\partial_l h\left(\mathbf{r} + l \cdot \mathbf{n}_l\right) = {} & n_l^x \partial_x h\left(r^x + l \cdot n_l^x, r^y + l \cdot n_l^y\right) \\ & + n_l^y \partial_y h\left(r^x + l \cdot n_l^x, r^y + l \cdot n_l^y\right),\end{aligned} \tag{9}$$

Author's personal copy

where $\mathbf{r} = (r^x, r^y)$ is the position vector of the corresponding point and $l$ is the differential length along the unit vector $\mathbf{n}_l$ direction. In the following, it is assumed that $l = 1$, without loss of generality and take $\mathbf{n}_l = (1, 0)$ and $u = 1$ as an example. Therefore, Eq. (8) becomes

$$\partial_t h = -u \cdot \partial_x h = -\partial_x h, \tag{10}$$

where the advection is homogeneous with a constant velocity field. Note that if we have calculated the FDM formula for $\partial_x h$, we can obtain the FDM formula for $\partial_t h$ and vice versa. For our simple example the time derivative is the space derivative.

According to Fig. 1c, the first-order directional derivatives at edge points 2, 3, 5, 7, 9, 10 will be called $\partial_t h = -\partial_l h(\mathbf{r}_{k,i,j}) = -\delta_x h(\mathbf{r}_{k,i,j})$ and for example $k = 7$ in Fig. 1c is computed by:

$$\delta_x h(\mathbf{r}_{7,i,j}) = h(\mathbf{r}_{6,i,j}) - h(\mathbf{r}_{8,i,j}), \tag{11}$$

which is given to compute flux divergences or derivatives in $x$- and $y$-directions in edge points.

The second-order derivative $\delta_x^2 h_{7,i',j'}$ at point $k = 7$ in Fig. 1c is computed by:

$$\delta_x^2 h(\mathbf{r}_{7,i,j}) = -8\left\{ h(\mathbf{r}_{7,i,j}) - \tfrac{1}{2}\left[h(\mathbf{r}_{6,i,j}) + h(\mathbf{r}_{8,i,j})\right] \right\}. \tag{12}$$

For the third-order derivative at the same point we define:

$$\delta_x^3 h(\mathbf{r}_{7,i,j}) = -\tfrac{7}{16}\left[h(\mathbf{r}_{6,i,j}) - h(\mathbf{r}_{8,i,j})\right] - \tfrac{1}{48}\left[h(\mathbf{r}_{11,i,j}) - h(\mathbf{r}_{11,i-1,j})\right]. \tag{13}$$

For the other edge points, the formula is analogous.

A third-order accurate formula for the directional derivative along the line $l$ between the points 8 and 6 in Fig. 1c is:

$$\partial_l h = \delta_x h(\mathbf{r}_{7,i,j}) + \delta_x^2 h(\mathbf{r}_{7,i,j}) w_{i,j}^2(\mathbf{r}_{7,i,j}) + \delta_x^3 h(\mathbf{r}_{7,i,j}) w_{i,j}^3(\mathbf{r}_{7,i,j}). \tag{14}$$

where $w_{i,j}^2$ and $w_{i,j}^3$ are the weight coefficients and $\delta_x^2 h(\mathbf{r}_{7,i,j}) w_{i,j}^2(\mathbf{r}_{7,i,j}) = 0$.

The derivations Eqs. (11)–(14) use the relations between third-, second- or first-order derivatives along straight lines combining points 8 and 6. Coordinate lines can be arbitrarily tilted to each other, but for the success of this method, it is necessary that some pieces of the lines are not curved, so that numerical differentiation along the lines is possible. The equations may appear rather difficult (see a considerable simplification and details in Appendix 1). The third-order derivative is obtained from point 7 in Fig. 1c and four more points being on a straight line. For practical modelers, it may be easier to differentiate in third-order approximation along this line, for which five points are sufficient. Unfortunately, the points are irregular, as the two points out of the hexagon have three times the distance from the center point $k = 7$ as the two interior points $k = 6$ and 8. Therefore, if weights for irregular differentiation are obtained numerically, the rather complicated derivations above could be replaced by simple finite differences on irregular grids on lines which are not necessarily in $x$- or $y$-directions. For example, the high-order derivative computed in Eq. (11) is obtained by

$$\begin{aligned}\partial_x h(\mathbf{r}_{7,i,j}) = {} & w_1 h(\mathbf{r}_{11,i,j}) + w_2 h(\mathbf{r}_{6,i,j}) \\ & + w_3 h(\mathbf{r}_{7,i,j}) + w_4 h(\mathbf{r}_{8,i,j}) + w_5 h(\mathbf{r}_{11,i-1,j}),\end{aligned} \tag{15}$$

whose indices are taken from Fig. 1c. The weights $w_1, w_2, w_3, w_4, w_5$ are obtained numerically for $l = 1$ as $w_1 = -\frac{1}{48}$, $w_2 = \frac{9}{16}$, $w_3 = -0.0$, $w_4 = -\frac{9}{16}$, $w_5 = \frac{1}{48}$. The numerical procedure can also handle the more irregular case, which would appear for discretizations on the sphere. It is in principle possible to compute fluxes in $x$- or $y$-directions using just two directional derivatives (see below). This was done by Steppeler et al. (2008) where the stability was achieved because the directional derivatives were taken as centered differences.

Here we have three directional derivatives available at corner points 1, 4, 6, 8. At corner points, as the derivatives are one-sided, an average of all three directional derivatives will be needed generally to have the chance of a stable scheme. To have a guide for the averaging of derivatives at corner points, infinitesimal control volumes are used. Figure 5 shows a FDM stencil at a corner point. It could be identical to the stencil with target point 4 in Fig. 1c. A heuristic consideration is done to obtain reasonable weights for averaging the derivative at the target point 4. We define the directional vectors as $\mathbf{n}_{k,tg} = \mathbf{r}_k - \mathbf{r}_{tg}$ (here, $k$ denotes one of three points linked to the target point $tg = 4$, $i$, $j$, see Fig. 5) and the corresponding directional derivative $\partial_{\mathbf{n}_{k,tg}} h_{tg}$ at the target point $tg$. We can obtain three directional vectors $\mathbf{n}_{1,i+1,j;4,i,j} = \mathbf{r}_{1,i+1,j} - \mathbf{r}_{4,i,j}, \mathbf{n}_{6,i,j;4,i,j} = \mathbf{r}_{6,i,j} - \mathbf{r}_{4,i,j}, \mathbf{n}_{6,i,j+1;4,i,j} = \mathbf{r}_{6,i,j+1} - \mathbf{r}_{4,i,j}$ observing in Fig. 5. Here we describe the $x$-direction advection equation. The $y$-direction is treated in

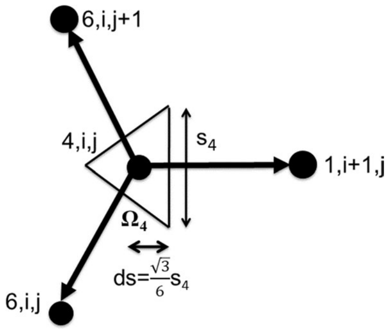

**Fig. 5** Infinitesimal control volume: the computation of $x$- and $y$-derivatives from directional derivatives along cell edges

Author's personal copy

an analogous way. Then, we obtain the integral of the flux $fl\left(\mathbf{r}_{4,i,j}\right)$ through the surface in Fig. 5:

$$\oint_{\Omega_4} fl\left(\boldsymbol{r}_{4,i,j}\right)\mathbf{n}\cdot d\mathbf{s} = \sum_{(k,i',j')=(6,i,j+1),(6,i,j),(1,i+1,j)} \left[\partial_{\mathbf{n}_{k,i',j';4,i,j}} h\left(\mathbf{r}_{4,i,j}\right)\cdot\left(\tfrac{1}{3}S_\Delta\right)\right]\mathbf{n}_{k,i',j';4,i,j}\cdot\mathbf{i}, \tag{16}$$

where the surface integral is taken over the triangle $\Omega_4$ with the size of $S_\Delta = \frac{\sqrt{3}}{4}s_4^2$ and $\mathbf{n}$ is the unit vector orthogonal to the boundary of the triangle $\Omega_4$ (Fig. 5).

For the $x$-derivative $d_x h_4$ we obtain:

$$\begin{aligned}\partial_x h\left(\mathbf{r}_{4,i,j}\right) &= \frac{\oint_{\Omega_4} fl\left(\boldsymbol{r}_{4,i,j}\right)\boldsymbol{n}\cdot ds}{S_\Delta} \\ &= \tfrac{1}{3}\Big[\partial_{\mathbf{n}_{1,i+1,j;4,i,j}} h\left(\mathbf{r}_{4,i,j}\right) - \tfrac{1}{2}\partial_{\mathbf{n}_{6,i,j;4,i,j}} h\left(\mathbf{r}_{4,i,j}\right) \\ &\quad -\tfrac{1}{2}\partial_{\mathbf{n}_{6,i,j+1;4,i,j}} h\left(\mathbf{r}_{4,i,j}\right)\Big],\end{aligned} \tag{17}$$

where the area of the triangle $S_\Delta$, $\mathbf{n}_{1,i+1,j;4,i,j}\cdot\mathbf{i} = 1$, $\mathbf{n}_{6,i,j;4,i,j}\cdot\mathbf{i} = \left|\mathbf{n}_{6,i,j;4,i,j}\right||\mathbf{i}|\cdot\cos 120^\circ = -\frac{1}{2}$ and $\mathbf{n}_{6,i,j+1;4,i,j}\cdot\mathbf{i} = \left|\mathbf{n}_{6,i,j+1;4,i,j}\right||\mathbf{i}|\cdot\cos 210^\circ = -\frac{1}{2}$. The other corner points and $y$-derivatives are obtained by analogous formula.

We aim for a third-order approximation and therefore any third-order FDM scheme may be used. So far, we have defined FDM schemes for all collocation points being on the boundary of a hexagon. Using the points on the grid boundary, the flux around the surface of the hexagon into the grid is equal to the time derivative of the mass inside the grid. The corner and edge amplitudes were obtained by above high-order FDM schemes for the hexagon. The approximation for the center amplitude will be then obtained by requiring mass balance. The center amplitude will be chosen in such a way that the time derivative of mass is consistent with the flux though the boundary. Mass conservation defines the implied amplitude uniquely.

We set the center point indexed $k=0$ in the element $(k, i, j)$ as an example. For any point $x$, $y$ inside the hexagon with center $(0, i, j)$, we have the piecewise quadratic representation using the basis functions defined in Sect. 2. The sum given in the following formula represents the field limited to the chosen hexagon. Using the collocation basis defined in Eq. (6), the field representation for the chosen hexagon with center $(0, i, j)$ is:

$$\begin{aligned}\partial_t h(\mathbf{r}) = \sum_{\Delta=1}^{6}\partial_t h_0\cdot e^c_{(0,i,j),\Delta} + \sum_{k=1}^{9}\sum_{\Delta=1}^{6}\partial_t h_k\cdot e^c_{(k,i,j),\Delta} \\ + \sum_{k=6}^{8}\sum_{\Delta=1}^{6}\partial_t h_k\cdot e^c_{(k,i,j+1),\Delta} + \sum_{k=12}^{17}\sum_{\Delta=1}^{6}\partial_t \bar{h}_k\cdot e^c_{(k,i,j),\Delta},\end{aligned} \tag{18}$$

where $\partial_t \bar{h}_k$ for $k=12, 13, \ldots, 17$ are diagnostically determined by the dynamic points $k=0, 1, \ldots, 9$ in the element $(k, i, j)$ and $k=6, 7, 8$ in the element $(k, i, j+1)$. Note that except $\partial_t h_0$ in Eq. (18), all other amplitudes $\partial_t h_i$ have already been computed. Therefore, it can easily be solved for $\partial_t h_0$. Let $m_k^0 = \sum_{\Delta=1}^{6}\iint e^c_{(k,i,j),\Delta}(\mathbf{r})dxdy$ be the mass of the collocation basis functions, as for the time derivative of mass $\partial_t M = \int \partial_t h\cdot dt$ of $\partial_t h$ inside the chosen hexagon we obtain:

$$\partial_t M = \sum_{k=0}^{9}\partial_t h_k m^0_{(k,i,j)} + \sum_{k=6}^{8}\partial_t h_k m^0_{(k,i,j+1)} + \sum_{k=12}^{17}\partial_t \bar{h}_k m^0_{(k,i,j)}, \tag{19}$$

where the $m_k^0$ is the mass contained in the characteristic basis functions: $m^0_{(k,i,j)} = \sum_{\Delta=1}^{6}\iint e^c_{(k,i,j),\Delta}(\mathbf{r})dxdy$ for $k=-6, \ldots, 12$. $\partial_t\, M$ in Eq. (19) is known from the flux balance equation. Let $fl(\mathbf{r})$ be the flux of field $h$ at the center point $k=0$. Then, the surface integral of the flux is equal to $\partial_t M$:

$$\oint_{\Omega_0} fl(\mathbf{r})\mathbf{n}_0\cdot d\mathbf{s} = \partial_t M, \tag{20}$$

where the surface integral is taken over the boundary $\Omega_0$ which are the edges of the hexagon and $\mathbf{n_0}$ is the unit vector orthogonal to the hexagonal surface. Combining Eqs. (18)–(20), the only unknown $\partial_t h_0$ can therefore be solved.

The hexagonal example described above may be better understood by a 1-D example, which is given in the following:

Consider the 1-D homogeneous advection equation $\partial_t h = -\partial_x h$. The grid is $0, \frac{1}{2}, 1, \frac{3}{2}, 2, \ldots$. The basis functions are defined in Eqs. (3)–(5). The grid is the interval $\left(x_i, x_{i+1}\right)$ where $i=0, 1, 2, 3, \ldots$. The time derivative of the mass inside $\left(x_i, x_{i+1}\right)$ is the flux difference.

Let the field $h(x)$ be defined by

$$h(x) = \sum_i h_i e_i(x) + A_i b^2_{i+\frac{1}{2}}(x) \tag{21}$$

with $e_i(x)$ being the piecewise linear function being 1 at point $x = x_i$ and being 0 at all other points ($x \in \left(x_{i-1}, x_{i+1}\right)$), $b^2_{i+\frac{1}{2}}(x) = \frac{1}{2}\left(x^2 - \frac{ds^2}{4}\right)$ where $ds$ is the distance of a cell interval, $A_i$ is the spectral amplitude belonging to the basis $b_i^2(x), i = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \ldots$. The integral of $b^2(x)$ is $-\frac{1}{12}ds^3$. The integral of the linear basis functions is $\frac{1}{2}ds$. The surface flux difference is $-\left(h_{i+1} - h_i\right)$ $(i=0, 1, 2, \ldots)$. Then, Eq. (21) becomes:

$$-\left(h_{i+1} - h_i\right) = \tfrac{1}{2}\left(\partial_t h_{i+1} + \partial_t h_i\right)ds - \tfrac{\partial_t A_i}{12}ds^3 \tag{22}$$

Author's personal copy

As $b^2(0) = -\frac{1}{8}ds^2$ and $\partial_t h_{i+\frac{1}{2}} = \frac{1}{2}\left(\partial_t h_{i+1} + \partial_t h_i\right) + \partial_t A_i b^2(0)$, we have

$$\partial_t h_{i+\frac{1}{2}} = -\frac{3}{2ds}\left(h_{i+1} - h_i\right) - \frac{1}{4}\left(\partial_t h_{i+1} + \partial_t h_i\right) \tag{23}$$

In Eq. (23) $\partial_t h_i$ can be chosen arbitrarily and we use:

$$\partial_t h_i = \frac{a}{ds}\left(h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}}\right) + \frac{b}{2ds}\left(h_{i+1} - h_{i-1}\right) \tag{24}$$

where coefficients $a$ and $b$ are defined in Steppeler et al. (2008).

Equations (23) and (24) are a 1-D version of the o2o3 scheme used with the hexagon. This is the scheme defined as the o2o3 scheme in Ahlberg et al. (1967) without any tests.

## 4 Advection experiments

The aim of these advection experiments is to demonstrate that the sparse SEM describes the process properly. This means that the accuracy expected from the third-order discretization scheme is achieved and that the results do not indicate a serious shortcoming due to the sparseness. This should also be the case for marginal resolution. Second- and third-order LGMs were proposed and used on squares in Steppeler (1976). In this study, the LGM operation has further implemented on hexagons. We want to confirm that the methods presented here give results of about similar quality.

Before the initialization of the advection test, we list the steps of the time loop for further explanation of LGM.

- Compute directional derivatives at corner and edge points [$k = 1, 2, \ldots, 9$ in the element $(k, i, j)$ and $k = 6, 7, 8$ in the element $(k, i, j{+}1)$ in Fig. 1c].
- Compute time derivatives at dynamic corner and edge points [$k = 1, 2, \ldots, 9$ in the element $(k, i, j)$ and $k = 6, 7, 8$ in the element $(k, i, j{+}1)$ in Fig. 1c]. At the corner points [$k = 1, 4, 6, 8$ in the element $(k, i, j)$ and $k = 6, 8$ in the element $(k, i, j{+}1)$ in Fig. 1c], average the three directional derivatives of the lines meeting here.
- Perform ISE LGM at center points [$k = 0$ in the element $(k, i, j)$ of Fig. 1c].
- Perform fourth-order Runge–Kutta (RK4) time stepping in gridpoint space.

No transformation of spectral coefficients to grid point space is necessary, as the grid point at the center ($k = 0$) is both spectral coefficient and spectral amplitude. Spectral coefficients which are different from gridpoint values occur only at edge amplitudes [$k = 2, 3, 5, 7, 9$ in the element $(k, i, j)$ and $k = 7$ in the element $(k, i, j{+}1)$ in Fig. 1c].

Now we perform the advection test. For the experiments done here, we assume the side length $s$ of the hexagon to be $s = 2$. Figure 6 gives the result of advection for a computational domain of 400 * 600 grid point where $dx = 1$ and $dy = \frac{\sqrt{3}}{2}$ between every two points, and the timestep $dt$ is 1.0 s with 90 timesteps. The components of advection velocity $u$ and $v$ are (1, 1) for the first thirty timesteps, (0, − 1) for the second thirty timesteps and (− 1, 0) for the last thirty timesteps. To obtain a solution where the initial field is reproduced after ninety timesteps, this velocity field is changed in time. In Fig. 6, thirty timesteps are done between plots and the plots are produced to show the sparse grid structure. The unused points are assigned a time derivative of zero. The initial values are shown in Fig. 6a and are given by:

$$\rho^h_{i,j} = \rho^h_{i,j}\left(x^h_i, y^h_j\right) = \exp\left[-\left(\frac{x^h_i - x^h_0}{dx_{mid}}\right)^2 - \left(\frac{y^h_j - y^h_0}{dx_{mid}}\right)^2\right], \tag{25}$$

where $\rho^h_{i,j}$ is the density of the tracer, $\left(x^h_i, y^h_j\right)$ is the location of the density, $\left(x^h_0, y^h_0\right) = (110, 65)$ is the initial location of the density and $dx_{\mathrm{mid}}$ is the radii of the density. In Steppeler and Klemp (2017), it was determined by 1-D experiments that for $dx_{\mathrm{mid}} = 8$ the second-order FDM scheme based on centered differences used there produced reasonable dispersion-free results, whereas heavy dispersion occurred for smaller values of $dx_{\mathrm{mid}}$. Since the third-order approximation is used in this study, we may expect to obtain a reasonably dispersion-free results for $dx_{\mathrm{mid}} = 4$.

Figure 6a shows the initial values for $dx_{\mathrm{mid}} = 4$. This is the same resolution as used in Steppeler and Klemp (2017). Of course, we may have better dispersion properties because using the third-order method. In Fig. 6a, Eq. (25) was used to compute all twenty-four points (twelve dynamic points and twelve unused points) for each $(i, j)$. Note that according to Fig. 1, the discretization group of points has twelve dynamic points and twelve diagnostic points which all are independent. These are more than the points belonging to a hexagon, as the discretization group is larger than one hexagon, as seen from Fig. 1. These plots are done using the NCAR Command Language (NCL) package. NCL is able to plot irregular distributions of amplitudes, as may occur with hexagonal grids. This way of plotting is called unconventional plotting. When by advection the structure is moving away from its initial position, unconventional plotting will make the hexagonal grid visible. Initial values at the unused points always remain visible. When the structure has moved away from its initial position this means that the unused points have amplitude 0, at the position of the structure and at the position of the initial field we see a negative picture of the solution, showing the values according to Eq. (25) at the unused points.

Author's personal copy

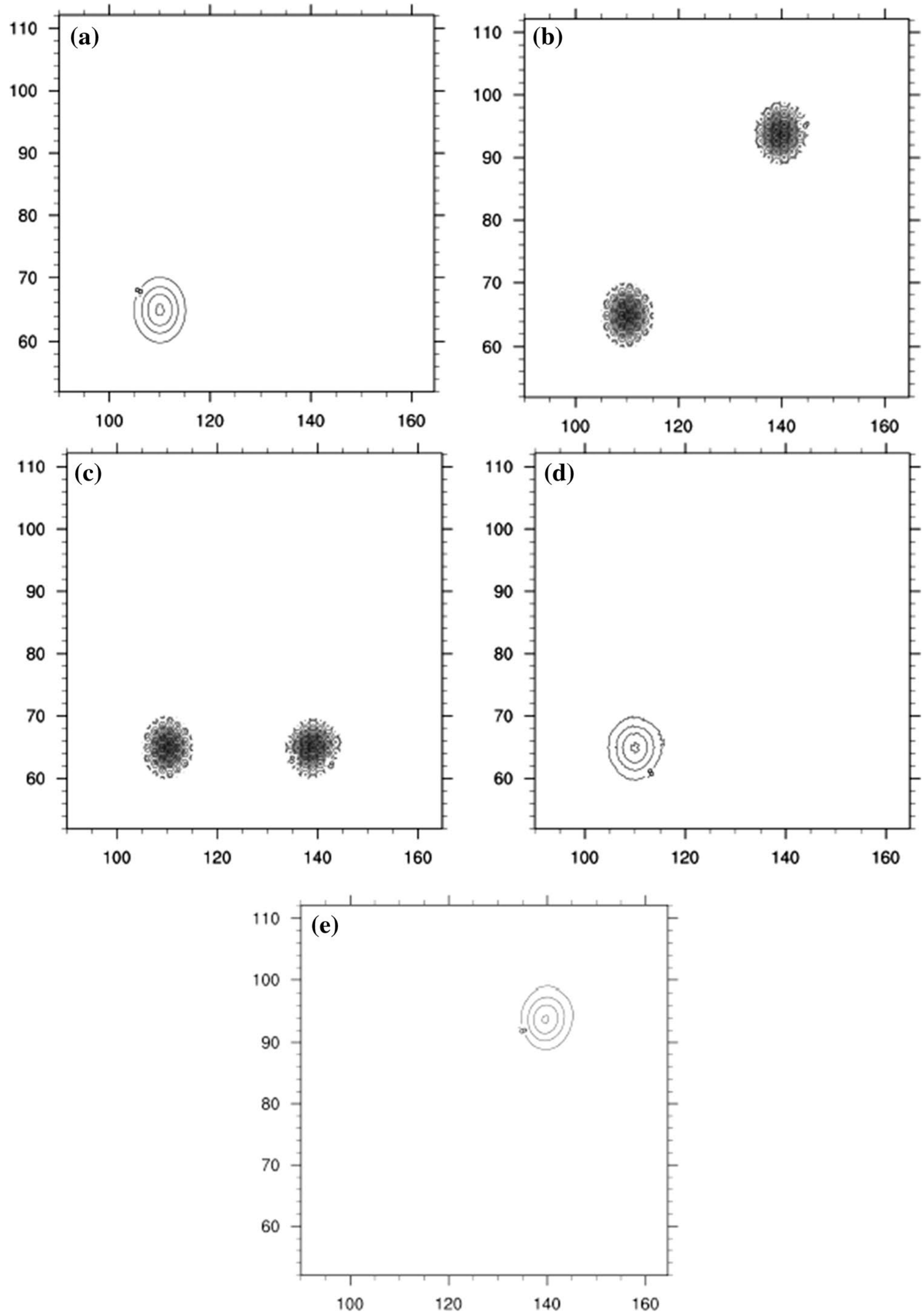

**Fig. 6** The advection results with time varying velocity field for $dx_{mid}$ = 4. The initial value of the density field is shown in (**a**). The advection results at thirty steps, sixty steps and ninety steps are shown in (**b**–**d**). In **b**, **c** the values of the density field are the initial one at the left-hand side and the numerical result at the right-hand side. The values of the field are set as 0 at the unused points where sparse grids are revealed. **e** is the results at thirty steps except that the field for dynamically unused points is interpolated from the dynamic points. In this way, the decrease of the maximum due to numerical dispersion can be seen. As expected for a third-order method, this decrease is not to be noticed for a transport of 30 s

We use the RK4 scheme for time stepping with $dt = 1$. Note that the RK4 time scheme with spatial centered differences has a Courant–Friedrichs–Levy (CFL) stability condition of 2.8. Together with geometric considerations given by Steppeler and Klemp (2017), the CFL condition for o2o3 will be 0.75 × 2.8. The fourth-order spatial differencing in comparison with lower order centered differences leads to a smaller CFL condition (see Durran 2010). This leads to an estimate of the CFL number 1.4. Therefore, we may expect a stable solution using a timestep $dt = 1$. We use thirty timesteps with $u = 1$, $v = 1$, thirty timesteps with $u = 0$, $v = -1$ and thirty timesteps with $u = -1$, $v = 0$. After this the analytic solution is identical to the initial values. Experiments with varying timesteps confirmed this estimate.

The result is shown in Fig. 6b–e. The result shown in Fig. 6d have a slight difference from Fig. 6a by a small

Author's personal copy

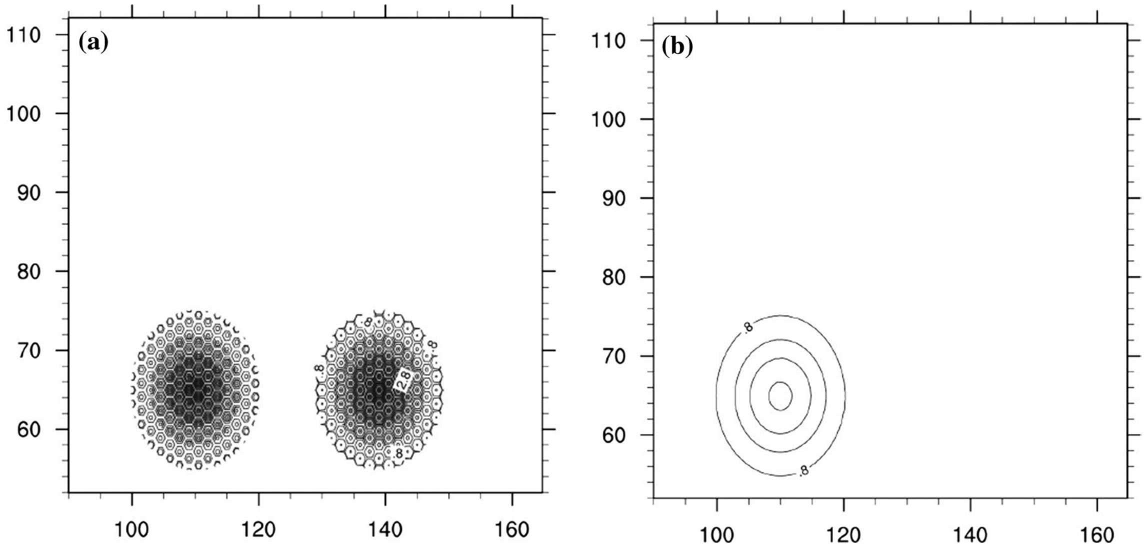

**Fig. 7** The advection results with time varying velocity field for a larger scale field ($dx_{\text{mid}} = 8$). **a** The advection results at sixty steps where the values of the density field are the initial one at the left-hand side and the numerical result at the right-hand side. The values of the field are set as 0 at the unused points where sparse grids are revealed. **b** The results at ninety steps except that the field for dynamically unused points is interpolated from the dynamic points. In **d** no interpolation is used, as the structure has been back to its initial place where the two parts of the field combine to give a smooth structure. **e** Corresponds to **b** and contains the interpolation of the unused points

numerical error. However, the plots given in Fig. 6b do not easily allow to see this, as the field is plotted on the sparse grid. Figure 6e is the same as Fig. 6b, but plotted on the full grid, where the points unused for dynamics are interpolated. The unused points for Fig. 6d are not interpolated. They have the values they had in the beginning, as the structure has reached the initial position again. The two parts of the field combine to give a smooth structure which allows to see an eventual change of maximum due to a dispersion.

Figures 7 and 8 show the same advection tests with a different $dx_{\text{mid}}$ for the initial density in Eq. (25) where $dx_{\text{mid}} = 8$ and $dx_{\text{mid}} = 1$. Figures 7a, b show only the last two steps for a smooth solution. The error in reproducing the initial field is reduced. Figure 8 is the same for a small-scale field, which also reasonably reproduces the initial field. The prediction of a very small-scale field is shown in Fig. 8. To indicate the small resolution, the hexagonal grid points are shown in Fig. 8. It is clear that in each linear cross section there are very few points supporting the structure, but the 2-D grid provides more points. In Appendix 1, we investigate the resolution of plane waves in the hexagonal grid in comparison with a square grid. It is shown in Appendix 1 that plane waves in the hexagonal grid are better resolved than in square grids.

Then, we conducted another advection experiment where the components of advection velocity $u$ and $v$ are (1, 0) and the initial density field is defined as follows:

$$\rho_{i,j}^{h} = \rho_{i,j}^{h}\left(x_i^h, y_j^h\right)$$
$$= \begin{cases} 4\exp\left[-\left(\frac{x_i^h-60}{dx_{\text{mid}}}\right)^2 - \left(\frac{y_j^h-210}{dx_{\text{mid}}}\right)^2\right], \left(x_0^h < 60, y_0^h < 210\right) \\ 4\exp\left[-\left(\frac{x_i^h-80}{dx_{\text{mid}}}\right)^2 - \left(\frac{y_j^h-210}{dx_{\text{mid}}}\right)^2\right], \left(x_0^h > 80, y_0^h < 210\right) \\ 4\exp\left[-\left(\frac{x_i^h-60}{dx_{\text{mid}}}\right)^2 - \left(\frac{y_j^h-230}{dx_{\text{mid}}}\right)^2\right], \left(x_0^h < 60, y_0^h > 230\right) \\ 4\exp\left[-\left(\frac{x_i^h-80}{dx_{\text{mid}}}\right)^2 - \left(\frac{y_j^h-230}{dx_{\text{mid}}}\right)^2\right], \left(x_0^h > 80, y_0^h > 230\right) \\ 4\exp\left[-\left(\frac{x_i^h-80}{dx_{\text{mid}}}\right)^2\right], \left(x_0^h > 80, 210 < y_0^h < 230\right) \\ 4\exp\left[-\left(\frac{x_i^h-60}{dx_{\text{mid}}}\right)^2\right], \left(x_0^h < 60, 210 < y_0^h < 230\right) \\ 4\exp\left[-\left(\frac{y_j^h-210}{dx_{\text{mid}}}\right)^2\right], \left(60 < x_0^h < 80, y_0^h < 210\right) \\ 4\exp\left[-\left(\frac{y_j^h-230}{dx_{\text{mid}}}\right)^2\right], \left(60 < x_0^h < 80, y_0^h > 230\right) \\ 4, \left(60 < x_0^h < 80, 210 < y_0^h < 230\right) \end{cases}, \tag{26}$$

which is used to test the sensitivity of the new LGM scheme. The density field is constructed to have sharper contours and is nearly the characteristic function of a rectangle with

Author's personal copy

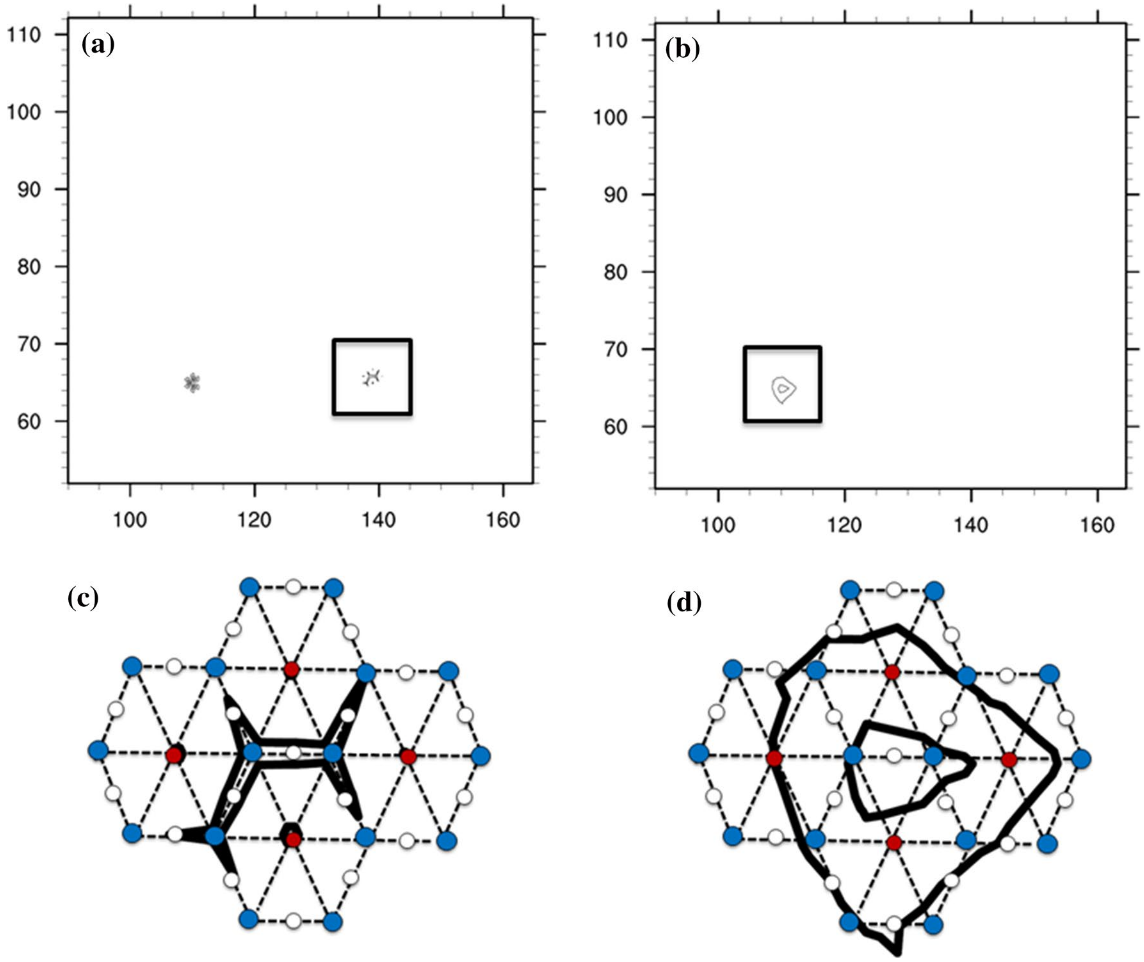

**Fig. 8** As Fig. 7, for a small-scale field ($dx_{\mathrm{mid}} = 1$), which represents a rather poor resolution. **a** initial field and forcasted field at sixty step, **b** the forcasted step at ninety step where the initial field is recreated again, **c** is a part of (**a**) marked with a black rectangle, but enlarged and with the hexagonal grid, **d** as (**c**), but using interpolated valued at the unused points. The red points in **c** and **d** carry amplitudes and possess small isolines which are covered by the red dots and a little black shadow is seen

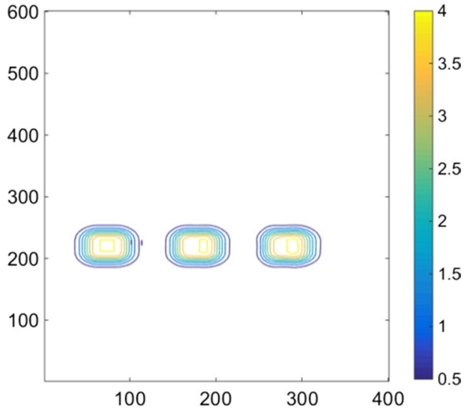

**Fig. 9** Advection of a field with strong gradients of a transport over 210 $dx$ ($dx = 1$) by an integration of 210 timesteps. The center of the initial tracer is located at (70, 220) and the other two results show the tracers at $t = 105$ s and $t = 210$ s. The deformation of the field by dispersion is small. A standard way of plotting is used, interpolating the unused amplitudes from the dynamic points

slightly rounded contours. The timestep $dt$ is 1.0 s with 210 timesteps while all other parameters share with the previous advection experiments. The advection in $x$-direction is shown in Fig. 9 which shows that the hexagonal grid is able to handle strong gradients. A standard way of plotting is used, interpolating the unused amplitudes from the dynamic points.

The scale of the structures used in the hexagonal grid was determined from 1-D experiments. The initial values were such that a small reduction of amplitude due to dispersion was presented. The same choice as in Steppeler and Klemp (2017) was used. We investigate this using a 1-D version of a o2o3 scheme. For comparison, we plot the result of the second-order spectral element method (SEM2, Taylor et al. 1997) which is known to converge in second order. A convergence test was done by computing the 1-D advection while systematically changing the resolution.

Author's personal copy

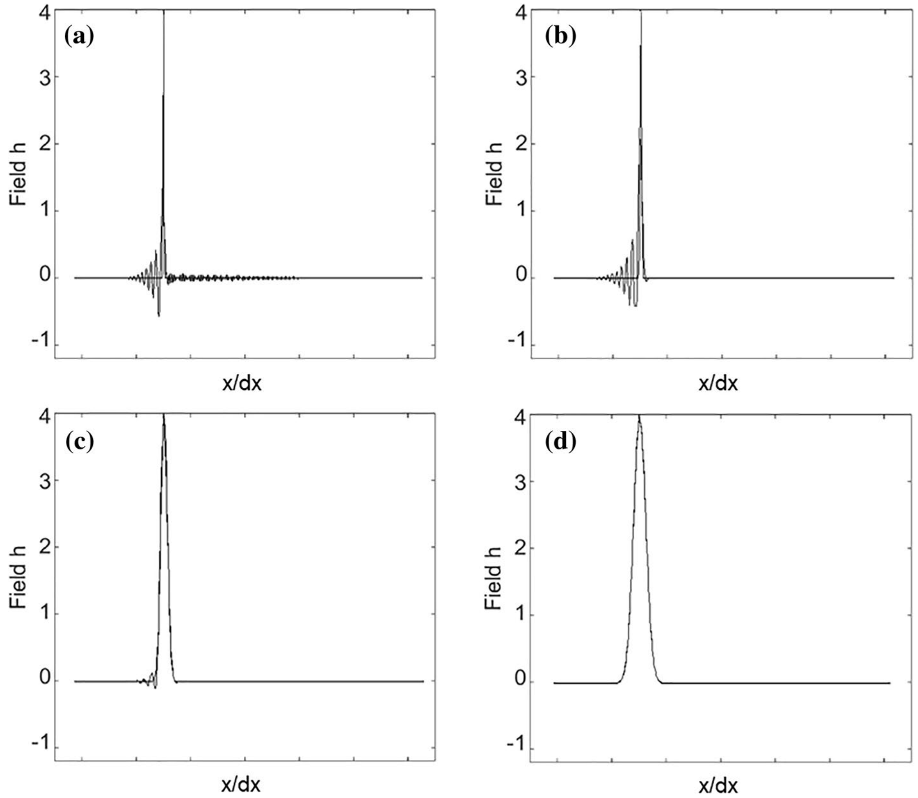

**Fig. 10** The advection over *30dx* for different resolutions. The *x*-axis is *x*/*dx*. **a** *dx* = 1.0, **b** *dx* = 0.5, **c** *dx* = 0.25 and **d** *dx* = 0.125

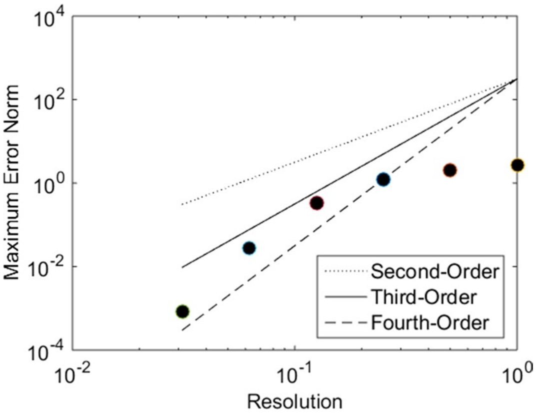

**Fig. 11** Maximum error norms (dotted scatters) of third-order sparse SEM for different resolutions, starting from minimum resolution of *dx* = 0.03125 and doubling it by factors 2, 4, 8, 16, 32. The other three lines are the second-order (dotted), third-order (solid) and fourth-order (dashed) convergences for comparison

The parameters are the same as the experiments in Fig. 6 except the resolution *dx*. The results are shown in Fig. 10. The resolution was changed systematically starting to 0.25 of the resolution used in Fig. 6 to four times this resolution. The errors, using the maximum norm, are given as Fig. 11. The convergence as seen in Fig. 11 is certainly third and appears to reach fourth order by super-convergence, which is observed also with SEM3 methods. From Fig. 11, it can be seen that the high-order scheme at poor resolutions retains a rather high level of accuracy. The level of accuracy seen with the diagrams for the highest resolution is certainly more than needed in practice. For practical purposes, it is less important how accurate a scheme is at very high resolutions, but rather, how soon a reasonable accuracy is obtained.

## 5 Conclusions

Some simple experiments show that advection using sparse hexagonal grids is possible in third order with a piecewise quadratic polynomial representation. Using the o2o3 LGM, the 2-D sparse grid used only half the number of grid points as the full grid, meaning six points per cell against twelve points. The particular mapping of the cells and grid points to a structured *i*, *j* grid turned out to lead to a rather complicated computer program, which may be difficult to develop to further applications. Future developments will depend on defining a more convenient numbering of cells and grid points (see details in Appendix 1).

**Acknowledgements** No authors reported any potential conflicts of interest. The paper was made possible by a number of research visits of the first author at NCAR, financed by NCAR. The city of Bad Orb supported this cooperation by providing office space. Dr. J. Li acknowledges the support of the National Natural Science Foundation of China (Grant No. 41905093), the China Postdoctoral Science Foundation

Author's personal copy

Funded Project (Grant No. 2016M601101) and China Scholarship Council (No. 201904910136). Dr. F. Fang acknowledges the support of the Innovation of the Chinese Academy of Sciences International Partnership Project (Grant No. Y56601M601) and the EPSRC grant: Managing Air for Green Inner Cities (MAGIC) (EP/N010221/1).

## Appendix 1

### A system of indices for second-order hexagonal cells creating a compact grid representation and the 1-D grids for plane waves

This appendix introduces an alternative system of indices to that used in Sects. 2 and 3. The grid is shown in Fig. 12 and it differs from Fig. 1 by assigning different numbers to the points. Each point has two indices $i$ and $k$, where $i$ indicates the hexagon and $k$ indicated the position of a point within the hexagon. $k=0$ indicates the center point of the hexagonal cell. The index $i$ can be a structured index $i=(i', j')$ $(i', j'=0, 1, 2, 3, \ldots)$ or unstructured. The structured case is shown in Fig. 12, where the grids and their center points are indicated by the double index $(i', j')$. The advantage of a structured grid notation is that neighboring grids are always known. For example, in Fig. 12 the cell $(i', j'+1)$ is always above the cell $(i', j')$. Our definition can also be applied to the unstructured grid case. In the unstructured case a list is needed indicating the six neighbouring hexagons to the hexagon $i$. The grid based compact grid pointed out in the following can be used with structures and unstructured index $i$, where Fig. 12 refers to the structured case $i=(i', j')$.

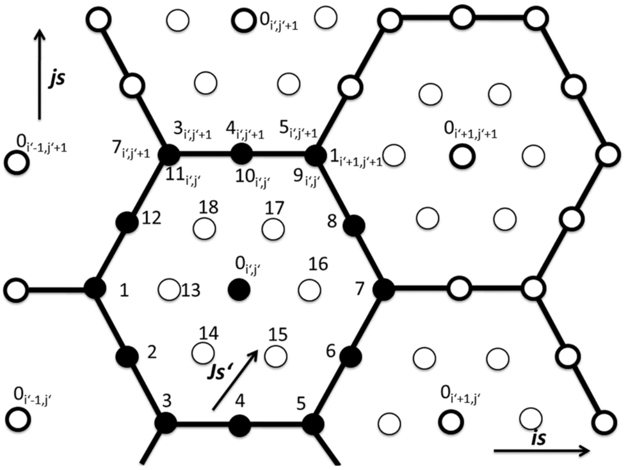

**Fig. 12** As Fig. 1 except that with new indices creating a compact grid. Black points and points with thick circles are dynamic points, those with thin circles are diagnostic points. The points shown in black are the points for compact representation. They do not belong to any other hexagon and each dynamic point appears as a compact point of some hexagons

Let the side length of the hexagon be $s$. For the corner nodes $\mathbf{r}_{i,k}$ $(k=1, 3, 5, 7, 9, 11)$ we have

$$\mathbf{r}_{i,k} = \mathbf{r}_{i,0} + \left(s \cdot \cos\left(\pi + (k-1)\frac{\pi}{3}\right), s \cdot \sin\left(\pi + (k-1)\frac{\pi}{3}\right)\right) \tag{27}$$

The edge nodes $k=2, 4, 6, 8, 12$ are obtained by

$$\mathbf{r}_{i,k} = \mathbf{r}_{i,k-1} + \frac{\sqrt{3}}{2}\left(s \cdot \cos\frac{\pi}{3}, s \cdot \sin\frac{\pi}{3}\right) \tag{28}$$

For the diagnostic inner points $k=13$ to 18, we have

$$\begin{aligned}\mathbf{r}_{i,k} &= \mathbf{r}_{i,0} + \frac{1}{2}\left(s \cdot \cos\left(\pi + (2k-25-1)\frac{\pi}{6}\right),\right.\\ &\quad \left. s \cdot \sin\left(\pi + (2k-25-1)\frac{\pi}{6}\right)\right)\\ &= \mathbf{r}_{i,0} - \frac{1}{2}\left(s \cdot \cos\left(\frac{k-1}{3}\pi\right), s \cdot \sin\left(\frac{k-1}{3}\pi\right)\right)\end{aligned} \tag{29}$$

The full grid consists for each $i$ of all points $k=0$ to 18. The points numbering 13, 14, 15, 16, 17, 18 are unused in the sparse grid. The sparse grid collocation therefore consists of the points $i$, $k$ $(k=0$ to $12)$ which are the mid-point $k=0$ and points on the boundary. The sparse grid is redundant, as all points with $k>0$ belong to more than one hexagon. For example in Fig. 12, the point $9_{i',j'}$, is also point $1_{i'+1,j'+1}$ and $5_{i',j'+1}$. The point $10_{i',j'}$ belongs only to one other hexagon: $i', j'+1$. These redundant points of the collocation grid can be used to represent discontinuous functions.

The index is defined with respect to the orthogonal basis $\mathbf{i}_s$ and $\mathbf{j}_s$ shown in Fig. 12. This method of indexing is suitable for model areas being approximately square. A rhomboidal area can be achieved by changing the definition of the basis $\mathbf{i}_s$ and $\mathbf{j}_s$ shown in Fig. 12.

It was shown by Steppeler (1979) that the anisotropy of the square grid can lead to the deformation of structures, such as the destruction of the symmetry of a circular wave. It may be useful to show the connection to 1-D discretizations. In Fig. 13, two vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ are given and for each of these directions grid lines are shown in Fig. 13b, c which meet hexagonal grid points and thus determine a 1-D grid. Figure 13 shows the lines and the 1-D grids created by them. $\mathbf{n}_1$ and $\mathbf{n}_2$ form a 30° angle. The situation is similar as with square grids, where two 1-D grids are generated by lines forming a 45° angle. For square grids, the implied grids have grid lengths of $dx$ in $x$ direction and for the 45° line the grid length $dx' = \frac{dx}{\sqrt{2}}$. For the square the grids seen by a plane wave in the direction of either $x$ or the diagonal are all the same, as each line determines the same grid.

For 1-D grids in the direction of $\mathbf{n}_1$ generated by hexagonal grids, three types of irregular grid appear, shown in Fig. 13b. A plane wave in the directions $\mathbf{n}_1$ or $\mathbf{n}_2$ sees one of the two grids (Fig. 13d, e). They have the average resolution $dx'$ ($\frac{3}{8}s$ for Fig. 13d and $\frac{\sqrt{3}}{4}s$ for Fig. 13e). This is of higher

Author's personal copy

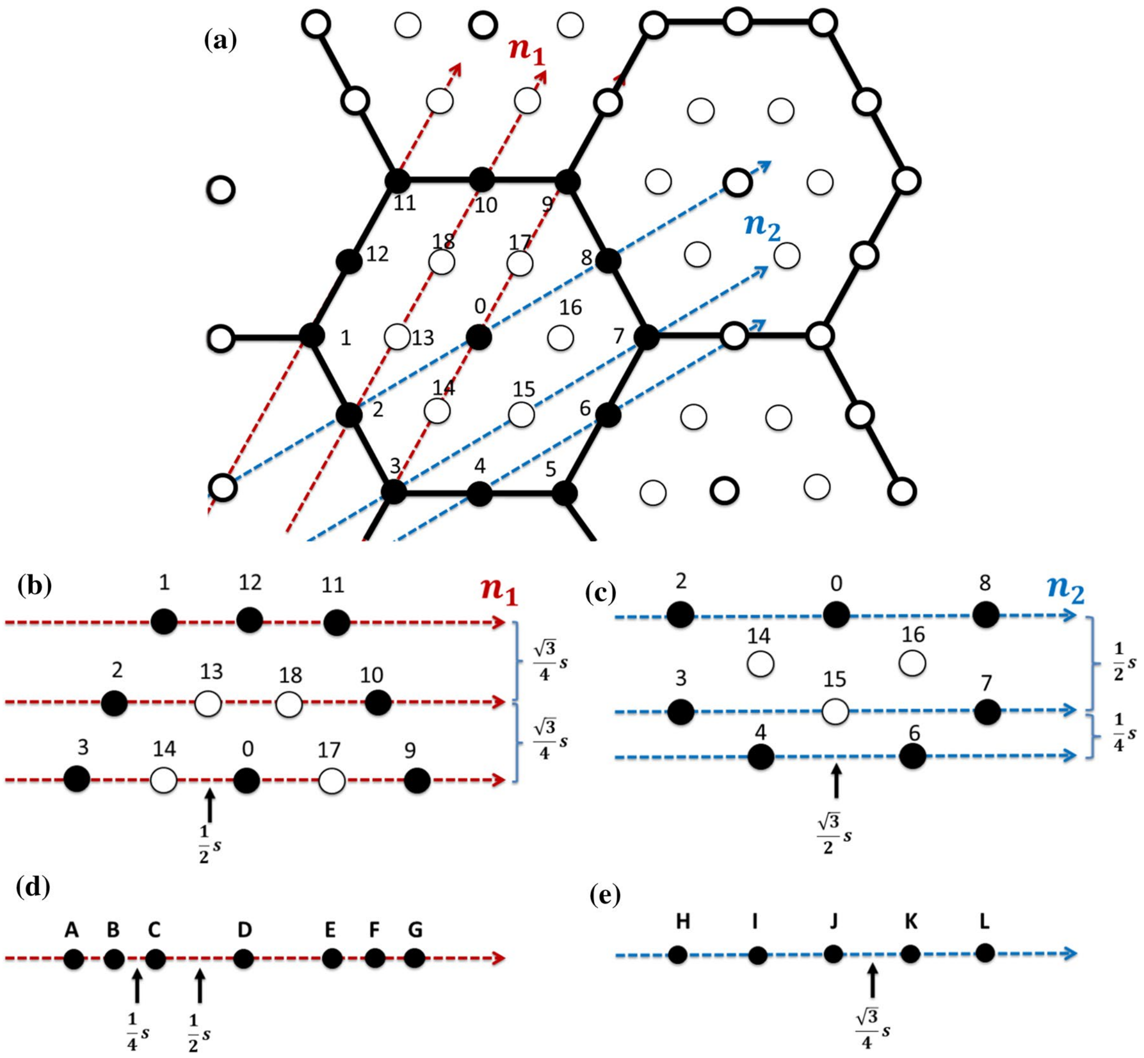

**Fig. 13** Two examples of lines resulting in 1-D grids for two directions. The combined grid which is seen by a plane wave in the directions of the arrows $\mathbf{n}_1 = (\cos 60°, \sin 60°) = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$ and $\mathbf{n}_2 = (\cos 30°, \sin 30°) = \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$ is shown in (**d**) and (**e**)

resolution than each of the original grids shown in Fig. 13b, c. It has also a higher resolution than the underlying regular square grid of resolution $\frac{1}{2}s$. For the gradients of the fluxes at corner points in $\mathbf{n}_1$ or $\mathbf{n}_2$ direction, the o2o3 scheme proposed in this paper uses one-sided derivatives and averages them over the three directions shown in Fig. 5. For a structure being smooth in the direction vertical to $\mathbf{n}_1$ or $\mathbf{n}_2$, the derivative at one point uses points forming a different grid for each of the lines indicated in Fig. 13. Thus, the resolution of a plane wave is that of the 1-D grids shown in Fig. 13d, e. The irregularity of the plane wave grids is of a small scale which means that while some points are near to each other, they have large differences in grid length. The pattern of small and larger grid lengths repeats itself. These grids have a higher resolution than the plane wave grids for the square.

It was shown by Steppeler and Klemp (2017) for the example of centered differences that such small-scale irregularity gives a quality of simulation corresponding to the average grid length. It can be seen that the average grid for plane waves is smaller than the grid used on the hexagonal sides. The values of the average grids given above should be compared to those corresponding to the square grid. For the square grid, we obtain $\frac{s}{2}$ and $\frac{s}{2\sqrt{2}}$ for the averaged resolutions in two directions. The relation of the two grids is 1.17 for the hexagon and 1.47 for the square. These numbers are a measure of the isotropy of the grid, which is higher for hexagons. The maximum plane wave grid length is also smaller for the hexagon (0.433 s) than for the square (0.5 s). The maximum of the two grids for different directions of plane waves is the resolution of the scheme. A small resolution in just one direction is not useful except for the case that the nature of the solution requires a deformed grid.

Author's personal copy

The indices of the inherent 1-D grids and their computation from the hexagonal indices can be seen from Fig. 13b, d or c, e. The hexagonal version of the grid for the method o3o3 (Steppeler et al. 2019) is obtained from the o2o3 hexagonal grid by having four instead three points on each edge and the center point no longer carries an amplitude.

## References

Ahlberg JH, Nilson EN, Walsh JL (1967) The theory of splines and their application. Academic, New York

Baumgardner JR, Frederickson PO (1985) Icosahedral discretization of the two-sphere. SIAM J Numer Anal 22:1107–1115

Cockburn B, Shu CW (2001) RungeKutta discontinuous Galerkin methods for convection-dominated problems. J Sci Comput 16(3):173–261

Cote J, Beland M, Staniforth A (1983) A spectral element shallow water model on spherical geodesic grids. Mon Wea Rev 111:1189–1207

Durran D (2010) Numerical methods of fluid dynamics: with applications to geophysics. Springer, New York

Giraldo FX (2001) A spectral element shallow water model on spherical geodesic grids. Int J Num Method Fluids 35:869–901

Kalnay E, Bayliss A, Storch J (1977) The 4th order GISS Model of the global atmosphere. Beitrage Phys Atmos 50:299–311

Li J, Li J, Zheng J, Zhu J, Fang F, Pain CC, Steppeler J, Navon IM, Xiao H (2018) Performance of adaptive unstructured mesh modelling in idealized advection cases over steep Terrains. Atmosphere 9:444

Marras S, Kelly JF, Moragues M, Müller A, Kopera MA, Vázquez M, Giraldo FX, Houzeaux G, Jorba O (2016) A review of element-based Galerkin methods for numerical weather prediction: finite elements, spectral elements, and discontinuous Galerkin. Arch Comput Method E 23(4):673–722

Navon IM, Alperson Z (1978) Application of fourth-order finite differences to a baroclinic model of the atmosphere. Arch Meterol Geophyd Biol Serie A 27:1–19

Peixoto PS, Barros SR (2013) Analysis of grid imprinting on geodesic spherical icosahedral grids. J Comput Phys 237:61–78

Rancic M, Purser RJ, Mesinger F (1996) A global shallow water model using an expanded spherical cube: gnomonic versus conformal coordinate. Q J R Meteorol Soc 122:959–982

Ringler TD, Heikes RH, Randall DA (2000) Modeling the atmospheric general circulation using a spherical geodesic grid: a new class of dynamical cores. Mon Wea Rev 128:2471–2485

Sadourny R (1972) Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. Mon Wea Rev 100:136–144

Satoh M, Masuno T, Tomita H, Miura H, Nasuno T, Iga S (2008) Nonhydrostatic icosahedaral atmosphericmodel (NICAM) for global cloud resolving simulations. J Comput Phys 227:3486–3514

Skamarock WC, Klemp JB (2008) A time-split nonhydrostatic atmospheric model for weather research and forecasting applications. J Comput Phys 227(7):3465–3485

Skamarock WC, Klemp JB, Duda MG, Fowler LD, Park SH (2012) A multiscale nonhydrostatic atmospheric model using centroidal voronoi tesselations and C-Grid staggering. Mon Wea Rev 140:3090–3105

Staniforth A, Thuburn J (2012) Horizontal grids for global weather and climate prediction models: a review. Q J R Meteorol Soc 138(662):1–26

Steppeler J (1976) The application of the second and third degree methods. J Comp Phys 22:295–318

Steppeler J (1979) Difference schemes with uniform second and third order accuracy and reduced smoothing. J Comput Phys 31:428–449

Steppeler J (1987) Galerkin and finite element methods in numerical weather prediction. Duemmler, Bonn

Steppeler J, Klemp JB (2017) Advection on cut-cell grids for an idealized mountain of constant slope. Mon Wea Rev 145:1765–1777

Steppeler J, Prohl P (1996) Application of finite volume methods to atmospheric models. Beitr Phys Atmos 69:297–306

Steppeler J, Ripodas P, Thomas S (2008) Third order finite difference schemes on isocahedral-type grids on the sphere. Mon Wea Rev 136:2683–2698

Steppeler J, Li J, Fang F, Zhu J, Ullrich PA (2019) o3o3: a variant of spectral elements with a regular collocation grid. Mon Wea Rev 147(6):2067–2082. https://doi.org/10.1175/MWR-D-18-0288.1

Taylor M, Tribbia J, Iskandarani M (1997) The spectral element method for the shallow water equations on the sphere. J Comput Phys 130:92–108

Tomita H, Tsugawa M, Satoh M, Goto K (2001) Shallow water model on a modified isosahedral geodesic grid by using spring dynamics. J Comp Phys 174:579–613

Weller H, Thuburn J, Cotter CJ (2012) Computational modes and grid imprinting on five quasi-uniform spherical C grids. Mon Wea Rev 140(8):2734–2755

Williamson DL (1968) Integrations of the barotropic vorticity equation on a spherical geodesic grid. Tellus 20:643–653

Williamson DL (2007) The evolution of dynamical cores for global atmospheric models. J Meteorol Soc Jpn Ser II 85:241–269

Zängl G, Reinert D, Rípodas P, Baldauf M (2015) The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: description of the non-hydrostatic dynamical core. Q J R Meteorol Soc 141(687):563–579

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.