Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 213 (2006) 358–390

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Numerical solution of the reaction–advection–diffusion equation on the sphere

Janusz A. Pudykiewicz *

Meteorological Service of Canada, Recherche en Prévison Numérique, 2121 Trans-Canada Highway, Dorval, Que., Canada H9P 1J3

Received 4 May 2005; received in revised form 10 August 2005; accepted 15 August 2005
Available online 5 October 2005

Abstract

A finite volume algorithm for the solution of the reaction–advection–diffusion equation on the sphere is derived and evaluated using analytical solutions. The proposed approach is based on the principle of semidiscretization. The convective and diffusive fluxes are approximated first, and then the resulting set of the ordinary differential equations (ODEs) is solved using the appropriate time stepping algorithm. In the first part of the paper, solutions to both the linear advection and the advection–diffusion problems for a single conservative scalar are discussed. The monotonicity of the scheme is achieved with the explicit adaptive dissipation. The development as well as the selected applications of the method are illustrated using a finite volume mesh constructed on the basis of geodesic icosahedral grid, which, in the past 40 years, has been frequently applied in different models of geophysical fluid dynamics. The performance of the solver is assessed using a suite of standard tests based on solid body rotation for different initial conditions. After analysis of the advection–diffusion problem, the extension of the method for the equations with reactive terms is presented. The performance of the solver is assessed by comparing the results to the analytical solution of the linearized reaction–diffusion system. In the final part of the paper, the application of the solver for studies of nonlinear reactions on the sphere is illustrated. The main intended application of the proposed method includes the simulation of transport of chemical constituents in the Earth's atmosphere as well as the forecasting of moisture and cloud water fields in numerical weather prediction and climate models.

© 2005 Elsevier Inc. All rights reserved.

Keywords: Advection on the sphere; Icosahedral grid; Atmospheric transport; Finite volume method; Reaction–diffusion waves

1. Introduction

An accurate numerical solution of the reaction–advection–diffusion equation is crucial for the simulation of a large class of physical systems described by interdependent scalar fields. Specific examples include such different problems as the transport of water vapour in the Earth's atmosphere [42] and the formation of complex spatial structures in systems of interacting chemical species [19]. In mathematical terms, the

* Tel.: +1 514 421 4744; fax: +1 514 421 2106.
E-mail address: Janusz.Pudykiewicz@ec.gc.ca.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.

doi:10.1016/j.jcp.2005.08.021

problem considered in all those different studies consists of solving of a set of partial differential equations of the form:

$$\frac{\partial \varphi^k}{\partial t} = -\nabla \mathbf{u}\varphi^k + \nabla \mathbf{K}\nabla\varphi^k + F_c^k(\varphi^1, \ldots, \varphi^{N_s}), \tag{1}$$

where φ^k is the kth scalar field; $\varphi^k : \mathscr{M}_s \to \mathbf{R}^1$, $\mathscr{M}_s \subset \mathbf{R}^3$ is the computational domain, $k = 1,\ldots,N_s$, N_s is number of scalar fields, $\mathbf{u}$ is the three-dimensional velocity field, ($\mathbf{u} : \mathscr{M}_s \to \mathbf{R}^3$), $\mathbf{r} \equiv (x,y,z)$, $\mathbf{K}$ is the diffusion tensor, ∇ is the nabla operator, and F_c^k are the functions. In general case, F_c^k can be written as $\alpha_{klm}\varphi^l\varphi^m + \beta_{kl}\varphi^l$, where α_{klm} and β_{kl} are the kinetic coefficients. We seek the solution of (1) with the mixed Dirichlet–von Neumann boundary conditions describing absorption and emission processes at the boundary of $\mathscr{M}_s$. The system of Eqs. (1) is inherently nonlinear since the interaction term F_c^k involves the product of concentrations. As opposed to hydrodynamic systems, the nonlinearity in the reaction–advection–diffusion equation is associated with the dissipative terms in the balance equation. Furthermore, since the F_c^k term is not explicitly dependent on $\mathbf{u}$, Eq. (1) can exhibit complex nonlinear behaviour even for spatially homogeneous systems.

The system of the reaction–advection–diffusion equations (1) admits a rich class of solutions ranging from a relatively simple linear advection [44] to nonlinear reaction–diffusion waves leading to the formation of complex, spatially nonhomogeneous patterns [18]. Consequently, there is a large body of literature describing specialized techniques for the numerical solution of (1). The available algorithms are based on different methodologies including finite differences, finite volumes, finite element and semi-Lagrangian techniques on various structured and unstructured meshes (see [21] for the review of general aspects).

The main objective of this paper is to explore numerical algorithms for the solution of the reaction–advection–diffusion equation (1) on the sphere; the main intended application being a simulation of the interacting scalar fields in the planetary atmospheres. Representative examples of possible problems in this area include transport of chemical constituents in the Earth's atmosphere [24] as well as forecasting of moisture and cloud water fields in numerical weather prediction and climate models [27,25].

In the past, such problems have often been solved using semi-Lagrangian techniques which offer certain advantages when applied to the linear advection on structured meshes in computational domains with simple geometries [33]. These advantages are, however, less evident when solving (1) in complex geometries with nonlinear reaction terms. In order to provide an alternative to the semi-Lagrangian techniques for solution of the reaction–advection–diffusion equation, we decided to develop a simple and accurate Eulerian algorithm build upon the principle of finite volumes. The approach adopted here exploits the concept of semidiscretization (see [13, p. 191]); we first approximate the convective and diffusive fluxes in (1) and then, we solve the resulting set of the ordinary differential equations (ODEs) using the appropriate time stepping algorithm. This methodology has been selected because of both its flexibility with respect to the mesh selection and its inherent ability to represent subgrid-scale processes and discontinuities in the solution. In the design of the method for the numerical solution of (1), we have acknowledged several major problems including the technique employed to discretize spatial terms, the choice of the time integration algorithm, and the methodology required to assure the monotonicity of the entire scheme.

The development and the selected applications of the method are illustrated using a finite volume mesh constructed on the basis of a geodesic icosahedral grid which has been introduced to geophysical fluid dynamics by Williamson [41] and Sadourny et al. [28]. In the past few years, various variants of spherical geodesic grids have been used in the numerical models of atmospheric flows [15,26,36].

The paper is organized as follows. Following the construction of the finite volumes on the sphere using the icosahedral grid, we discuss the solution of the linear advection and the advection–diffusion problems for a single conservative scalar. The monotonicity of the scheme is achieved with the explicit adaptive dissipation proposed by Shchepetkin and McWilliams [30]. The rationale for the selection of this method is motivated by the fact that classical split-directional algorithms such as van Leer algorithm [40] are not applicable for the icosahedral grid. It is worth mentioning, however, that there are several other viable alternatives to the strategy presented here. They include Smolarkiewicz MPDATA [31,34] and Osher's (W)ENO [14].

The performance of the advection–diffusion solver is assessed using the suite of standard tests based on solid body rotation [16] of cosine hill, cylinder, and multiscale signal as suggested by Smolarkiewicz and Grabowski [32]. After the analysis of the advection–diffusion problem, we present the solution of the transport equation with reactive terms. The performance of the solver in this case is assessed using the analytical solution of the reaction–diffusion system on the sphere introduced by Turing [38] in the context of the investigation of pattern formation in biological systems. The ideas presented by Turing are particularly useful for the investigation of reactive systems in a spherical geometry both in the general context and from the point of view of geophysical problems. In the final part of the paper, we illustrate the application of the solver for studies of nonlinear reaction–diffusion systems on the sphere.

2. Definition of the set of control volumes on the sphere

The set of finite volumes on the sphere is constructed in two stages. In the first stage, the geodesic mesh which covers the sphere in a quasi-uniform manner is defined. The mesh is generated starting from an icosahedron imbedded in the sphere. We begin by bisecting the edges of the icosahedron and projecting the bisection points on the surface of the sphere. The projected points are then connected to form a set of new triangles (see [6] for further details). The entire algorithm can be repeated an arbitrary number of times leading to the connected set of triangles approximating the spherical surface with desired accuracy (Fig. 1). The nodes of

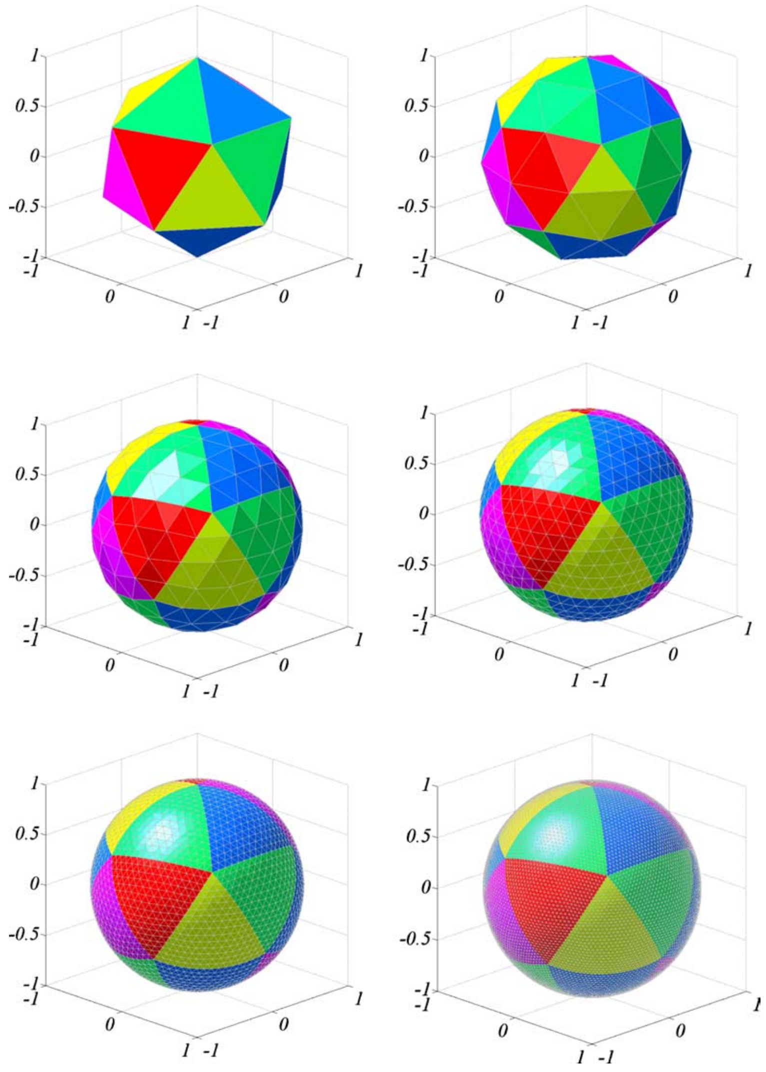

Fig. 1. Generation of the geodesic grid based on an icosahedron imbedded in the sphere.

these triangles define the connected planar graph $\mathcal{G}$ with N_{p} vertices, N_{e} edges, and N_{f} faces. The number of faces $N_{\mathrm{f}}(k)$ and the number of edges $N_{\mathrm{e}}(k)$ of $\mathcal{G}$ at each iteration k are given by the following relations for $k \geqslant 0$: $N_{\mathrm{f}}(k) = 2^{2k}N_{\mathrm{f}}(0)$, and $N_{\mathrm{e}}(k) = \frac{3}{2}N_{\mathrm{f}}(k)$, where $N_{\mathrm{f}}(0) \equiv 20$ is the number of faces in the icosahedron. The number of vertices of $\mathcal{G}$ can be derived from the Euler theorem for graphs [8]: $N_{\mathrm{p}}(k) = 2 + N_{\mathrm{e}}(k) - N_{\mathrm{f}}(k)$. After substitution of the relations for $N_{\mathrm{e}}(k)$ and $N_{\mathrm{f}}(k)$, we obtain $N_{\mathrm{p}}(k) = \frac{1}{2}N_{\mathrm{f}}(k) + 2$ (see Table 1 for a summary of the properties of geodesic grids).

In the second stage, the geodesic mesh, obtained after performing the desired number of iterations, is used to construct a set of finite volumes on the sphere. For each ith vertex of $\mathcal{G}$ we define a set of coincident edges $\mathcal{E}_i \equiv \{i, j(i)\}$. Each edge $(i,j(i))$ from this set is shared by two triangles (Fig. 2):

$$\mathcal{T}^1_{ij(i)} \equiv \triangle(i, j(i), x^1_{ij(i)}), \quad \mathcal{T}^2_{ij(i)} \equiv \triangle(i, j(i), x^2_{ij(i)}),$$

where $x^1_{ij(i)}$ and $x^2_{ij(i)}$ are the two exterior points defining triangles build on edge $(i,j(i))$.

We denote the mid-point of edge $(i,j(i))$ as $E_{ij(i)}$ and the mass centers of $\mathcal{T}^1_{ij(i)}$ and $\mathcal{T}^2_{ij(i)}$ as $M^1_{ij(i)}$ and $M^2_{ij(i)}$, respectively (see Fig. 2). The positions of these points can be defined explicitly as: $\mathbf{r}^E_{ij(i)} = (\mathbf{r}_i + \mathbf{r}_{j(i)})/2$, $\mathbf{r}^{M_1}_{ij(i)} = \frac{1}{3}(\mathbf{r}_i + \mathbf{r}_{j(i)} + \mathbf{r}^{x_1}_{ij(i)})$, and $\mathbf{r}^{M_2}_{ij(i)} = \frac{1}{3}(\mathbf{r}_i + \mathbf{r}_{j(i)} + \mathbf{r}^{x_2}_{ij(i)})$, where $\mathbf{r}_i$, $\mathbf{r}_{j(i)}$, $\mathbf{r}^{x_1}_{ij(i)}$, and $\mathbf{r}^{x_2}_{ij(i)}$ are positions of points i, $j(i)$, $x^1_{ij(i)}$, and $x^2_{ij(i)}$, respectively.

The control volume Ω_i associated with the ith node is then defined by a two step procedure. In the first step, we project the centers of the triangles and the mid-points of the edges on the surface of the sphere:

$$\mathbf{r}^{m_1}_{ij(i)} = \Pi\left(\mathbf{r}^{M_1}_{ij(i)}\right), \quad \mathbf{r}^{m_2}_{ij(i)} = \Pi\left(\mathbf{r}^{M_2}_{ij(i)}\right), \quad \mathbf{r}^{e}_{ij(i)} = \Pi\left(\mathbf{r}^{E}_{ij(i)}\right), \tag{2}$$

where Π is the projection operator defined as: $\Pi(\mathbf{r}) = a\mathbf{r}/|\mathbf{r}|$, and a is the radius of the sphere; we assume that $a = 1$ hereafter. In the second step, we define Ω_i as a polygon bound by a closed contour connecting all projected points

Table 1
Properties of geodesic grids used in the experiments

k	2^{2k}	Number of faces, $N_{\mathrm{f}}(k) = 2^{2k}N_{\mathrm{f}}(0)$	Number of edges, $N_{\mathrm{e}}(k) = \frac{3}{2}N_f(k)$	Number of vertices, $N_{\mathrm{p}}(k) = \frac{1}{2}N_f(k) + 2$
0	1	20	30	12
1	4	80	120	42
2	16	320	480	162
3	64	1280	1920	642
4	256	5120	7680	2562
5	1024	20,480	30,720	10,242
6	4096	81,920	122,880	40,962

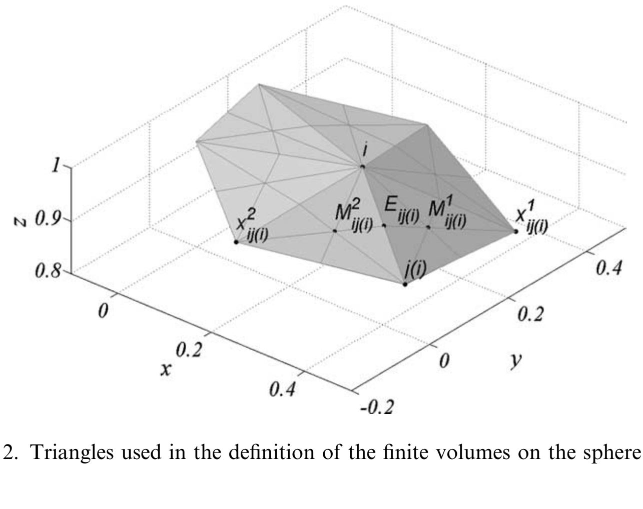

Fig. 2. Triangles used in the definition of the finite volumes on the sphere.

$$\partial\Omega_i \equiv \{\mathbf{r}^{m_1}_{ij(i)}, \mathbf{r}^{e}_{ij(i)}, \mathbf{r}^{m_2}_{ij(i)}\}|_{j(i)\in\{j(i)\}}, \tag{3}$$

where $\{j(i)\}$ is the set of incident nodes at node i (Fig. 3). An example of the set of finite volumes after two iterations is depicted in Fig. 4.

The most important characteristic of each node of $\mathcal{G}$ is the degree indicating the number of incident edges. The vertices of $\mathcal{G}$ corresponding to the nodes of the original icosahedron have degree five, whereas all new vertices obtained by the division of edges of triangles have degree six. The graph $\mathcal{G}$ is uniquely defined by specifying for each node number "i" a set of incident nodes $\{j(i)\}$ or, equivalently, by the adjacency matrix defined as follows [8]:

$$\mathscr{A}_{ij} = \begin{cases} 0 & \text{if points } i \text{ and } j \text{ are not connected,} \\ 1 & \text{if points } i \text{ and } j \text{ are connected.} \end{cases} \tag{4}$$

The structure of the adjacency matrix depends on the selected node ordering. In the case of the algorithm based on the bisection of triangles, new nodes obtained in each iteration are numbered consecutively following the old nodes. The sample adjacency matrix for this labelling of mesh points is depicted in Fig. 5 (right upper

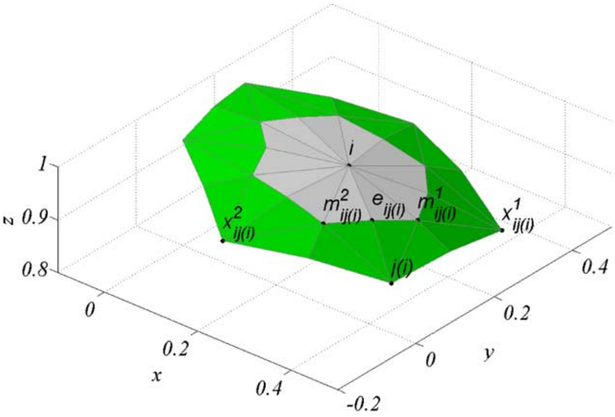

Fig. 3. The finite volume Ω_i associated with a single node i.

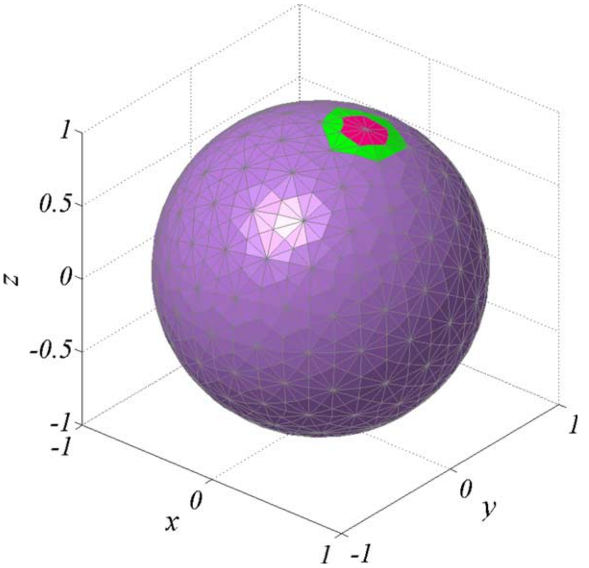

Fig. 4. Example of the finite volume mesh associated with the geodesic mesh after second iteration.

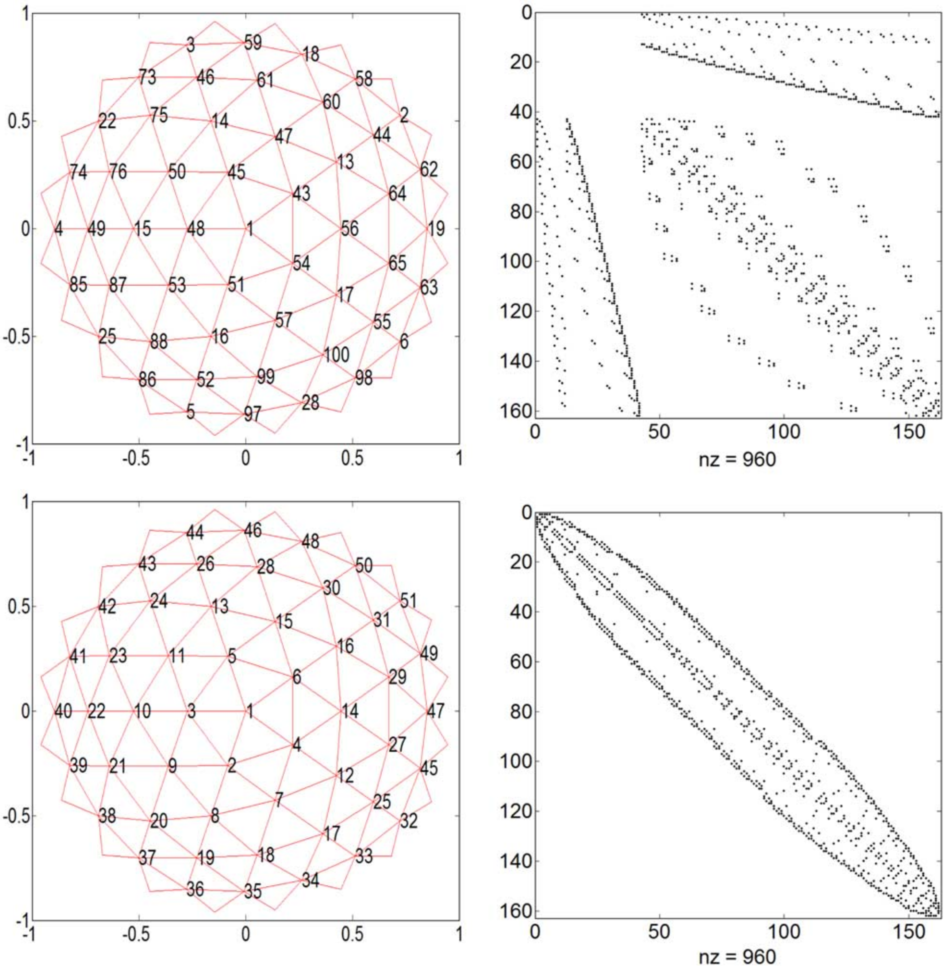

Fig. 5. The distribution of nodes of the geodesic mesh after the second iteration (left upper panel) and the corresponding connectivity matrix (right upper panel). The distribution of nodes after application of the Cuthill–McKee algorithm (left lower panel) and the corresponding connectivity matrix (right lower panel).

panel). The main conclusion from the analysis of this sparsity pattern is that the bandwidth of $\mathscr{A}$ is quite large. This is not desirable from the point of view of a construction of the compact discrete representation of the advection operator on $\mathscr{G}$. The bandwidth of $\mathscr{A}$ can be reduced upon the node reordering according to the

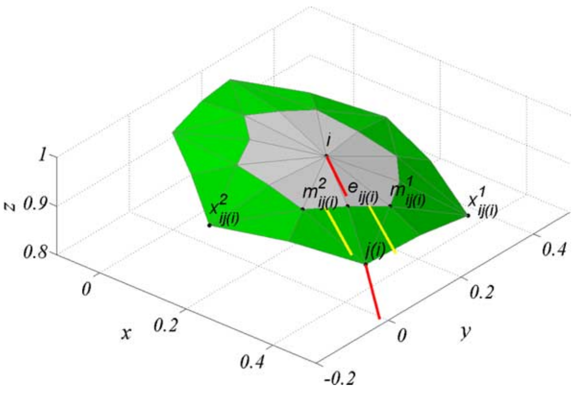

Fig. 6. Vectors normal to the boundary of Ω_i (yellow) and vectors tangential to the circle connecting points i and $j(i)$ (red).

Cuthill–McKee algorithm (see [39, p. 98]). The adjacency matrix for the new ordering of mesh points can be obtained from the transformation $\tilde{\mathscr{A}} = \mathbf{P}^{-1}\mathscr{A}\mathbf{P}$; where $\mathbf{P}$ is the permutation operator reordering the nodes following application of the Cuthill–McKee algorithm. The adjacency matrix, after reordering of nodes, is shown in Fig. 5 (right lower panel). The bandwidth of $\tilde{\mathscr{A}}$ is quite small; this property will lead to the definition of the differential operators with a compact stencil.

3. Discretization of the advection equation

We start our considerations from the advection equation for a single, conservative tracer written in a flux form:

$$\begin{aligned} &\frac{\partial\varphi}{\partial t} = -\nabla\mathbf{f}, \\ &\mathbf{f} = \mathbf{u}\varphi. \end{aligned} \tag{5}$$

After integration of (5) over Ω_i defined by (3) and applying the Stokes theorem (see [5, p. 64]) we obtain the set of ODE's governing the grid cell average values of φ:

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = -\frac{1}{S(\Omega_i)}\int_{\partial\Omega_i}\mathbf{f}\circ\mathbf{n}\,\mathrm{d}l, \tag{6}$$

where ϕ_i is the grid cell average of φ over the finite volume $\Omega_i(\phi_i = \langle\varphi\rangle_i = \int_{\Omega_i}\varphi(\mathbf{r},t)\,\mathrm{d}\mathbf{r}/S(\Omega_i))$, $i = 1,\ldots,N_\mathrm{p}$, $\circ$ denotes the scalar product, $S(\Omega_i)$ is the surface of Ω_i, and $\partial\Omega_i$ denotes the boundary of Ω_i.

After approximating of the integral term in (6), we obtain the following set of linear equations:

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = -\frac{1}{S(\Omega_i)}\sum_{j(i)}\sum_{k=1}^{2}\mathbf{f}\left(\varphi(\mathbf{r}^{(m_k)}_{ij(i)},t)\right)\circ\left(\delta l^{(k)}_{ij(i)}\mathbf{n}^{(k)}_{ij(i)}\right), \tag{7}$$

where $\partial\Omega_i$ is defined by (3), $\sum_{j(i)}\sum_{k=1}^{2}$ denotes the summation over elements constituting boundary of Ω_i, $\mathbf{f}(\varphi(\mathbf{r}^{(m_k)}_{ij(i)},t))$ is the flux through the segment of the boundary connecting points $m^1_{ij(i)}$, $e_{ij(i)}$ and $m^2_{ij(i)}$, $\delta l^{(k)}_{ij(i)}$ is the length of the segment of $\delta\Omega_i$ connecting points $e_{ij(i)}$ and $m^k_{ij(i)}(\delta l^{(k)}_{ij(i)} = |\mathbf{r}^e_{ij(i)} - \mathbf{r}^{m_k}_{ij(i)}|,\ k = 1,2)$, and $\mathbf{n}^{(k)}_{ij(i)}$ is the vector normal to the segment of $\delta\Omega_i$ connecting points $e_{ij(i)}$ and $m^k_{ij(i)}$, $(\mathbf{n}^{(1)}_{ij(i)} = \mathbf{r}^e_{ij(i)}\times\mathbf{r}^{m_1}_{ij(i)}, \mathbf{n}^{(2)}_{ij(i)} = \mathbf{r}^{m_2}_{ij(i)}\times\mathbf{r}^e_{ij(i)})$ (see Fig. 6).

In order to complete the discretization, we need to express flux $\mathbf{f}(\varphi(\mathbf{r}^{(m_k)}_{ij(i)},t))$ in terms of grid cell averages. One of the best known approximations is that based on the upwind scheme in which the flux through each segment $(\mathbf{r}^{(e)}_{ij(i)},\mathbf{r}^{(m_k)}_{ij(i)})$ is:

$$\mathbf{f}(\varphi(\mathbf{r}^{(m_k)}_{ij(i)},t))\circ\mathbf{n}^{(k)}_{ij(i)} = \begin{cases} \phi_i\mathbf{u}_{ij(i)}\circ\mathbf{n}^{(k)}_{ij(i)} & \text{for } (\mathbf{u}_{ij(i)}\circ\mathbf{n}^{(k)}_{ij(i)}) > 0 \text{ (outflow)}, \\ \phi_{j(i)}\mathbf{u}_{ij(i)}\circ\mathbf{n}^{(k)}_{ij(i)} & \text{for } (\mathbf{u}_{ij(i)}\circ\mathbf{n}^{(k)}_{ij(i)}) \leqslant 0 \text{ (inflow)}, \end{cases} \tag{8}$$

where $\mathbf{u}_{ij(i)} = (\mathbf{u}_i + \mathbf{u}_{j(i)})/2$.

After substitution of (8) to (7) and some algebra, we obtain the explicit form of the upwind approximation:

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = -\frac{1}{2}\sum_{j(i)}\sum_{k=1}^{2}\phi_i\left(\mathbf{u}_{ij(i)}\circ\mathbf{n}^{(k)}_{ij(i)} + |\mathbf{u}_{ij(i)}\circ\mathbf{n}^{(k)}_{ij(i)}|\right)\zeta^{(k)}_{ij(i)} + \phi_{j(i)}\left(\mathbf{u}_{ij(i)}\circ\mathbf{n}^{(k)}_{ij(i)} - |\mathbf{u}_{ij(i)}\circ\mathbf{n}^{(k)}_{ij(i)}|\right)\zeta^{(k)}_{ij(i)}, \tag{9}$$

where $\zeta^{(k)}_{ij(i)} = \delta l^{(k)}_{ij(i)}/S(\Omega_i)$.

The set of equations (9) defines a Cauchy problem for the vector of cell averages. In order to perform further the analysis, it is more convenient to rewrite (9) in a vector form:

$$\frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} = -\mathbf{A}_\mathrm{u}\boldsymbol{\phi}, \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \tag{10}$$

where $\boldsymbol{\phi}(t) = [\phi_1(t), \phi_2(t), \ldots, \phi_{N_p}(t)]^{\mathrm{T}}$ is the vector of grid cell average values of the scalar field, and $\mathbf{A}_u$ is the sparse matrix with diagonal elements of the form

$$A_{uii} = \sum_{j(i)} \sum_{k=1}^{2} \frac{1}{2} \left(\mathbf{u}_{ij(i)} \circ \mathbf{n}^{(k)}_{ij(i)} + |\mathbf{u}_{ij(i)} \circ \mathbf{n}^{(k)}_{ij(i)}| \right) \zeta^{(k)}_{ij(i)}, \tag{11}$$

and off-diagonal elements

$$A_{uij(i)} = \sum_{k=1}^{2} \frac{1}{2} \left(\mathbf{u}_{ij(i)} \circ \mathbf{n}^{(k)}_{ij(i)} - |\mathbf{u}_{ij(i)} \circ \mathbf{n}^{(k)}_{ij(i)}| \right) \zeta^{(k)}_{ij(i)}. \tag{12}$$

The set of equations (10) can be integrated using the Euler-forward scheme:

$$\boldsymbol{\phi}(t + \Delta t) = \boldsymbol{\phi}(t) - \Delta t \mathbf{A}_u \boldsymbol{\phi}(t), \tag{13}$$

where Δt is the time step.

The scheme expressed by (13) is stable provided that the norm of $\boldsymbol{\phi}(t + \Delta t)$ does not exceed the norm of $\boldsymbol{\phi}(t)$, i.e.,

$$\frac{|\boldsymbol{\phi}(t + \Delta t)|}{|\boldsymbol{\phi}(t)|} \leqslant 1. \tag{14}$$

After substitution of (13) to (14), we obtain the following criterion for the stability of (13):

$$|\mathbf{I} - \Delta t \mathbf{A}_u| \leqslant 1, \tag{15}$$

where $\mathbf{I}$ is a matrix representing unit operator.

It is well known that the upwind approximation of the advection equation produces nonoscillatory results [13]; unfortunately, the scheme has significant numerical diffusion which is manifested as a very strong damping of the amplitude of the advected field. In order to obtain more accurate results than those provided by the upwind scheme, we have to find a different formula for the approximation of the flux term in (7). In principle, there are two main approaches to the problem. The first one is based on the extension of the basic ideas of Godunov [7] which were introduced in the context of gas dynamics (upwinding with the slope limiting). The discussion of this approach in the context of unstructured meshes is presented by Hubbard [9].

The second approach to the problem is based on the use of central approximation of fluxes in conjunction with high order time integration in the spirit of Kreiss and Oliger [11]. This methodology requires an appropriate dissipation scheme to suppress the numerical noise. In the following section, we perform the explicit construction of the operators corresponding to the high order central scheme based on the second approach. The discussion of the dissipation mechanism is presented in Section 6.

4. Construction of advection operators based on central approximation of fluxes

Flux term in (7) can be approximated using the relation:

$$\mathbf{f}(\varphi(\mathbf{r}^{(m_k)}_{ij(i)}, t)) = \psi_{ij(i)} \mathbf{u}_{ij(i)}, \tag{16}$$

where $\psi_{ij(i)}$ is the interface value. In order to evaluate $\psi_{ij(i)}$, we assume that the variation of φ over the triangles $\triangle_1 = (\mathbf{r}_i, \mathbf{r}^{m_1}_{ij(i)}, \mathbf{r}^{e}_{ij(i)})$ and $\triangle_2 = (\mathbf{r}_i, \mathbf{r}^{m_2}_{ij(i)}, \mathbf{r}^{e}_{ij(i)})$ is described:

$$\varphi = \begin{cases} p(l) q_1(h_1) & \text{for } \triangle_1, \\ p(l) q_2(h_2) & \text{for } \triangle_2, \end{cases}, \tag{17}$$

where (l, h_1) and (l, h_2) are the local cartesian coordinate systems with the origin at $\mathbf{r}_i$, on the planes defined by $\triangle_1$ and $\triangle_2$, respectively (with l-direction determined by $(\mathbf{r}^{e}_{ij(i)} - \mathbf{r}_i)$), and p, q_1, q_2 are the polynomials satisfying the following relations:

$$\begin{aligned}
&\frac{1}{S(\triangle_1)}\int_{\triangle_1} pq_1\,\mathrm{d}l\,\mathrm{d}h_1 = \phi_i,\\
&\frac{1}{S(\triangle_2)}\int_{\triangle_2} pq_2\,\mathrm{d}l\,\mathrm{d}h_2 = \phi_i,\\
&q_1(0) = q_2(0) = 1,
\end{aligned}\tag{18}$$

where $S(\Delta_i)$ is the area of ith triangle. The idea of using polynomials for reconstruction of the field within the control volume was originally proposed by Colella and Woodward in the piecewise parabolic method (PPM) [4]. Here we assume that polynomial $p(l)$ is of the form

$$p(l) = w_3(l) = al^3 + bl^2 + cl + d. \tag{19}$$

We can evaluate the coefficients of (19) by fitting it to the variable values and first derivatives at two nodes $\mathbf{r}_i$ and $r_{j(i)}$. The coefficients can be thus obtained by solving the following set of linear equations:

$$\begin{aligned}
&w_3(0) = \phi_i,\\
&w_3'(0) = \phi_i',\\
&w_3(L_{ij(i)}) = \phi_{j(i)},\\
&w_3'(L_{ij(i)}) = \phi_{j(i)}',
\end{aligned}\tag{20}$$

where $\phi_i' = \mathbf{h}^a_{ij(i)}\nabla\phi|_i$, $\phi_{j(i)}' = \mathbf{h}^b_{ij(i)}\nabla\phi|_{j(i)}$, and $l \in [0, L_{ij(i)}]$, and $L_{ij(i)} = \arccos(\mathbf{r}_i \circ \mathbf{r}_{j(i)})$. The normalized vectors $\mathbf{h}^a_{ij(i)}$, $\mathbf{h}^b_{ij(i)}$ are tangential to the great circle $\mathscr{C}(L_{ij(i)})$ passing through the points $\mathbf{r}_i$ and $\mathbf{r}_{j(i)}$. These vectors can be expressed explicitly by the double vector products, $\mathbf{h}^a_{ij(i)} = (\mathbf{r}_i \times \mathbf{r}_{j(i)}) \times \mathbf{r}_i$ and $\mathbf{h}^b_{ij(i)} = (\mathbf{r}_i \times \mathbf{r}_{j(i)}) \times \mathbf{r}_{j(i)}$. Eq. (20) can be transformed, after substitution of the explicit form of $w_3(l)$, into

$$\begin{bmatrix} 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ L^3_{ij(i)} & L^2_{ij(i)} & L_{ij(i)} & 1\\ 3L^2_{ij(i)} & 2L_{ij(i)} & 1 & 0 \end{bmatrix}\begin{bmatrix} a\\ b\\ c\\ d\end{bmatrix} = \begin{bmatrix}\phi_i\\ \phi_i'\\ \phi_{j(i)}\\ \phi_{j(i)}'\end{bmatrix}. \tag{21}$$

Upon solving (21), we obtain the explicit expressions for coefficients in $w_3(l)$ and, consequently, the interface value $\psi_{ij(i)}$ can be calculated from the following relation:

$$\psi_{ij(i)} = w_3\left(\frac{L_{ij(i)}}{2}\right)q_1(0) = w_3\left(\frac{L_{ij(i)}}{2}\right)q_2(0) = \frac{1}{2}\left(\phi_i + \phi_{j(i)}\right) + \frac{1}{8}L_{ij(i)}\left(\phi_i' - \phi_{j(i)}'\right). \tag{22}$$

Alternatively, the interface value can be evaluated assuming that $p(l)$ is a a fifth order polynomial. The corresponding expression for the interface value in this case is:

$$\psi_{ij(i)} = \frac{1}{2}\left(\phi_i + \phi_{j(i)}\right) + \frac{5}{32}L_{ij(i)}\left(\phi_i' - \phi_{j(i)}'\right) + \frac{1}{64}L^2_{ij(i)}\left(\phi_i'' + \phi_{j(i)}''\right), \tag{23}$$

where $\phi_i'' = \mathbf{h}^a_{ij(i)} \circ \nabla(\mathbf{h}^a_{ij(i)}\nabla\phi|_i)$, and $\phi_{j(i)}'' = \mathbf{h}^b_{ij(i)} \circ \nabla(\mathbf{h}^b_{ij(i)}\nabla\phi|_{j(i)})$.

After calculation of flux (16) using the formula for the interface value obtained from cubic interpolation (22), we substitute the resulting expression in (7) in order to obtain the following set of equations:

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = -\sum_{j(i)}\left(\frac{1}{2}\left(\phi_i + \phi_{j(i)}\right) + \frac{1}{8}L_{ij(i)}\left(\mathbf{h}^a_{ij(i)}\nabla\phi|_i - \mathbf{h}^b_{ij(i)}\nabla\phi|_{j(i)}\right)\right)\gamma_{ij(i)}, \tag{24}$$

where

$$\gamma_{ij(i)} = \frac{\mathbf{u}_{ij(i)}\mathbf{n}^{(1)}_{ij(i)}\delta l^{(1)}_{ij(i)} + \mathbf{u}_{ij(i)}\mathbf{n}^{(2)}_{ij(i)}\delta l^{(2)}_{ij(i)}}{S(\Omega_i)}.$$

The analogical set of equations for the higher order interpolation scheme can be obtained after calculation of flux (16) using the interface value obtained from (23). After substitution of the resulting flux in (7) we obtain:

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = -\sum_{j(i)}\left(\frac{1}{2}\left(\phi_i + \phi_{j(i)}\right) + \frac{5}{32}L_{ij(i)}\left(\mathbf{h}^a_{ij(i)}\nabla\phi|_i - \mathbf{h}^b_{ij(i)}\nabla\phi|_{j(i)}\right) + \frac{1}{64}L^2_{ij(i)}\left(\mathbf{h}^a_{ij(i)}\circ\nabla\left(\mathbf{h}^a_{ij(i)}\nabla\phi|_i\right)\right.\right.$$
$$\left.\left. + \mathbf{h}^b_{ij(i)}\circ\nabla\left(\mathbf{h}^b_{ij(i)}\nabla\phi|_{j(i)}\right)\right)\right)\gamma_{ij(i)}. \tag{25}$$

In order to approximate the gradient operator in (24) and (25) we apply the Stokes theorem as follows:

$$\int_{\Omega_i}\nabla\varphi\,\mathrm{d}\mathbf{r} = \int_{\partial\Omega_i}\varphi\mathbf{n}\,\mathrm{d}l. \tag{26}$$

After division of both sides of (26) by $S(\Omega_i)$, we obtain

$$\langle\nabla\varphi\rangle_i = \nabla\langle\varphi\rangle_i = \nabla\phi|_i = \frac{1}{S(\Omega_i)}\int_{\partial\Omega_i}\varphi\mathbf{n}\,\mathrm{d}l. \tag{27}$$

The integral on the right-hand side of (27) is discretized as follows:

$$\int_{\partial\Omega_i}\varphi\mathbf{n}dl = \sum_{j(i)}\sum_{k=1}^{2}w(\varphi_i,\varphi_{j(i)})\delta l^{(k)}_{ij(i)}\mathbf{n}^{(k)}_{ij(i)}, \tag{28}$$

where $w(\varphi_i, \varphi_{j(i)})$ is the interface value that can be approximated by either linear or quadratic interpolation. For both cases of interpolation, the explicit expression for the interface value is $w(\varphi_i, \varphi_{j(i)}) = \alpha\phi_i + \beta\phi_{j(i)}$, with $\alpha = \beta = 0.5$ for the linear interpolation and $\alpha = 0.25$, $\beta = 0.75$ for the quadratic one. In the calculations involving (24) and (25) we used the latter set of values whereas the former set was used for the evaluation of the ∇^2 operator in the next section. The explicit expression for the gradient operator can be obtained by substituting (28) in (27)

$$\nabla\phi|_i = \sum_{j(i)}\sum_{k=1}^{2}(\alpha\phi_i + \beta\phi_{j(i)})\mathbf{n}^{(k)}_{ij(i)}\zeta^{(k)}_{ij(i)}. \tag{29}$$

To assure that the gradient of the constant field be exactly zero, the expression in (29) can be modified by addition of the correction term suggested by Tomita et al. [37].

$$\nabla\phi|_i = \sum_{j(i)}\sum_{k=1}^{2}(\alpha\phi_i + \beta\phi_{j(i)})\mathbf{n}^{(k)}_{ij(i)}\zeta^{(k)}_{ij(i)} - \phi_i\sum_{j(i)}\sum_{k=1}^{2}\mathbf{n}^{(k)}_{ij(i)}\zeta^{(k)}_{ij(i)}. \tag{30}$$

In the calculations reported here the corrective term is not included since it has no observable impact on the results when using (30) with (24) or (25), particularly on grids obtained after six or more iterations.

The expression for gradient (29) can be compactly written in a vector form as

$$\nabla\boldsymbol{\phi} = [\mathbf{G}_x\boldsymbol{\phi}, \mathbf{G}_y\boldsymbol{\phi}, \mathbf{G}_z\boldsymbol{\phi}], \tag{31}$$

where $\boldsymbol{\phi}(t) = [\phi_1(t), \phi_2(t), \ldots, \phi_{N_p}(t)]^{\mathrm{T}}$, and $\mathbf{G}_\delta$ $(\delta = x, y, z)$ are the gradient operators represented by sparse matrices with nonzero elements described by the following relations:

$$\begin{cases} G_{\delta ii} = \sum_{j(i)}\sum_{k=1}^{2}\alpha n^{(k)}_{\delta ij(i)}\zeta^{(k)}_{ij(i)} & \text{diagonal elements,} \\ G_{\delta ij(i)} = \sum_{k=1}^{2}\beta n^{(k)}_{\delta ij(i)}\zeta^{(k)}_{ij(i)} & \text{off-diagonal elements.} \end{cases} \tag{32}$$

Further analysis of the problem will be significantly simplified if Eqs. (24) and (25) are rewritten in the operator form. In particular, (24) can be cast in the vector form as follows:

$$\frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} = -\left[\underbrace{\mathbf{A}_0 + \mathbf{H}_x\mathbf{G}_x + \mathbf{H}_y\mathbf{G}_y + \mathbf{H}_z\mathbf{G}_z}_{\mathbf{A}_3}\right]\boldsymbol{\phi}. \tag{33}$$

The operators $\mathbf{A}_0$ and $\mathbf{H}_\delta$ ($\delta = x, y, z$) are ($N_p \times N_\mathrm{p}$) sparse matrices with nonzero elements defined by the following relations:

$$\begin{cases} A_{0ii} = \frac{1}{2}\sum\limits_{j(i)} \gamma_{ij(i)}, \quad H_{\delta ii} = \frac{1}{8}\sum\limits_{j(i)} L_{ij(i)} h^a_{\delta ij(i)} \gamma_{ij(i)} & \text{diagonal elements,} \\ A_{0ij(i)} = \frac{1}{2}\gamma_{ij(i)}, \quad H_{\delta ij(i)} = -\frac{1}{8} L_{ij(i)} h^b_{\delta ij(i)} \gamma_{ij(i)} & \text{off-diagonal elements.} \end{cases} \tag{34}$$

A sample structure of the operator $\mathbf{A}_3$ appearing in Eq. (33) is depicted in Fig. 7 (the effect of reordering of nodes is also depicted in this figure). Further, the corresponding vector version of scheme (25) is analogous to (33) with operator $\mathbf{A}_3$ replaced by operator A_5 of the following form:

$$\begin{aligned} \mathbf{A}_5 = \mathbf{A}_0 + \frac{5}{4}(\mathbf{H}_x\mathbf{G}_x + \mathbf{H}_y\mathbf{G}_y + \mathbf{H}_z\mathbf{G}_z) + \mathbf{H}_{xx}\mathbf{G}_x\mathbf{G}_x + \mathbf{H}_{yy}\mathbf{G}_y\mathbf{G}_y + \mathbf{H}_{zz}\mathbf{G}_z\mathbf{G}_z + \mathbf{H}_{xy}(\mathbf{G}_x\mathbf{G}_y + \mathbf{G}_y\mathbf{G}_x) \\ + \mathbf{H}_{xz}(\mathbf{G}_x\mathbf{G}_z + \mathbf{G}_z\mathbf{G}_x) + \mathbf{H}_{yz}(\mathbf{G}_y\mathbf{G}_z + \mathbf{G}_z\mathbf{G}_y), \end{aligned} \tag{35}$$

where elements of $\mathbf{H}_{v\mu}$ ($\mu = x, y, z, v = x, y, z$) are defined as follows:

$$\begin{cases} H_{v\mu ii} = \frac{1}{64}\sum\limits_{j(i)} L^2_{ij(i)} h^a_{vij(i)} h^a_{\mu ij(i)} \gamma_{ij(i)} & \text{diagonal elements,} \\ H_{v\mu ij(i)} = \frac{1}{64} L^2_{ij(i)} h^b_{vij(i)} h^b_{\mu ij(i)} \gamma_{ij(i)} & \text{off-diagonal elements.} \end{cases} . \tag{36}$$

A sample structure of operator $\mathbf{A}_5$ is depicted in Fig. 8. Operator $\mathbf{A}_5$, after node reordering, has a very regular structure, similar to that of operator $\mathbf{A}_3$ but with a larger number of diagonals.

The numerical solution of (33) can be accomplished with different ODE solvers. Because of its simplicity and effectiveness for advection problem the fourth-order Runge–Kutta scheme has been selected here [44]. The solution of the advection equation can be expressed as

$$\begin{aligned} &\boldsymbol{\phi}(t + \Delta t) = \boldsymbol{\phi}(t) + \frac{1}{6}(\mathbf{f}_1 + 2\mathbf{f}_2 + 2\mathbf{f}_3 + \mathbf{f}_4), \\ &\mathbf{f}_1 = \Delta t \mathscr{L}(\boldsymbol{\phi}), \quad \mathbf{f}_2 = \frac{\Delta t}{2}\mathscr{L}(\boldsymbol{\phi} + \mathbf{f}_1), \quad \mathbf{f}_3 = \frac{\Delta t}{2}\mathscr{L}(\boldsymbol{\phi} + \mathbf{f}_2), \quad \mathbf{f}_4 = \Delta t \mathscr{L}(\boldsymbol{\phi} + \mathbf{f}_3), \end{aligned} \tag{37}$$

where $\mathscr{L}(\boldsymbol{\phi}) = \mathbf{A}\boldsymbol{\phi}$, with $\mathbf{A} = -\mathbf{A}_3$ or $\mathbf{A} = -\mathbf{A}_5$. (The selection of the ODE solvers can be expanded by the addition of semi-implicit and implicit methods.)

Further, in order to obtain an explicit formula for the advancement of the solution in time, we substitute $\mathscr{L}(\boldsymbol{\phi}) = \mathbf{A}\boldsymbol{\phi}$ in (37) and after some algebra

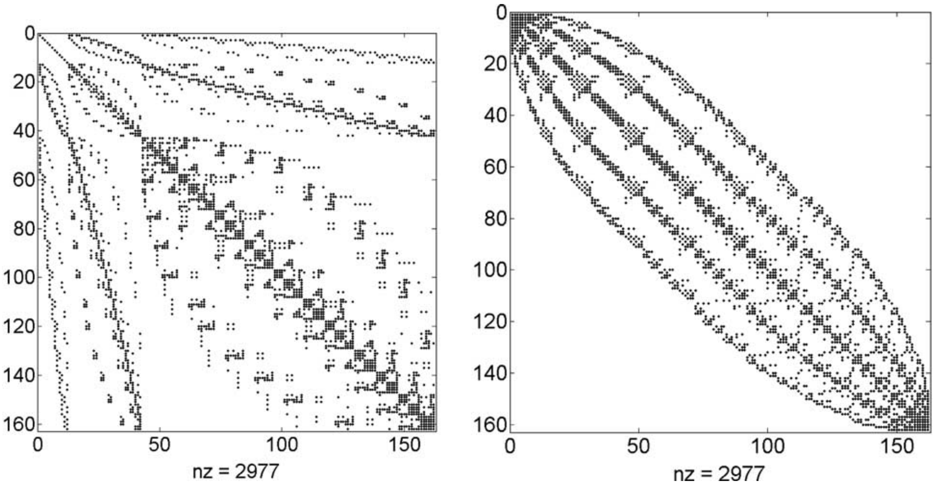

Fig. 7. The structure of $\mathbf{A}_3$ operator for the geodesic mesh obtained after two iterations, left panel without node reordering, right panel after node reordering.

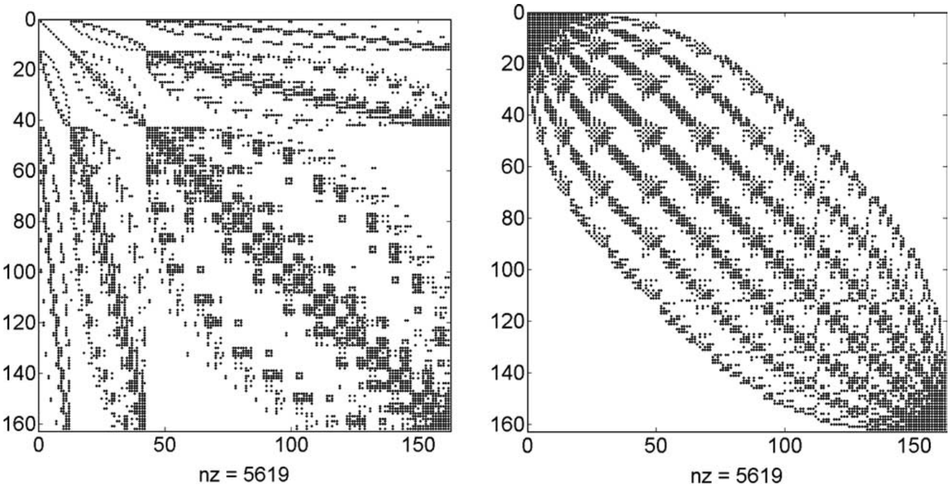

Fig. 8. Same as in Fig. 7 but for $\mathbf{A}_5$ operator.

$$\boldsymbol{\phi}(t+\Delta t) = \left[\mathbf{I} + \Delta t\mathbf{A} + \frac{\Delta t^2}{2}\mathbf{A}^2 + \frac{\Delta t^3}{6}\mathbf{A}^3 + \frac{\Delta t^4}{24}\mathbf{A}^4\right]\boldsymbol{\phi} = \boldsymbol{\Lambda}_\tau\boldsymbol{\phi}(t). \tag{38}$$

The time advancement of the solution can be thus accomplished with one multiplication of vector $\boldsymbol{\phi}(t)$ by a sparse matrix $\boldsymbol{\Lambda}_\tau$. The stability of this scheme is assured for $|\boldsymbol{\Lambda}_\tau| \leqslant 1$.

Eq. (38) can be alternatively derived from the Taylor expansion of $\boldsymbol{\phi}$ as

$$\boldsymbol{\phi}(t+\Delta t) \approx \boldsymbol{\phi}(t) + \Delta t\frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} + \frac{\Delta t^2}{2}\frac{\mathrm{d}^2\boldsymbol{\phi}}{\mathrm{d}t^2} + \cdots + \frac{\Delta t^n}{n!}\frac{\mathrm{d}^n\boldsymbol{\phi}}{\mathrm{d}t^n}\cdots \tag{39}$$

After substitution of $\mathrm{d}^n\boldsymbol{\phi}/\mathrm{d}t^n = \mathbf{A}^n\boldsymbol{\phi}$ in (39) we obtain:

$$\boldsymbol{\phi}(t+\Delta t) \approx \boldsymbol{\phi}(t) + \Delta t\mathbf{A}\boldsymbol{\phi} + \frac{\Delta t^2}{2}\mathbf{A}^2\boldsymbol{\phi} + \cdots + \frac{\Delta t^n}{n!}\mathbf{A}^n\boldsymbol{\phi} + \cdots \tag{40}$$

Expression (40) for $n = 4$ is formally equivalent to (38) derived from the fourth order Runge–Kutta scheme, (37).

5. Incorporation of the diffusion term

A semidiscrete approximation of the advection–diffusion equation is obtained after substituting $\mathbf{f} = \mathbf{u}\varphi - \mathbf{K}\nabla\varphi$ in (6):

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}t} = -\frac{1}{S(\Omega_i)}\int_{\partial\Omega_i}\mathbf{u}\varphi\mathbf{n}\,\mathrm{d}l + \frac{1}{S(\Omega_i)}\int_{\partial\Omega_i}\mathbf{K}\nabla\varphi\mathbf{n}\,\mathrm{d}l. \tag{41}$$

The discretization of the first integral on the right-hand side of (41) is given explicitly either by (33) or by (35). The discretization of the second integral is derived from the Stokes theorem as

$$\frac{1}{S(\Omega_i)}\int_{\partial\Omega_i}\mathbf{K}\nabla\varphi\mathbf{n}\,\mathrm{d}l = \nabla\langle\mathbf{K}\varphi\rangle_i. \tag{42}$$

Assuming that $\nabla\langle\mathbf{K}\varphi\rangle_i = \nabla\langle K\rangle_i\langle\nabla\rangle_i$ we obtain

$$\frac{1}{S(\Omega_i)}\int_{\partial\Omega_i}\mathbf{K}\nabla\varphi\mathbf{n}\,\mathrm{d}l = \nabla\langle\mathbf{K}\rangle_i\nabla\phi_i. \tag{43}$$

After substituting (43) in (41) and using the general expression for gradient given by (31) with $\alpha = \beta = 0.5$, we obtain

$$\frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} = -\mathbf{A}_3\boldsymbol{\phi} + \sum_{\mu\nu}\mathbf{G}_\mu\left[\boldsymbol{\kappa}_{\mu\nu}\circ(\mathbf{G}_\nu\boldsymbol{\phi})\right], \tag{44}$$

where $\boldsymbol{\kappa}_{\mu\nu} = [\langle K_{\mu\nu}\rangle_1, \ldots, \langle K_{\mu\nu}\rangle_{N_p}]$ are the vectors of grid cell average values of the components of the diffusion tensor, $\mu = x, y, z$, and $\nu = x, y, z$.

The selection of the solver for the time integration of (44) is flexible and can include both semi-implicit and implicit algorithms. The discussion of the possible strategies of integration of (44) is presented in numerous texts on the ODEs (see for example [12] for the discussion of general aspects). For the large scale geophysical problems with relatively weak horizontal mixing, the time integration of (44) can be accomplished with the help of the technique discussed in the context of the advection equation. In particular, one can use the scheme described by (38) with operator $\mathbf{A}$ replaced by operator $\mathbf{A}_{\mathscr{D}}$ defined as

$$\mathbf{A}_{\mathscr{D}} = -\mathbf{A}_3 + \sum_{\mu\nu}\mathbf{G}_\mu\left[\boldsymbol{\kappa}_{\mu\nu}\circ\mathbf{G}_\nu\right]. \tag{45}$$

For a more general case with stronger diffusion, the time integration scheme can be changed to the implicit or semi-implicit in order to assure the solver stability.

6. Monotonic version of the advection scheme

The solution of the advection equation obtained using scheme (38) is not monotone in accordance with the Godunov theorem [7]. The most common problem in the approximate solution of (1) reported in the literature is preserving the amplitude of the advected scalar field while avoiding numerical oscillations which are detrimental to the nonlinear term F_c^{k} in (1). A complete elimination of these oscillations is particularly difficult to achieve for scalar fields exhibiting large gradients or discontinuities. In order to correctly represent the approximate solution of (1) under these conditions, most of the known general algorithms for the advection equation rely either on some kind of upwinding with slope limiting [13] or on flux corrected transport (FCT) techniques [2,46]. In the geophysical context, the class of techniques designed to suppress numerical oscillations in the advected scalar fields is extended by the method known as the explicit local adaptive dissipation (ELAD) [30]. For the advection equation, the ELAD methodology is particularly attractive because it assures quasi-monotonicity in the high order central schemes for the advection equation in a natural manner without excessive dissipation. There are also several other approaches that complement the upwinding, FCT and in particular, ELAD techniques. One among the most notable ones is the MPDATA algorithm [31,34] which relies on the upwind technique with an iterative antidiffusion procedure.

In order to eliminate spurious oscillations which can be generated by a numerical scheme, we decided to employ ELAD [30]. The generalization of the method for the grid system discussed here follows.

In the first step, we compute the solution for advection using the algorithm described by (38):

$$\boldsymbol{\phi}^{(0)} = \boldsymbol{\Lambda}_\tau\boldsymbol{\phi}(t). \tag{46}$$

Contrary to the analytical solution, the approximate solution given by (46) is not constrained by the bounds defined as

$$\forall_{i\in[1,N_\mathrm{p}]}\begin{cases}\phi_i^{(\max)} = \max\left[\phi_i(t), \phi_i^{(\max u)}\right],\\ \phi_i^{(\min)} = \min\left[\phi_i(t), \phi_i^{(\min u)}\right],\end{cases} \tag{47}$$

where

$$\forall_{i\in[1,N_\mathrm{p}]}\begin{cases}\phi_i^{(\max u)} = \max\left[\phi_j(t) : j\in\{Ju(i)\}\right],\\ \phi_i^{(\min u)} = \min\left[\phi_j(t) : j\in\{Ju(i)\}\right]\end{cases}$$

and $\{Ju(i)\}$ is the set of nodes located upstream to the ith node.

The measure of the deviation of the numerical solution from the correct, bounded solution can be provided by the excess field defined as

$$\epsilon_i^{(0)} = \max(0, \phi_i^{(0)} - \phi_i^{(\max)}) + \min(0, \phi_i^{(0)} - \phi_i^{(\min)}). \tag{48}$$

The excess field ϵ is zero for the solution of the equation within the bounds ($\phi_i^{\min} \leqslant \phi_i^{(0)} \leqslant \phi_i^{\max}$).

In order to obtain the corrected value of the solution, the excess field is diffused:

$$\boldsymbol{\phi}^{(1)} = \boldsymbol{\phi}^{(0)} + \mathbf{D}_\nu \boldsymbol{\epsilon}^{(0)}, \tag{49}$$

where $\boldsymbol{\epsilon}^{(0)} = [\epsilon_1^{(0)}, \ldots, \epsilon_{N_\mathrm{p}}^{(0)}]^\mathrm{T}$, $\mathbf{D}_\nu$ is a diffusion operator defined as $\mathbf{D}_\nu = \nu\,(\mathbf{G}_x\mathbf{G}_x + \mathbf{G}_y\mathbf{G}_y + \mathbf{G}_z\mathbf{G}_z)$, and operators $\mathbf{G}_x, \mathbf{G}_y, \mathbf{G}_z$ are defined by (32). Coefficient ν is selected in such a manner that $\nu \leqslant \nu_{\max}$; $\nu_{\max}$ is the value for which $|\mathbf{I} + \mathbf{D}_\nu| < 1$.

The procedure described above leads to a significant reduction of spurious oscillations. In general, a single application of (49) is not sufficient to eliminate all noise and, consequently, all steps described above are repeated in an iterative manner. Formally, the iteration procedure can be written as

$$\begin{cases} \epsilon_i^{(k)} = \max(0, \phi_i^{(k)} - \phi_i^{(\max)}) + \min(0, \phi_i^{(k)} - \phi_i^{(\min)}) & \text{evaluate the excess field,} \\ \boldsymbol{\phi}^{(k+1)} = \boldsymbol{\phi}^{(k)} + \mathbf{D}_\nu \boldsymbol{\epsilon}^{(k)} & \text{diffuse the excess field.} \end{cases} \tag{50}$$

The solution for the next time step is then given as:

$$\boldsymbol{\phi}(t + \Delta t) = \boldsymbol{\phi}^{(n_{\max})},$$

where $n_{\max}$ is the maximum number of iterations. The selection of the number of iterations and the value of the coefficient ν is discussed in detail by Shchepetkin and McWilliams [30].

It is suggested that the the value of ν should be close to that permitted by the stability considerations ($\nu_{\max}$). However, in most practical cases, it is sufficient to use a smaller value of ν. The large values of ν, close to $\nu_{\max}$, should be reserved for regions with very sharp gradients. This procedure has significant impact on the further reduction of the dissipation of the method as well as on the preservation of the maximum values of the advected tracer fields.

The other possible improvement of the method consists of the modification of the procedure used to evaluate the permissible bounds (47). In the current formulation, the bounds are determined as minima and maxima among previous-time-step values at the point itself and the nearest values in every upstream direction regardless of the magnitude of the velocity in that direction. This definition can contribute to inaccurate results in situations when the flow is perpendicular to the gradient of advected field. In such a case, the bounds are set predominantly by the points located on the line perpendicular to the flow, which actually do not affect much the change of the advected quantity. Therefore, it is advisable to define bounds (47) in such a manner that they depend on differences in concentrations multiplied by the corresponding edge Courant numbers, so that points which only weakly influence the net change due to advection will not affect permissible bounds.

7. Evaluation of the algorithm for advection

The performance of numerical schemes for the advection equation is evaluated by performing numerical tests with a prescribed velocity field as in [43]:

$$\begin{aligned} u_x &= -\widetilde{u}\sin(\lambda) - \widetilde{v}\sin(\theta)\cos(\lambda), \\ u_y &= \widetilde{u}\cos(\lambda) - \widetilde{v}\sin(\theta)\sin(\lambda), \\ u_z &= \widetilde{v}\cos(\theta), \end{aligned} \tag{51}$$

where $\widetilde{u} = \Omega(\cos(\theta)\cos(\alpha) + \sin(\theta)\cos(\lambda)\sin(\alpha))$, $\widetilde{v} = -\Omega\sin(\lambda)\sin(\alpha)$, θ is the latitude, λ is the longitude, α is the angle between the sphere axis and the axis of rotation, and Ω is the angular frequency of rotation. We consider first the initial condition described by a cosine hill:

$$\phi(\mathbf{r}_i) = \begin{cases} g(\mathbf{r}_i) + c_\phi & \text{for } d(\mathbf{r}_i, \mathbf{r}_o) \leqslant r_a, \\ c_\phi & \text{for } d(\mathbf{r}_i, \mathbf{r}_o) > r_a, \end{cases} \tag{52}$$

where $g(\mathbf{r}_i) = (1 + \cos(\pi d(\mathbf{r}_i,\mathbf{r}_o)/r_a))/2$, $\mathbf{r}_i$ is the position of the center of the control volume, $\mathbf{r}_o$ is the position of the center of the mass distribution, r_a and c_ϕ are parameters, and $d(\mathbf{r}_i,\mathbf{r}_o)$ is the distance between $\mathbf{r}_o$ and $\mathbf{r}_i$. All advection tests are performed on a geodesic grid with 40,962 nodes (see Table 1 for additional parameters of this grid). In the experiment, we assumed $\Omega = 2\pi/12$, $\Delta t = 0.025$, $r_a = 7\ \pi/64$, and $c_\phi = 2$. For this set of parameters, the maximum value of the edge Courant number, $\mathscr{C} = \mathbf{u}_{j(i)}\Delta t/(|\mathbf{r}_i - \mathbf{r}_{j(i)}|)$ is 0.75.

The first test consisted of advecting a cosine hill in the velocity field (51) through the North and South Poles ($\alpha = \pi/2$ in (51)). The numerical solution obtained with the scheme defined by Eq. (33) (labeled $\mathscr{S}_1$) is depicted in Fig. 9. It is evident that the method maintains the maximum value of the distribution and is almost free of any computational noise. In order to evaluate the numerical solution quantitatively, we introduce l_1, l_2 and l_∞ norms defined as:

$$l_1 = \frac{\mathscr{I}(|\phi - \phi_a|)}{\mathscr{I}(|\phi_a|)}, \tag{53}$$

$$l_2 = \sqrt{\frac{\mathscr{I}((\phi - \phi_a)^2)}{\mathscr{I}((\phi_a)^2)}}, \tag{54}$$

$$l_\infty = \frac{\max(|\phi - \phi_a|)}{\max(|\phi_a|)}, \tag{55}$$

where $\mathscr{I}(f) = \int_{\mathscr{S}} f\, \mathrm{d}s$ is the integral over $\mathscr{S}$, ϕ_0 is the initial condition, ϕ_a is the analytical solution, and ϕ is the numerical solution. The mass conservation is measured by either a loss or a gain of mass normalized by the initial mass:

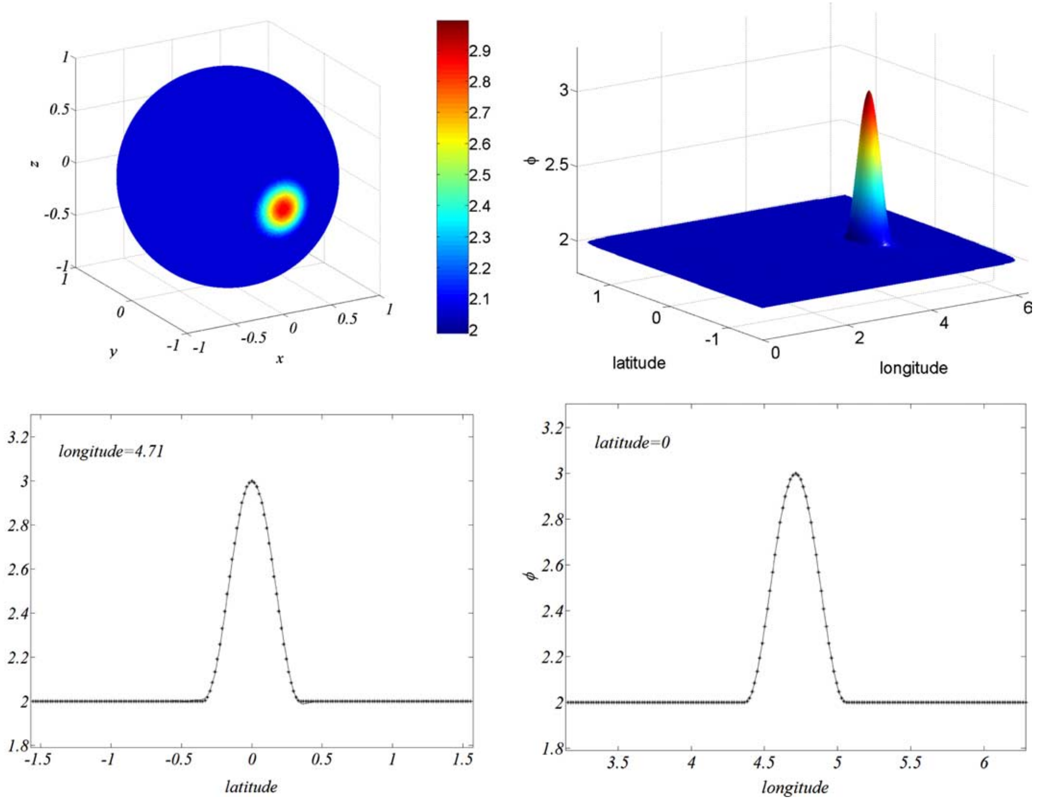

Fig. 9. Solid body rotation of a cosine hill after one revolution using scheme $\mathscr{S}_1$. The upper right panel shows results plotted on the sphere directly from the polygons defining the control volume mesh, and the left upper panel the numerical solution interpolated from the geodesic mesh to the latitude–longitude grid. The right lower and the left lower panels compare the numerical solutions (solid line) and the analytical solution (line marked by stars) along the lines passing respectively through the north and south poles and along the equator.

$$\mathscr{M} = \frac{\mathscr{I}(\phi) - \mathscr{I}(\phi_a)}{\mathscr{I}(\phi_0)}. \tag{56}$$

The ability of the scheme to preserve minimum and maximum values is measured by two quantities,

$$f_{\min} = \frac{\min(\phi) - \min(\phi_0)}{\Delta\phi_0}, \tag{57}$$

$$f_{\max} = \frac{\max(\phi) - \min(\phi_0)}{\Delta\phi_0}, \tag{58}$$

where $\Delta\phi_0 = \max(\phi_0) - \min(\phi_0)$ (for an ideal scheme for passive advection we should have $f_{\min} = 0$ and $f_{\max} = 1$).

The data in Table 2 contain the summary of the experiments with advection of a cosine hill. For brevity, we label schemes (33) and (35) by $\mathscr{S}_1$ and $\mathscr{S}_2$, respectively. The numerical data confirm an overall good accuracy of both schemes. In particular, the maximum and minimum values are very well preserved. Both methods are also mass conserving; quantity $\mathscr{M}$ defined by (56) is equal to zero with the precision of machine

Table 2
Summary of the experiments with the cosine hill

Method	l_1	l_2	l_∞	$\mathscr{M}$	$f_{\min}$	$f_{\max}$
Upwind	0.0037	0.0205	0.2581	0.0000	−0.0001	0.3846
$\mathscr{S}_1$	0.0003	0.0008	0.0065	0.0000	−0.0125	0.9989
$\mathscr{S}_2$	0.0002	0.0005	0.0034	0.0000	−0.0101	0.9980

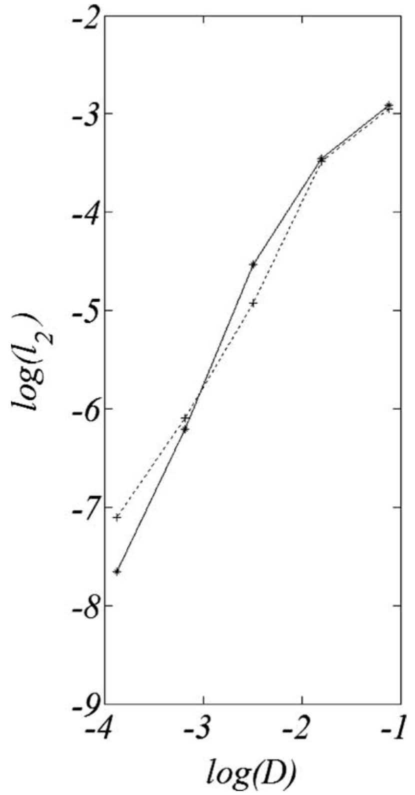

Fig. 10. The logarithm of l_2 as a function of the logarithm of the average distance between the nodes of the geodesic mesh (scheme $\mathscr{S}_1$ is represented by the dashed line, and scheme $\mathscr{S}_2$ by the solid line).

accuracy. Method $\mathscr{S}_2$ is more accurate than method $\mathscr{S}_1$ with respect to l_2 and l_∞ norms. For reference purposes, we also include results obtained with the upwind scheme (13). The examination of $f_{\min}$ and $f_{\max}$ values clearly points to the fact that the upwind scheme leads to a rather severe reduction of the initial amplitude and, for this reason, is not practical for solving problems where preservation of the extreme values is important.

In order to illustrate the convergence of the scheme, we solved the advection equation on grids with different resolutions. After each of the experiments, we calculated l_2 norm. The logarithm of l_2 versus the logarithm of the average distance between nodes of the grid is depicted in Fig. 10. Examination of the average slope of the lines shown in this figure indicates that both schemes achieve approximately the second order of accuracy; scheme $\mathscr{S}_2$ is converging slightly faster, particularly in the range of finer resolutions. The fact that schemes $\mathscr{S}_1$ and $\mathscr{S}_2$ exhibit the same rate of convergence despite the different interpolation formulae is caused by the use of the mid-point rule for the integral term approximation in (6) (see Eq. (7) for illustration). The sample solutions for the scheme $\mathscr{S}_1$ on grids number 3, 4, 5, and 6 are depicted in Fig. 11. Indeed, the method converges quite quickly as shown in the plot of l_2 norm in Fig. 10.

In addition to the standard test with cosine hill, we also investigated advection of a cylindrical mass distribution which can be obtained from (52) after substituting $g(\mathbf{r}_i) = 1$. The velocity field in the latter advection test was the same as that for the cosine hill test. The summary of the experiments is presented in Table 3 where, for reference, the results obtained for the upwind scheme are also included.

The examination of $f_{\min}$ and $f_{\max}$ values for schemes $\mathscr{S}_1$ and $\mathscr{S}_2$ indicates clearly that the method is not monotonic for the cylindrical mass distribution as predicted by the Godunow theorem [7]. However, this undesirable property can be eliminated by adding dissipation as discussed in Section 6. Sample results for the advection of a cylindrical mass distribution performed using scheme $\mathscr{S}_1$ with the with three corrective iterations (50) and $v = 10^{-3}$ is presented in Fig. 12. Examination of the shape of the solution confirms that the

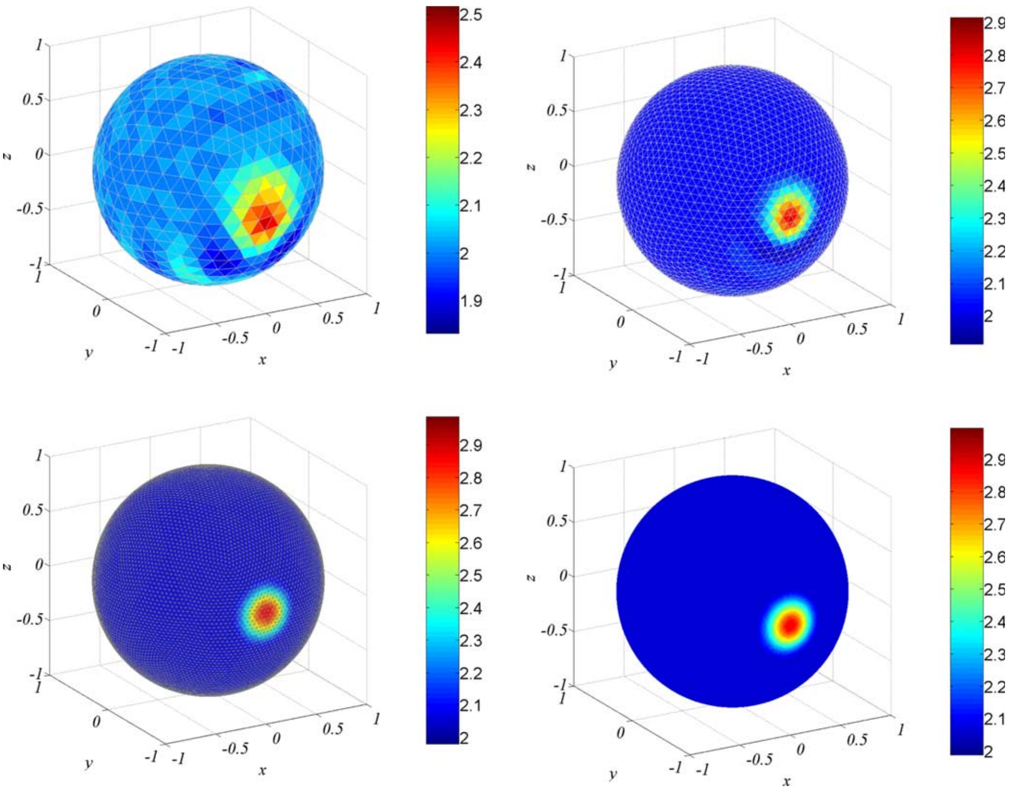

Fig. 11. The illustration of the convergence of $\mathscr{S}_1$ scheme. The left upper, right upper, left bottom, and right bottom panels show the solution obtained after one revolution on grids number 3, 4, 5, and 6, respectively.

Table 3
Summary of the experiments with the cylindrical mass distribution

Method	l_1	l_2	l_∞	$\mathscr{M}$	$f_{\min}$	$f_{\max}$
Upwind	0.0115	0.0430	0.2334	0.0000	−0.0001	0.8341
$\mathscr{S}_1$	0.0114	0.0213	0.1819	0.0000	−0.2320	1.2930
$\mathscr{S}_2$	0.0117	0.0237	0.2197	0.0000	−0.3280	1.3060
$\mathscr{S}_1 +$ ELAD	0.0027	0.0167	0.1869	0.0000	−0.0183	1.0263
$\mathscr{S}_2 +$ ELAD	0.0028	0.0174	0.1919	0.0000	−0.0196	1.0425
FCT	0.0024	0.0181	0.2150	0.0000	−0.0005	1.0006

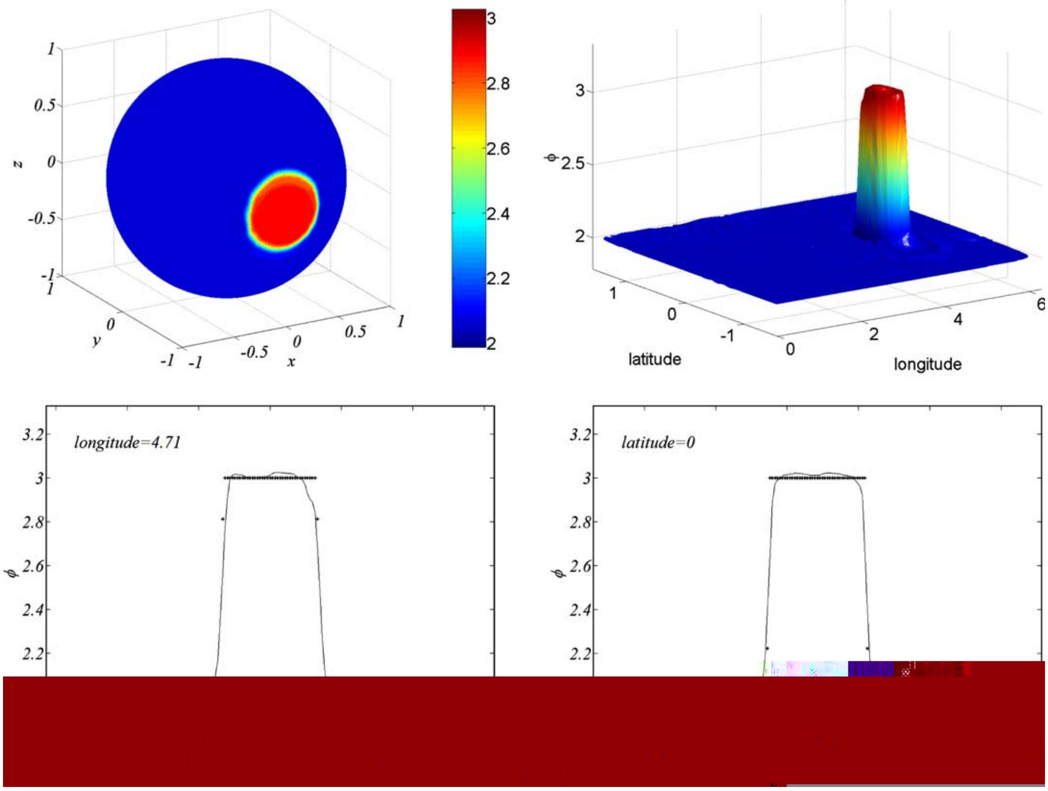

Fig. 12. Same as in Fig. 9 but for a cylindrical mass distribution.

results are quasi-monotonic. Consequently, the minimum and maximum values are well preserved. The most important property of the schemes with adaptive dissipation is that mass is conserved down to the machine accuracy, similarly as for the the original schemes, $\mathscr{S}_1$ and $\mathscr{S}_2$, without the dissipation mechanism.

In order to evaluate the accuracy of the monotonic versions of $\mathscr{S}_1$ and $\mathscr{S}_2$, we compared them to the flux corrected transport (FCT) algorithm ([46] and [20], p. 285). The solution of the advection equation for a cylindrical mass distribution obtained with FCT is depicted in Fig. 13. The comparison of the results shown in Figs. 12 and 13 indicates clearly that both schemes produce almost equivalent results. Furthermore, the set of norms in Table 3 provides some additional evidence of the nearly similar performance of $\mathscr{S}_1 + ELAD$, $\mathscr{S}_2 + ELAD$ and *FCT*. This fact confirms that various approaches to the monotonicity of advective schemes are in many ways equivalent despite the different types of constraints employed to assure the monotonicity of the solutions.

The tests with both cosine hill and cylindrical mass distribution are not sufficient to comprehensively evaluate a numerical scheme. In order to complete the investigation of the accuracy of the method, the additional

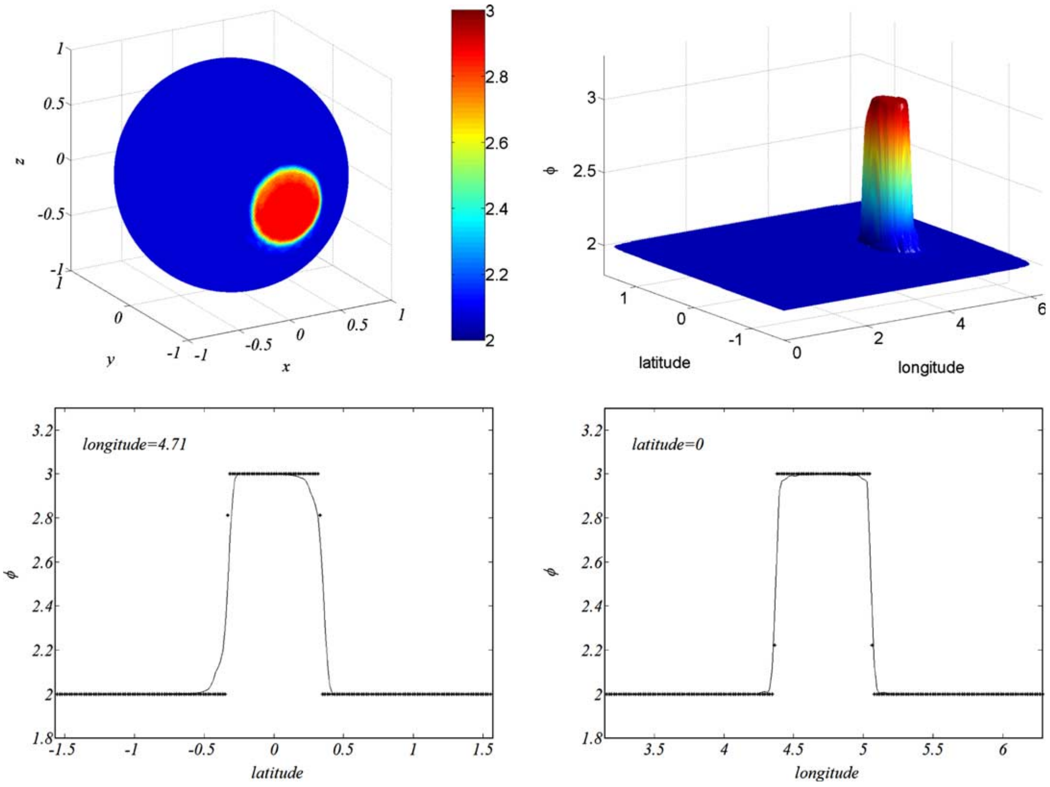

Fig. 13. Same as in Fig. 12 but for results obtained with the flux corrected transport (FCT) algorithm.

test with an initial condition containing both smooth and discontinuous parts was performed. A good example of such an initial condition is the multiscale signal shown in Fig. 14 and described by the following function of θ and λ:

$$\phi(\theta,\lambda) = \cos^4(\theta)(f_1(\lambda)+f_2(\lambda)+2)\left(1+0.3\sin\left(\frac{50}{9}\lambda\right)\right)\left(1+0.4\sin\left(\frac{50}{10}\lambda\right)\right), \tag{59}$$

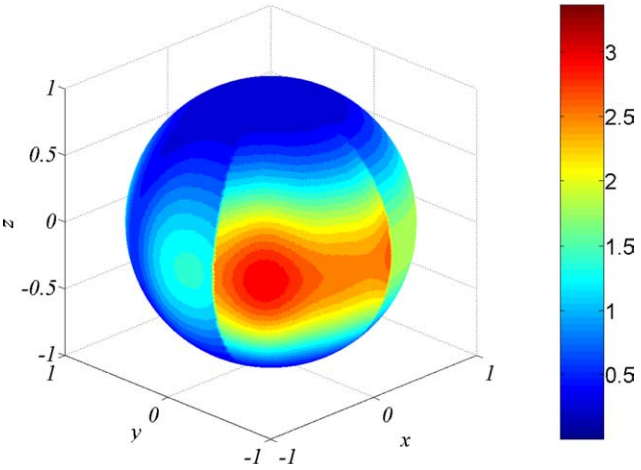

Fig. 14. The multiscale signal at the initial time.

where $f_1(\lambda) = -1$ for $\lambda \in \mathscr{D}_1 = [8\pi/25, 28\pi/25]$; $f_1(\lambda) = 0$ for $\lambda \in [0, 2\pi] - \mathscr{D}_1$, and $f_2(\lambda) = 1$ for $\lambda \in \mathscr{D}_2 = (28\pi/25, 39\pi/50]$; $f_2(\lambda) = 0$ for $\lambda \in [0, 2\pi] - \mathscr{D}_2$.

The function described by (59) is a product of $\cos^4(\theta)$ and the formula suggested by Smolarkiewicz and Grabowski [32] for testing monotonic advection schemes. The main property of the function depicted in Fig. 14 is a discontinuous manner in which smooth 2-D distributions on the spherical surface are connected with each other.

Considering the symmetry of the initial condition depicted in Fig. 14, we selected the velocity field (51), with parameter $\alpha = 0$ (this corresponds to the uniform flow around the Equator). In order to assure the monotonicity of the solution, we used two corrective iterations and $v = 0.25 \times 10^{-3}$. The results from the advection of profile (59) obtained from $\mathscr{S}_1 + ELAD$ are depicted in two panels on the left-hand side of Fig. 15. Evidently, the numerical solution preserves general shape of the initial condition. In particular, small-scale transition zones between continuous parts of the field are maintained with a reasonable accuracy. The solution obtained from the FCT algorithm is shown on the right-hand side of Fig. 15. The comparison of the two solutions shown in Fig. 15 indicates clearly that both schemes produce quite equivalent results. The main noticeable difference is that the shape of small scale transition zones between continuous parts of the field is maintained by the monotonic version of $\mathscr{S}_1$ method with significantly better accuracy than by the FCT algorithm. Examination of data in Table 4 further indicates that the monotonic version of $\mathscr{S}_1$ has also smaller l_2 and l_∞ norms than the FCT algorithm.

The results of the suite of the solid body rotation tests indicate that the presented method preserves relatively well the shape of the advected profiles. In particular, circular shapes are advected without characteristic elongation of the original shape. This effect was especially evident in the solutions obtained with the linear

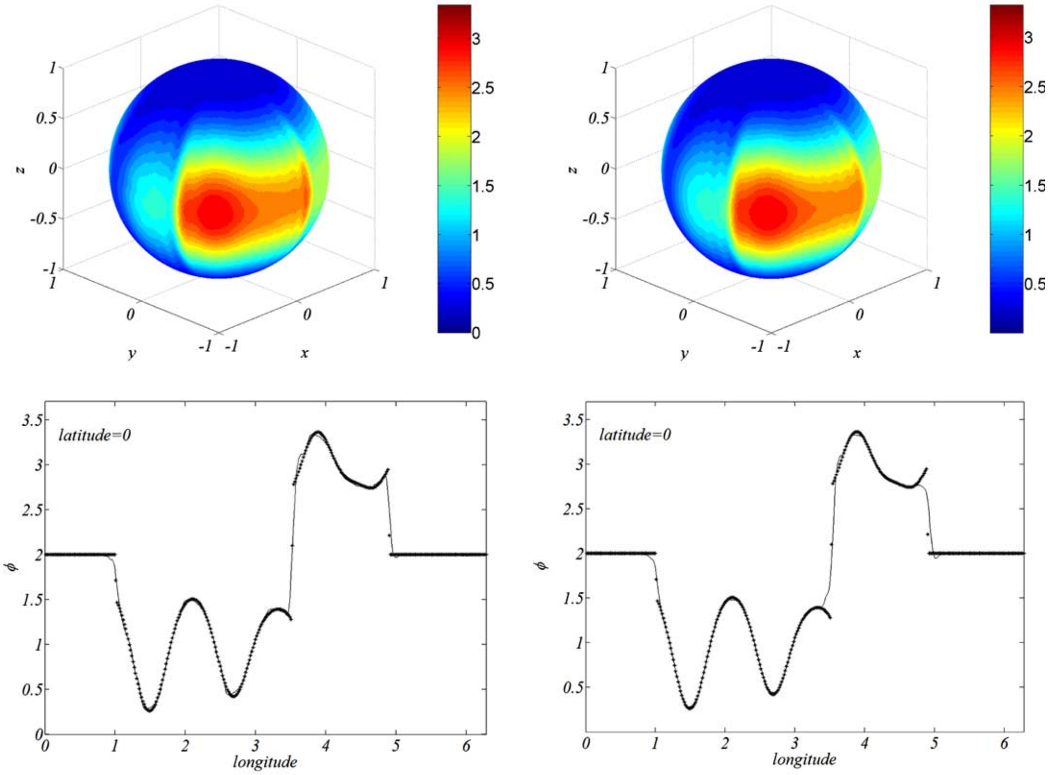

Fig. 15. The solid body rotation test of the multiscale signal; results after one revolution along the equator. The left upper panel shows results obtained using $\mathscr{S}_1$ scheme with ELAD whereas the left lower panel depicts both the analytical solution (line marked by stars) and the numerical solution (line marked by stars) along the equator. The right upper and lower panels show the analogical information obtained with the FCT algorithm.

Table 4
Summary of the experiments with the multiscale signal

Method	l_1	l_2	l_∞	$\mathscr{M}$	$f_{\min}$	$f_{\max}$
Upwind	0.1246	0.1301	0.2665	0.0000	0.0000	0.8692
$\mathscr{S}_1$	0.0266	0.0392	0.2223	0.0000	−0.0101	1.0109
$\mathscr{S}_2$	0.0289	0.0445	0.2607	0.0000	−0.0119	1.0174
$\mathscr{S}_1$ + ELAD	0.0180	0.0354	0.2250	0.0000	−0.0004	0.9942
$\mathscr{S}_2$ + ELAD	0.0196	0.0393	0.2587	0.0000	−0.0009	0.9916
FCT	0.0145	0.0386	0.2646	0.0000	0.0000	0.9882

interpolation applied for the evaluation of the interface value. The adaptive dissipation is sufficient to suppress numerical noise without significant damping of the amplitude of the main signal. Furthermore, the dissipation scheme does not affect the conservation properties.

The algorithm is also very compact and efficient. The entire calculation is accomplished by a single matrix vector multiplication followed by an iterative ELAD procedure. In order to evaluate the timing of the algorithm, it is convenient to do this with respect to the upwind scheme (13). The algorithm without the monotonic option is four times slower than simple upwind scheme. With the addition of the ELAD option with two iterations, the scheme is about 10 times slower when compared to the upwind algorithm.

8. Incorporation of terms representing coupling between scalar fields

The semidiscrete approximation to the advection–diffusion equation with reaction terms can be written in a manner analogous to (6) as

$$\frac{\mathrm{d}\phi_i^k}{\mathrm{d}t} = -\frac{1}{S(\Omega_i)}\int_{\partial\Omega_i}(\mathbf{u}\varphi^k - \mathbf{K}\nabla\varphi^k)\mathbf{n}\,\mathrm{d}l + \frac{1}{S(\Omega_i)}\int_{\Omega_i} F_c^k\,\mathrm{d}\mathbf{r}, \tag{60}$$

where $i \in [1,N_\mathrm{p}]$, and $k \in [1,N_\mathrm{s}]$, N_s is the number of interacting scalar fields. After applying the discretization defined by (44), we can rewrite (60) in the vector form as

$$\frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} = -(\mathbf{I}_{N_\mathrm{s}} \otimes \mathbf{A}_{\mathscr{D}})\boldsymbol{\phi} + \mathbf{F}(\boldsymbol{\phi}), \tag{61}$$

where $\mathbf{A}_{\mathscr{D}}$ is defined by (45), $\mathbf{I}_{N_\mathrm{s}}$ is the $N_\mathrm{s} \times N_\mathrm{s}$ unit diagonal matrix, $\otimes$ denotes the Kronecker product, and

$$\boldsymbol{\phi} = [\phi_1^1, \phi_2^1, \ldots, \phi_{N_\mathrm{p}}^1, \phi_1^2, \phi_2^2, \ldots, \phi_{N_\mathrm{p}}^2, \ldots, \phi_1^{N_\mathrm{s}}, \phi_2^{N_\mathrm{s}}, \ldots, \phi_{N_\mathrm{p}}^{N_\mathrm{s}}]^\mathrm{T},$$
$$\mathbf{F} = \left[\langle F_{c1}^1\rangle, \ldots, \langle F_{cN_\mathrm{p}}^1\rangle, \langle F_{c1}^2\rangle, \ldots, \langle F_{cN_\mathrm{p}}^2\rangle, \ldots, \langle F_{c1}^{N_\mathrm{s}}\rangle, \ldots, \langle F_{cN_\mathrm{p}}^{N_\mathrm{s}}\rangle\right]^\mathrm{T}.$$

There are several possible techniques for the numerical integration of (61). In the majority of applications this equation is solved using the fractional steps method [20,45]. The advantage of this approach is its flexibility in combining the advection algorithm with a specialized chemical solver (various advection schemes can be linked to different chemical solvers in order to create the algorithm suited for a specific application). For the system of Eqs. (61), the fractional step method amounts to splitting the equation into two parts:

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} &= -(\mathbf{I}_{N_\mathrm{s}} \otimes \mathbf{A}_{\mathscr{D}})\boldsymbol{\phi},\\ \frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} &= \mathbf{F}(\boldsymbol{\phi}).\end{aligned} \tag{62}$$

At each time step, the advection–diffusion problem is solved first using the algorithm described in the previous sections. The obtained solution serves then as an initial condition for the second of equations in (62) which is integrated with the help of the appropriate ODE solver. Such an approach has already been used in models of atmospheric chemistry (see for example [24]). The selection of a specific method for the numerical integration of the chemistry part depends primarily on the form of **F** in (62). In instances when the coupling of equations involves very different time scales, specialized stiff solvers need to be used [21]. In the context of atmospheric sciences, the list of possible techniques include both specialized explicit solvers and highly accurate implicit methods. The brief description of these techniques and their evaluation is presented in [29].

An alternative approach to the operator splitting is the direct solution of Eq. (61) using an appropriate ODE solver. The selection of a scheme depends mainly on values of the diffusion coefficient as well as the stiffness of the chemical terms. In the simplest case of the advection dominated problem with weak diffusion and nonstiff chemical terms, we can apply the explicit fourth order Runge–Kutta method, so the compactly written solution of (61) is

$$\begin{aligned}
&\boldsymbol{\phi}(t+\Delta t) = \boldsymbol{\phi}(t) + \frac{1}{6}(\mathbf{f}_1 + 2\mathbf{f}_2 + 2\mathbf{f}_3 + \mathbf{f}_4),\\
&\mathbf{f}_1 = -\Delta t(\mathbf{I}_{N_s}\otimes\mathbf{A}_{\mathscr{D}})\boldsymbol{\phi}(t) + \Delta t\mathbf{F}(\boldsymbol{\phi}(t)),\\
&\mathbf{f}_2 = -\frac{\Delta t}{2}(\mathbf{I}_{N_s}\otimes\mathbf{A}_{\mathscr{D}})(\boldsymbol{\phi}+\mathbf{f}_1) + \frac{\Delta t}{2}\mathbf{F}(\boldsymbol{\phi}+\mathbf{f}_1),\\
&\mathbf{f}_3 = -\frac{\Delta t}{2}(\mathbf{I}_{N_s}\otimes\mathbf{A}_{\mathscr{D}})(\boldsymbol{\phi}+\mathbf{f}_2) + \frac{\Delta t}{2}\mathbf{F}(\boldsymbol{\phi}+\mathbf{f}_2),\\
&\mathbf{f}_4 = -\Delta t(\mathbf{I}_{N_s}\otimes\mathbf{A}_{\mathscr{D}})(\boldsymbol{\phi}+\mathbf{f}_3) + \Delta t\mathbf{F}(\boldsymbol{\phi}+\mathbf{f}_3).
\end{aligned}\tag{63}$$

When the stiffness of the chemical terms is a factor but diffusion is relatively small we can apply a semi-implicit technique in which the transport operator is treated explicitly whereas chemical terms are treated implicitly. The example of such a technique is provided by the class of additive semi-implicit Runge–Kutta schemes [47]. The coefficients of these schemes are derived in such a manner that the methods are both high order accurate and strongly A-stable for the implicit terms. The specific variant of the method of this type is a third order additive semi-implicit Runge–Kutta scheme derived in [47]. The corresponding algorithm can be written compactly as:

$$\begin{aligned}
&\boldsymbol{\phi}(t+\Delta t) = \boldsymbol{\phi}(t) + \omega_1\mathbf{f}_1 + \omega_2\mathbf{f}_2 + \omega_3\mathbf{f}_3,\\
&\mathbf{f}_1 = \mathbf{P}_c^{\mathrm{T}}\{[\mathbf{I} - \alpha_1\Delta t\mathbf{J}_c(\boldsymbol{\phi}_0)]^{-1}\mathbf{P}_c[\Delta t\mathbf{F}_{\mathrm{T}}(\boldsymbol{\phi}_0) + \Delta t\mathbf{F}(\boldsymbol{\phi}_0)]\},\\
&\mathbf{f}_2 = \mathbf{P}_c^{\mathrm{T}}\{[\mathbf{I} - \alpha_2\Delta t\mathbf{J}_c(\boldsymbol{\phi}_0)]^{-1}\mathbf{P}_c[\Delta t\mathbf{F}_T(\boldsymbol{\phi}_0 + b_{21}\mathbf{f}_1) + \Delta t\mathbf{F}(\boldsymbol{\phi}_0 + c_{21}\mathbf{f}_1)]\},\\
&\mathbf{f}_3 = \mathbf{P}_c^{\mathrm{T}}\{[\mathbf{I} - \alpha_3\Delta t\mathbf{J}_c(\boldsymbol{\phi}_0)]^{-1}\mathbf{P}_c\Delta t[\mathbf{F}_{\mathrm{T}}(\boldsymbol{\phi}_0 + b_{31}\mathbf{f}_1 + b_{32}\mathbf{f}_2) + \mathbf{F}(\boldsymbol{\phi}_0 + c_{31}\mathbf{f}_1 + c_{32}\mathbf{f}_2)]\},
\end{aligned}\tag{64}$$

where α_i, ω_i, b_{ij}, c_{ij} are constants, $\mathbf{F}_{\mathrm{T}}(\boldsymbol{\phi}) = -(\mathbf{I}_{N_s}\otimes\mathbf{A}_{\mathscr{D}})\boldsymbol{\phi}$, $\mathbf{I}$ is the unit diagonal matrix of size $N_pN_s \times N_pN_s$, $\mathbf{J} = \{\partial\mathbf{F}/\partial\boldsymbol{\varphi}\}|_{\boldsymbol{\phi}_0}$ is the Jacobian of the chemical system, $\boldsymbol{\phi}_0 \equiv \boldsymbol{\phi}(t=0)$, $\mathbf{P}_c$ is the permutation operator changing ordering of elements of $\boldsymbol{\phi}$ to the chemical sequence: $\tilde{\boldsymbol{\phi}} = [\phi_1^1, \phi_1^2, \ldots, \phi_1^{N_s}\phi_2^1, \phi_2^2, \ldots, \phi_2^{N_s}\phi_{N_p}^1, \phi_{N_p}^2, \ldots, \phi_{N_p}^{N_s}]^{\mathrm{T}}$. (Note that $\tilde{\boldsymbol{\phi}} = \mathbf{P}_c\boldsymbol{\phi}$, $\mathbf{P}_c$ is unitary and $\mathbf{P}_c^{-1} \equiv \mathbf{P}_c^{\mathrm{T}}$.)

The strategy of integration of reactive flow problems using the semi-implicit Runge–Kutta schemes is becoming more popular in the area of atmospheric chemistry [10]. The main reason being the simplicity of algorithms and good performance in the integration of stiff chemical systems. The selection of the methods for integration of (61) without operator splitting can be further expanded by the modification of (64) by including the diffusion terms implicitly. This is particularly important when the significance of the diffusion process is comparable to that of chemical reactions.

In the following section, we will investigate the properties of the numerical solution of the coupled set of the mass conservation equations using schemes (63) and (64). The numerical solutions will be evaluated using the analytical solutions of the Turing [38] system.

9. Evaluation of the algorithm for the reaction–diffusion system

The addition of chemical reactions considerably enriches the class of possible solutions as compared to a simple advection. Considering this fact, it is important to perform the evaluation of a numerical solution for a system of reactions which is relatively simple but still admits interesting and complex solutions. These requirements are well satisfied for the following system of reactions describing the autocatalytic conversion of species $\mathscr{X}$ to $\mathscr{Y}$:

$$\begin{aligned}
&\mathscr{A} \rightarrow \mathscr{X}, \\
&\mathscr{B} + \mathscr{X} \rightarrow \mathscr{Y} + \mathscr{C}, \\
&2\mathscr{X} + \mathscr{Y} \rightarrow 3\mathscr{X}, \\
&\mathscr{X} \rightarrow \mathscr{D}.
\end{aligned} \tag{65}$$

In the above system, the concentrations of species $\mathscr{A}$ and $\mathscr{B}$ are fixed and species $\mathscr{C}$, and $\mathscr{D}$ are byproducts. The system of reactions, (65), known as Brusselator was introduced in the context of the investigation of oscillations in chemical systems [22]. In terms of kinetic equations, the set of reactions, (65), can be described as

$$\begin{aligned}
&\frac{\mathrm{d}\varphi^1}{\mathrm{d}t} = k_1 A - k_2 B\varphi^1 + k_3(\varphi^1)^2\varphi^2 - k_4\varphi^1, \\
&\frac{\mathrm{d}\varphi^2}{\mathrm{d}t} = k_2 B\varphi^1 - k_3(\varphi^1)^2\varphi^2,
\end{aligned} \tag{66}$$

where k_1, k_2, k_3 and k_4 are the reaction rates, φ^1, φ^2, A and B are concentrations of species $\mathscr{X}$, $\mathscr{Y}$, $\mathscr{A}$ and $\mathscr{B}$, respectively. In the subsequent discussion it is assumed, without the loss of generality, that all reaction rates are equal to one. The system (66) has a single stationary point with coordinates $\varphi_\mathrm{s}^1 = A$ and $\varphi_\mathrm{s}^2 = B/A$. The sample phase trajectories of (66) are depicted in Fig. 16; notice that the point with coordinates φ_s^1 and φ_s^2 is the attractor of the system.

After the reaction terms defined by (66) are included in the diffusion equation, the following system is obtained

$$\frac{\partial \boldsymbol{\varphi}}{\partial t} = \mathbf{F}(\boldsymbol{\varphi}) + \mathbf{K}\nabla^2\boldsymbol{\varphi}, \tag{67}$$

where

$$\boldsymbol{\varphi} = \begin{pmatrix} \varphi^1 \\ \varphi^2 \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} A - B\varphi^1 + (\varphi^1)^2\varphi^2 - \varphi^1 \\ B\varphi^1 - (\varphi^1)^2\varphi^2 \end{pmatrix}, \quad \mathbf{K} = \begin{pmatrix} \mu & 0 \\ 0 & v \end{pmatrix},$$

and μ and v are the diffusion coefficients for φ^1 and φ^2, respectively.

The behaviour of system (67) is quite complex (see [23] for the general discussion). The most interesting feature is the emergence of different spatial patterns which appear even when equations are integrated with simple homogeneous initial conditions. The second salient feature is the occurrence of inhomogeneous stationary states which can be obtained for some values of the control parameters, A, B, μ, and v. The investigation of structures governed by (67) illustrates the existence of complex nonlinear behaviour in the dissipative system.

Before proceeding to the analysis of some aspects of the numerical solutions, we first analyse the linear version of (67). Let us start with the Taylor series expansion around the stationary point

$$\frac{\partial(\boldsymbol{\varphi}_\mathrm{s} + \delta\boldsymbol{\varphi})}{\partial t} = \mathbf{F}(\boldsymbol{\varphi}_\mathrm{s}) + \left\{\frac{\partial \mathbf{F}}{\partial \boldsymbol{\varphi}}\right\}\bigg|_{\boldsymbol{\varphi}_\mathrm{s}} \delta\boldsymbol{\varphi} + \cdots + \mathbf{K}\nabla^2(\boldsymbol{\varphi}_\mathrm{s} + \delta\boldsymbol{\varphi}), \tag{68}$$

where

$$\delta\boldsymbol{\varphi} = \boldsymbol{\varphi} - \boldsymbol{\varphi}_\mathrm{s} = \begin{pmatrix} \delta\varphi^1 \\ \delta\varphi^2 \end{pmatrix}, \quad \boldsymbol{\varphi}_\mathrm{s} = \begin{pmatrix} \varphi_\mathrm{s}^1 \\ \varphi_\mathrm{s}^2 \end{pmatrix}, \quad \left\{\frac{\partial \mathbf{F}}{\partial \boldsymbol{\varphi}}\right\}\bigg|_{\boldsymbol{\varphi}_\mathrm{s}} = \begin{pmatrix} B-1 & A^2 \\ -B & -A^2 \end{pmatrix}.$$

Considering that $\nabla^2\boldsymbol{\varphi}_s \equiv 0$ and $\partial\boldsymbol{\varphi}_s/\partial t \equiv 0$, we can rewrite the set of equations (68) in the following linearized form:

$$\begin{aligned}
&\frac{\partial\delta\varphi^1}{\partial t} = a\delta\varphi^1 + b\delta\varphi^2 + \mu\nabla^2\delta\varphi^1, \\
&\frac{\partial\delta\varphi^2}{\partial t} = c\delta\varphi^1 + d\delta\varphi^2 + v\nabla^2\delta\varphi^2,
\end{aligned} \tag{69}$$

where $a = B - 1$, $b = A^2$, $c = -B$, and $d = -A^2$.

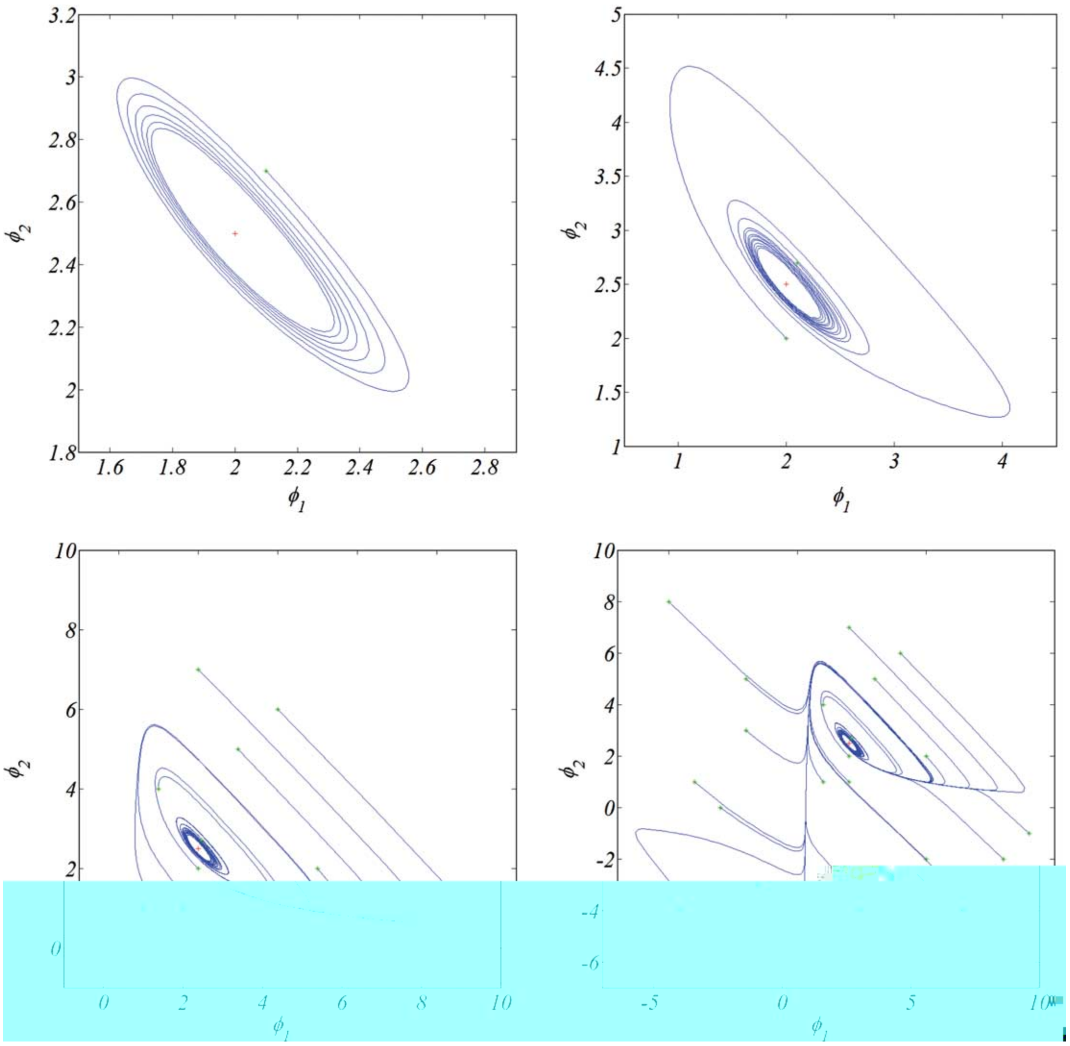

Fig. 16. Phase trajectories of the Brusselator system for $A = 2$, $B = 5$, and $k_1 = k_2 = k_3 = k_4 = 1$. The graphs in the left upper, right upper, left lower, and right lower panels show the trajectories for the progressively larger basin of influence.

The system of equations, (69), has the same form as that used by Turing [38] in the study of the reaction–diffusion waves on the sphere. Here we use the methodology suggested by Turing [38] in order to find the analytical solution of (69) and to evaluate the numerical solution of the reaction–diffusion system on a geodesic grid. In particular, we seek the solutions of (69) which can be expressed as a linear combination of spherical harmonics:

$$\delta\varphi^1(\theta,\lambda,t) = \sum_{n=0}^{\infty}\sum_{m=-n}^{m=n} A_n^m(t)Y_n^m(\theta,\lambda), \quad \delta\varphi^2(\theta,\lambda,t) = \sum_{n=0}^{\infty}\sum_{m=-n}^{m=n} B_n^m(t)Y_n^m(\theta,\lambda), \tag{70}$$

where $A_n^m(t), B_n^m(t) \in \mathbf{R}^1$, and $Y_n^m(\theta,\lambda)$ is the spherical harmonic.

After substituting (70) into (69), applying $\nabla Y_n^m = -\frac{n(n+1)}{r^2}Y_n^m$ and taking into account orthogonality of Y_n^m, we obtain the following system of the ODEs for A_n^m and B_n^m:

$$\begin{aligned}\frac{\mathrm{d}A_n^m}{\mathrm{d}t} &= \left(a - \frac{\mu}{r^2}n(n+1)\right)A_n^m + bB_n^m,\\ \frac{dB_n^m}{dt} &= cA_n^m + \left(d - \frac{v}{r^2}n(n+1)\right)B_n^m.\end{aligned} \tag{71}$$

The analytical solution of (71) can be obtained from the theory of the linear ODEs (see for example [1, p. 123]) as:

$$\begin{pmatrix} A_n^m \\ B_n^m \end{pmatrix} = \sum_{k=1}^{2} \mathscr{C}_k \mathbf{X}_k^n \exp \omega_k^n t, \tag{72}$$

where $\mathscr{C}_k$ are the constants evaluated from the initial conditions, and $\mathbf{X}_k^n$ are the eigenvectors of $\mathscr{J}_{\mathscr{D}}^n$, the latter being defined as:

$$\mathscr{J}_{\mathscr{D}}^n = \begin{pmatrix} a_n & b \\ c & d_n \end{pmatrix}, \quad a_n = a - \frac{\mu}{r^2}n(n+1), \quad d_n = a - \frac{v}{r^2}n(n+1),$$

ω_k^n are the roots of the characteristic equation of (71):

$$\det(\omega\mathbf{I} - \mathscr{J}_{\mathscr{D}}^n) = \omega^2 - \operatorname{tr}(\mathscr{J}_{\mathscr{D}}^n)\omega + \det(\mathscr{J}_{\mathscr{D}}^n) = 0, \tag{73}$$

where $\mathbf{I}$ is a 2×2 unit matrix, $\operatorname{tr}(\mathscr{J}_{\mathscr{D}}^n) = a_n + d_n$, and $\det(\mathscr{J}_{\mathscr{D}}^n) = a_n d_n - bc$. The expressions for the roots can be written compactly as

$$\begin{aligned}&\omega_{n(1,2)} = \gamma_n \pm \delta_n,\\ &\gamma_n = \frac{1}{2}\operatorname{tr}(\mathscr{J}_{\mathscr{D}}^n), \quad \delta_n = \frac{1}{2}\sqrt{\Delta^n},\\ &\Delta^n = \left(\operatorname{tr}^2(\mathscr{J}_{\mathscr{D}}^n) - 4\det(\mathscr{J}_{\mathscr{D}}^n)\right).\end{aligned} \tag{74}$$

The nature of the roots of (74) and, consequently, the type of the solution of (72) depend on the signs of Δ^n, $\operatorname{tr}(\mathscr{J}_{\mathscr{D}}^n)$, and $\det(\mathscr{J}_{\mathscr{D}}^n)$ (see [1,3] for the discussion of possible cases). In particular, the solutions of the form of damped harmonic oscillations are obtained for $\Delta^n < 0$ and $\operatorname{tr}(\mathscr{J}_{\mathscr{D}}^n) < 0$. In terms of the relation between A and B, these conditions can be expressed by the following requirements:

$$\begin{aligned}&A^2 - 2A\eta_n + \eta_n^2 < B < A^2 + 2A\eta_n + \eta_n^2,\\ &B < A^2 + \zeta_n,\end{aligned} \tag{75}$$

where

$$\eta_n^2 = 1 + \frac{n(n+1)}{r^2}(\mu - v), \quad \zeta_n = 1 + \frac{n(n+1)}{r^2}(\mu + v).$$

After evaluation of γ_n, δ_n and constants in the general solution, we obtain the explicit formulae for the analytical solution of (69):

$$\begin{aligned}\delta\varphi^1 &= \sum_{n=0}^{\infty}\sum_{m=-n}^{m=n} \mathrm{e}^{\gamma_n t}(\mathscr{A}_n^m \cosh(\delta_n t) + \mathscr{B}_n^m \sinh(\delta_n t))Y_n^m(\theta, \lambda),\\ \delta\varphi^2 &= \sum_{n=0}^{\infty}\sum_{m=-n}^{m=n} \mathrm{e}^{\gamma_n t}(\mathscr{C}_n^m \cosh(\delta_n t) + \mathscr{D}_n^m \sinh(\delta_n t))Y_n^m(\theta, \lambda),\end{aligned} \tag{76}$$

where

$$\begin{aligned}&\mathscr{A}_n^m = A_n^m(0),\\ &\mathscr{B}_n^m = (2\widetilde{\alpha}_n A_n^m(0) + 2bB_n^m(0))/\delta_n,\\ &\mathscr{C}_n^m = B_n^m(0),\\ &\mathscr{D}_n^m = (2cA_n^m(0) + 2\widetilde{\beta}_n B_n^m(0))/\delta_n\end{aligned}$$

and

$$\widetilde{\alpha}_n = \frac{1}{2}(a_n - d_n), \qquad \widetilde{\beta}_n = -\widetilde{\alpha}_n.$$

In his seminal paper, Turing [38] referred to the solution of the form of (76) as reaction–diffusion waves.

The solution expressed by (76) can be used to evaluate the accuracy of the numerical algorithm on the icosahedral grid. The initial conditions used in the test were described by (70) with $n = 5$; coefficients $A_5^m(0), B_5^m(0)$ were selected randomly as $A_5^m = [-0.5839, -0.8436, -0.4764, 0.6475, 0.1886, 0.8709, -0.8338, 0.1795, -0.7873, 0.8842, 0.2943]$, and $B_5^m = [-0.6263, 0.9803, 0.7222, 0.5945, 0.6026, -0.2076, 0.4556, 0.6024, 0.9695, -0.4936, 0.1098]$ for $m \in [-5, 5]$. The corresponding initial conditions $\varphi^{1,2}(\theta, \lambda, 0)$ are depicted in Fig. 17. The following parameters characterizing the reaction–diffusion system are assumed: $A = 2$, $B = 5$, $\mu = 10^{-3}$, and $v = 2.0 \times 10^{-3}$; these values lead to $\gamma_5 = -0.0450$ and $\delta_5 = 1.9697i$ which corresponds to a slightly damped harmonic oscillations of both fields.

The numerical solution of (69) is obtained with the help of (63). Considering the fact that the system of equations, (69) is linear, we can write the term $\mathbf{F}(\delta\boldsymbol{\phi})$ in the following simpler form:

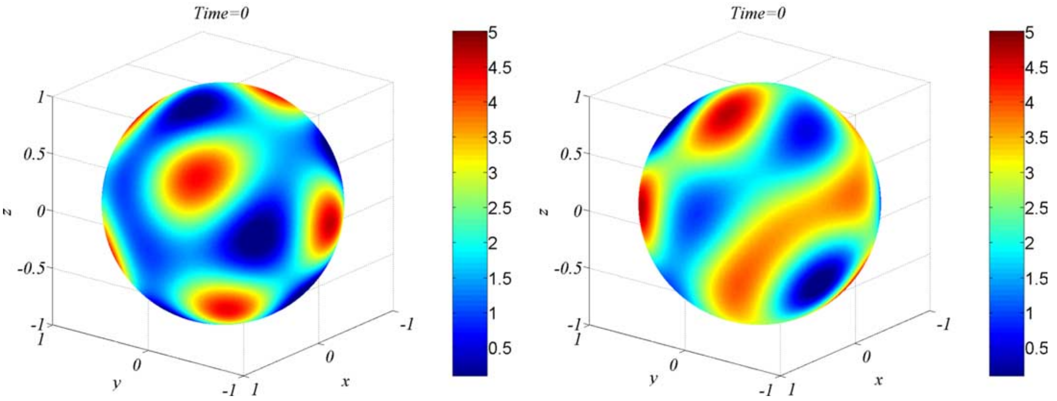

Fig. 17. Initial conditions for the test of the reaction–diffusion equation; left-hand side and right-hand side panels show the first and the second field, respectively.

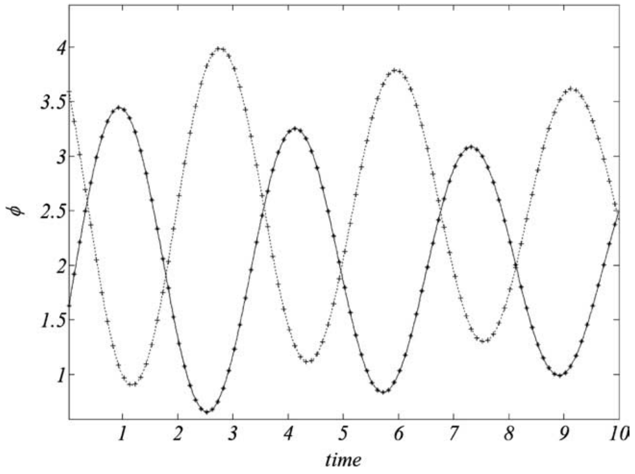

Fig. 18. Numerical versus analytical solution for a selected point; analytical solution for φ^1: solid line, analytical solution for φ^2: dashed line, numerical solution for φ^1: $*$, and numerical solution for φ^2: $+$.

$$\mathbf{F}(\delta\boldsymbol{\phi}) = \left[\boldsymbol{P}_c^{-1}(\mathbf{I}_{N_\mathrm{p}} \otimes \mathbf{J})\boldsymbol{P}_c\right]\delta\boldsymbol{\phi}, \tag{77}$$

where $\mathbf{I}_{N_\mathrm{p}}$ is the $N_p \times N_\mathrm{p}$ unit matrix, and $\mathbf{J} = \{\partial\mathbf{F}/\partial\boldsymbol{\varphi}\}|_{\boldsymbol{\varphi}_\mathrm{s}}$ is the Jacobian defined during the linearization of Eq. (67). The numerical solution is obtained for the time of 10 units with the time step of 0.025. These values are comparable to those used in the advection tests in the previous sections.

The comparison of the analytical solution (76) and the numerical solution for a selected point on the sphere (latitude = 0.22π, longitude = 1.32π) is depicted in Fig. 18. The solution and the solution error at $T = 4.25$ are displayed in Figs. 19 and 20. The analysis of the data from Fig. 20 indicates that the numerical solution is quite accurate with the maximum error not exceeding 0.01%. However, the spatial pattern of the error field depicted in Fig. 20 has some residual structure reflecting the nonhomogeneity of the icosahedral geodesic mesh. Still, the effect is not more pronounced than in a typical solution obtained on a constant resolution rectangular grid.

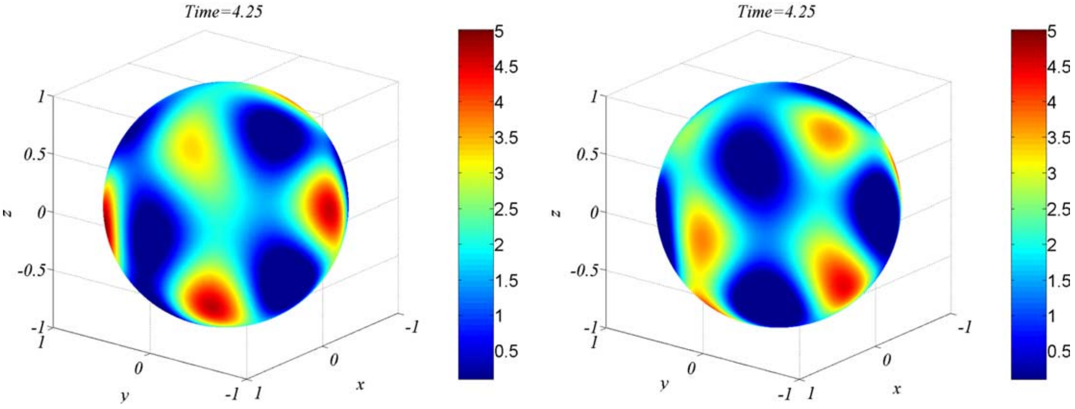

Fig. 19. Numerical solution of the Turing system at $T = 4.25$ (φ^1 left-hand side panel, φ^2 right-hand side panel).

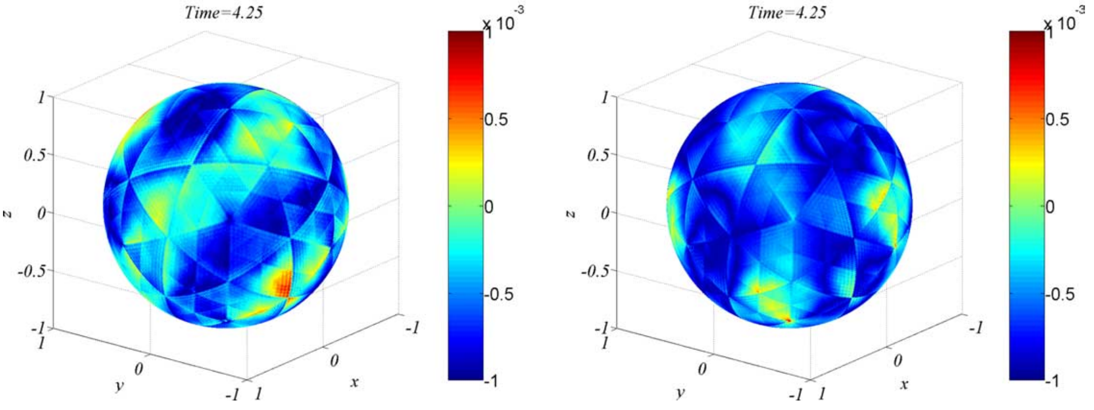

Fig. 20. Numerical solution error for Turing system at $T = 4.25$ (error of φ^1 left-hand side panel, error of φ^2 right-hand side panel).

Table 5
Error norms for the Turing system

Field	l_1	l_2	l_∞
φ^1	0.0002	0.0002	0.0007
φ^2	0.0003	0.0003	0.0009

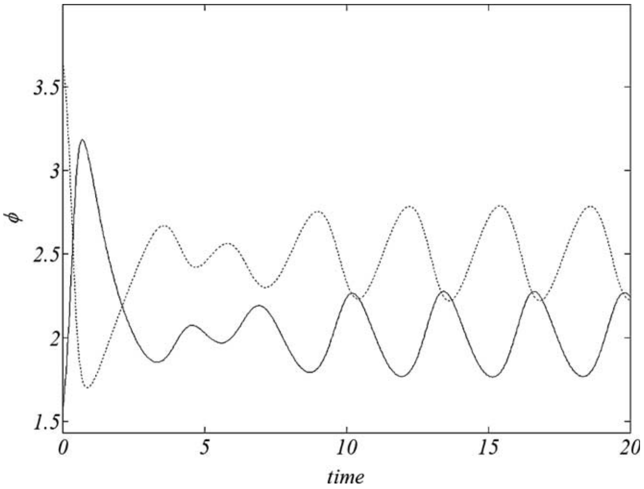

Fig. 21. Numerical solution of the Brusselator system at the selected point on the sphere. The solution was obtained with $A = 2$, $B = 5$, $\mu = 10^{-3}$, $v = 2.0\mu$. The first field is depicted by the solid line, the second field by the dashed line.

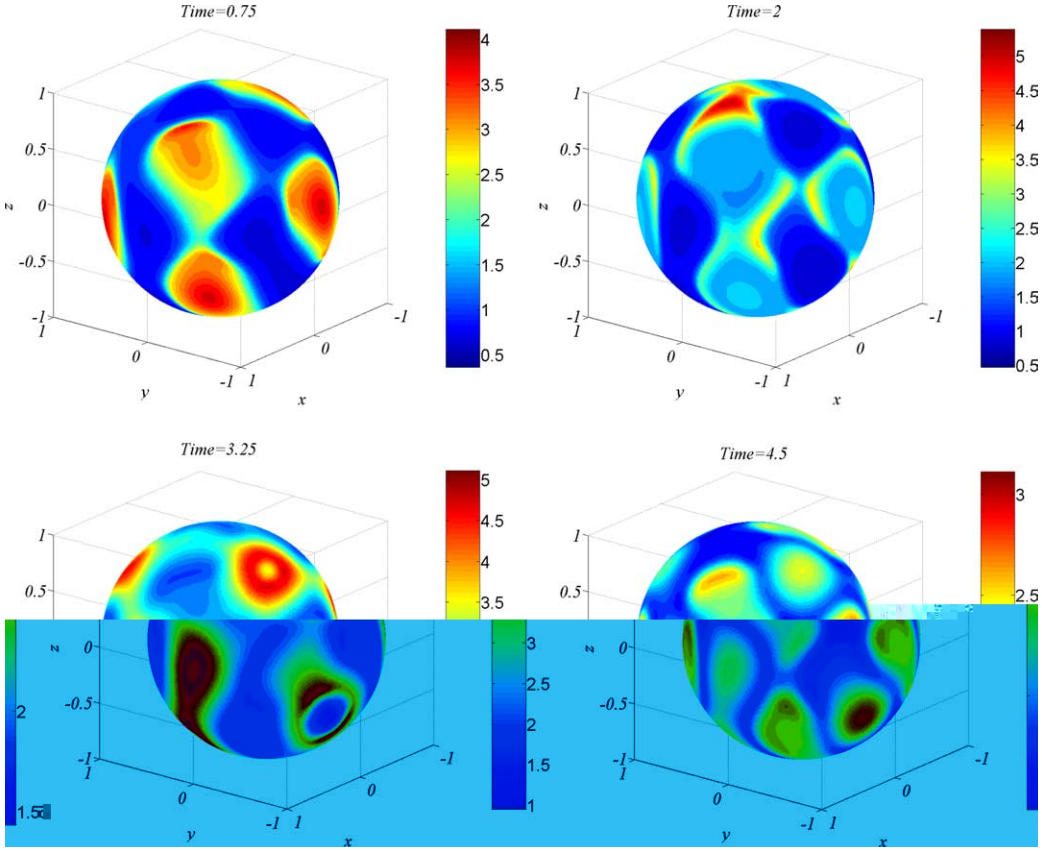

Fig. 22. The illustration of the spatial pattern of the first field of the Brusselator system during the initial aperiodic stage; time dependence of the solution is depicted in Fig. 21.

The possible ways to reduce the effect of nonhomogeneity of the geodesic mesh have been recently discussed in the literature in the context of the numerical solution of the shallow water equations (see [35,37] for the discussion). A quantitative summary of the solution accuracy is presented in Table 5. The overall level of the scheme accuracy indicates that the described algorithm performs well when solving classical reaction–diffusion problem suggested by Turing [38].

The linearized solution is certainly a good indication of the potential applicability of the method. It is natural, however, to question the performance of the scheme for the nonlinear system (67). Despite that the analytical solution for this case is not known, it is still useful to examine the solutions in a general qualitative

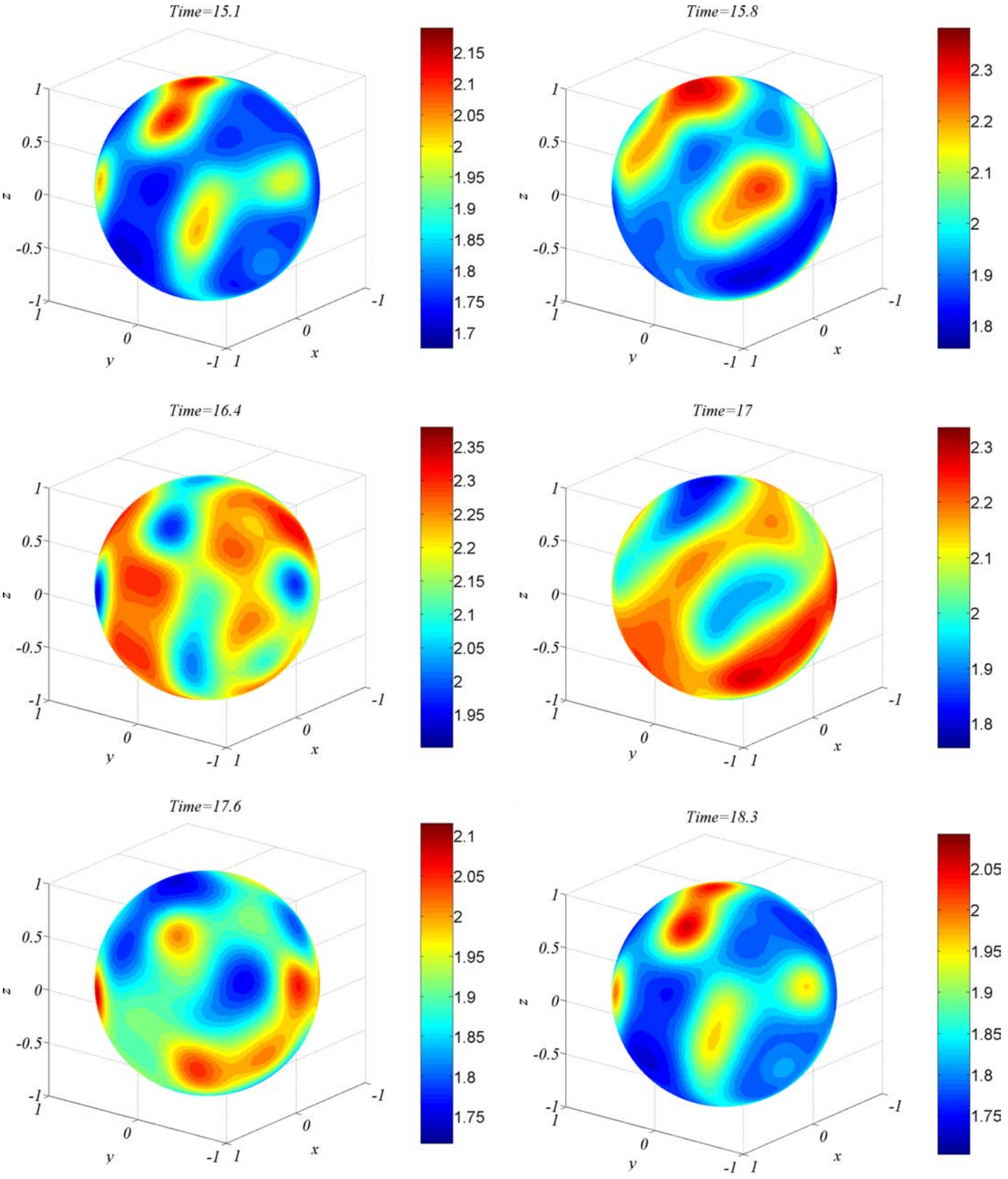

Fig. 23. The illustration of the cycle of spatial patterns of the first field of the Brusselator system in the oscillatory stage; time dependence of the solution is depicted in Fig. 21.

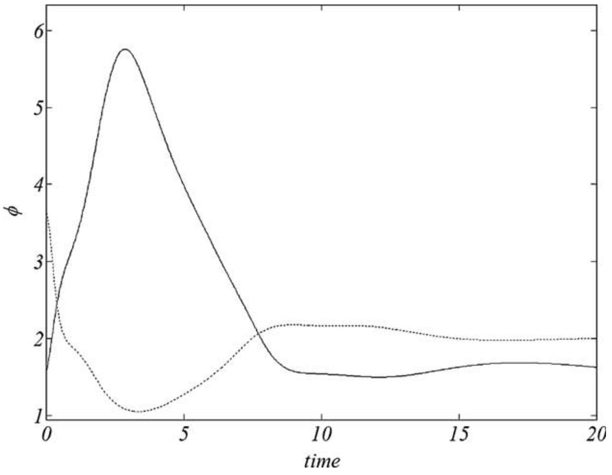

Fig. 24. Numerical solution of the Brusselator system at the selected point on the sphere. The solution was obtained with $A = 2$, $B = 5$, $\mu = 10^{-3}$, $\nu = 8.0\mu$. The first field is depicted by the solid line, the second field by the dashed line.

manner. It is particularly important to verify whether or not the grid nonhomogeneity will contaminate the numerical solution of the nonlinear system.

Initially for the nonlinear reaction–diffusion system we assumed the same values of parameters as that for the linearized system ($A = 2$, $B = 5$, $\mu = 10^{-3}$ and $\nu = 2.0 \times 10^{-3}$). The numerical solution obtained using scheme (63) is depicted in Figs. 21–23. Examination of the time dependence at (latitude = 0.22π, longitude = 1.32π) depicted in Fig. 21 shows that the numerical solution is periodic in time. However, the time dependence in the nonlinear case has slightly different character than that in Fig. 18. The most notable difference is that the solution has initially an aperiodic stage (Figs. 21 and 22) which is absent in the solution of the linearized equations. After the initial time of the adjustment, however, the system starts to exhibit regular oscillations. A sample sequence of spatial patterns of these oscillations is depicted in Fig. 23. The main conclusion from the examination of the reactive wave patterns on the sphere is the lack of the grid effects associated with grid inhomogeneity. This fact could be significant while solving other partial differential equations on the sphere by assuring that the grid structure effects do not distort the solution when nonlinear reaction terms are included. In order to investigate the role of the specific selection of the time integration method, we have repeated the calculations for the nonlinear case with the semi-implicit scheme (64). It is interesting to note that the obtained solution is almost identical to those shown in Figs. 21–23.

The solution of the nonlinear equations can diverge significantly from the solution obtained for the linearized system (69). This fact is particularly evident after solving (67) for the set of parameters, $A = 2$, $B = 5$, $\mu = 10^{-3}$, and $\nu = 8.0 \times 10^{-3}$. According to the results discussed in the literature [17] the Brusselator system with the above listed parameters should converge, independently of the geometry of the domain, to the stationary pattern in the form of stripes. Thus, this type of the solution depends only on internal parameters such as the reaction rates, the values of A and B and the diffusion coefficients. The numerical solution at (latitude = 0.22π, longitude = 1.32π) depicted in Fig. 24 shows that the system indeed converges quickly to the quasi-stationary state.

The examination of the structure of the field shown in Fig. 25 furthermore indicates that this state is spatially inhomogeneous with characteristic network of belt patterns on the surface of the sphere as expected [17]. The behaviour of the system is thus very different than that exhibited by the linearized model (76) that, for the same set of parameters, predicts the usual pattern of slightly damped oscillations. It is interesting that despite the drastic change of the type of solution, the effects of grid irregularity are not present; this fact is particularly important for the solution of other systems of the nonlinear partial differential equations on the sphere.

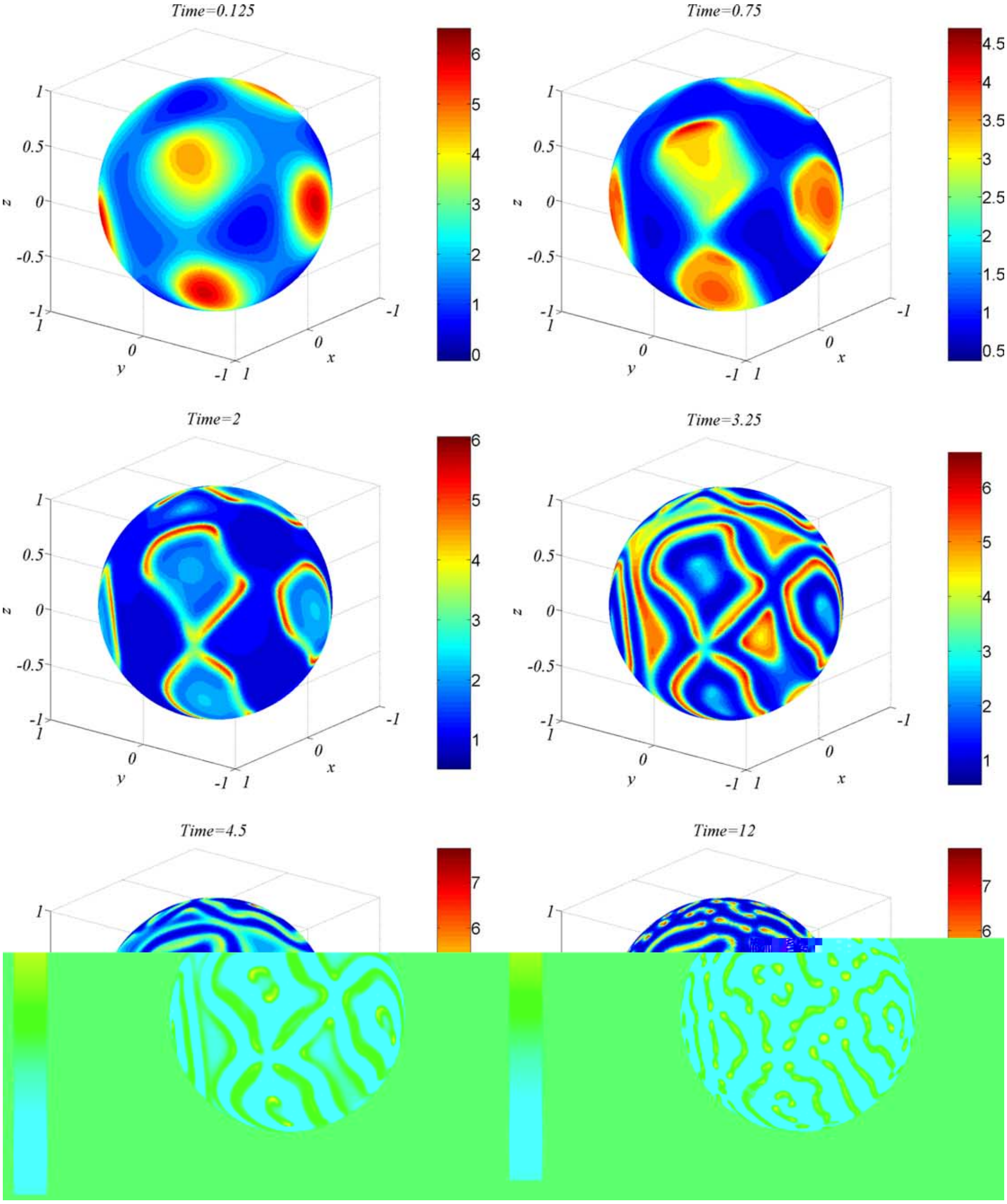

Fig. 25. The illustration of the convergence of the spatial patterns of the first field to the spatially nonhomogeneous stationary state; time dependence of the solution for this case is depicted in Fig. 24.

10. Conclusions

The finite volume solver for the reaction–advection–diffusion equation presented in this paper is straightforward and efficient, yet assures high accuracy. The presented method offers mass conservation, quasi-monotonicity and good accuracy when applied to reactive scalar fields. In the case of linear advection, the algorithm is able to retain the maximum value of the field even for the initial conditions with a discontinuity in the first derivative. The general evaluation of the method presented in Tables 2–4 shows that the scheme is accurate. The explicit local adaptive dissipation applied with the algorithm has a small dissipation and produces results which are, at least, equivalent to those obtained with FCT technique, but in a more efficient manner. In addition to the good accuracy for conservative tracer, the performance of the scheme is confirmed by the analytical solutions of the reaction–diffusion system suggested by Turing [38]. This indicates that the method can be used for other reactive systems in spherical geometry subject to some additional testing. The most important subject to be addressed in the future testing of the method is its performance with stiff chemical systems. The scheme described in this paper can be also easily applied on arbitrary differentiable manifolds. This property is a direct consequence of the fact that the operators defined by (33) and (35) are dependent only on the connectivity matrix and the general parameters defining grid geometry.

References

[1] V.I. Arnold, Ordinary Differential Equations, MIT Press, Cambridge, MA and London, England, 1998.

[2] J.P. Boris, D.L. Book, Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works, J. Comput. Phys. 11 (1972) 397–431.

[3] L. Cesari, Asymptotic Behavior and Stability Problems in Ordinary Differential Equations, Academic Press, New York, 1963.

[4] P. Colella, P.R. Woodward, The Piecewise Parabolic Method (PPM) for gas-dynamics simulation, J. Comput. Phys. 54 (1984) 174–201.

[5] H. Flanders, Differential Forms with Applications to the Physical Sciences, Academic Press, New York, London, 1963.

[6] F.X. Giraldo, Lagrange–Galerkin methods on spherical geodesic grids, J. Comput. Phys. 136 (1997) 197–213.

[7] S.K. Godunov, A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics, Math. Sb. 47 (1959) 271–306.

[8] F. Harary, Graph Theory, Perseus Books, Reading, MA, 1969.

[9] M.E. Hubbard, Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids, J. Comput. Phys. 155 (1999) 54–74.

[10] O. Knoth, R. Wolke, Implicit–explicit Runge–Kutta methods for computing atmospheric reactive flows, Appl. Numer. Math. 28 (1998) 327–341.

[11] H.O. Kreiss, J. Oliger, Comparison of accurate methods for the integration of hyperbolic equations, Tellus 24 (1972) 199–215.

[12] J.D. Lambert, Computational Methods in Ordinary Differential Equations, Wiley, New York, 1973.

[13] R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, Cambridge, 2002.

[14] X. Liu, D.S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.

[15] D. Majewski, D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hannish, G. Paul, W. Wergen, The operational global icosahedral-hexagonal gridpoint model gme: description and high resolution tests, Mon. Weather Rev. 130 (2002) 319–338.

[16] C.R. Molenkamp, Accuracy of finite difference methods applied to the advection equation, J. Appl. Metall. 7 (1968) 160–167.

[17] G. Nicolis, Introduction to Nonlinear Science, Cambridge University Press, Cambridge, 1995.

[18] G. Nicolis, J. Auchmuty, Dissipative structures, catastrophes and pattern formation: a bifurcation analysis, Proc. Natl. Acad. Sci. USA 71 (1977) 2748–2751.

[19] G. Nicolis, I. Prigogine, Self-organization in Nonequilibrium Systems, Wiley, New York, 1977.

[20] E.S. Oran, J.P. Boris, Numerical Simulation of Reactive Flow, Elsevier, New York, Amsterdam, London, 1987.

[21] E.S. Oran, J.P. Boris, Numerical Simulation of Reactive Flow, second ed., Cambridge University Press, Cambridge, 2001.

[22] I. Prigogine, R. Lefever, Symmetry-breaking instabilities in dissipative systems, J. Chem. Phys. 48 (1968) 1695–1700.

[23] I. Prigogine, From Being to Becoming, Freeman, San Francisco, 1980.

[24] J. Pudykiewicz, A predictive tracer model, J. Met. Soc. Jpn. 68 (1990) 213–225.

[25] P.J. Rash, D.L. Williamson, Computational aspects of moisture transport in global models, Quart. J. Roy. Meteorol. Soc. 116 (1990) 1071–1090.

[26] T.D. Ringler, R.H. Heikes, D.A. Randall, Modelling the atmospheric general circulation using a spherical geodesic grid: a new class of dynamical cores, Mon. Weather Rev. 128 (2000) 2471–2490.

[27] H. Ritchie, Application of a semi-Lagrangian integration scheme to the moisture equation in a regional forecast model, Mon. Weather Rev. 113 (1985) 424–435.

[28] R. Sadourny, A. Arakawa, Y. Mintz, Integration of the nondivergent barotropic vorticity equation with an icosahedral hexagonal grid for the sphere, Mon. Weather Rev. 96 (1968) 351–356.

[29] A. Sandu, J.G. Verwer, M. Van Loon, G.R. Carmichael, F.A. Potra, D. Dabdub, J.H. Seinfeld, Benchmarking stiff ODE solvers for atmospheric chemistry problems – I. Implicit vs. explicit, Atmos. Environ. 31 (1997) 3151–3166.

[30] A. Shchepetkin, J.C. McWilliams, Quasi-monotone advection schemes based on explicit locally adaptive dissipation, Mon. Weather Rev. 126 (1998) 1541–1580.

[31] P.K. Smolarkiewicz, A fully multidimensional advection transport algorithm with small implicit diffusion, J. Comput. Phys. 67 (1984) 396.

[32] P.K. Smolarkiewicz, W.W. Grabowski, The multidimensional positive definite advection transport algorithm: nonoscillatory option, J. Comput. Phys. 86 (1990) 355–375.

[33] P.K. Smolarkiewicz, P.J. Rash, Monotone advection on the sphere: an Eulerian versus semi-Lagrangian approach, J. Atmos. Sci. 48 (1991) 793–810.

[34] P.K. Smolarkiewicz, J. Szmelter, MPDATA: an edge based unstructured-grid formulation, J. Comput. Phys. 206 (2005) 624–649.

[35] H. Tomita, M. Satoh, K. Goto, An optimization of the icosahedral grid modified by spring dynamics, J. Comput. Phys. 183 (2002) 307–331.

[36] H. Tomita, M. Satoh, A new dynamical framework of nonhydrostatic global model using the icosahedral grid, Fluid Dyn. Res. 34 (2004) 357–400.

[37] H. Tomita, M. Tsugawa, M. Satoh, K. Goto, Shallow water model on a modified icosahedral geodesic grid by using spring dynamics, J. Comput. Phys. 174 (2001) 579–613.

[38] A.M. Turing, The chemical basis of morphogenesis, Philos. Trans. R. Soc. Lond. B 237 (1952) 37–72.

[39] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer, New York, 2000.

[40] B. van Leer, Towards the ultimate conservative difference scheme, J. Comput. Phys. 135 (1997) 229–248.

[41] D.L. Williamson, Integration of the barotropic vorticity equation on a spherical geodesic grid, Tellus 20 (1968) 642–653.

[42] D.L. Williamson, P.J. Rash, Two-dimensional semi-Lagrangian transport with shape-preserving interpolation, Mon. Weather Rev. 117 (1989) 102–129.

[43] D.L. Williamson, J.B. Drake, J.J. Hack, R. Jakob, P.N. Shwartzrauber, A standard test set for numerical approximations to the shallow water equations in spherical geometry, J. Comput. Phys. 102 (1992) 211.

[44] C.B. Vreugdenhill, Linear central-difference methods, in: C.B. Vreugdenhill, B. Koren (Eds.), Notes on Numerical Fluid Mechanics, Numerical Methods for Advection–Diffusion Problems, vol. 45, Braunschweig, Viesbaden, 1993, pp. 27–54.

[45] N.A. Yanenko, The Method of Fractional Steps, Springer, Berlin, 1971, 160p.

[46] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (1979) 335–362.

[47] X. Zhong, Additive semi-implicit Runge–Kutta methods for computing high-speed nonequilibrium reactive flows, J. Comput. Phys. 128 (1996) 19–31.