Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 209 (2005) 322–339
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Modified-truncation finite difference schemes

Don A. Jones *

Department of Mathematics, Arizona State University, Tempe, AZ 85287-1804, United States

Received 13 October 2004; received in revised form 7 March 2005; accepted 8 March 2005
Available online 23 May 2005

Abstract

A method is presented for modifying the truncation error of a given finite difference scheme approximating a nonlinear evolution equation. The new scheme has several advantages over the original. It is higher order, in the absence of time derivatives, has the same time-step requirements, it removes nonphysical oscillations, and it is not less accurate than the original scheme. The idea applied to a finite difference scheme approximating a geophysical flow produces a scheme consistent with the accuracy of the original scheme, but on a mesh three times more refined.

1. Introduction

The difference between the nodal values of a solution of a nonlinear evolution equation (PDE) and the values computed from a finite difference scheme approximating the PDE are, of course, driven apart by the truncation error – that is, the error produced by replacing continuous derivatives with discrete derivatives. In special cases the true nodal values and the computed values can be shown to be close for all time (as in the autonomous heat equation). In general however, the truncation error leads to an instability which drives the two apart exponentially in time. Reducing the size of the truncation error leads to an increase in the accuracy of finite-time trajectories. Moreover, the long-time behavior can be address from a dynamical systems point of view: a smaller truncation error, in appropriate norms, implies invariant manifolds of the truncated system and those of the PDE are closer. This implies the structure of attractors is better preserved [6,8,9].

In this paper we consider a method of modifying the truncation error of a given finite difference scheme approximating a PDE. The original scheme can be any order. We modify the original scheme, but we insist

* Tel.: +1 480 965 7412.
E-mail address: dajones@math.asu.edu.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.03.012

several constraints apply. First, the new scheme cannot, in any parameter regime, be less accurate than the original; the new scheme must have the same or better time-step restrictions.

The results here are similar to those obtained using compact difference schemes [1–3]. In particular, we derive in many cases a higher-order scheme which uses the same spatial stencil as the lower-order scheme from which the modified scheme is derived. The compact differences are constructed by using the form of the PDE at steady state to express certain higher-order derivatives in the truncation error in terms of lower-order derivatives. If the time derivative is not part of the principal balance, as in a quasi-stationary flow, then the new scheme will be higher order. It will have the accuracy, and possess similar characteristics of the original scheme but on a more refined mesh.

The structure of the paper closely follows [4]. In that paper we demonstrated an algorithmic procedure for modifying any finite difference scheme so that the new scheme has the accuracy of the original scheme but on a twice-fine mesh when time derivatives are not in the main balance of terms. The new scheme resides on the same coarse-grid spatial stencil, has the same time step requirements as the original scheme from which is was derived, is never less accurate than the original scheme, and in certain circumstances removes nonphysical oscillations. The algorithmic approach was first introduced in [7].

The procedure discussed here is equivalent to iterating the algorithmic approach an infinite number of times. However, rather than determining the fixed point of the iterations, which is possible, we modify the truncation error directly and obtain the same result. We will see the limit scheme, which we call the modified scheme, inherits the same properties held by schemes derived from finite iterations of the algorithmic approach, only it is higher order at steady state. If the time derivatives associated with the physical problem are large however, the modified scheme will be less computationally efficient. When the time derivatives are small, the extra cost is more than compensated by the increase in accuracy.

In Section 2 we apply our ideas to the one-dimensional heat equation. We repeat the same process in Section 3 for the two-dimensional heat equation. In Section 4, we apply the procedure to the advection diffusion equation. In this setting we see the mechanism for the removal of nonphysical oscillations. Similar results are found for Burgers equation in Section 5. We also compare the modified scheme with finite iterations of the algorithmic approach in [4] in this section. Section 6 concludes with an application to the shallow water equations.

2. One-dimensional heat equation

To illustrate the basic ideas we consider the one-dimensional heat equation with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} &\frac{\partial u}{\partial t} - \lambda u_{xx} = f(x,t), \quad 0 < x < L,\ t > 0, \\ &u(0,t) = 0 = u(L,t), \quad t > 0. \end{aligned} \tag{2.1}$$

The constant λ is positive. We assume a uniform spatial grid in all cases, and to shorten the exposition, we employ the notation throughout

$$\delta_x^2 u_i = \frac{1}{\Delta x^2}(u_{i+1} - 2u_i + u_{i-1}), \quad \delta_x u_i = \frac{1}{2\Delta x}(u_{i+1} - u_{i-1}).$$

Suppose we are given the second-order finite difference scheme approximating the heat equation

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} - \lambda \delta_x^2 u_i = f_i, \quad 1 \leqslant i \leqslant N-1,$$

where N is given. The numbers u_i approximate the nodal values of the solution to (2.1) at $x_i = iL/N = i\Delta x$. Moreover, we leave the time derivative continuous since the specific time integrator used is not important – we are only modifying the spatial truncation error. Temporal errors are assumed to be small.

Starting with the heat equation, the starting scheme leads to

$$\begin{aligned} 0 &= u_t - \lambda u_{xx} - f \\ &= u_t - \lambda \delta_x^2 u_i - f + (\lambda \delta_x^2 u_i - \lambda u_{xx}) \\ &= u_t - \lambda \delta_x^2 u_i - f + \frac{\lambda}{12} u_{xxxx}(i\Delta x)\Delta x^2 + \mathcal{O}(\Delta x^4). \end{aligned} \tag{2.2}$$

The truncation error is $\Delta x^2 u_{xxxx}/12$.

A more accurate scheme could be constructed by approximating the u_{xxxx} term. Such an approximation however would widen the stencil, adversely affect the time stability of an explicit time scheme, and complicate the implementation of an implicit scheme. However, at steady state $\lambda u_{xxxx} = -f_{xx}$. If we use this in (2.2), we find the scheme

$$\frac{\mathrm{d}a_i}{\mathrm{d}t} - \frac{\lambda}{\Delta x^2}\delta_x^2 a_i = f_i + \frac{1}{12}(f_{i+1} - 2f_i + f_{i-1}) = \frac{1}{12}(f_{i+1} + 10f_i + f_{i-1})$$

is second order for all time and fourth order at steady state. At steady state, it agrees with the compact differencing technique described in [2]. Moreover, since the discrete Laplacian is the same, the stability of an explicit scheme time integration is the same. The proof the modified scheme is never less accurate than the original is lengthy, and rather than presenting it, numerical verification of the constraints will be provided in later sections.

3. Two-dimensional heat equation

The same procedure applies in higher dimensions. Consider the two-dimensional heat equation on a square, Ω, with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} &\frac{\partial u}{\partial t} - \kappa \nabla^2 u = f, \\ &u|_{\partial\Omega} = 0. \end{aligned}$$

Suppose N is a given natural number. On a uniform mesh, we take the starting scheme to be

$$\frac{\mathrm{d}u_{i,j}}{\mathrm{d}t} - \frac{\kappa}{\Delta x^2}(u_{i+1,j} + u_{i-1,j} - 4u_{i,j} + u_{i,j+1} + u_{i,j-1}) = f_{i,j}$$

for $1 \leqslant i, j \leqslant N-1$. At i, j the scheme has truncation error $\frac{1}{12}(u_{xxxx}(i\Delta x, j\Delta x) + u_{yyyy}(i\Delta x, j\Delta x))\Delta x^2$. To replace the higher-order derivative with lower-order terms, we again use the steady-state form of the PDE. Indeed, the steady state equation $-\kappa\nabla^2 u = f$ implies $-\kappa(u_{xxxx} + 2u_{xxyy} + u_{yyyy}) = \nabla^2 f$. In order to use this, the truncation error of the Laplacian must contain cross derivatives. We must modify the original differences.

Remark. Since the difference schemes we derive here are an infinite number of iterates of the algorithmic approach in [4], and the algorithmic approach is a procedure which acts on a given finite difference scheme and creates a new one, we commonly refer to the result of our procedure as a modified scheme-even though the procedure sometimes gives the appearance of constructing a difference scheme rather than modifying a given scheme.

We set

$$\bigoplus u = \frac{1}{\Delta x^2}(u_{i+1,j} + u_{i-1,j} - 4u_{i,j} + u_{i,j+1} + u_{i,j-1}),$$
$$\bigotimes u = \frac{1}{2\Delta x^2}(u_{i+1,j+1} + u_{i+1,j-1} - 4u_{i,j} + u_{i-1,j+1} + u_{i-1,j-1}).$$

Then

$$\bigoplus u = \nabla^2 u + \frac{\Delta x^2}{12}(u_{xxxx} + u_{yyyy}) + \mathcal{O}(\Delta x^4),$$
$$\bigotimes u = \nabla^2 u + \frac{\Delta x^2}{12}(u_{xxxx} + 6u_{xxyy} + u_{yyyy}) + \mathcal{O}(\Delta x^4),$$

and, at steady state,

$$\begin{aligned} 0 &= -\kappa\nabla^2 u - f \\ &= -\kappa\left(\frac{2}{3}\bigoplus u + \frac{1}{3}\bigotimes u\right) + \frac{\kappa\Delta x^2}{12}u_{xxxx} + 2u_{xxyy} + u_{yyyy}) - f + \mathcal{O}(\Delta x^4) \\ &= -\kappa\left(\frac{2}{3}\bigoplus u + \frac{1}{3}\bigotimes u\right) - \frac{\Delta x^2}{12}\nabla^2 f - f + \mathcal{O}(\Delta x^4). \end{aligned}$$

This is the key estimated needed in the usual error analysis revealing that the scheme

$$\frac{\mathrm{d}a_{i,j}}{\mathrm{d}t} - \kappa\left(\frac{2}{3}\bigoplus u + \frac{1}{3}\bigotimes u\right) = f + \frac{1}{12}\nabla^2 f$$

is second order for all time and fourth order at steady state.

4. Linear advection diffusion and nonoscillatory properties

Before presenting numerical studies showing the improved accuracy of the modified scheme, we show its ability to remove unphysical oscillations. We apply the method to the linear equation $u_t - vu_{xx} + Cu_x = f$ with homogeneous Dirichlet boundary conditions. Suppose the starting second-order finite difference scheme is

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} - v\delta_x^2 u_i + C\delta_x u_i = f_i. \tag{4.1}$$

The truncation error is given in

$$\begin{aligned} 0 &= ut - vu_{xx} + Cu_x - f \\ &= ut - v\delta_x^2 u_i + C\delta_x u_i - f + (v\delta_x^2 u_i - C\delta_x u_i - vu_{xx} + Cu_x) \\ &= u_t - v\delta_x^2 u_i + C\delta_x u_i - f + \left(\frac{v}{12}u_{xxxx} - \frac{C}{6}u_{xxx}\right)\Delta x^2 + \mathcal{O}(\Delta x^4). \end{aligned}$$

To make the scheme more accurate, and not adversely affect the time step restrictions, we need to approximate u_{xxxx} and u_{xxx} without widening the stencil. At steady state

$$u_{xxx} = \frac{1}{v}(Cu_{xx} - f_x), \quad vu_{xxxx} = Cu_{xxx} - f_{xx}.$$

Replacing the higher-order derivatives produces a scheme using the same stencil and is fourth order at steady state. In discrete form the modified scheme is

$$\frac{da_i}{dt} - \left(v + \frac{C^2\Delta x^2}{12v}\right)\delta_x^2 a_i + C\delta_x a_i = \frac{1}{12}(f_{i-1} + 10f_i + f_{i+1}) - \frac{\Delta x^2}{12v} C\delta_x f_i. \tag{4.2}$$

4.1. Nonoscillatory properties

For a fixed spatial resolution and viscosity, v, sufficiently low, an explicit Euler time scheme, for example, produces nonphysical oscillations. The modified scheme however will be diffusive in this case. It will be nonoscillatory at all times and for any value of v or initial data. Moreover, following the same analysis in [4] the modified scheme is sign preserving for the choice of a forward Euler time scheme ($u_i^n \geqslant 0$ implies $u_i^{n+1} \geqslant 0$). The relative accuracy of the modified scheme is not relevant in this parameter regime since the original scheme is order one. For v sufficiently large that no nonphysical oscillations occur in the original scheme, the new dissipative term conspires with the other corrections to make the scheme fourth order when the time derivative is not the dominant term.

We illustrate the nonoscillatory properties in Fig. 1. We use $v = 0.00003$, $C = .1$, $\Delta x = 1/60$, and the scheme is integrated to $T = 1.5$, $\Delta t = 0.01$ using an explicit Euler scheme.

Next we consider the advection–diffusion equation on a square, Ω, with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} &\frac{\partial u}{\partial t} + \mathbf{C}\cdot\nabla u = v\nabla^2 u + f, \\ &u|_{\Omega} = 0. \end{aligned} \tag{4.3}$$

The velocity $\mathbf{C}$ is constant. We modify the standard scheme

$$\frac{du_{i,j}}{dt} + (C_1, C_2)\cdot(\delta_x u, \delta_y u) = v(\delta_x^2 + \delta_y^2)u_{i,j} + f_{i,j}. \tag{4.4}$$

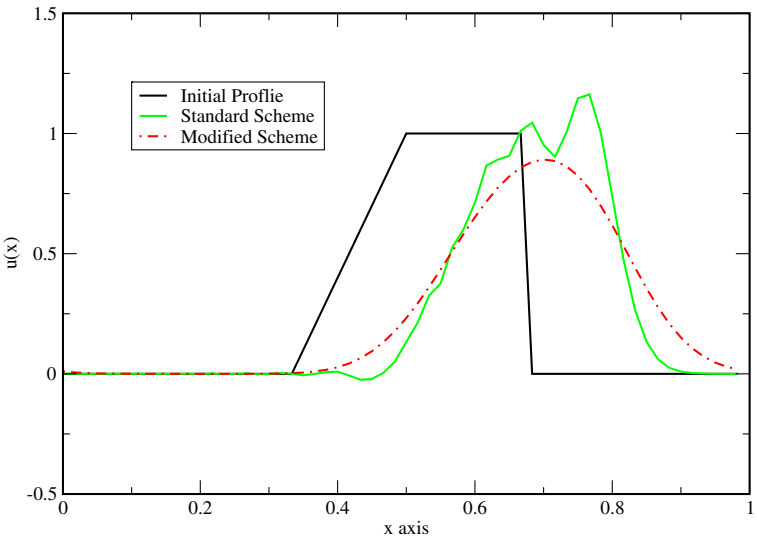

Fig. 1. Standard scheme (4.1) and modified scheme (4.2).

At steady state (4.3) implies

$$\begin{aligned} u_{xxx} + u_{xyy} &= \frac{C_1}{v} u_{xx} + \frac{C_2}{v} u_{xy} - \frac{f_x}{v}, \\ u_{xxy} + u_{yyy} &= \frac{C_1}{v} u_{xy} + \frac{C_2}{v} u_{yy} - \frac{f_y}{v}. \end{aligned} \tag{4.5}$$

Taking the Laplacian of the steady state,

$$v(u_{xxxx} + 2u_{xxyy} + u_{yyyy}) = C_1 u_{xxx} + C_2 u_{xxy} + C_1 u_{xyy} + C_2 u_{yyy} - \nabla^2 f. \tag{4.6}$$

To use (4.5) the third-order derivatives contained in the truncation error of the advective term should have matching coefficients. This requires the use of averages in the approximation of the advective term. The averages required are

$$\begin{aligned} \frac{C_1}{6}(\delta_x u_{i,j+1} + 4\delta_x u_{i,j} + \delta_x u_{i,j-1}) &= C_1 u_x + C_1 \frac{u_{xxx}}{6}\Delta x^2 + C_1 \frac{u_{xyy}}{6}\Delta x^2 + \mathcal{O}(\Delta x^4), \\ \frac{C_2}{6}(\delta_y u_{i+1,j} + 4\delta_y u_{i,j} + \delta_y u_{i-1,j}) &= C_2 u_y + C_2 \frac{u_{yyy}}{6}\Delta x^2 + C_2 \frac{u_{xxy}}{6}\Delta x^2 + \mathcal{O}(\Delta x^4). \end{aligned}$$

As in the analysis of the previous equations, we start by replacing the error produced by the Laplacian with (4.6). We find

$$\begin{aligned} 0 &= u_t - v\nabla^2 u + \mathbf{C}\cdot\nabla u - f \\ &= u_t - v\left(\frac{2}{3}\bigoplus u + \frac{1}{3}\bigotimes u\right) + \frac{C_1}{6}(\delta_x u_{i,j+1} + 4\delta_x u_{i,j} + \delta_x u_{i,j-1}) + \frac{C_2}{6}(\delta_y u_{i+1,j} + 4\delta_y u_{i,j} + \delta_y u_{i-1,j}) \\ &\quad - \frac{v}{12}(C_1 u_{xxx} + C_2 u_{xxy} + C_1 u_{xyy} + C_2 u_{yyy})\Delta x^2 - f - \frac{\Delta x^2}{12}\nabla^2 f + \mathcal{O}(\Delta x^4). \end{aligned}$$

Next we use the formulas for $u_{xxx} + u_{xyy} + u_{yyy} + u_{xxy}$ in (4.5) to express the third-order derivatives in terms of lower-order derivatives, and approximate the second-order corrections so that the new scheme is fourth order when the time derivative is small. The new scheme, with corrections in continuous form, is

$$\begin{aligned} &\frac{da_{i,j}}{dt} - v\left(\frac{2}{3}\bigoplus a + \frac{1}{3}\bigotimes a\right) - \frac{\Delta x^2}{12v}(C_1^2 a_{xx} + 2C_1 C_2 a_{xy} + C_2^2 a_{yy}) + \frac{C_1}{6}(\delta_x a_{i,j+1} + 4\delta_x a_{i,j} + \delta_x a_{i,j-1}) \\ &\quad + \frac{C_2}{6}(\delta_y a_{i+1,j} + 4\delta_y a_{i,j} + \delta_y a_{i-1,j}) = f + \frac{\Delta x^2}{12}\nabla^2 f = \frac{\Delta x^2}{12v}(\mathbf{C}\cdot\nabla f). \end{aligned} \tag{4.7}$$

Fig. 2 illustrates, as in the one-dimensional case, the positive definiteness of the modified scheme. A uniform mesh is used with $\Delta x = \Delta y = 1/80$, and the initial data is a Gaussian-shaped hill centered at the origin; specifically, $u(x, y, 0) = 0.005e^{-500(x^2+y^2)}$. We plot contours of the advected scalar computed from (4.4) and (4.7) on a unit box with periodic boundary conditions. The time integration uses a second-order modified Euler discretization with $\Delta t = 0.001$ and is integrated to $T = 0.5$. The velocity has components (1, 1), and the diffusion constant is $v = 0.0009$. The modified scheme is highly diffusive, but remains positive definite and monotone.

5. Burgers equation

In this section we consider the one-dimensional Burgers equation

$$\begin{aligned} &\frac{\partial u}{\partial t} - \lambda u_{xx} + u u_x = f, \quad 0 < x < L,\ t > 0, \\ &u(0, t) = 0 = u(L, t), \quad t > 0, \end{aligned}$$

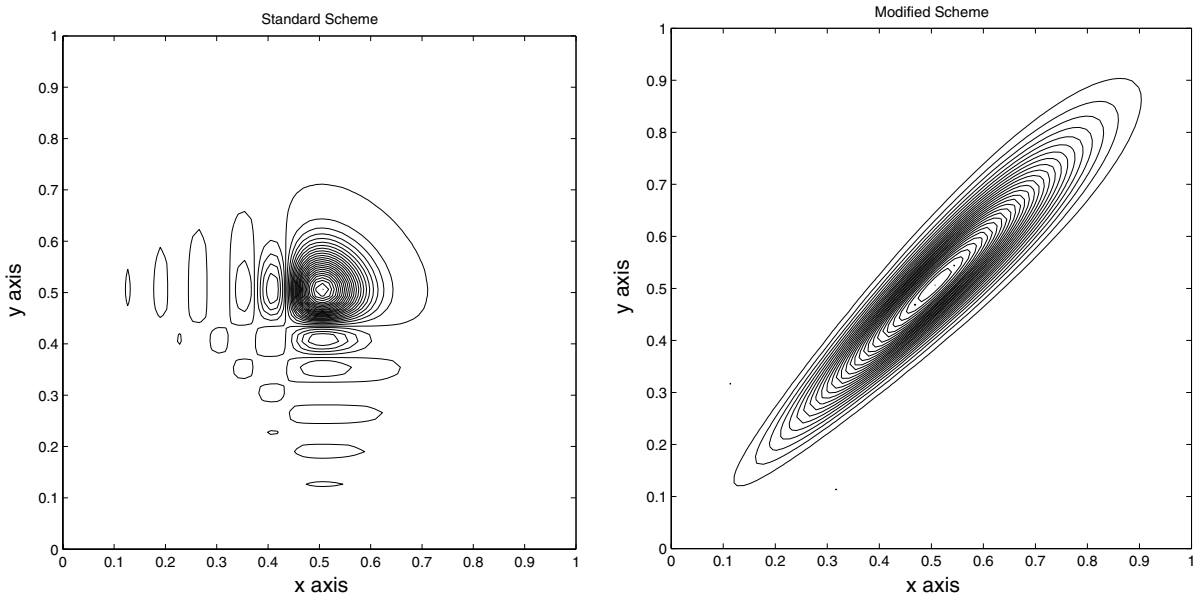

Fig. 2. Left: standard scheme (4.4). Right: modified scheme (4.7).

and we suppose we wish to modify the given second-order scheme

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} - \lambda\delta_x^2 u_i + \frac{u_{i+1}^2 - u_{i-1}^2 + u_i(u_{i+1} - u_{i-1})}{6\Delta x} = f_i. \tag{5.1}$$

We proceed essentially the same way as in previous examples.

Burgers equation at steady state implies

$$u_{xx} = \frac{1}{\lambda}(uu_x - f), \quad u_{xxx} = \frac{1}{\lambda}((u_x)^2 + uu_{xx} - f_x), \quad \lambda u_{xxxx} = 3u_x u_{xx} + uu_{xxx} - f_{xx}. \tag{5.2}$$

We first replace the truncation error produced by the Laplacian using (5.2). We find

$$\begin{aligned} 0 &= u_t - \lambda u_{xx} + uu_x - f \\ &= u_t - \lambda\delta_x^2 u_i + \frac{1}{3}(\delta_x u_i^2 - u_i\delta_x u_i) - f + \left(\lambda\frac{u_{xxxx}}{12} - \frac{uu_{xxx}}{6} - \frac{u_x u_{xx}}{3}\right)\Delta x^2 + \mathcal{O}(\Delta x^4) \\ &= u_t - \lambda\delta_x^2 u_i + \frac{1}{3}(\delta_x u_i^2 - u_i\delta_x u_i) - f - \frac{u_x u_{xx} + uu_{xxx}}{12}\Delta x^2 - \frac{f_{xx}}{12}\Delta x^2 + \mathcal{O}(\Delta x^4). \end{aligned}$$

Next we use the first two equations in (5.2) to find

$$u_t - \lambda\delta_x^2 u_i + \frac{1}{3}(\delta_x u_i^2 - u_i\delta_x u_i) - \frac{\Delta x^2}{12\lambda}\left((u^2 u_x)_x - (uf)_x\right) = f + \frac{f_{xx}}{12}\Delta x^2 + \mathcal{O}(\Delta x^4).$$

In discrete form the new scheme, fourth order at steady state, is

$$\begin{aligned} &\frac{\mathrm{d}a_i}{\mathrm{d}t} - \delta_x^+(\lambda_i^e\delta_x^- a_i) + \frac{a_{i+1}^2 a_{i-1}^2 + a_i(a_{i+1} - a_{i-1})}{6\Delta x} \\ &\quad = f_i + \frac{\delta_x^2 f_i}{12} - \left(\frac{\Delta x}{48\lambda}\right)((a_{i+1} + a_i)(f_{i+1} + f_i) - (a_i + a_{i-1})(f_i + f_{i-1})), \end{aligned} \tag{5.3}$$

where

$$\lambda_i^e := \lambda\left(1 + \frac{\Delta x^2}{12\lambda^2}\left(\frac{a_i + a_{i-1}}{2}\right)^2\right), \quad \delta_x^+ a_i = \frac{1}{\Delta x}(a_{i+1} - a_i), \quad \delta_x^- a_i = \frac{1}{\Delta x}(a_i - a_{i-1}).$$

In Fig. 3 we use the exact solution $u(x, t) = \varepsilon\,(\sin(\pi x)\cos(\omega t) + (1 - x)\sin(\omega t)) + \eta\sin(\pi x)$. This solution determines the forcing in Burgers equation. The left side of Fig. 3 shows the error ratio for the modified scheme (5.3) and the enslaved scheme given in algorithmic approach of [4]. The error ratio is

$$R(t) = \frac{\max_i |u_i - u(i\Delta x, t)|}{\max_i |a_i - u(i\Delta x, t)|},$$

where the u_i solve (5.1), and the a_i solve (5.3) or the scheme in [4], both using an improved Euler time discretization. Note that the ratio never goes below one for either scheme – the modified scheme is never less accurate than the original. The right side of Fig. 3 shows the average ratio as ω in the solution increases. Large ω implies a highly time-dependent solution with large time derivatives. The ratio approaches one for increasing ω as expected. The modified scheme's ratio approaches $1/\Delta x$ for small ω since it is higher order at steady state. Meanwhile, the enslaved scheme approaches four – the accuracy of the twice-fine standard scheme.

5.1. Nonoscillatory properties

As in the advection–diffusion equation, the modified scheme (5.3) is sign preserving for all values of the viscosity, λ, and at all times. The simple verification of this property follows exactly the one given in [4] for the enslaved scheme. In addition, extensive studies of the nonoscillatory properties of the enslaved scheme are given in [4]. Results for the modified scheme (5.3) are similar and are not repeated here.

5.2. Computational efficiency

We compute a critical error ratio in Fig. 3 which reveals whether or not the modified scheme is more efficient than the standard scheme. Since the enslaved scheme in [4] is constructed from the original scheme (5.1) on a twice-fine mesh, the cost per iteration is expected to be twice the cost of the original scheme at any given resolution. The modified scheme (5.3) has exactly the same structure as the

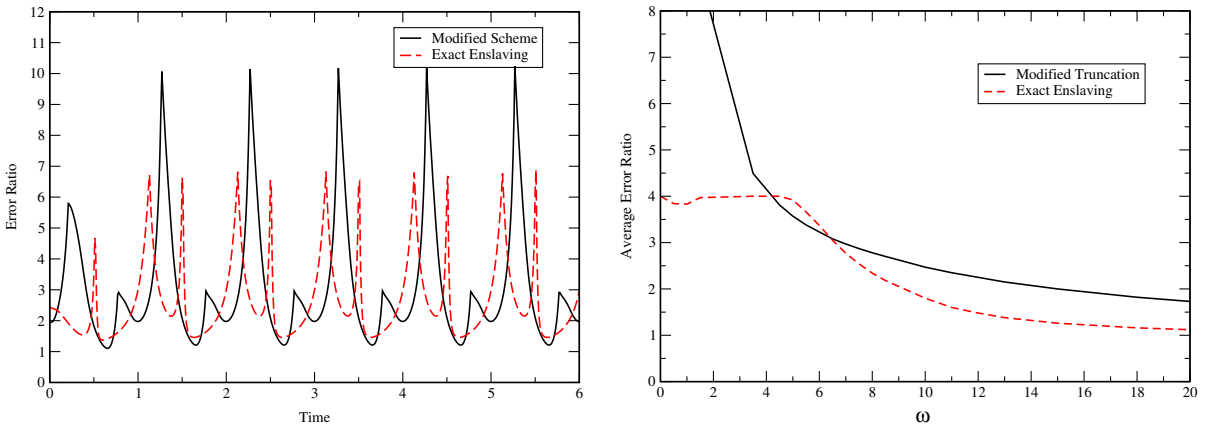

Fig. 3. Left: error ratio versus time with $\epsilon = \eta = 0.5$, $\omega = 2\pi$, $\lambda = 0.5$, $\Delta x = 1/60$. Right: long-time average error ratio as ω varies.

enslaved scheme, and hence the same expected cost. In practice, the modified scheme is roughly 70% the cost. If the cost of the standard scheme is $C_s/\Delta x$ per iterate for some constant C_s, the cost of the modified scheme is $1.7C_s/\Delta x$.

To determine the relative efficiency of the modified scheme, assumptions about the behavior of Δt with changing mesh size must be made. Suppose the time step is diffusionally limited. That is, $\Delta t \leqslant C_1\Delta x^2$ for the scheme to remain stable. Given a time T and N such that $T = N\Delta t$ and a grid resolution Δx_0, the cost to integrate the modified scheme to time T is $NC_m/\Delta x_0 = TC_m/(\Delta t\Delta x_0) = 1.7TC_s/(C_1\Delta x_0^3)$.

Now the resolution of the standard scheme (5.1) may be increase until it matches the cost of the modified scheme. The cost of the standard scheme iterated to time T is $TC_s/(C_1\Delta x_0^3)$. Setting the two equal implies the standard scheme can be computed on a grid with spacing $\Delta x = (1.7)^{-1/3}\Delta x_0$. Since the scheme is second order, the error ratio of the standard scheme computed on Δx_0 to the error computed on $(1.7)^{-2/3}\Delta x_0$ is $(1.7)^{2/3} \approx 1.42$.

This leads to the following conclusion. If the error ratio in Fig. 3 falls below 1.42, the same accuracy can be achieved cheaper by running the standard scheme at a higher resolution. If it is above 1.42 the modified scheme is more efficient. Whether or not the modified scheme is more efficient depends on the size of the time derivatives in the problem. If the time step is limited by the advection term, $\Delta t \leqslant C\Delta x$, the critical ratio is 2.

6. 2D shallow-water equations

In this section we apply the method to an equation in which the diffusion is not part of the balance of terms in the PDE. In advective form the basin scale, double gyre, wind-driven, reduced-gravity shallow-water equations (SWE) are

$$\begin{aligned} &u_t - D + uu_x + vu_y = -g'h_x + fv + F_u,\\ &v_t - D + uv_x + vv_y = -g'h_y - fu + F_v,\\ &h_t + (uh)_x + (vh)_y = 0. \end{aligned}$$

We consider the equations on a rectangular basin on a beta plane; the Coriolis force is by $f = f_0(1 + \beta y)$ with y in the north–south direction. The operator D is a diffusion operator (Laplacian, BiLaplacian, ...), with appropriate boundary conditions on u, v – the fluid velocity in the east–west, north–south directions respectively. h is the fluid depth, g' is the reduced gravity, and F_u, F_v are external forcing functions. Here, we take $F_u = -\tau\cos(\pi u/L_y)$ and $F_v = 0$.

We consider a leapfrog time differencing combined with a B-grid centered in space algorithm. Specifically, the standard scheme is taken to be

$$\begin{aligned} \frac{u_{i,j}^{n+1} - u_{i,j}^{n-1}}{2\Delta t} &= -g'\delta_x\bar{h}_{i,j}^{n} - (u_{i,j}^{n}\delta_x u_{i,j}^{n} + v_{i,j}^{n}\delta_y u_{i,j}^{n}) + f_{i,j}v_{i,j}^{n} + \nu(\delta_x^2 + \delta_y^2)u_{i,j}^{n-1} + F_{i,j}^{u},\\ \frac{v_{i,j}^{n+1} - v_{i,j}^{n-1}}{2\Delta t} &= -g'\delta_y\bar{h}_{i,j}^{n} - (u_{i,j}^{n}\delta_x v_{i,j}^{n} + v_{i,j}^{n}\delta_y v_{i,j}^{n}) + f_{i,j}u_{i,j}^{n} + \nu(\delta_x^2 + \delta_y^2)v_{i,j}^{n-1} + F_{i,j}^{v},\\ \frac{h_{i,j}^{n+1} - h_{i,j}^{n-1}}{2\Delta t} &= -\bar{\delta}_x\frac{(u_{i,j}^{n} + u_{i,j-1}^{n})(h_{i,j}^{n} + h_{i+1,j}^{n})}{4\Delta x} - \bar{\delta}_y\frac{(v_{i,j}^{n} + v_{i-1,j}^{n})(h_{i,j}^{n} + h_{i,j+1}^{n})}{4\Delta y}, \end{aligned} \tag{6.1}$$

where $\bar{\delta}_x g_{i,j} = g_{i,j} - g_{i-1,j}$, $\bar{\delta}_y g_{i,j} = g_{i,j} - g_{i,j-1}$, and the gradient of layer thickness in the momentum equations is

$$\delta_x\bar{h}_{i,j} = (h_{i+1,j+1} + h_{i+1,j} - h_{i,j+1} - h_{i,j})/2.$$

To damp the computational mode we use a temporal filter that mixes a small amount of forward time integration at each time step. Specifically, we set

$$u^{n+1} = (1-\chi)u^{n-1} + \chi u^n + (2-\chi)R^n,$$

where R is the right-hand side of the momentum equations, and χ is a small parameter, typically near 1%.

The results of a high-resolution run of the equations, using the Earth-like parameters in Table 1, are shown in Fig. 4. The initial data is $u = v = 0$, $h = H_0$. The figure plots the maximum norm of each term in the SWE versus time. Here, we take $D = \nabla^2 u$ with homogeneous Dirichlet boundary conditions on u and v. As can be seen, the main balance is the geostrophic balance between the Coriolis term and the pressure gradient.

The same procedure applies only, based on Fig. 4, we ignore the dissipative operator D. The nonlinear and pressure gradient terms in the east–west momentum equation in (6.1) produce the truncation error $-uu_{xxx}/6 - vu_{yyy}/6 - g'h_{xxx}/24 - g'h_{xyy}/8$ with a similar error for the v equation. To find formulas for the third-order derivatives, the steady state is again used. At steady state

$$\nabla^2(uu_x + vu_y) = \nabla^2(fv - g'h_x + F_u)$$

Table 1
Physical parameters for the simulations

Coriolis parameter	$f_0 = 5.0 \times 10^{-5}\ \mathrm{s}^{-1}$
$f = f_0(1 + \beta y)$	$\beta = 0.5$
Wind stress	$\tau = 1 \times 10^{-7}\ \mathrm{N\ m}^{-2}$
Viscosity parameter	$\nu = 900\ \mathrm{m}^2\ \mathrm{s}^{-1}$
Effective gravity	$g' = 0.03\ \mathrm{m\ s}^{-2}$
Initial upper layer depth	$H_0 = 500$ m
Domain	
East–west	$L_x = 3000$ km
North–south	$L_y = 2000$ km

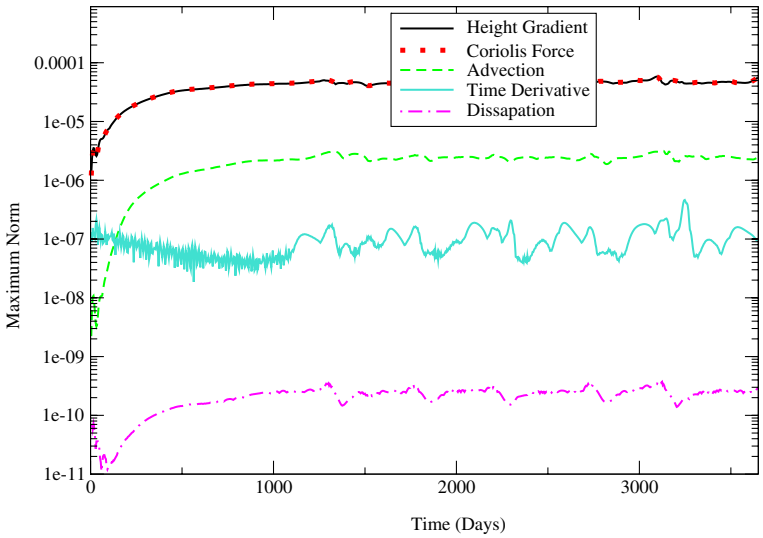

Fig. 4. Maximum of each term in SWE. The legend is in the same order from top to bottom as the graphs in the figure with the first two, the height gradient and Coriolis force, essentially overlapping.

or

$$\frac{1}{6}(uu_{xxx} + uu_{xyy} + vu_{xxy} + vu_{yyy} + g'h_{xxx} + g'h_{xyy}) = -\frac{1}{3}(\nabla u \cdot \nabla u_x + \nabla v \cdot \nabla u_y) - \frac{1}{6}(u_x\nabla^2 u + u_y\nabla^2 u - \nabla^2(fv + F_u)). \tag{6.2}$$

The discrete form of the nonlinear and pressure gradient terms needs to be modified so that the truncation error matches the left side of (6.2)

We use the following approximations: for uu_x,

$$\langle uu_x\rangle := \frac{1}{6}(u_{i,j+1}\delta_x u_{i,j+1} + 4u_{i,j}\delta_x u_{i,j} + u_{i,j-1}\delta_x u_{i,j-1}) = uu_x + \frac{\Delta x^2}{6}(uu_{xxx} + uu_{xyy}) + \Delta x^2\left(\frac{u_y u_{xy}}{3} + \frac{u_x u_{yy}}{6}\right) + \mathcal{O}(\Delta x^4),$$

for vu_y,

$$\langle vu_y\rangle := \frac{1}{6}(v_{i+1,j}\delta_y u_{i+1,j} + 4v_{i,j}\delta_y u_{i,j} + v_{i-1,j}\delta_y u_{i-1,j}) = vu_y + \frac{\Delta x^2}{6}(vu_{yyy} + vu_{xxy}) + \Delta x^2\left(\frac{v_x u_{xy}}{3} + \frac{u_y v_{xx}}{6}\right) + \mathcal{O}(\Delta x^4),$$

and for h_x,

$$\langle h_x\rangle := \frac{1}{48\Delta x}(h_{i+1,j+2} - h_{i,j+2}) + (h_{i+1,j-1} - h_{i,j-1}) + \frac{7}{24\Delta x}(h_{i+1,j} - h_{i,j}) + (h_{i+1,j+1} - h_{i,j+1}) + \frac{1}{16\Delta x}(h_{i+2,j+1} - h_{i-1,j+1}) + (h_{i+2,j} - h_{i-1,j}) = h_x + \frac{\Delta x^2}{6}(h_{xxx} + h_{xyy}) + \mathcal{O}(\Delta x^4).$$

With these approximations the truncation error of the nonlinear term and the height field includes the terms on the left side of (6.2). The higher-order derivatives in the truncation error are replaced with the lower-order derivatives from the right side of (6.2) and combine with the other lower-order terms. In this way the second-order correction is approximated, and the following scheme is second order for all time and fourth order at steady state

$$u_t + \langle uu_x\rangle + \langle vu_y\rangle = -g'\langle h_x\rangle + \frac{f}{6}(v_{i+1,j} + v_{i,j+1} + v_{i-1,j} + v_{i+1,j} + 2v_{i,j}) + F_u + \frac{\Delta x^2}{6}\nabla^2 F_u - \frac{\Delta x^2}{6}(3u_x u_{xx} + u_y v_{yy} + 2v_y u_{yy}). \tag{6.3}$$

The approximation for the v equation is found by switching u and v, x and y, f with $-f$, and F_u with F_v.

The height equation works in a similar way. If we call the negative of the right side of (6.1) $\bar{\delta}\cdot(\mathbf{u}h)$, then

$$\begin{aligned} 0 &= h_t + \nabla\cdot(\mathbf{u}h) = \frac{\mathrm{d}h_{i,j}}{\mathrm{d}t} + \bar{\delta}\cdot(\mathbf{u}h) + \left(\nabla\cdot(\mathbf{u}h) - \bar{\delta}\cdot(\mathbf{u}h)\right) \\ &= \{\text{the original scheme}\} \\ &\quad - \frac{\Delta x^2}{24}(4(uh_{xx})_x + (u_{xx}h)_x + 2(u_x h_x)_x + 3(u_{yy}h)_x) \\ &\quad - \frac{\Delta x^2}{24}(4(vh_{yy})_y + (v_{yy}h)_y + 2(v_y h_y)_y + 3(v_{xx}h)_y). \end{aligned}$$

Because the truncation error is written in flux form, all of the terms are easy to approximate near a boundary except the terms $(u_{xx}h)_x$ and $(v_{yy}h)_y$. They will be replaced using the steady state equation. At steady state $\nabla\cdot(\mathbf{u}h)=0$ or $\nabla^2\nabla\cdot(\mathbf{u}h)=0$. This implies

$$-\frac{(u_{xx}h)_x+(v_{yy}h_y)_y}{24}=\frac{1}{24}(2(u_xh_x)_x+2(u_yh_y)_x+(uh_{xx})_x+(u_{yy}h)_x+(uh_{yy})_x)+\frac{1}{24}(2(v_yh_y)_y+2(v_xh_x)_y+(vh_{yy})_y+(v_{xx}h)_y+(vh_{xx})_y).$$

A scheme second order, fourth order at steady state, with corrections in continuous form, is

$$\{\text{the original scheme}\}=\frac{\Delta x^2}{24}(3(uh_{xx})_x+2(u_{yy}h)_x+3(vh_{yy})_y+2(v_{xx}h)_y-2(u_yh_y)_x-2(v_xh_x)_y-(uh_{yy})_x-(vh_{xx})_y). \tag{6.4}$$

6.1. Computational efficiency

As argued above for Burgers equation, the cost per iteration of the modified scheme (6.3), (6.4) is approximately equal to the cost of the standard scheme (6.1) but on a twice-fine mesh. The extra cost is observed in this case. The computational savings arise from the time step properties of the modified scheme. At a given spatial resolution it has the same time step restrictions as the standard scheme, but produces dynamics consistent with original scheme computed on a more resolved spatial mesh. The CFL condition implies $\Delta t\leqslant\Delta x/V$. If the cost of the standard scheme run to time T is C at some resolution, the cost of the standard scheme on a twice-fine mesh is $8C$, on a three-times fine mesh is $27C\ldots$, while the cost of the modified scheme is $4C$. However, since the modified scheme's dynamics is neither that of the twice or three-times fine standard scheme, a precise estimate of the CPU savings is difficult.

6.2. Numerical studies

We use (6.3) and (6.4) and the values in Table 1. We take $D=\nabla^2u$ with homogeneous Dirichlet boundary conditions on u and v. The brief statistical analysis presented is similar to the one given in [5].

We set the domain averaged kinetic energy to

$$E(t)=\int_\Omega\left[\frac{1}{2}h(x,y,t)(u^2(x,y,t)+v^2(x,y,t))\right]\mathrm{d}\Omega.$$

In Fig. 5 we plot $E(t)$ versus time and the associated histograms for various resolutions. As can be seen in the histograms, the low $\Delta x=40$ km resolution standard scheme has a much higher and broader energy distribution compared to the standard scheme with resolution $\Delta x=10$ km. A doubling of the spatial mesh, $\Delta x=20$ km, still produces a scheme with incorrect kinetic energy distribution. Alternatively, the modified scheme has a histogram in close agreement with the standard scheme computed on spatial meshes three, $\Delta x=13.3$ km, and four times refined, 10 km.

In the final two figures we examine the mean and the first three eigenvectors of the upper layer depth field correlation matrix (the first three Empirical Orthogonal Functions) computed over a 60-year time span beginning at year 10. The left side of Fig. 6 shows the mean of the standard scheme computed on a grid with $\Delta x=40$ km. The modified scheme with resolution $\Delta x=40$ km, the standard scheme with resolution $\Delta x=20$ km, and at the bottom the standard scheme with $\Delta x=13.3$ km. Notice the modified scheme matches the 13.3 km run – the standard scheme with a three-times refined mesh.

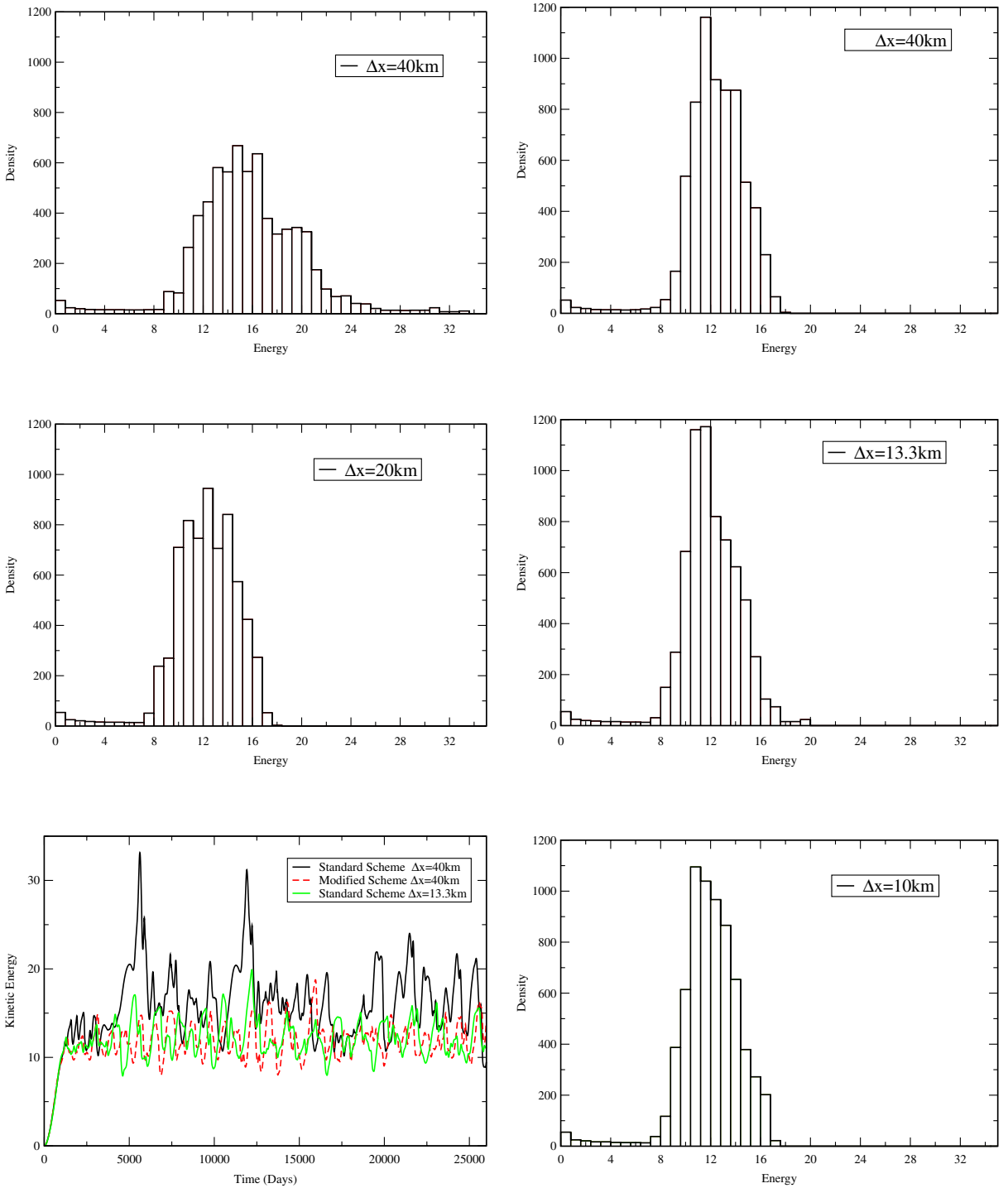

Fig. 5. Upper left: low-resolution standard scheme. Upper right: low-resolution modified scheme. Middle left, Middle right and Bottom right: high-resolution standard scheme.

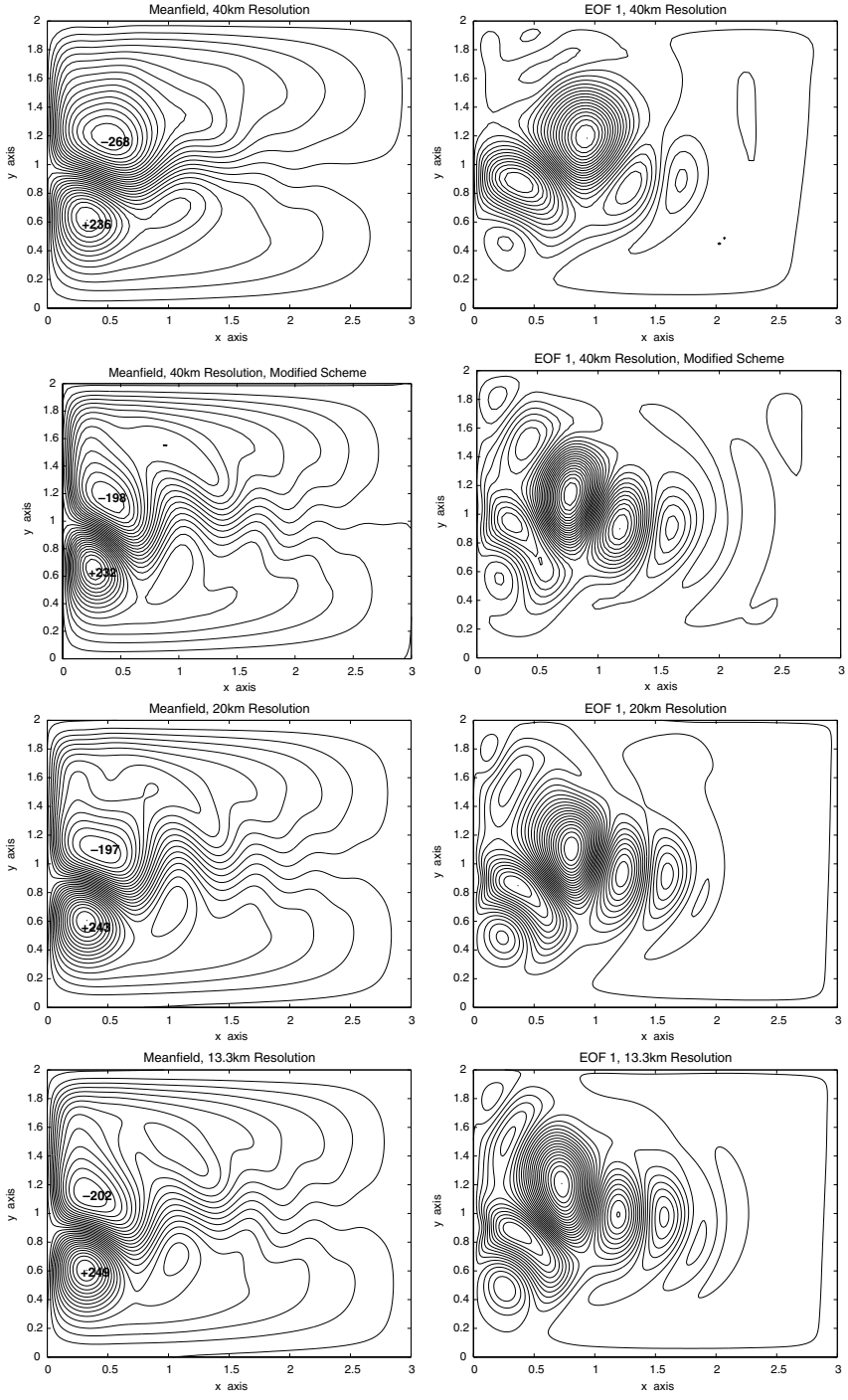

Fig. 6. Left side: mean field. Right side EOF 1. Top: low-resolution standard scheme. Second down: low-resolution modified scheme. Next two: high-resolution standard scheme.

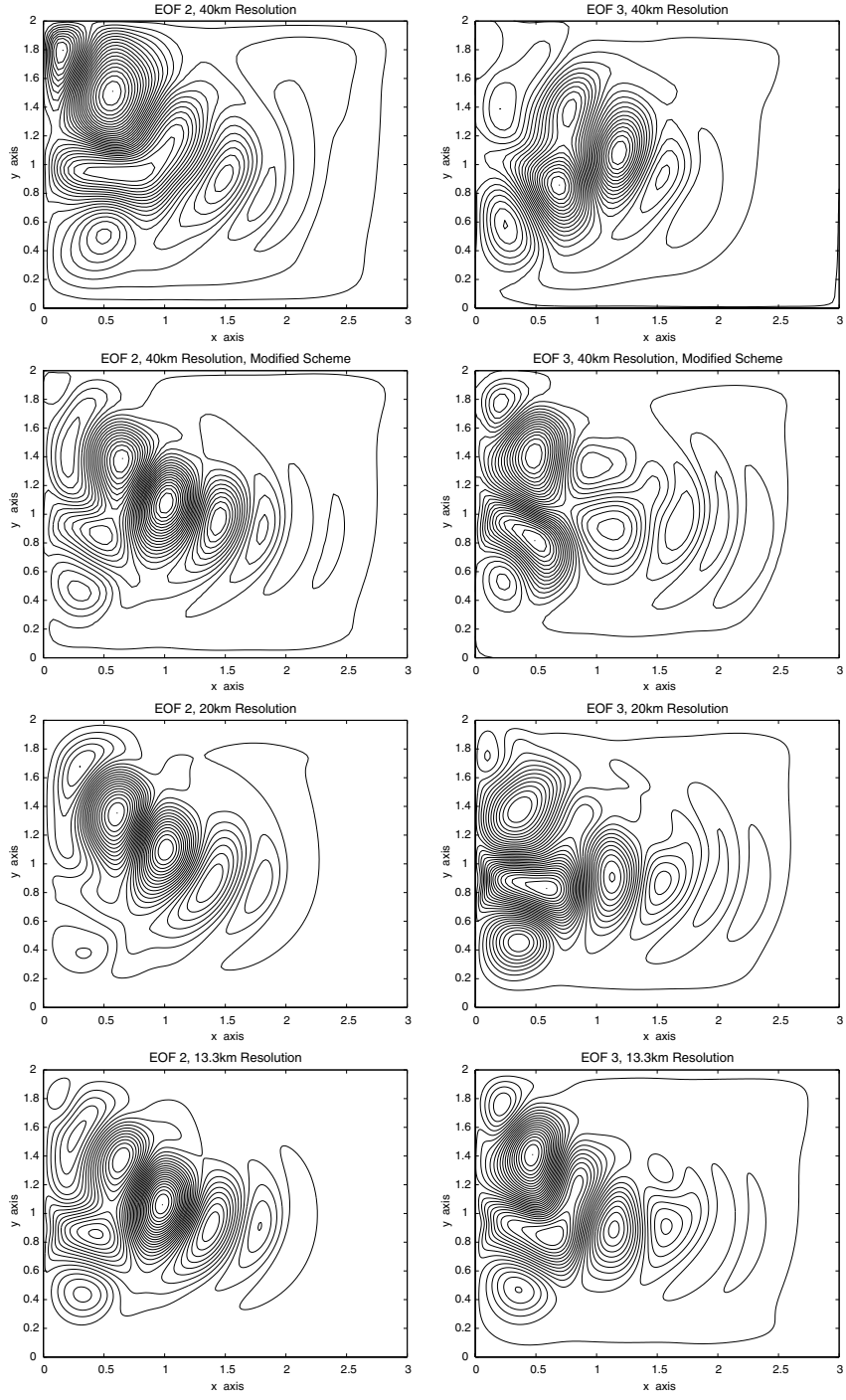

Fig. 7. Left side: EOF 2. Right side EOF 3. Top: low-resolution standard scheme. Second down: low-resolution modified scheme. Next two: high-resolution standard scheme.

The right side of Fig. 6 shows the first eigenvector. Fig. 7 shows the second and third eigenvectors with the resolutions in the same order as in the previous figure. Again the modified scheme using at the 40 km resolution most closely matched the standard scheme in the three-times refined mesh.

6.3. Conclusions

We have effectively iterated an infinite number of times an algorithm which both improves the accuracy and stability properties of any given finite difference scheme. The fixed point of the iteration is a scheme which is higher order in the absence of time derivatives. This procedure reduces the overall truncation error of the given finite-difference scheme by taking advantage of balances present in the governing equations.

We have shown, numerically, that the modified scheme is never less accurate than the original scheme. Moreover, the stability of the modified scheme is the same as the original scheme, in part, because the procedure does not increase the size of the spatial stencil on which it is computed. The larger time step more than compensates for the added computational expense when time derivatives are not part of the balance of terms in the governing equations.

The procedure applied to a chaotic geophysical flow indicates the modified scheme, run on coarse grids, captures the dynamics and statistical features of more resolved flows.

Acknowledgments

The author wishes to express a sincere gratitude to Andrew Poje and Len Margolin for many useful discussions. The author gratefully acknowledges the support of the Institute for Geophysics and Planetary Physics (IGPP) at Los Alamos National Laboratory.

Appendix A. 2D Burgers equation

To illustrate the procedure on an equation functionally similar to the shallow water equations but one in which the diffusion is a significant term, we consider the two component, two-dimensional Burgers equation. We ignore questions of global existence of solutions etc. Specifically, the procedure is applied to

$$\frac{\partial \mathbf{u}}{\partial t} - \lambda \nabla^2 \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \mathbf{F} + f\mathbf{k} \times \mathbf{u}.$$

As before, we consider the equation with homogeneous Dirichlet boundary conditions on a square. The general algorithm for producing the modified scheme is as follows:

- Express the highest-order derivatives in terms of lower-order derivatives using the steady-state PDE. This will create many new lower-order derivatives.
- Re-difference, typically using averages, the lower derivatives in the original PDE in such a way that the formulas found in step one may be used.
- Replace higher-order derivatives using the steady-state PDE.

We work with the u component. At steady state

$$\nabla^2 u = \frac{1}{\lambda}(uu_x + vu_y - fv - F^u),$$

$$u_{xxx} + u_{xyy} = \frac{1}{\lambda}(u_x^2 + uu_{xx} + v_x u_y + vu_{xy} - fv_x - F_x^u), \tag{A.1}$$

$$u_{yyy} + u_{xxy} = \frac{1}{\lambda}(u_y u_x + uu_{xy} + v_y u_y + vu_{yy} - fv_y - F_y^u),$$

$$\lambda(u_{xxxx} + 2u_{xxyy} + u_{yyyy}) = \nabla^2(uu_x + vu_y) - f\nabla^2 v - \nabla^2 F^u. \tag{A.2}$$

The truncation error of the Laplacian will produce a term proportional to (A.2). Its replacement, using (A.2) will in turn produce 16 new terms in $\nabla^2(uu_x + vu_y)$. They need to combine with the truncation terms produced by the re-differencing of the nonlinear term. Several constraints apply. We suppose the new differencing should reduce to the one-dimensional Burgers equation, (5.1), when the solution does not depend on y or v. In particular, it should have the Arakawa differencing of the nonlinear term. Second, like in the linear 2D advection diffusion case, certain third-order derivatives need to balance so that (A.1) may be used.

We consider three approximations of the nonlinear term. They are constructed as follows. The computational grid is the same as we used for the two-dimensional heat equation.

$$\langle u\rangle_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}(u_{i,j} + u_{i+1,j} + u_{i+1,j+1} + u_{i,j+1})$$

and

$$\langle u_x\rangle_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2\Delta x}(u_{i+1,j+1} + u_{i+1,j} - u_{i,j+1} - u_{i,j}).$$

Like the one-dimensional burgers equations, we difference uu_x two ways,

$$\langle uu_x\rangle_1 = \frac{u_{i,j}}{2\Delta x}(u_{i+1,j} - u_{i-1,j}),$$

$$\begin{aligned}\langle uu_x\rangle_2 &= \frac{1}{4}\left(\langle u\rangle_{i+\frac{1}{2},j+\frac{1}{2}}\langle u_x\rangle_{i+\frac{1}{2},j+\frac{1}{2}} + \langle u\rangle_{i-\frac{1}{2},j+\frac{1}{2},j}\langle u_x\rangle_{i-\frac{1}{2},j+\frac{1}{2},j} + \langle u\rangle_{i+\frac{1}{2},j-\frac{1}{2},j}\langle u_x\rangle_{i+\frac{1}{2},j-\frac{1}{2},j} + \langle u\rangle_{i-\frac{1}{2},j-\frac{1}{2},j}\langle u_x\rangle_{i-\frac{1}{2},j-\frac{1}{2},j}\right)\\ &= uu_x + \left(\frac{uu_{xxx}}{6} + \frac{u_x u_{xx}}{2} + \frac{uu_{xyy}}{4} + \frac{u_y u_{xy}}{4} + \frac{u_x u_{yy}}{4}\right)\Delta x^2 + \mathcal{O}(\Delta x^4).\end{aligned}$$

Similarly, for the approximation of the vu_y term, we set $\langle v\rangle_{i+\frac{1}{2},j+\frac{1}{2}}$ in the same way the $\langle u\rangle$ counterpart was constructed. In addition,

$$\langle u_y\rangle_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2\Delta x}(u_{i+1,j+1} + u_{i,j+1} - u_{i+1,j} - u_{i,j}).$$

As before, we consider two approximations. They are

$$\langle vu_y\rangle_1 = \frac{v_{i,j}}{2\Delta x}(u_{i,j+1} - u_{i,j-1}),$$

$$\begin{aligned}\langle vu_y\rangle_2 &= \frac{1}{4}\left(\langle v\rangle_{i+\frac{1}{2},j+\frac{1}{2}}\langle u_y\rangle_{i+\frac{1}{2},j+\frac{1}{2}} + \langle v\rangle_{i-\frac{1}{2},j+\frac{1}{2},j}\langle u_y\rangle_{i-\frac{1}{2},j+\frac{1}{2},j} + \langle v\rangle_{i+\frac{1}{2},j-\frac{1}{2},j}\langle u_y\rangle_{i+\frac{1}{2},j-\frac{1}{2},j} + \langle v\rangle_{i-\frac{1}{2},j-\frac{1}{2},j}\langle u_y\rangle_{i-\frac{1}{2},j-\frac{1}{2},j}\right)\\ &= vu_y + \left(\frac{vu_{yyy}}{6} + \frac{v_x u_{xy}}{4} + \frac{vu_{xxy}}{4} + \frac{v_{xx}u_y}{4} + \frac{v_y u_{yy}}{4} + \frac{v_{yy}u_y}{4}\right)\Delta x^2 + \mathcal{O}(\Delta x^4).\end{aligned}$$

Finally, using (A.2) and the truncation error for the nonlinear terms,

$$\begin{aligned}0 &= u_t - \lambda\nabla^2 u + (uu_x + vu_y) - fv - F^u \\ &= u_t - \lambda\left(\frac{2}{3}\bigoplus u + \frac{1}{3}\bigoplus u\right) + \frac{1}{3}(\langle uu_x\rangle_1 + \langle vu_y\rangle_1) + \frac{2}{3}(\langle uu_x\rangle_2 + \langle vu_y\rangle_2) - fv - F^u \\ &\quad - \frac{1}{12}\left(uu_{xxx} + uu_{xyy} + vu_{yyy} + vu_{xxy} + u_x\nabla^2 u + u_y\nabla^2 v + \nabla^2(vf) + \nabla^2 F^u\right)\Delta x^2 + \mathcal{O}(\Delta x^4).\end{aligned}$$

Now we use (A.1) and the steady state formulas for $\nabla^2 u$ and $\nabla^2 v$ to find the scheme

$$\begin{aligned}&u_t - \lambda\left(\frac{2}{3}\bigoplus u + \frac{1}{3}\bigotimes u\right) + \frac{1}{3}(\langle uu_x\rangle_1 + \langle vu_y\rangle_1) + \frac{2}{3}(\langle uu_x\rangle_2 + \langle vu_y\rangle_2) - \frac{\Delta x^2}{12\lambda}\left((u^2u_x)_x + (v^2u_y)_y + 2(uvu_y)_x\right) \\ &\quad + \frac{\Delta x^2}{24\lambda}\left(2f(uv)_x + f(v^2 - u^2)_y\right) = fv + f\frac{\nabla^2 v}{12}\Delta x^2 + F^u + \frac{\nabla^2 F^u}{12}\Delta x^2 - \frac{\Delta x^2}{12\lambda}\left((uF^u)_x + vF^u_y + u_yF^v\right).\end{aligned}$$

We immediately notice the new dissipative term and correction to the Coriolis term conserve momentum. The equation for v is found by switching u with v, x with y, and changing the sign of f. The result is

$$\begin{aligned}&v_t - \lambda\left(\frac{2}{3}\bigoplus v + \frac{1}{3}\bigoplus v\right) + \frac{1}{3}(\langle uv_x\rangle_1 + \langle vv_y\rangle_1) + \frac{2}{3}(\langle uv_x\rangle_2 + \langle vv_y\rangle_2) - \frac{\Delta x^2}{12\lambda}\left((v^2v_y)_y + (u^2v_x)_x + 2(uvv_x)_y\right) \\ &\quad - \frac{\Delta x^2}{24\lambda}\left(2f(uv)_y + f(u^2 - v^2)_x\right) = -fu - f\frac{\nabla^2 u}{12}\Delta x^2 + F^v + \frac{\nabla^2 F^v}{12}\Delta x^2 - \frac{\Delta x^2}{12\lambda}((vF^v)_y + uF^v_x + v_xF^u).\end{aligned}$$

References

[1] E. Blayo, Compact finite difference scheme for ocean models, J. Comput. Phys. 164 (2000) 241–257.
[2] R.S. Hirsh, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, J. Comput. Phys. 19 (1975) 90–109.
[3] S.K. Lele, Compact finite difference scheme with spectral like resolution, J. Comput. Phys. 103 (1992) 16–43.
[4] D.A. Jones, L.G. Margolin, A.C. Poje, Accuracy and nonoscillatory properties of enslaved difference schemes, J. Comput. Phys. 181 (2002) 705–728.
[5] D.A. Jones, A.C. Poje, L.G. Margolin, Resolution effects and enslaved finite difference schemes for a double gyre, shallow water model, J. Theor. Comput. Fluid Dynamics 9 (1997) 269–280.
[6] D.A. Jones, S. Shkoller, Persistence of invariant manifolds for nonlinear PDEs, Stud. Appl. Math. 102 (1999) 27–67.
[7] L.G. Margolin, D.A. Jones, An approximate inertial manifold for computing Burgers equation, Physica D 60 (1992) 175–184.
[8] G.R. Sell, Y. You, Dynamics of Evolutionary Equations, vol. 143, Spring-Verlag, 2002.
[9] A.M. Stuart, A.R. Humphries, Dynamical Systems and Numerical Analysis, Cambridge University Press, New York, 1996.