

Filters for geodesy data based on linear and nonlinear diffusion

Róbert Čunderlík, Michal Kollár & Karol Mikula

**GEM - International Journal on
Geomathematics**

ISSN 1869-2672

Volume 7

Number 2

Int J Geomath (2016) 7:239-274


DOI 10.1007/s13137-016-0087-y



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



Filters for geodesy data based on linear and nonlinear diffusion

Róbert Čunderlík¹ · Michal Kollár¹  · Karol Mikula¹

Received: 1 September 2016 / Accepted: 4 September 2016 / Published online: 15 September 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract The article deals with filtering of data on closed surfaces by using the linear and nonlinear diffusion equations. The linear diffusion filtering is given by the Laplace–Beltrami operator representing linear diffusion along the surface. For the nonlinear diffusion filtering, we introduce nonlinear diffusion equations with diffusion coefficient depending on surface gradient and/or surface Laplacian of solution. This allows adaptive filtering respecting edges and local extrema in the data. For numerical discretization we develop a surface finite-volume method to approximate the partial differential equations on surfaces like sphere, ellipsoid or the Earth surface. The surfaces are approximated by a polyhedral mesh created by planar triangles representing subdivision of an initial icosahedron or octahedron grids. Numerical experiments illustrate behaviour of the linear and nonlinear diffusion filters on testing data and on real measurements, namely the GOCE satellite observations and the satellite-only mean dynamic topography. They show advantages of the nonlinear filters which, on the contrary to the linear one, preserve important structures in processed geodesy data.

Keywords Data filtering · Nonlinear diffusion equation · Surface finite volume method · GOCE data · Satellite-only mean dynamic topography

Mathematics Subject Classification 35R01 · 65M08

✉ Michal Kollár
michalkollar27@gmail.com

Róbert Čunderlík
cunderli@svf.stuba.sk

Karol Mikula
karol.mikula@gmail.com

¹ Department of Mathematics and Descriptive Geometry, Faculty of Civil Engineering, Slovak University of Technology, Radlinského 11, 810 05 Bratislava, Slovakia

1 Introduction

In many applications it is necessary to analyse data, e.g. classical images given on 2D or 3D regular grid structure or geoscience data given on 2D surfaces like Earth topography or satellite orbits, which are often contaminated by noise, and their quality can be rather poor. Nonlinear partial differential equations (PDEs) can be used to automatically produce output of higher quality, enhance the sharpness, filter out the noise, extract shapes, etc. From the mathematical point of view, the input processed data can be modelled by a real function $u^0(x)$, $u^0 : \Omega \rightarrow R$, where $\Omega \subset R^d$ represents a spatial domain. In image analysis, Ω is typically rectangular and $d = 2$ or 3, in surface data analysis Ω is a closed or open surface in R^3 . In the presented approach, we are strongly inspired by ideas from classical image processing which we transfer to geodesy surface data analysis.

Image processing operations based on PDEs involve such important tasks as image filtering, edge detection, deblurring and image enhancement, restoration, image inpainting, shape extraction and analysis, image segmentation, motion analysis, motion based filtering etc. (Alvarez et al. 1993; Alvarez and Morel 1994; Romeny 1994; Sethian 1999; Weickert 1998; Caselles et al. 1998; Nielsen et al. 1999; Sapiro 2001; Osher and Fedkiw 2003). Typical 2D examples are given by a large variety of medical and bioengineering images, satellite images, texts pre-processed for automatic reading, old corrupted photographs or any other digital images of poor quality. 3D or even 4D image analysis applications arise in biology, medicine or in material quality control, where 3D volumetric acquisition methods are widely used nowadays, see e.g. (Faure et al. 2016).

The first step to use PDEs for image processing was done in the beginning of eighties (Koenderink 1984; Witkin 1983). By the simple observation that the Gauss function

$$G_\sigma(x) = \frac{1}{(4\pi\sigma)^{d/2}} e^{-|x|^2/4\sigma} \quad (1)$$

is the fundamental solution of the linear heat (diffusion) equation, it has been possible to replace the classical image processing operation—convolution of an image with the Gauss kernel (normal probability distribution function) with variance $v = \sqrt{2\sigma}$ (Gaussian smoothing)—by solving the linear heat (diffusion) equation

$$u_t = \Delta u \quad (2)$$

for time $t = \sigma$ with initial condition given by the processed image u^0 . Here Δ denotes the classical Laplace operator in cartesian coordinates and u_t denotes the time derivative of the unknown function $u(t, x)$, $u : [0, T] \times \Omega \rightarrow R$.

It is well known that Gaussian smoothing (linear diffusion) blurs edges and moves their positions in images as well as smooths out local extrema of data. Although such phenomena can cause no problems in some examples of data analysis, in applications where, e.g. a visual impression is crucial and a precise localization of edges and values at extrema are necessary, the linear (Gaussian) smoothing is generally not the best choice. A way has been found to overcome these shortcomings, namely to switch to nonlinear diffusion models.

Due to the evolutionary character of the process which controls the processing using diffusion equations, application of any PDE to an initially given image is understood as its embedding in the so-called *scale space*. The Gaussian smoothing represents *linear scale space*. In the case of nonlinear PDEs one speaks about *nonlinear scale space*. The axioms and fundamental properties of such embeddings have been summarized and studied by Alvarez et al. (1993), Alvarez and Morel (1994), and Lions (1994), where the notion of *image multiscale analysis* has been introduced. The image multiscale analysis associates with a given image $u^0(x)$ a family $u(t, x)$ of smoothed-simplified images depending on an abstract parameter $t \in [0, T]$, the *scale*. As has been proved in Alvarez et al. (1993), if such a family fulfills certain basic assumptions—pyramidal structure, regularity and local comparison principle—then $u(t, x), u : [0, T] \times \Omega \rightarrow R$, can be represented as the unique viscosity solution [in the sense of Crandall et al. (1992)] of a general second order (degenerate) parabolic partial differential equation. This theoretical result has also an important practical counterpart. The equations of (degenerate) parabolic type have a smoothing property, so they are a natural tool for filtering (image simplification) by removing spurious structures, e.g. noise. Moreover, the simplification should be “data oriented”, e.g. it should respect edges and not blur them. Such requirements, or even more sophisticated ones related to the differential characteristics of the image, bring strong nonlinearity into the parabolic PDEs, and make this field interesting not only because of the applications but also from a mathematical and numerical point of view.

Since the end of the 80s, the nonlinear diffusion equations have been used for processing of 2D and 3D images. After the pioneering work of Perona and Malik (1987) who modified the linear heat equation (2) to nonlinear diffusion preserving edge positions, there has been a great deal of interest in the application and analysis of such equations. At present, the following nonlinear PDE suggested by Catté et al. (1992), often called regularized Perona–Malik model, is widely used in various practical image processing applications:

$$u_t - \nabla \cdot (g(|\nabla G_\sigma * u|)\nabla u) = 0, \tag{3}$$

where $u(t, x)$ is an unknown function. The equation is accompanied by the zero Neumann boundary conditions, the initial condition is given by the processed image $u^0 \in L_\infty(\Omega)$, the following assumptions on diffusion coefficient function g and smoothing kernel G_σ are prescribed

$$g : R_0^+ \rightarrow R^+ \text{ is a nonincreasing function, } g(0) = 1, g(s) \rightarrow 0 \text{ for } s \rightarrow \infty, \tag{4}$$

$$G_\sigma \in C^\infty(R^d) \text{ is a smoothing kernel, } \int_{R^d} G_\sigma(x) dx = 1, \int_{R^d} |\nabla G_\sigma| dx \leq C_\sigma, \tag{5}$$

$$G_\sigma(x) \rightarrow \delta_x \text{ for } \sigma \rightarrow 0, \text{ where } \delta_x \text{ is Dirac function localized at point } x, \tag{6}$$

and

$$\nabla G_\sigma * u(x) = \int_{R^d} \nabla G_\sigma(x - \xi)\tilde{u}(\xi) d\xi, \tag{7}$$

where \tilde{u} is an extension of u to R^d . One can consider the extension of u by 0 outside Ω or the reflective periodic extension of the image (Catté et al. 1992).

The Eq. (3) represents a modification of the original Perona–Malik model (Perona and Malik 1987; Nitzberg and Shiota 1992; Kichenassamy 1997)

$$u_t - \nabla \cdot (g(|\nabla u|)\nabla u) = 0, \quad (8)$$

called also *anisotropic diffusion* in the computer vision community. Perona and Malik introduced (8) in the context of edge enhancement. The equation selectively diffuses the image in the regions where the signal has small variance in intensity in contrast with those regions where the signal changes its tendency. Such a diffusion process is governed by the shape of the diffusion coefficient given by the function g in (8) and by its dependence on ∇u , which is understood as an edge indicator (Perona and Malik 1987). Since $g \rightarrow 0$ for large gradients, the diffusion is strongly slowed down on edges, while outside them it provides averaging of pixel intensities as in the linear case. From a mathematical point of view, for practical choices of g (e.g. $g(s) = 1/(1 + s^2)$, $g(s) = e^{-s^2}$), the original Perona–Malik equation (8) can behave locally like the backward heat equation. It is, in general, an ill-posed problem which suffers from non-uniqueness and whose solvability is a difficult problem (Kichenassamy 1997). One way to overcome this disadvantage was proposed in (Catté et al. 1992), where the convolution with the Gaussian kernel G_σ was introduced into the decision process for the value of the diffusion coefficient, cf. (3). Since convolution with the Gaussian is equivalent to linear diffusion, their model combines ideas of linear and nonlinear scale space equations. Such a slight modification made it possible to prove the existence and uniqueness of solutions for the modified equation, and to keep the practical advantages of the original formulation (Catté et al. 1992). Moreover, usage of the *Gaussian gradient* $\nabla G_\sigma * u$ combines the theoretical and implementation aspects of the model. The convolution (with prescribed σ) gives a unique way to compute gradients of a piecewise constant image. It also bounds (depending on σ) the gradient of the solution as input of the function g in the continuous model—which corresponds to the situation in numerical implementations where gradients evaluated on a discrete grid are finite. Also, the local edge enhancement is more understandable in the presence of noise.

In this paper we present linear and nonlinear diffusion filtering methods for geodesy data given on closed surfaces. For readers interested in image processing numerical algorithms on regular grids we refer, e.g. to (Mikula and Ramarosy 2001; Kriva et al. 2010) and further references therein. In order to process surface data by PDEs like (2) and (3), instead of standard gradient and Laplacian, we have to consider the surface gradient and surface Laplacian—so called Laplace–Beltrami operator. For any scalar function u defined on an open subset G of R^d containing surface Ω the surface (tangential) gradient is defined by

$$\nabla_s u = \nabla u - (\nabla u \cdot \nu)\nu \quad (9)$$

where ν is outer unit normal to the closed surface Ω and ∇u denotes the usual gradient and \cdot denotes the usual scalar product on R^d . The tangential gradient $\nabla_s u$ only depends

on the values of u restricted to surface Ω and $\nabla_s u \cdot \nu = 0$. The Laplace–Beltrami operator on surface Ω is then defined as the tangential divergence of the tangential gradient, i.e.

$$\Delta_s u = \nabla_s \cdot \nabla_s u. \tag{10}$$

We also recall the Green (integration by parts) formula for scalar functions on surfaces and its consequences (Gilbarg and Trudinger 1988; Dziuk and Elliott 2007, 2013), which will be used in derivation of numerical schemes later in the paper. Let Γ be a subset of Ω having a boundary $\partial\Gamma$ whose unit outer normal, tangential to Γ , is denoted by η . Then

$$\int_{\Gamma} \nabla_s u \, dA = \int_{\partial\Gamma} u \eta \, da - \int_{\Gamma} H u \nu \, dA \tag{11}$$

where H denotes the (scalar) mean curvature of Γ , and dA denotes 2D surface element measure and da denotes 1D curve element measure, for the proof see (Dziuk and Elliott 2013), Theorem 2.10. Let q be a tangential vector field to the surface Γ , i.e. $q \cdot \nu = 0$, a typical example used later is the so called surface diffusion flux given by $q = g \nabla_s u$, where g is a scalar function defined on the surface. Then from (11) we obtain divergence theorem on the surface

$$\int_{\Gamma} \nabla_s \cdot q \, dA = \int_{\partial\Gamma} q \cdot \eta \, da - \int_{\Gamma} H q \cdot \nu \, dA = \int_{\partial\Gamma} q \cdot \eta \, da. \tag{12}$$

and choosing $g = 1$ in surface diffusion flux we get

$$\int_{\Gamma} \Delta_s u \, dA = \int_{\partial\Gamma} \nabla_s u \cdot \eta \, da. \tag{13}$$

In the following sections, we introduce and discuss linear and nonlinear diffusion models depending on surface gradient and surface Laplacian. We present in details their numerical discretization by surface finite volume method and study behaviour of models and their numerical discretizations on testing examples as well as on real geodesy data.

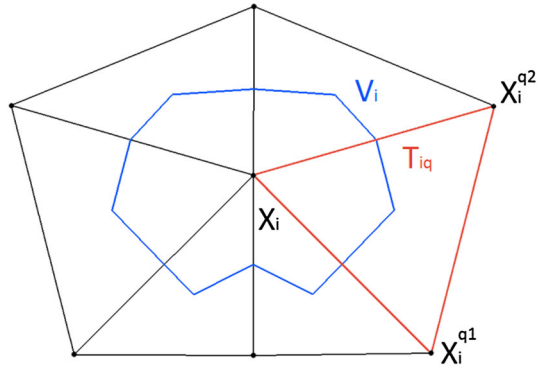
2 The linear diffusion filter on a surface

The linear diffusion of a scalar function u on a closed surface Ω is given by the equation

$$\partial_t u = \Delta_s u, \tag{14}$$

which is a direct generalization of Eq. (2) and serves for a uniform smoothing of data on surfaces. Since the surface is closed no boundary conditions have to be prescribed. Processed data $u^0(x)$ defined on Ω gives initial condition for (14).

Fig. 1 Finite volume V_i



2.1 Numerical discretization of linear diffusion on a surface

The differential equation (14) is numerically solved by the surface finite volume method (Čunderlík et al. 2012; Eymard et al. 2003). In this approach, the surface Ω is approximated by an appropriate triangulation defined by N representative nodes $X_i, X_i \subset \Omega, i = 1, \dots, N$. These nodes represent vertices of the triangular grid defined by planar triangles $T_{iq}, q = 1, \dots, Q_i, i = 1, \dots, N$, where Q_i is the number of triangles with the vertex X_i . Other two vertices of the triangle T_{iq} will be denoted by X_i^{q1} and X_i^{q2} . A value of function u in the node X_i is denoted by u_i . For the given triangulation we construct a finite volume grid. At each node X_i we create a co-volume (finite volume) V_i bounded by straight lines that connect midpoints between X_i and its neighbours X_i^{q1}, X_i^{q2} with centers of mass of all triangles joined in the node X_i (see Fig. 1).

By integrating Eq. (14) over the finite volume V_i and by applying (13) to its right hand side we obtain

$$\int_{V_i} \partial_t u \, dA = \int_{\partial V_i} \nabla_s u \cdot \eta_i \, da, \tag{15}$$

where $\nabla_s u$ represents the surface gradient of the function u and η_i is a unit outer normal, tangential to the co-volume boundary ∂V_i . Taking into account geometry of this boundary we get

$$\int_{V_i} \partial_t u \, dA = \sum_{q=1}^{Q_i} \int_{\partial V_{iq}} \nabla_s u \cdot \eta_{iq} \, da, \tag{16}$$

where ∂V_{iq} are parts of the co-volume boundary that belong to T_{iq} having normal vectors η_{iq} .

Equation (14) is solved in a time interval $[0, T]$. This interval is divided into M time steps $t_j, j = 1, \dots, M$ and the time derivative $\partial_t u$ is approximated by the backward difference

$$\partial_t u \approx \frac{u^j - u^{j-1}}{\tau}, \tag{17}$$

where $\tau = t_j - t_{j-1}$ denotes the time step and the value of u^j represents a solution in the j th time step. Then the left hand side of the Eq. (16) can be approximated by

$$\int_{V_i} \partial_t u \, dA \approx m(V_i) \frac{u_i^j - u_i^{j-1}}{\tau}, \tag{18}$$

where $m(V_i)$ denotes the area of co-volume V_i , and we get

$$m(V_i) \frac{u_i^j - u_i^{j-1}}{\tau} = \sum_{q=1}^{Q_i} \int_{\partial V_{iq}} \nabla_s u^j \cdot \eta_{iq} \, da. \tag{19}$$

For the right hand side of Eq. (19) let us consider a linear representation of u^j on each triangle. Then the surface gradient $\nabla_s u^j$ is a constant vector over each triangle T_{iq} and we can replace it by the mean value

$$\nabla_s u^j = \frac{1}{m(T_{iq})} \int_{T_{iq}} \nabla_s u^j \, da, \tag{20}$$

where $m(T_{iq})$ denotes the area of the triangle T_{iq} . Applying (11) to the right hand side of Eq. (20), since any triangle has the zero mean curvature, we obtain

$$\nabla_s u^j = \frac{1}{m(T_{iq})} \int_{\partial T_{iq}} u^j \cdot \mathbf{n}_{iq} \, da, \tag{21}$$

where \mathbf{n}_{iq} is the unit outer normal vector, tangential to the boundary of the triangle T_{iq} . For the linear representation of u^j , the integral over the triangle boundary can be expressed as a sum of average values from each triangle side, and denoting by $\mathbf{P}_{T_{iq}}^j$ the constant approximation of the surface gradient on the triangle T_{iq} , we get

$$\mathbf{P}_{T_{iq}}^j = \frac{1}{m(T_{iq})} \left(\frac{u_i^j + u_{q1}^j}{2} d_{iq1} \mathbf{n}_{iq1} + \frac{u_i^j + u_{q2}^j}{2} d_{iq2} \mathbf{n}_{iq2} + \frac{u_{q1}^j + u_{q2}^j}{2} d_{q1q2} \mathbf{n}_{q1q2} \right) \tag{22}$$

where $u_i^j, u_{q1}^j, u_{q2}^j$ denotes the nodal values of the solution in triangle nodes X_i, X_i^{q1}, X_i^{q2} , see Figs. 2 and 3, where all geometrical quantities appearing in (22) are plotted.

Consequently, the approximation of Eq. (19) can be written in the form

$$m(V_i) \frac{u_i^j - u_i^{j-1}}{\tau} = \sum_{q=1}^{Q_i} \int_{\partial V_{iq}} \mathbf{P}_{T_{iq}}^j \cdot \eta_{iq} \, da. \tag{23}$$

Since $\mathbf{P}_{T_{iq}}^j$ is a constant vector and

Fig. 2 Plot of the triangle T_{iq} sides with lengths d_{iq1} , d_{iq2} , d_{q1q2} , and portions of the co-volume boundary e_{iq}^1 , e_{iq}^2 belonging to the triangle T_{iq}

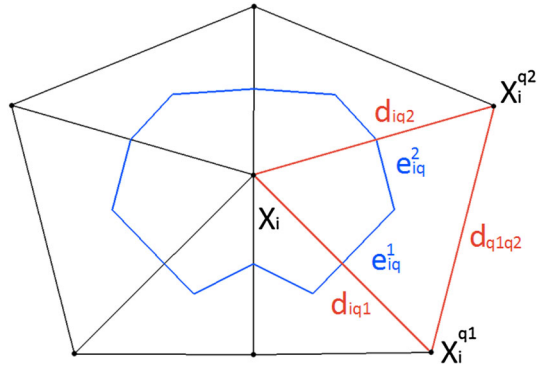
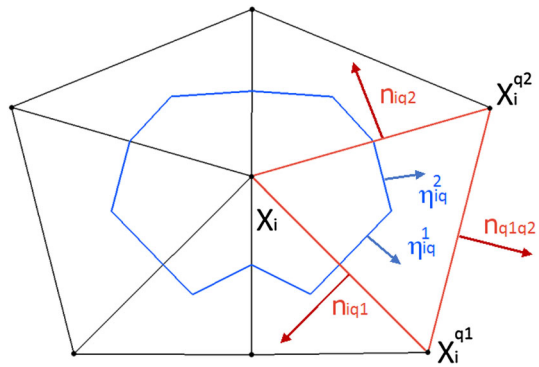


Fig. 3 Plot of the outer unit normal vectors to the triangle T_{iq} sides, \mathbf{n}_{iq1} , \mathbf{n}_{iq2} , \mathbf{n}_{q1q2} , and to the co-volume boundary, η_{iq}^1 , η_{iq}^2



$$\int_{\partial V_{iq}} \eta_{iq} = m(e_{iq}^1) \eta_{iq}^1 + m(e_{iq}^2) \eta_{iq}^2, \tag{24}$$

where $m(e_{iq}^1)$ and $m(e_{iq}^2)$ are lengths of the parts of the co-volume boundaries inside the triangle T_{iq} , see Fig. 2, we get

$$m(V_i) \frac{u_i^j - u_i^{j-1}}{\tau} = \sum_{q=1}^{Q_i} \left[m(e_{iq}^1) \eta_{iq}^1 \cdot \mathbf{P}_{T_{iq}}^j + m(e_{iq}^2) \eta_{iq}^2 \cdot \mathbf{P}_{T_{iq}}^j \right] \tag{25}$$

which can be, for every $i = 1, \dots, N$, written in the form

$$u_i^j - \frac{\tau}{m(V_i)} \sum_{q=1}^{Q_i} \left[m(e_{iq}^1) \eta_{iq}^1 \cdot \mathbf{P}_{T_{iq}}^j + m(e_{iq}^2) \eta_{iq}^2 \cdot \mathbf{P}_{T_{iq}}^j \right] = u_i^{j-1} \tag{26}$$

and represents the implicit numerical scheme for solving linear diffusion equation (14) on the closed surface Ω .

Equation (26) represent a linear system of equations which can be written in the form

$$\mathbf{A}\mathbf{u}^j = \mathbf{u}^{j-1} \tag{27}$$

where \mathbf{A} represents the system matrix and $\mathbf{u}^j = [u_1^j, \dots, u_N^j]^T$ is a vector of nodal values of the solution in the j th time step. The system matrix \mathbf{A} is a sparse non-symmetric matrix and its properties depend on the time step τ and geometry of the triangulation. Let us define a local numbering of nodal values in co-volume and its neighbourhood: the nodal value in the co-volume centre will be denoted by u_{i0}^j and its neighbouring unknown values are denoted by $u_{i1}^j, \dots, u_{iQ_i}^j$. Then the equation corresponding to co-volume V_i contains $Q_i + 1$ unknowns $u_{i0}^j, u_{i1}^j, \dots, u_{iQ_i}^j$ and i th row of the linear system (27) is given by

$$\sum_{q=0}^{Q_i} a_{iq}^j u_{iq}^j = u_{i0}^{j-1}, \tag{28}$$

where $a_{i0}^j, a_{i1}^j, \dots, a_{iQ_i}^j$ represent non-zero coefficients in the i th row of the matrix \mathbf{A} (let us note that for the linear diffusion approximation the upper (time) index j is dummy, but it will play a role in nonlinear diffusion approximation later). The exact column locations of these non-zero coefficients depends on a global indexing of the corresponding nodal value. However, one can easily see that the diagonal coefficient a_{i0}^j can be expressed in the form

$$\begin{aligned} a_{i0}^j = 1 - \frac{\tau}{m(V_i)} \sum_{q=1}^{Q_i} \frac{1}{2m(T_{iq})} & \left[m(e_{iq}^1) \boldsymbol{\eta}_{iq}^1 \cdot (d_{iq1} \mathbf{n}_{iq1} + d_{iq2} \mathbf{n}_{iq2}) \right. \\ & \left. + m(e_{iq}^2) \boldsymbol{\eta}_{iq}^2 \cdot (d_{iq1} \mathbf{n}_{iq1} + d_{iq2} \mathbf{n}_{iq2}) \right] \end{aligned} \tag{29}$$

and the non-diagonal coefficients $a_{iq}^j, q = 1, \dots, Q_i$, are given by

$$\begin{aligned} a_{iq}^j = -\frac{\tau}{m(V_i)} & \left(\frac{1}{2m(T_{iq})} \left[m(e_{iq}^1) \boldsymbol{\eta}_{iq}^1 \cdot (d_{iq1} \mathbf{n}_{iq1} + d_{q1q2} \mathbf{n}_{q1q2}) \right. \right. \\ & \left. \left. + m(e_{iq}^2) \boldsymbol{\eta}_{iq}^2 \cdot (d_{iq1} \mathbf{n}_{iq1} + d_{q1q2} \mathbf{n}_{q1q2}) \right] \right. \\ & \left. + \frac{1}{2m(T_{ik})} \left[m(e_{ik}^1) \boldsymbol{\eta}_{ik}^1 \cdot (d_{ik1} \mathbf{n}_{ik1} + d_{k1k2} \mathbf{n}_{k1k2}) \right. \right. \\ & \left. \left. + m(e_{ik}^2) \boldsymbol{\eta}_{ik}^2 \cdot (d_{ik1} \mathbf{n}_{ik1} + d_{k1k2} \mathbf{n}_{k1k2}) \right] \right). \end{aligned} \tag{30}$$

where two triangles with indexes q and k are involved and index $k = q - 1$ if $q > 1$ and $k = Q_i$ if $q = 1$.

2.2 Behaviour of the linear diffusion on a closed surface

In this subsection, we present a simple numerical experiment showing how an additive noise is filtered from an artificial function defined on a spherical computational domain. To approximate a unit sphere we use the seventh subdivision of the initial icosahedron grid. The process of subdivision is illustrated in Fig. 4.

In this experiment, we use the triangulation with 163,842 nodes and 327,680 triangles. On this triangulated sphere we defined an artificial piecewise linear function w in such a way that $w_i = 1$ for all nodes located on lands and $w_i = 0$ for all nodes at oceans (see Fig. 5).

Onto this function w , we put an additive noise up to 40 % at 16,661 nodes which represent approximately 10 % of all nodes randomly distributed over the sphere. The generated uniform non-Gaussian additive noise is from the interval $(-0.4, 0.4)$ and after adding the noise we get our piecewise linear initial condition u^0 for which $u_i^0 \in (0.6, 1.4)$ on lands and $u_i^0 \in (-0.4, 0.4)$ at oceans (see Fig. 6).

The system of linear equations (27) has to be solved in every filtering (time) step. To that goal, in all our numerical experiments, we use the SOR (successive-over-relaxation) method (Young 1971). In order to get convergence of SOR, we make the system matrix diagonally dominant choosing the time step τ proportional to an average area of the co-volumes (Čunderlík et al. 2012)

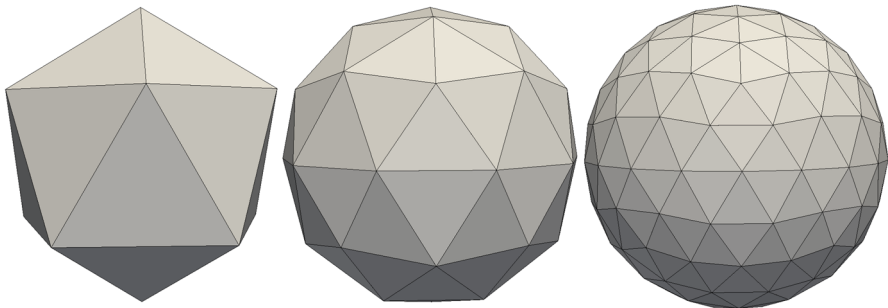


Fig. 4 Process of subdivision of the initial icosahedron

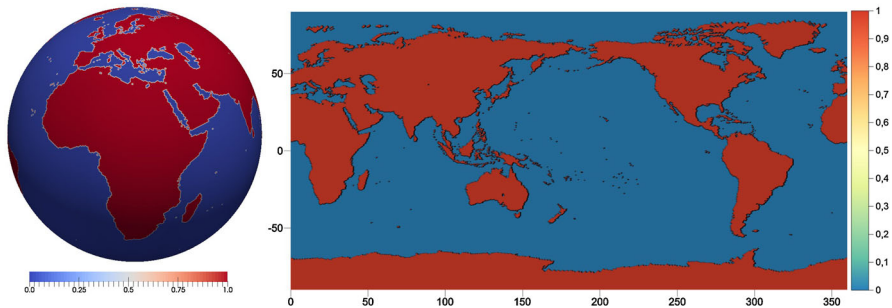


Fig. 5 Artificial function w

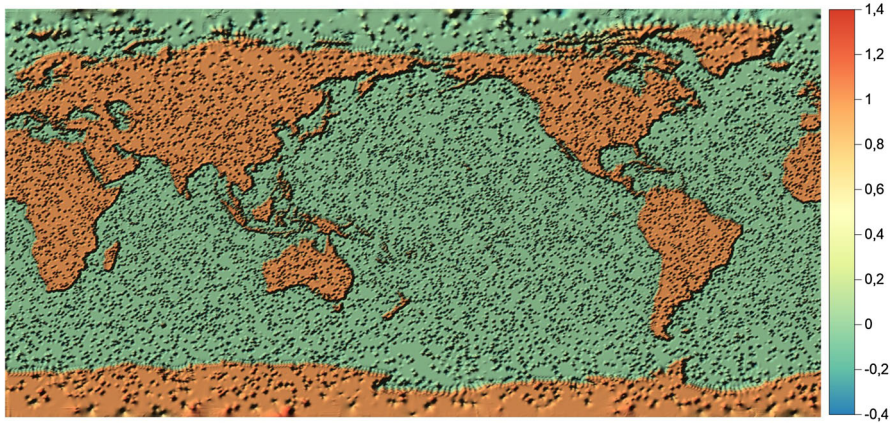


Fig. 6 Initial condition u^0

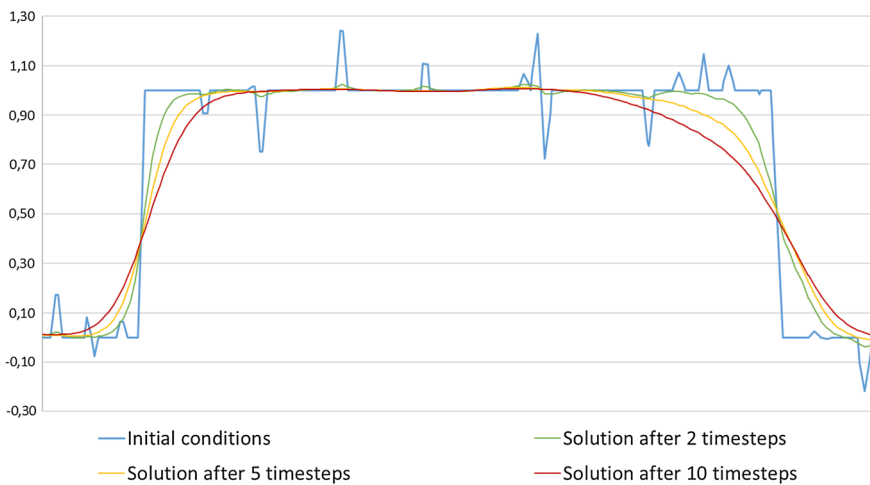


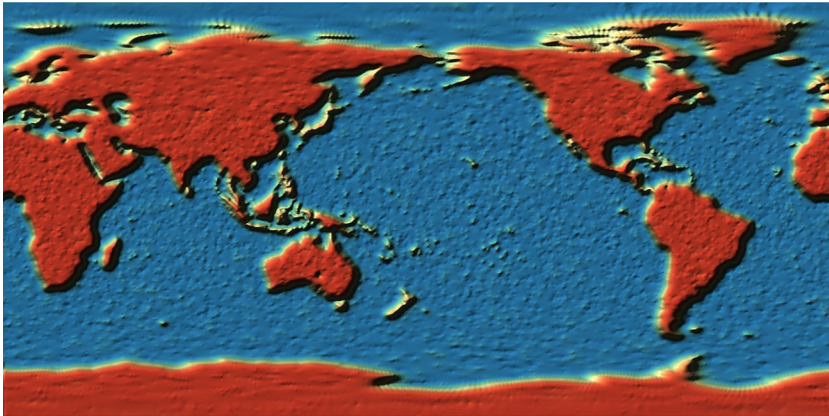
Fig. 7 Profiles across South America from the solution of filtering of the additive noise using the surface linear diffusion after 2, 5 and 10 time steps

$$\tau = \frac{1}{N} \sum_{i=1}^N m(V_i). \quad (31)$$

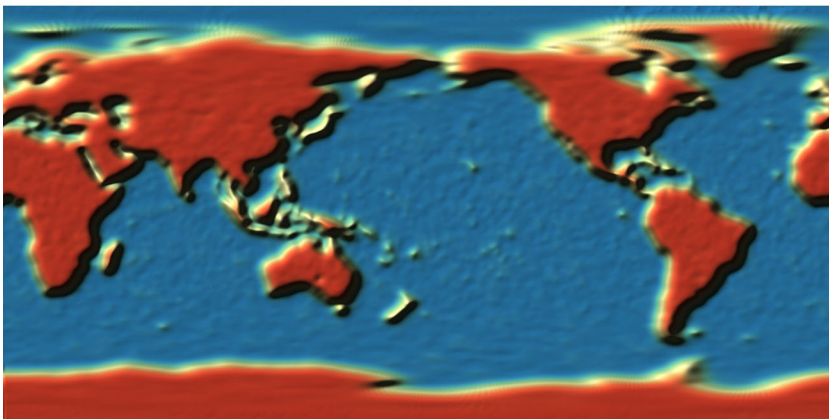
The choice of the time step is essential for the whole filtering process and implies how many filtering (time) steps (we call it also iterations) will be necessary to get reasonable results.

Our numerical results obtained after 2, 5 and 10 iterations of the linear diffusion filtering ($\tau = 7 \cdot 10^{-5}$) are depicted in Fig. 8. Figure 7 shows the corresponding profiles along the equator across South America.

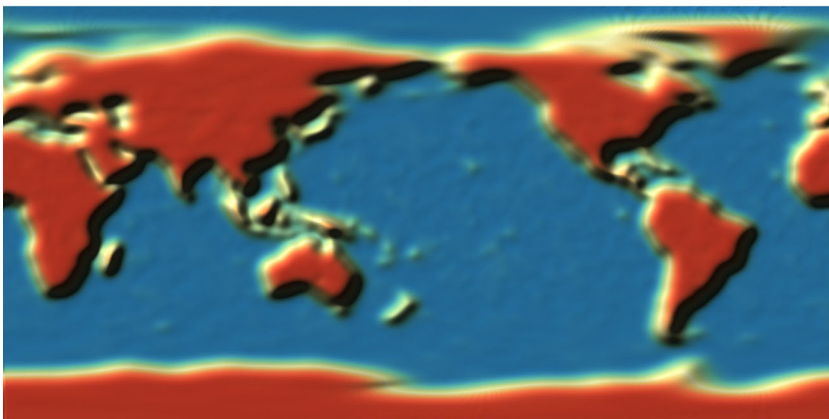
From the result, it is evident that this approach reduces the additive noise but it has a uniform smoothing effect similarly to other linear filters. Together with denoising



(a)



(b)



(c)

Fig. 8 The solution of filtering of the additive noise using the surface linear diffusion after **a** 2, **b** 5, and **c** 10 time steps

it smooths out the main structures represented by the edges of continents. This can produce serious inaccuracies in interpretation of processed data. A way how to avoid such inaccuracies is to use nonlinear filtering methods.

3 The nonlinear diffusion filters on a surface

To perform a nonuniform smoothing of data on surfaces we need to choose an appropriate diffusion coefficient in diffusion PDEs. The main idea is that the diffusion coefficient should not be a constant but a nonlinear function of differential characteristics of data.

3.1 The nonlinear diffusion depending on the surface gradient-regularized surface Perona–Malik model

In this approach, originally developed in Čunderlík et al. (2012), we use an analogy with the regularized Perona–Malik model (3) from the classical image processing and we suggest following PDE

$$\partial_t u = \nabla_s \cdot (g(|\nabla_s u^\sigma|) \nabla_s u) \tag{32}$$

for filtering the data on surfaces. The nonlinear diffusivity function g depends on the term $\nabla_s u^\sigma$, the surface gradient of solution u smoothed by the surface linear diffusion for a short time interval σ , and represents an edge detector for surface data. We consider g in the form

$$g(|\nabla_s u^\sigma|) = \frac{1}{1 + K|\nabla_s u^\sigma|^2}, \quad K \geq 0, \tag{33}$$

where constant K represents an edge sensitivity parameter. By this definition, g fulfils assumption (4) and returns values from the range

$$0 < g(|\nabla_s u^\sigma|) \leq 1. \tag{34}$$

The parameter K determines how sensitive will be the edge detector to high values of a smoothed surface gradient of u and gives us a decision capability which gradients to preserve. Large gradients which represent edges in processed data yield a small value of edge detector and vice versa. If the values of edge detector are close to zero, the diffusion process is strongly slowed down, on the other hand, if the values are close to 1, the process is similar to the linear diffusion. This allows the adaptive smoothing according to surface gradients of the solution. The parameter σ affects the solution u^σ of the surface linear diffusion from which the surface gradient is computed. This presmoothing of surface gradients causes that only un-noisy edges will be preserved in nonlinear adaptive smoothing process. An appropriate choice of the parameters K and σ plays an important role in the filtering process and needs to be tuned experimentally.

To approximate the Eq. (32) we again apply the surface finite volume method and use the same notation as in Sect. 2. By integrating (32) over the finite volume V_i and by applying (12) to its right hand side, we obtain

$$\int_{V_i} \partial_t u \, dA = \int_{\partial V_i} g(|\nabla_s u^\sigma|) \nabla_s u \cdot \eta_i \, da. \tag{35}$$

After analogous steps as described in Sect. 2, we obtain the numerical scheme

$$u_i^j - \frac{\tau}{m(V_i)} \sum_{q=1}^{Q_i} \left[m(e_{iq}^1) \eta_{iq}^1 \cdot \mathbf{P}_{T_{iq}}^j g(|\mathbf{P}_{T_{iq}}^{\sigma, j-1}|) \right. \\ \left. \times m(e_{iq}^2) \eta_{iq}^2 \cdot \mathbf{P}_{T_{iq}}^j g(|\mathbf{P}_{T_{iq}}^{\sigma, j-1}|) \right] = u_i^{j-1}, \tag{36}$$

$i = 1, \dots, N$, which represents a semi-implicit numerical scheme for solving the regularized Perona–Malik model on the closed surface Ω . The diffusivity function g inside the scheme (36) depends on $P_{T_{iq}}^{\sigma, j-1}$ that represents numerical approximation of the surface gradient on triangle T_{iq} of the solution u at the previous time step $j - 1$ smoothed by one step σ of the surface linear diffusion. In such a way the nonlinearity in the Eq. (32) is treated by using the smoothed gradients from the previous time step, thus leading to system of linear equations. Comparing the semi-implicit scheme (36) with the implicit scheme (26) for the surface linear diffusion, the difference is that now the surface gradients $P_{T_{iq}}^j$ are multiplied by the edge detector function g which allows the adaptive smoothing according to the smoothed surface gradients evaluated at the previous time step. Consequently, the edge detector is step by step evolving in time giving an opportunity to preserve main structures in the data and effectively reduce the noise. Our experience shows that the regularized surface Perona–Malik model (32) successfully reduces noise while preserves edges, but we have also observed that it slightly smooths out local extrema of filtered data (Čunderlík et al. 2012). It is due to the fact that the smoothed surface gradients are not high enough in areas of local extrema, but opposite, they are close to zero.

3.1.1 Behaviour of the regularized surface Perona–Malik model

To show behaviour and advantages of the nonlinear diffusion model (32) we use the same experiment as we used in Sect. 2.2. This experiment aims to demonstrate how diffusion filtering controlled by the edge detector can successfully remove an additive noise. We use the same space discretization of the unit sphere and the same initial condition, see Fig. 6. In case of the nonlinear surface diffusion, the linear system of equations is given by the semi-implicit scheme (36). To get this system, first we have to apply the linear diffusion filtering to the solution from the previous time step. Then we evaluate the corresponding surface gradients that indicate values of the edge detector. After that we are able to compute coefficients of the system matrix. This process is repeated in every filtering step. Therefore, the nonlinear surface diffusion filtering is more time consuming and usually requires more iterations than the linear one.

A significant part of data filtering is finding the optimal values for the edge detector function parameters K and σ and the time step τ and number of iterative (filtering) time steps. Since in this experiment the true (un-noisy) solution is known, we are able

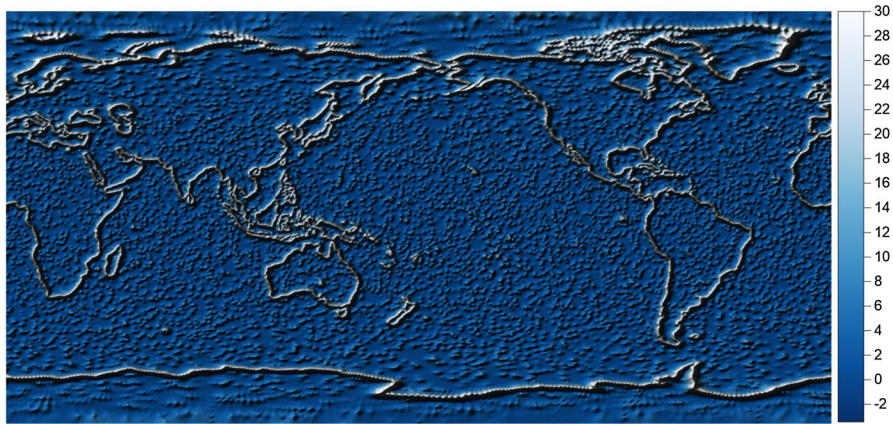


Fig. 9 Gradients of the data after the linear diffusion ($\sigma = 7 \cdot 10^{-5}$)

to tune the parameters considering the residuals between the true and filtered solutions. In general, the parameters for the edge detector depend on gradients of processed data. In Fig. 9, the gradients of solution obtained after one step of surface linear diffusion are plotted, $\sigma = \tau = 7 \cdot 10^{-5}$ in Eq. (31).

If we want to preserve the high gradients along the continents edges, we need to estimate the sensitivity parameter K in Eq. (33) according to them. We set the value of K with respect to the maximum value $\nabla_s u_{max}^\sigma$ of all gradients of the processed data. We want the edge detector function to have a small value S in the nodes with the highest gradient. Then, considering $|\nabla_s u_{max}^\sigma| > 0$, we can set

$$K = \frac{1 - S}{S|\nabla_s u_{max}^\sigma|^2}. \tag{37}$$

For this experiment, we use $S = 10^{-5}$ and the estimated $K = 48.26$. The time step $\tau = 7 \cdot 10^{-5}$ is chosen according to Eq. (31).

Figure 11 depicts the solution of filtering after 250, 500 and 1000 iterations of the nonlinear diffusion. Figure 10 shows the corresponding profiles along the equator across South America. From the result it is evident that the signal corresponding to the additive noise is step by step vanishing while high gradients and their positions remain preserved. Here we remind that the edge detector always depends on surface gradients computed from the solution in the previous iterative step, thus it is adapted to the filtered solution evolving in time. Such an adaptive smoothing effect is a main advantage of the nonlinear filtering. Comparing results of the linear and nonlinear diffusion filtering, Figs. 8 and 11, it is evident that both approaches reduce the additive noise, however, the linear filtering also smooths out the main structures, i.e. the high gradients on edges of continents. On the contrary, the nonlinear diffusion model preserves them almost unaltered, see Fig. 10.

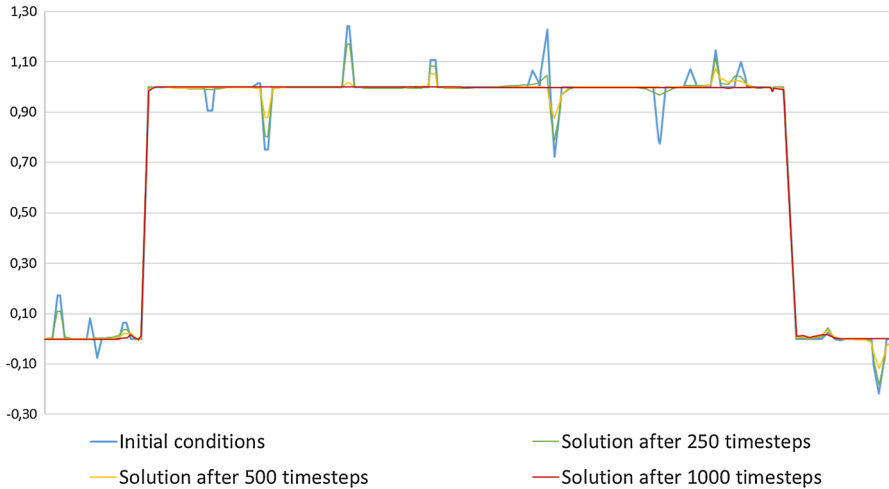


Fig. 10 Profiles across South America from the solution of filtering of the additive noise using the nonlinear surface diffusion after 250, 500 and 1000 time steps

3.2 The nonlinear diffusion influenced by the surface Laplacian

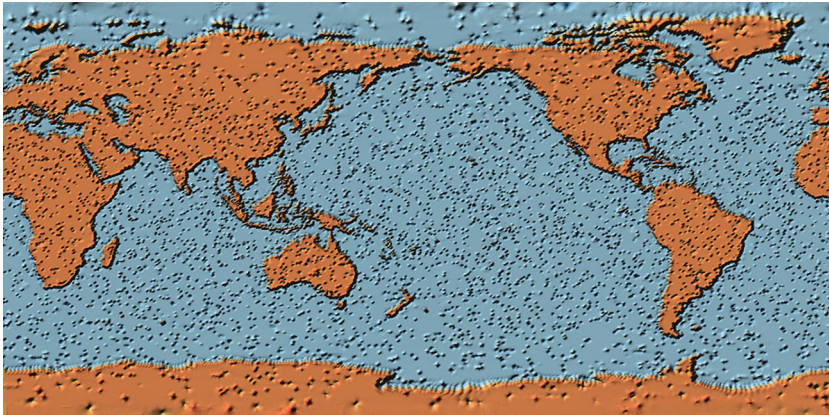
As mentioned earlier, the regularized Perona–Malik nonlinear diffusion filtering can slightly smooth the local extrema when processing data. Such drawback can be overcome by a suitable modification of diffusivity function g . We extend the edge detector function of the Perona–Malik model by another argument which will slow down diffusion process in areas of local extrema, i.e. in regions with large values of surface Laplacian—the Laplace–Beltrami operator. The diffusion coefficient which represents the edge and local extrema detector is defined by

$$g(|\nabla_s u^{\sigma^1}|, |\Delta_s u^{\sigma^2}|) = \frac{1}{1 + K_1 |\nabla_s u^{\sigma^1}|^2 + K_2 |\Delta_s u^{\sigma^2}|^2}, \quad K_1, K_2 \geq 0. \quad (38)$$

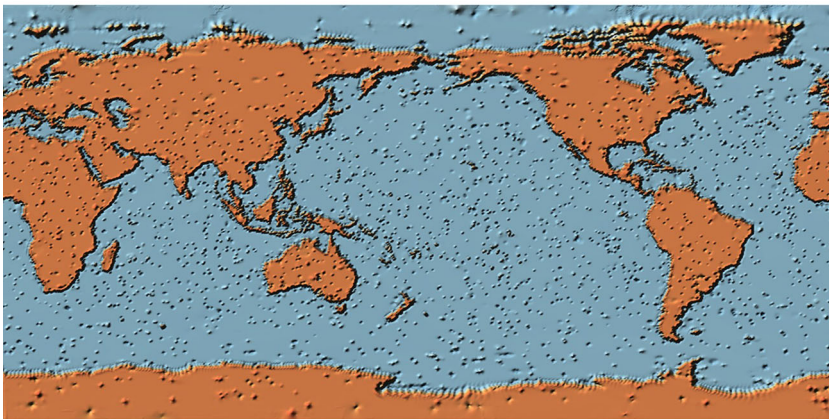
The function g depends on the surface gradient of solution u smoothed by the linear diffusion equation for a short time interval σ^1 as well as on the Laplace–Beltrami operator applied to solution u smoothed by the linear diffusion equation for a short time interval σ^2 . The parameter K_1 has the same meaning as in the edge detector in the regularized Perona–Malik model and the parameter K_2 affects sensitivity to high values of the surface Laplacian of the function u .

To approximate the Laplace–Beltrami operator on a co-volume we use similar steps as we have used in Sect. 2. We denote its approximate mean value on the co-volume V_i in time step j by C_i^j and using derivations yielding Eq. (25) we obtain

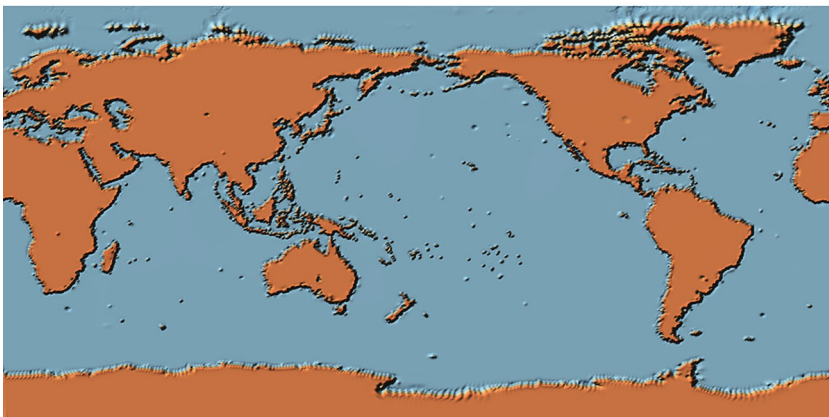
$$C_i^j = \frac{1}{m(V_i)} \sum_{q=1}^{Q_i} \left[m(e_{iq}^1) \eta_{iq}^1 \cdot P_{Tiq}^j + m(e_{iq}^2) \eta_{iq}^2 \cdot P_{Tiq}^j \right]. \quad (39)$$



(a)



(b)



(c)

Fig. 11 Solutions of filtering of the additive noise using the nonlinear surface diffusion after **a** 250, **b** 500 and **c** 1000 time steps

Since we need a value of diffusion coefficient on the edges of co-volume, i.e. on the triangles meeting in the center of co-volume, we compute the average value of the Laplace–Beltrami operator on the triangle T_{iq} by

$$C_{T_{iq}}^j = \frac{1}{3} (C_i^j + C_{q_1}^j + C_{q_2}^j), \tag{40}$$

where C_i^j , $C_{q_1}^j$ and $C_{q_2}^j$ represent the values in triangle vertices. Using the same approach as we have used in the Sect. 3.1, we obtain a semi-implicit numerical scheme for the nonlinear diffusion filtering method influenced by the surface Laplacian on the closed surface Ω

$$u_i^j - \frac{\tau}{m(V_i)} \sum_{q=1}^{Q_i} \left[m(e_{iq}^1) \eta_{iq}^1 \cdot \mathbf{P}_{T_{iq}}^j g \left(|\mathbf{P}_{T_{iq}}^{\sigma^1, j-1}|, |C_{T_{iq}}^{\sigma^2, j-1}| \right) \right. \\ \left. \times m(e_{iq}^2) \eta_{iq}^2 \cdot \mathbf{P}_{T_{iq}}^j g \left(|\mathbf{P}_{T_{iq}}^{\sigma^1, j-1}|, |C_{T_{iq}}^{\sigma^2, j-1}| \right) \right] = u_i^{j-1}, \tag{41}$$

$i = 1, \dots, N$, where $\mathbf{P}_{T_{iq}}^{\sigma^1, j-1}$ is an approximation of the smoothed gradient of the solution from the previous time step $j - 1$, and analogously, $C_{T_{iq}}^{\sigma^2, j-1}$ is an approximation of the smoothed surface Laplacian of the solution from the previous time step $j - 1$.

3.2.1 Comparison of behaviour of linear and nonlinear diffusion filters

In this testing experiment, we study a different behavior of three different filtering methods introduced above, namely, the linear diffusion model, the nonlinear diffusion depending on the surface gradient and the nonlinear model influenced by the surface Laplacian. We use the same icosahedron grid as in Sect. 2.2 and create an artificial initial data u^0 as a function with compact support defined by

$$u^0(\mathbf{x}) = v \cdot e^{\frac{-\sigma}{(r^2 - |\mathbf{x} - \mathbf{s}|^2) + \frac{\sigma}{r^2}}} \tag{42}$$

if $|\mathbf{x} - \mathbf{s}| < r$ and $u(\mathbf{x}) = 0$ if $|\mathbf{x} - \mathbf{s}| \geq r$, where $\mathbf{s} = (s_1, s_2, s_3) \in \Omega$ and $\mathbf{x} = (x_1, x_2, x_3) \in \Omega$. We set $v = 0.4$, $\sigma = 5r^2$, $r = 0.2$ and $\mathbf{s} = (0, -0.52573, 0.85065)$. The function u^0 is plotted in Fig. 12.

Figure 13 depicts an intersection through chosen nodes of the initial data u^0 and the results obtained after 50 and 100 time steps of each diffusion filter. In case of the nonlinear surface Perona–Malik model we use the edge sensitivity coefficient $K = 10$. In case of the surface Laplacian influence, we use the sensitivity coefficients $K_1 = 10$, $K_2 = 0.01$ for the edge and local extrema detector. In the plots corresponding to the surface Perona–Malik model and model influenced by the surface Laplacian, the initial values of the edge and local extrema detector before the first time step are visualized. The scale for detector values on the right hand side of both plots is reversed.

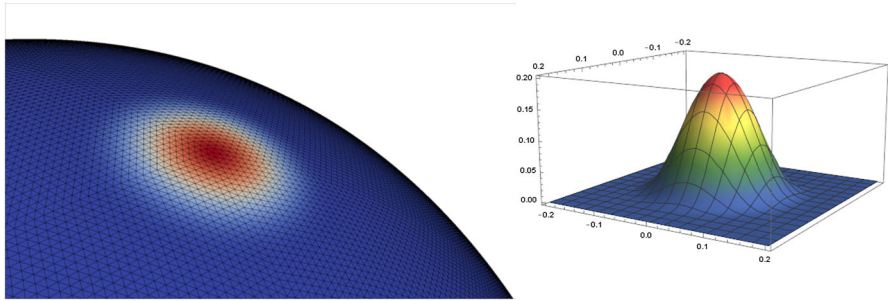


Fig. 12 The initial artificial function u^0

The profiles of results of the linear diffusion filter confirms a uniform smoothing effect, see in Fig. 13a. In Fig. 13b, we can see that the surface Perona–Malik model preserves edges of initial data. The red line in Fig. 13b represents values of the edge detector in corresponding nodes. On the edges, the values of the detector are almost zero and for this reason the diffusion process is remarkably slowed down. The values of the edge detector grow towards a top of the data, and, as we can see by the results after 50 and 100 time step, the local extremum of the data is pulled down. Figure 13c shows the profile of results of the nonlinear diffusion influenced by the surface Laplacian. As we can see, the values of the edge and local extrema detector are zero almost over whole testing data which causes that the original data are barely changed over the time steps.

4 Numerical experiments

In this section, we present two numerical experiments where we use all developed types of diffusion filtering, namely the linear diffusion, nonlinear Perona–Malik diffusion and nonlinear diffusion influenced by the surface Laplacian. The first experiment aims to demonstrate filtering of noise from the GOCE data, while showing advantages and disadvantages of each filtering model. The second experiment presents filtering of a satellite-only mean dynamic topography (MDT), where we try to reduce a typical stripping noise due to omission errors of the spherical harmonic approach. In this case we use the modified nonlinear diffusion influenced by the surface Laplacian to pre-filter local extrema of initial data and then we use the linear diffusion and nonlinear Perona–Malik diffusion to obtain improved MDT model.

4.1 Filtering of the GOCE data

In this experiment, we use our developed filters to reduce the noise from measurements of the GOCE satellite mission (ESA 1999), namely from the radial components T_{rr} of the gravity disturbing tensor available from the EGG_TRF_2 product. Our processed dataset represent data observed during June–July 2013. Due to the fact that the altitudes of observations vary considerably reaching differences more than 10 km between

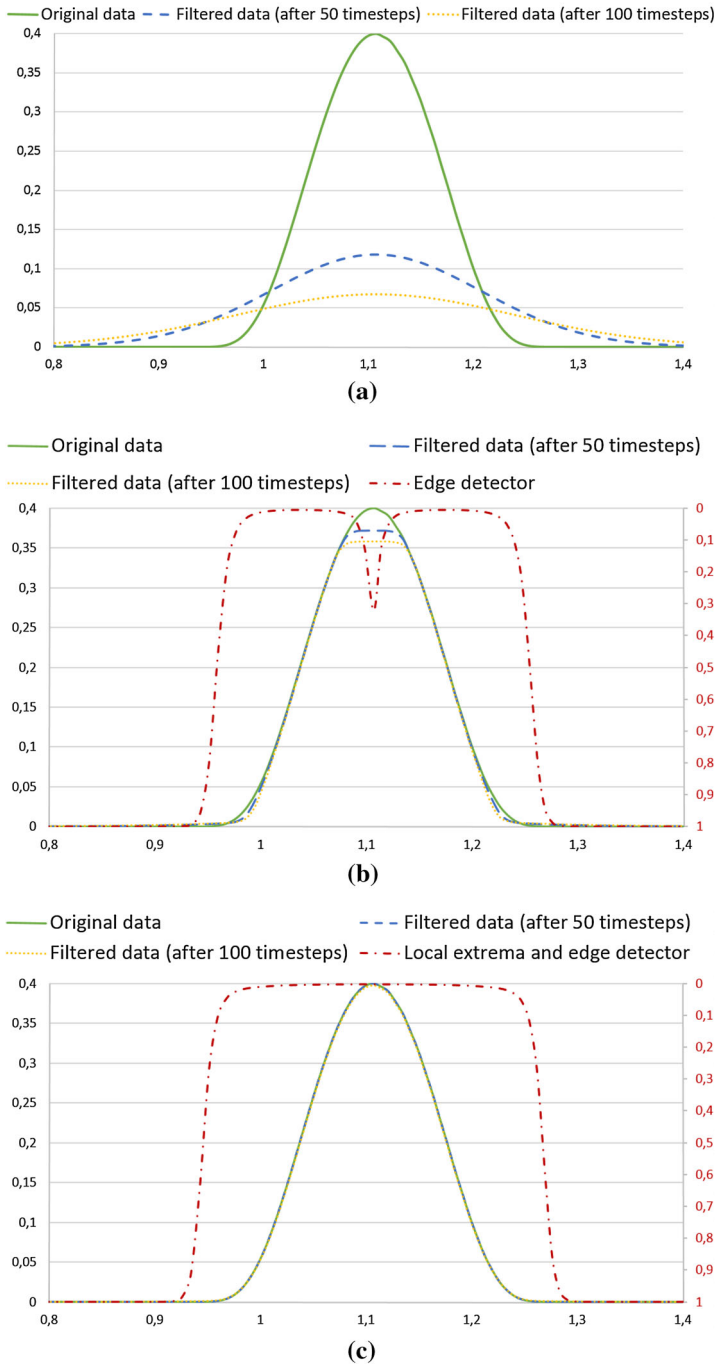


Fig. 13 Profiles of the solution of **a** the linear diffusion filtering, **b** the surface Perona–Malik diffusion filtering and **c** the nonlinear diffusion filtering influenced by the surface Laplacian. Subfigures **b**, **c** include also the plots of detector functions

“neighboring” points of observations (e.g. for distant observing time), we have reduced observed values from the observing altitudes into the reference altitude 245 km, for more details, see (Čunderlík 2015). It means that our computational domain represents a closed surface given by the constant altitude 245 km above a reference ellipsoid. For its discretization we use a very refined triangulation in order to capture a dense coverage of the processed GOCE data. Namely, an octahedral grid with the resolution of 0.05° is constructed to generate 3D positions of 12,960,002 nodes of the regular triangulation. In these nodes the values of the radial components T_{rr} reduced to the reference altitude are interpolated. The missing values in polar gaps (0.54 % of all nodes) are generated from the GOCO03S satellite-only model up to 250° (Mayer-Grr 2012). Such input data (see Fig. 14) are then subsequently filtered by the linear and nonlinear diffusion filtering method.

At first we use the linear diffusion to filter data. We use the time step τ according to Eq. (31). In this case $\tau = 4 \cdot 10^7$. The time step τ is so high because areas of covolumes are quantify in meters. The linear diffusion successfully has removed most of the noise, however, looking closely on the character of differences between the initial and filtered data after 1 and 10 time steps, we can see that the linear diffusion smooths out main structures of processed data. This drawback can be seen already since the first time step. A detail of differences over South America is depicted in Fig. 15.

To obtain a better solution we have used the regularized surface Perona–Malik model (see Sect. 3.1). At the beginning we need to set initial values of model parameters. The optimal parameter of the edge detector were tuned experimentally. The time interval for pre-smoothing by the linear diffusion was chosen as $\sigma = 2\tau$, so we use two steps of the linear diffusion with $\tau = 4 \cdot 10^7$. The reason for such an appropriate selection of the linear diffusion time interval is that we need to smooth the most of noise before we quantify the surface gradients. Before pre-filtering by the linear diffusion, high values of the surface gradients are almost everywhere. After applying the linear diffusion, the noise is removed (according to the previous part of the experiment) and values of the gradient preserves better structures of the Earth’s gravity field. Tuning

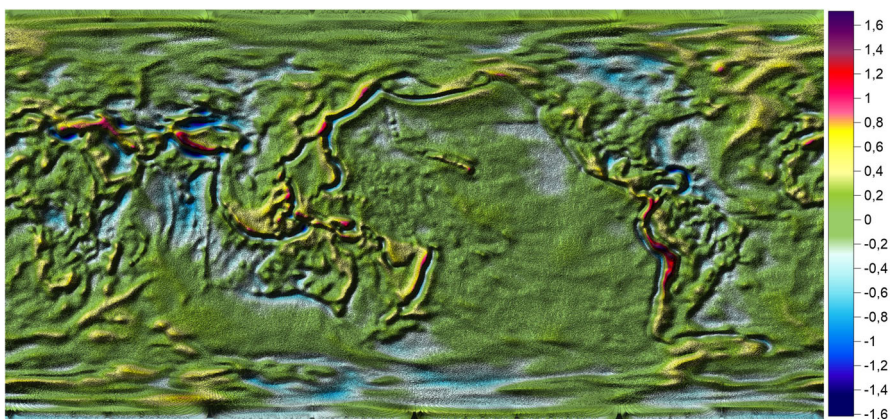


Fig. 14 Original GOCE measurement— T_{rr} component

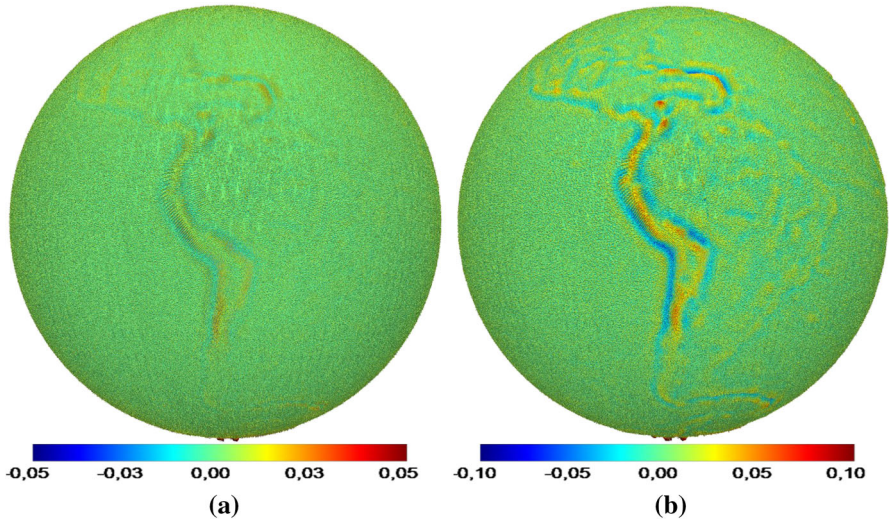


Fig. 15 Differences between initial data and solution after a 1 and b 10 time steps

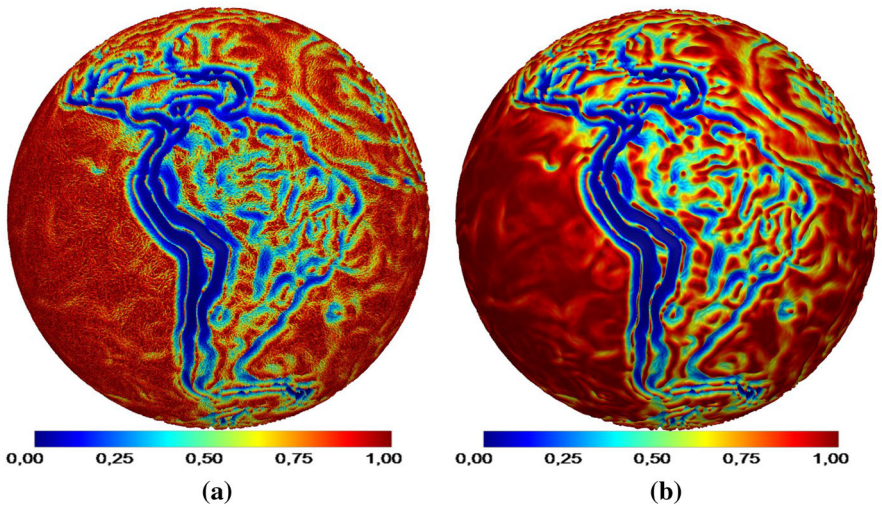
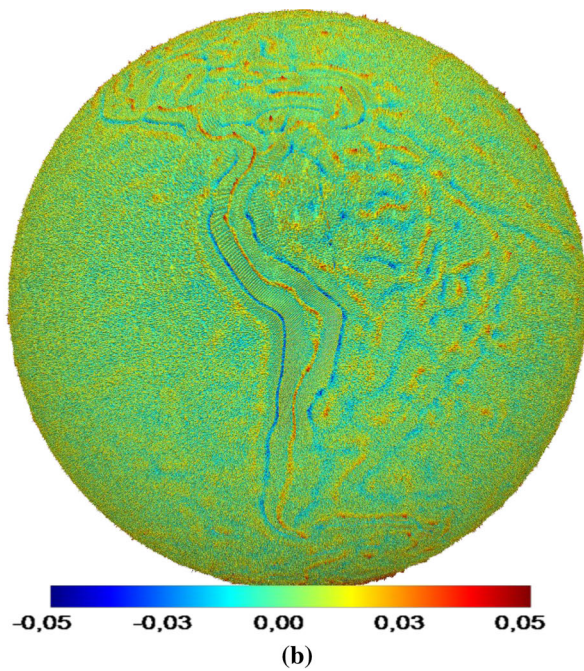
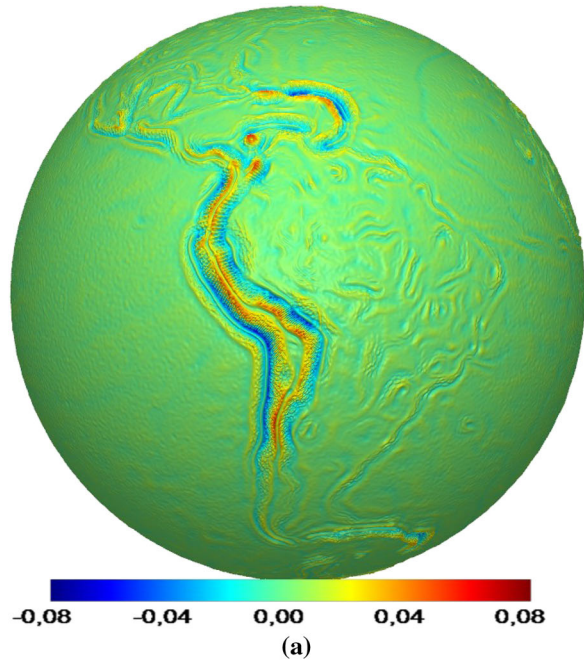


Fig. 16 Evolution of the edge detector function after a 1 and b 30 time steps

of the sensitivity coefficient K for the edge detector function, see Eq. (33), was based on values of the surface gradient from the solution u^σ . Experimentally (considering different sensitivity parameters) we estimate $K = 10^{12}$. The evolution of the edge detector function during filtering is depicted in Fig. 16.

To demonstrate advantages of the nonlinear Perona–Malik model, Fig. 17 shows differences between initial data and filtered data obtained after 30 time steps of the nonlinear Perona–Malik diffusion and differences between filtered data obtained by the linear diffusion after 10 time steps and filtered data obtained after 30 time steps by

Fig. 17 Differences between **a** filtered data obtained by the linear diffusion and filtered data obtained by the nonlinear Perona–Malik diffusion, **b** initial data and filtered data obtained by the nonlinear Perona–Malik model after 30 time steps



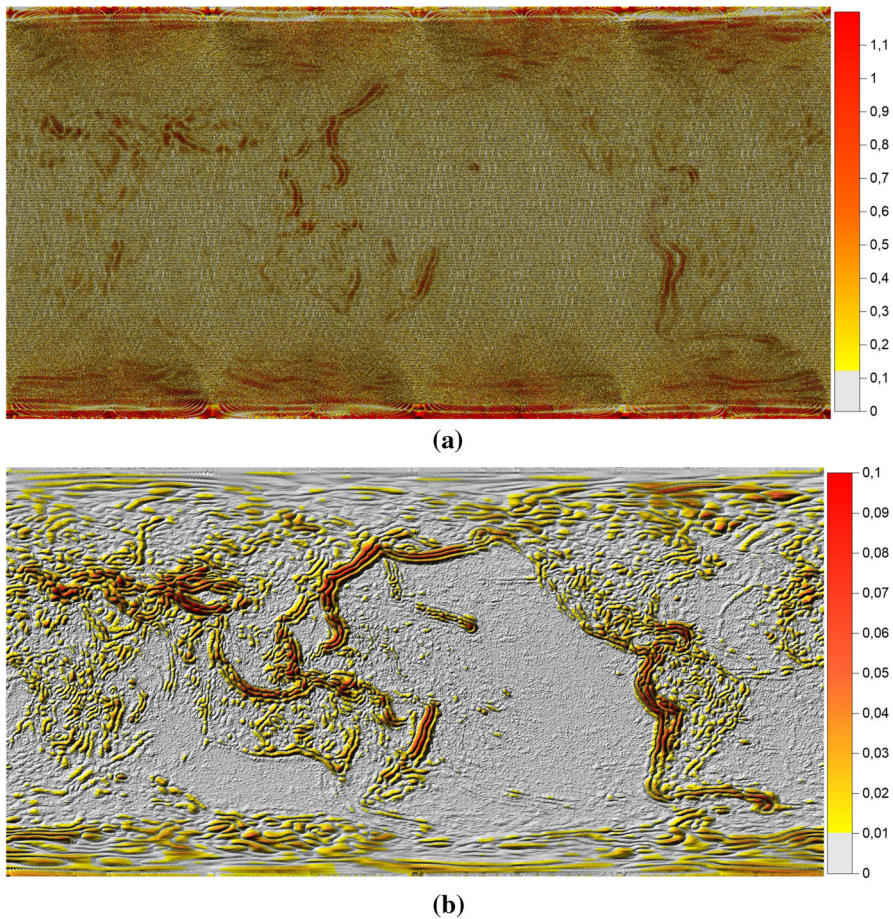


Fig. 18 The surface Laplacian of data **a** before the linear diffusion and **b** after pre-filtering by 20 time steps of the linear diffusion (values on the scale are 10^{-9})

the nonlinear Perona–Malik diffusion. We can see how the linear diffusion smooths also main structures, especially in areas where the gravity field is changing sharply, while the nonlinear diffusion preserves them and effectively reduce the noise (Fig. 17a). The only problem is in areas of the local extrema of processed data. In Fig. 17b, we can see that maxima of differences are along the main range of Andes and along its western and eastern edges. This area represents local extrema of initial data. Such a behaviour can be seen also in other regions indicating that the Perona–Malik model slightly smooths also local extrema, which represent important structures in the filtered data.

To void such smoothing we use the nonlinear diffusion influenced by the surface Laplacian. As input parameters σ^1 and K_1 for the edge and local extrema detector function (see Eq. 38) we use the same parameters as we use in the previous part. The tuning of the time interval σ^2 for pre-filtering by the linear diffusion is in this case very important. Figure 18 depicts surface Laplacians before the linear diffusion and after

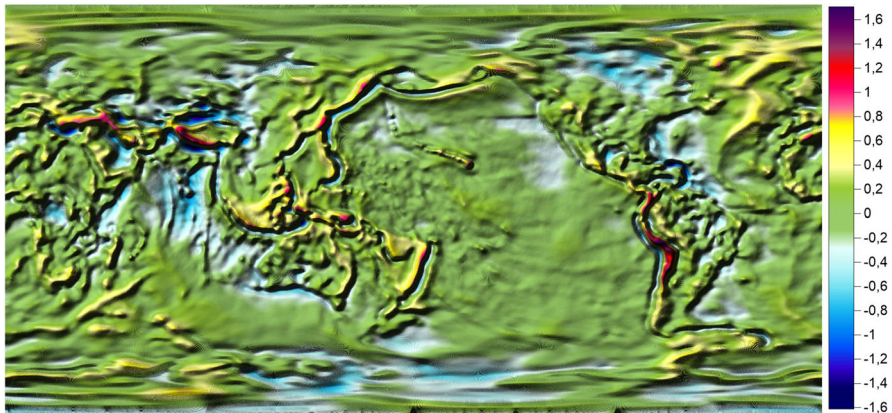


Fig. 19 Solution of the nonlinear diffusion influenced by the surface Laplacian after 30 time steps

20 τ time steps of the linear diffusion (values on the scale are 10^{-9}). We can see that before pre-filtering the surface Laplacian of processed data is high everywhere, even in noisy areas. After the significant pre-filtering by the linear diffusion, the surface Laplacian reflect structures of the gravity field. To obtain an appropriate solution we need to be sure that most of the noise is smoothed and then we can compute the surface Laplacian. From the initial surface Laplacian we estimate the sensitivity coefficient $K_2 = 10^{22}$.

Figure 19 presents final filtered data after 30 time steps using the nonlinear diffusion influenced by the surface Laplacian.

A remarkable advantage of the nonlinear approach can be seen in the Fig. 20. It shows differences between the filtered data obtained by the nonlinear Perona–Malik diffusion and those obtained by the nonlinear diffusion influenced by the surface Laplacian. Both solutions are obtained after 30 time steps.

Figure 20a depicts differences between the initial and filtered data obtained by the nonlinear diffusion influenced by the surface Laplacian. Comparing them the differences depicted in Figs. 15 and 17b, we can see that new differences (Fig. 20a) resembles white noise more closely, so we can assume that the nonlinear diffusion filtering influenced by the surface Laplacian preserves all the important structures including local extrema of the initial GOCE data.

4.2 Filtering of the satellite-only MDT

In this experiment, we present filtering of the GOCE-based satellite-only MDT. The satellite-only MDT as our initial data are given as a combination of the DTU13 mean sea surface model (Andersen et al. 2013) and the geoid model evaluated from the GO_CONS_GCF_2_DIR_R5 up to 300° (Bruinsma et al. 2003). Such a MDT model is significantly affected by the stripping noise due to omission errors of the spherical harmonics approach used for the geoid modelling (Fig. 21). Our aim is to reduce this stripping noise while to preserve important gradients that correspond to the main ocean

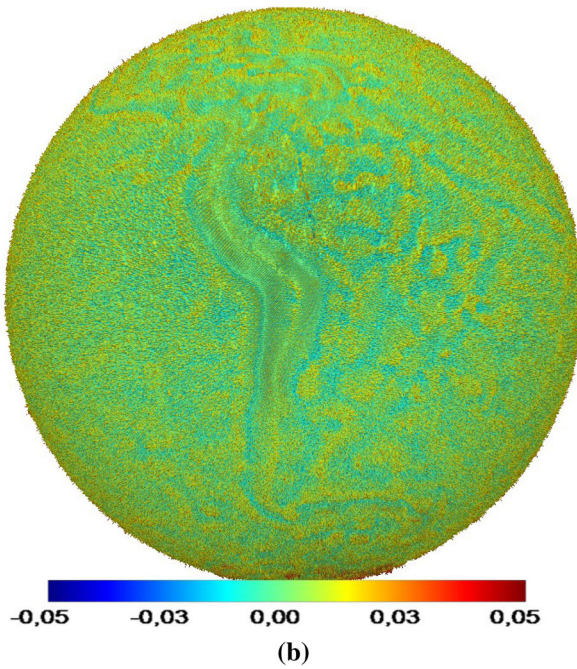
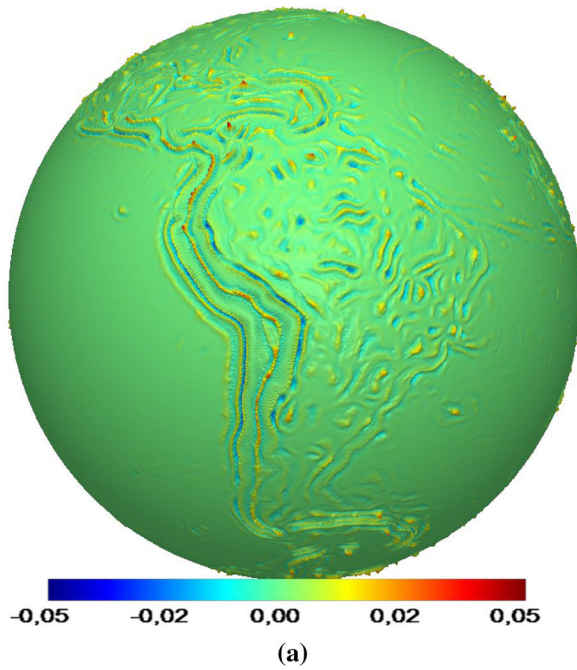


Fig. 20 Differences between **a** filtered data obtained by the nonlinear diffusion influenced by the surface Laplacian and filtered data obtained by the nonlinear Perona–Malik diffusion, **b** initial data and filtered data obtained by the nonlinear diffusion influenced by the surface Laplacian model after 30 time steps

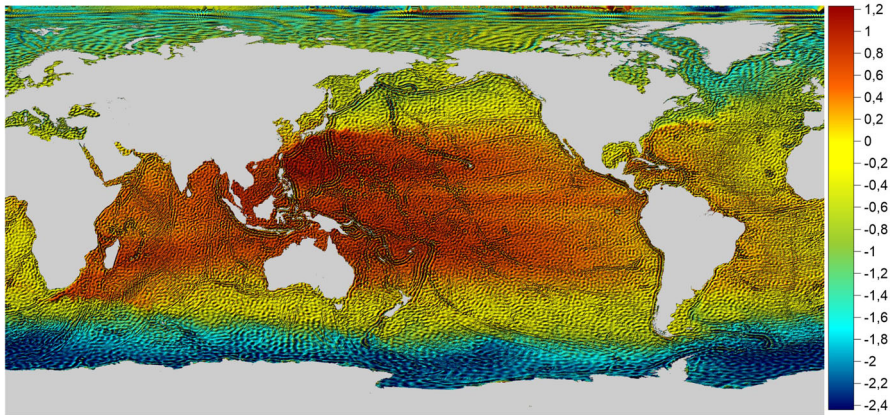


Fig. 21 Initial satellite-only MDT model

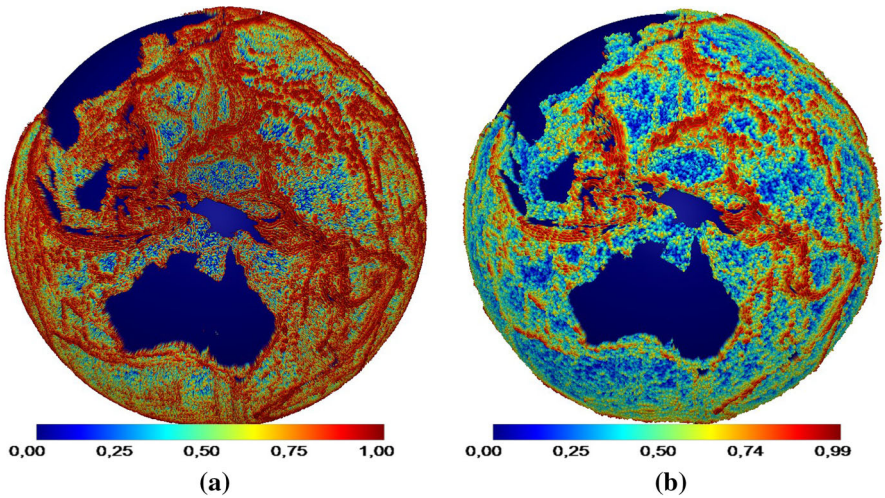


Fig. 22 Values of the detector function for the nonlinear diffusion influenced by the surface Laplacian: **a** for initial values, and **b** for values after 10 time steps

geostrophic surface currents. In this experiment we use the same octahedral grid with the resolution of 0.05° , but in this case generated 3D positions of 12,960,002 nodes of the regular triangulation are on the surface of the reference ellipsoid.

Taking into account the nature of noise, at first we have reduced the highest peaks of this noise. To achieve this we have used a modification of the nonlinear diffusion influenced by the surface Laplacian (see Sect. 3.2). This modification is based on the detector function in the form

$$g(|\Delta_s u^{\sigma^2}|) = \frac{1}{1 + \frac{1}{K_2 |\Delta_s u^{\sigma^2}|^2}}, \quad K_2 \geq 0. \quad (43)$$

Fig. 23 Differences between initial data and pre-filtered data by the modification of the nonlinear diffusion equation influenced by the surface Laplacian obtained after 50 time steps

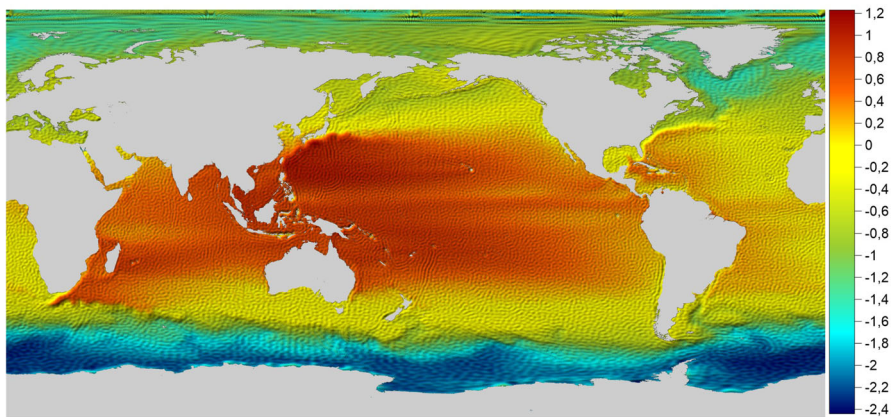
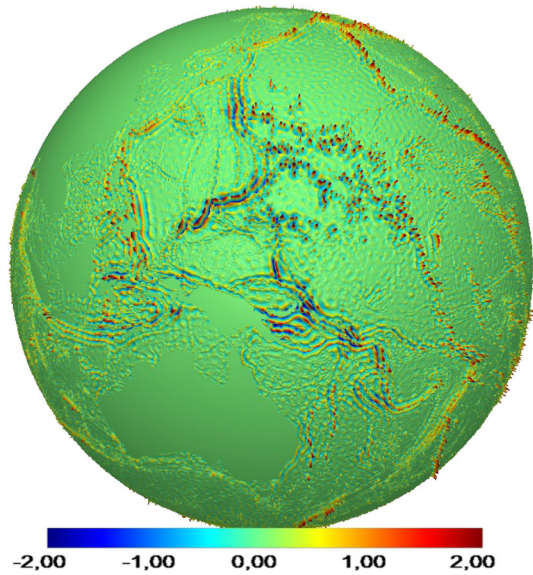
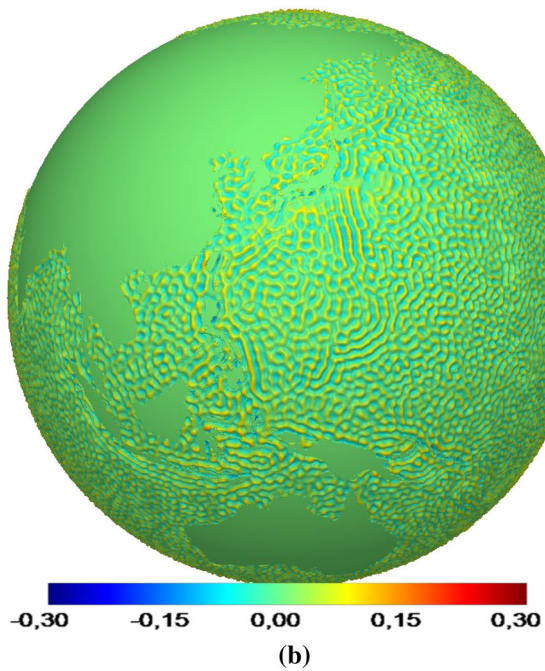
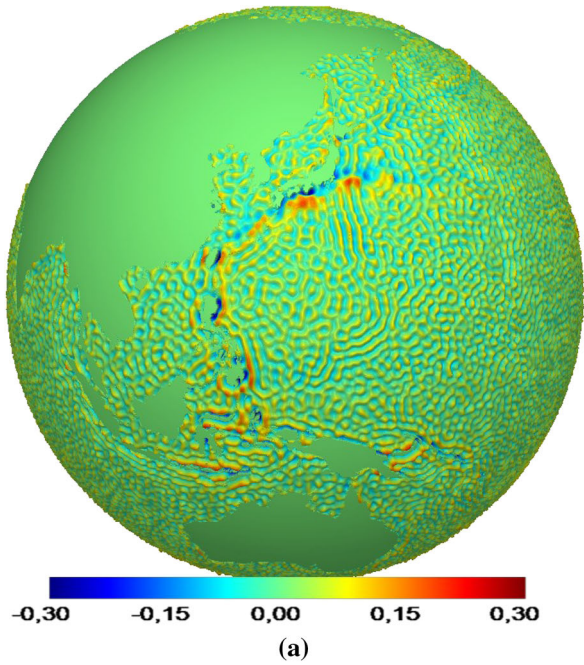


Fig. 24 Pre-filtered satellite-only MDT model

The detector function in this form speed up diffusion process in nodes with the highest surface Laplacian. Those nodes represent highest values of striping noise (Fig. 21). Pre-filtering by the nonlinear diffusion reduces local extrema of noise but at the same time preserves original structures from initial data. In the pre-filtering process, we use the model parameters $\sigma^2 = \tau = 4 \cdot 10^7$ and $K_2 = 10^{19}$. Figure 22 shows evolution of values of the detector function after 1 and 10 time steps. Figure 23 depicts differences between the initial and pre-filter data in detailed area after 50 time steps. If we look closely on Figs. 22 and 23 we can see that detector function successfully detects high values of the surface Laplacian and reduce only those values.

Fig. 25 Differences between initial data and **a** filtered data obtained by the linear diffusion, **b** filtered data obtained by the nonlinear Peron–Malik diffusion



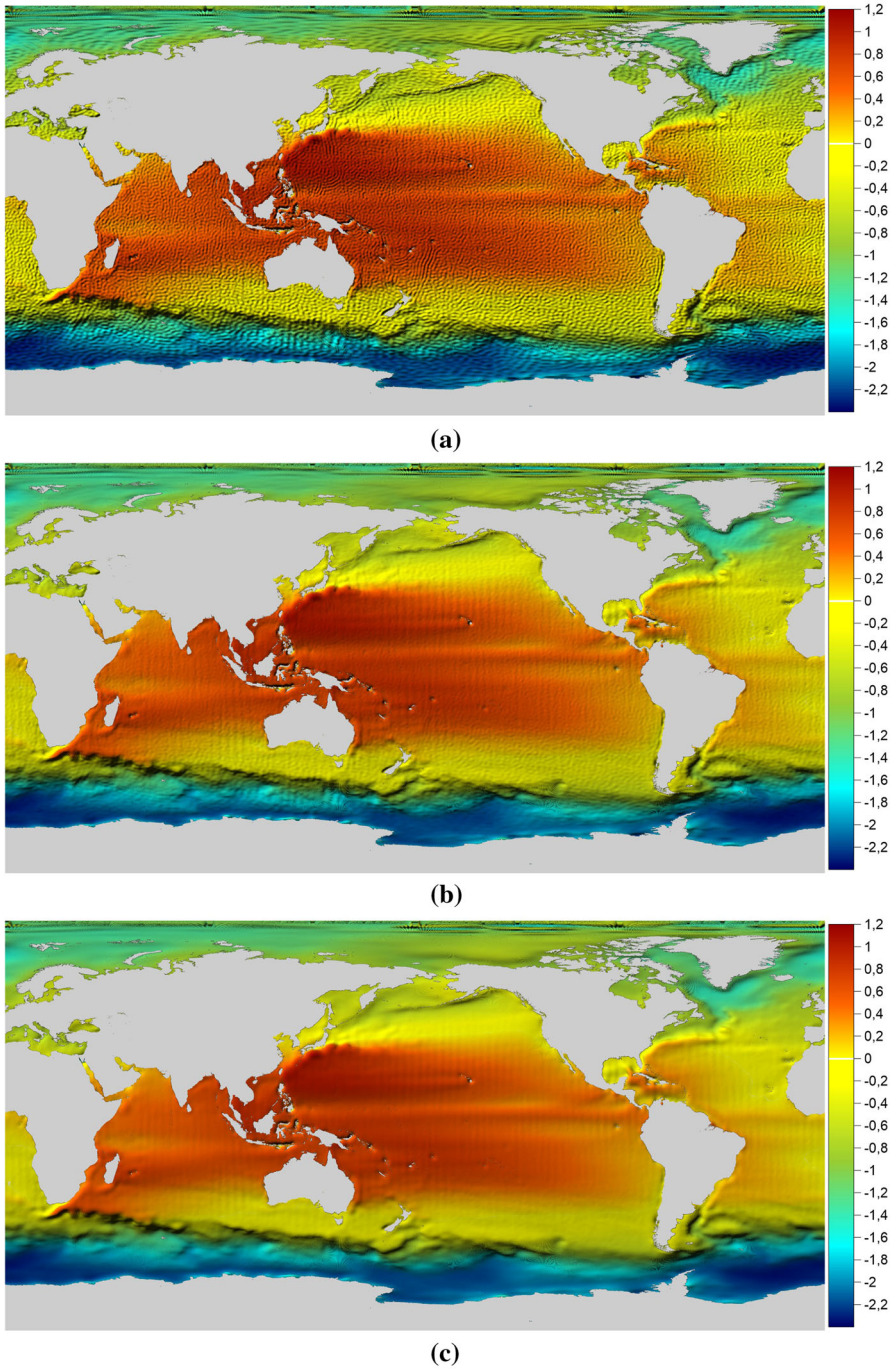


Fig. 26 Filtered MDT data obtained by the nonlinear Perona–Malik diffusion after **a** 10, **b** 50 and **c** 100 time steps

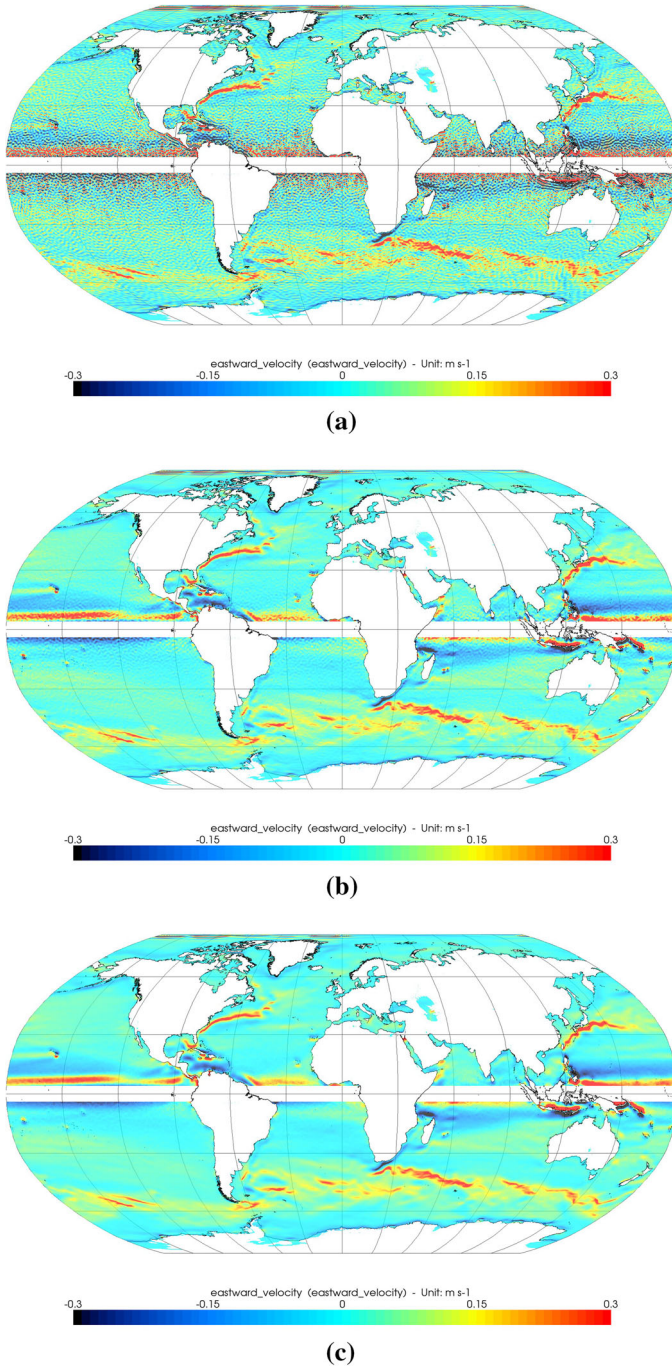
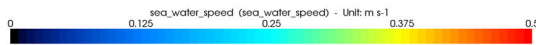
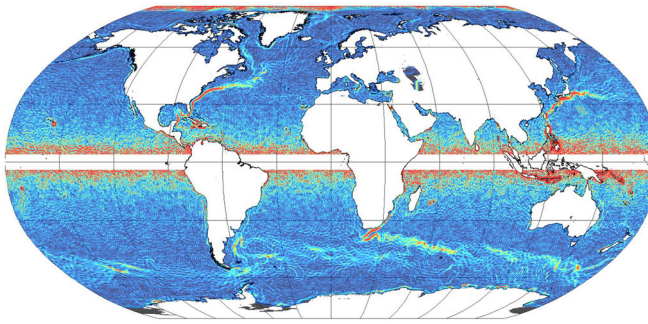
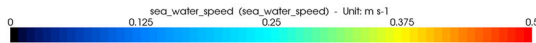
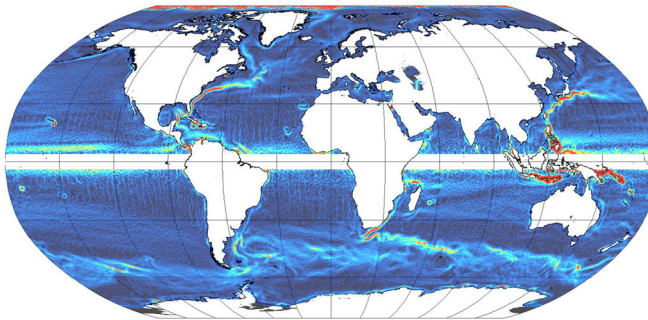


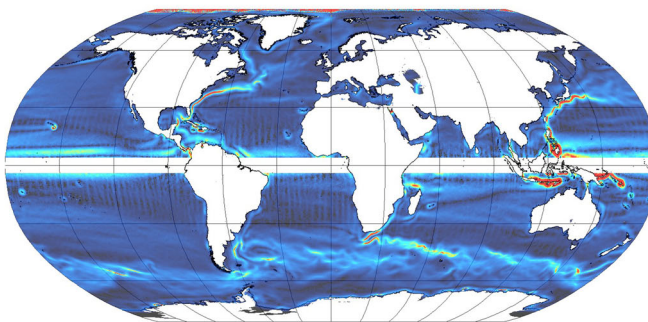
Fig. 27 Derived ocean geostrophic surface currents (zonal velocity components) obtained from data filtered by the nonlinear Perona–Malik filtering after **a** 10, **b** 50 and **c** 100 time steps



(a)



(b)



(c)

Fig. 28 Derived ocean geostrophic surface currents (sea water speed) obtained from data filtered by the nonlinear Perona–Malik model after **a** 10, **b** 50 and **c** 100 time steps

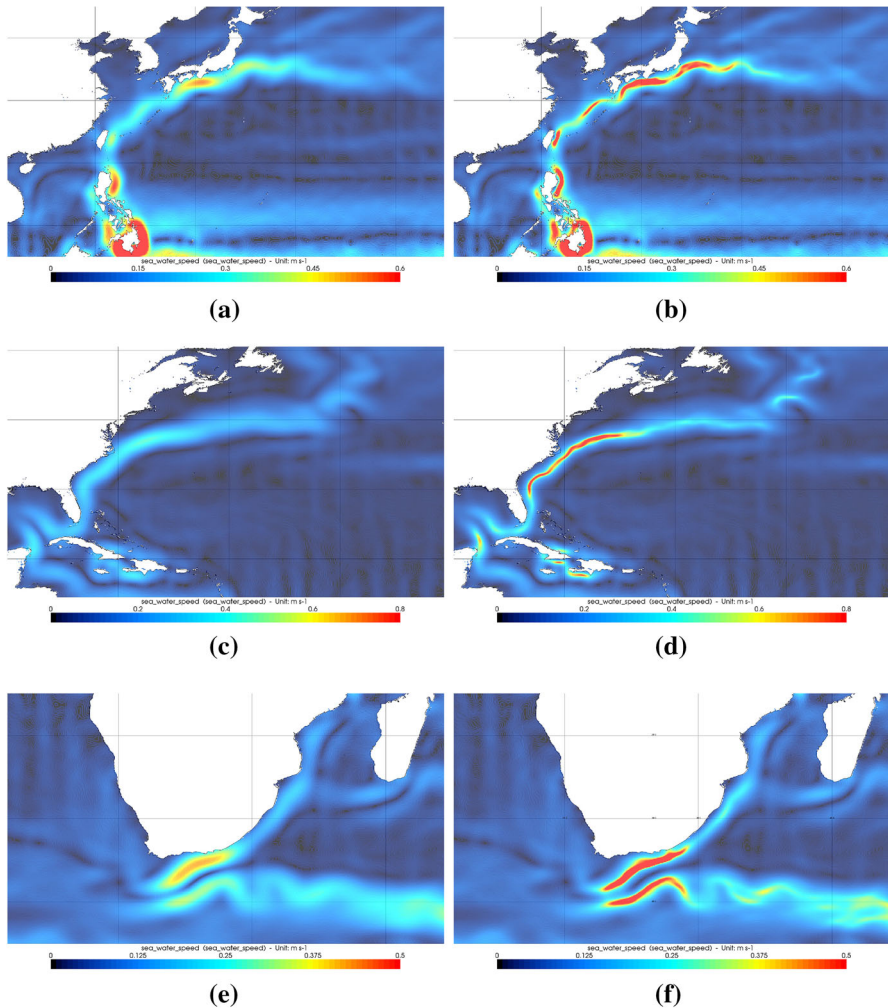


Fig. 29 Sea water speed from filtered data obtained by the linear diffusion and filtered data obtained by the nonlinear diffusion in regions of **a, b** Kuroshio, **c, d** Gulf Stream, **e, f** Agulhas

Figure 24 depicts the pre-filtered data after 50 time steps. This data have been used for a final filtering. To remove the remaining noise we have used the linear diffusion (see Sect. 2) and the nonlinear Perona–Malik diffusion (see Sect. 3.1). In case of the linear diffusion we use similar time step $\tau = 4 \cdot 10^7$ as we have used in pre-filtering. This time step is also used for the nonlinear Perona–Malik diffusion. The estimated sensitivity coefficient $K_1 = 10^{10}$ for the edge detector function (Eq. 33) is in relation to the surface gradients of processed data.

To compare the linear diffusion and the nonlinear diffusion, differences between the initial and filtered data after 100 time steps in the area of the Kuroshio current are

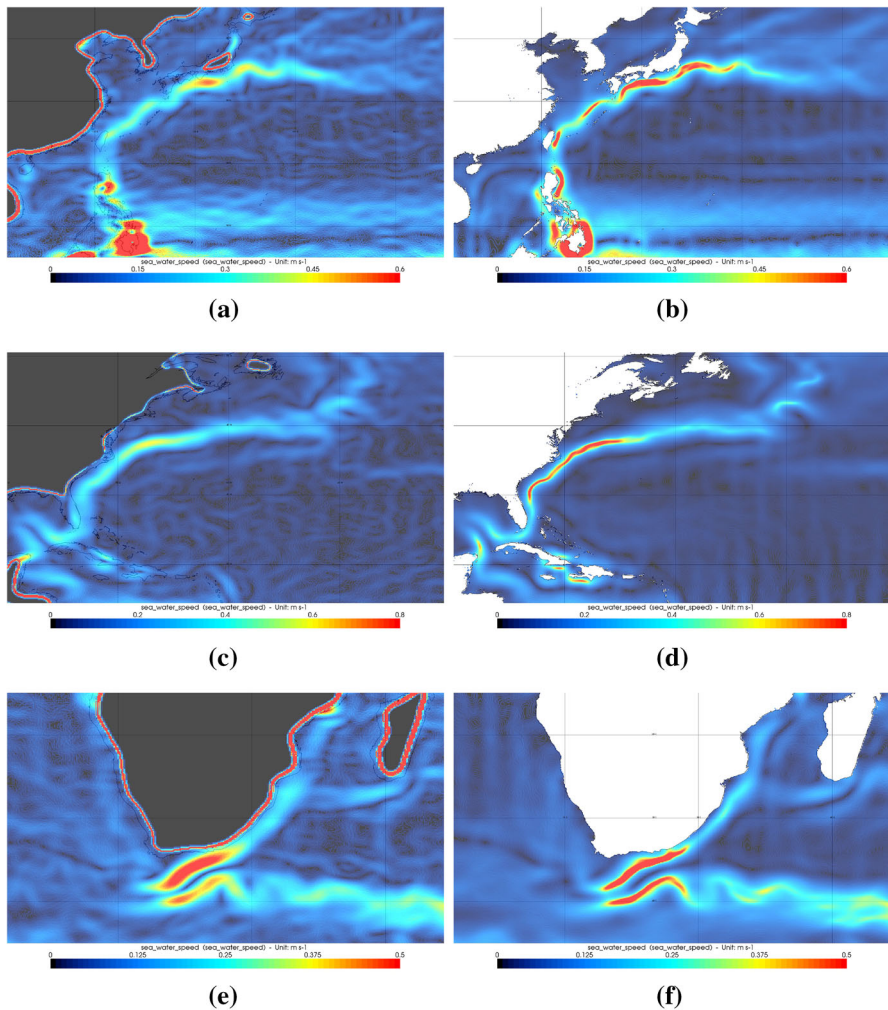


Fig. 30 Sea water speed from DTU13MDT_5MIN and from filtered data obtained by the nonlinear diffusion in regions of **a, b** Kuroshio, **c, d** Gulf Stream, **e, f** Aghulas

depicted in Fig. 25. We can see how the detector function of the nonlinear diffusion preserve the signal from original data, while the linear diffusion distorted it.

Figure 26 depicts filtered data after 10, 50 and 100 time steps. In these iterative steps we also have derived the ocean geostrophic surface currents, using GUT, namely their zonal velocity components (Fig. 27) as well as their sea water speed (Fig. 28).

Figure 29 shows filtered data obtained by the nonlinear diffusion together with those obtained by the linear diffusion. Figure 30 similarly, with DTU13MDT_5MIN model. Both Figures depict sea water speed in details in regions of the main currents like the Gulf Stream, Kuroshio or Aghulas current. We can see that currents generated from data filtered by the linear diffusion represent weaker signal, on the contrary those

generated from the nonlinear diffusion capture signal better. Stronger signal indicate that the nonlinear diffusion preserve gradient from the initial data.

Consequently, the nonlinear diffusion filtering on a closed surface using the regularized surface Perona–Malik model, prefiltered by nonlinear diffusion influenced by the surface Laplacian, seems to be an efficient tool for filtering the satellite-only MDT. The pre-filtering by the modification of the nonlinear diffusion influenced by the surface Laplacian have reduced the highest values of the noise. Then the opportunity for adaptive smoothing according to the main gradients in the filtered data allows us to reduce the stripping noise efficiently while preserving important gradients that correspond to the main ocean geostrophic surface currents. Derived velocities of the ocean geostrophic surface currents have clearly shown that preserving the important gradients by the nonlinear filtering have resulted in much stronger signal than in case of the linear filtering whose uniform smoothing effect also smoothes these structures.

Acknowledgments This work was supported by Grants APVV-15-0522, VEGA 1/0608/15 and VEGA 1/0714/15.

References

- Alvarez, L., Morel, J.M.: Formalization and computational aspects of image analysis. *Acta Numer.* **3**, 1–59 (1994)
- Alvarez, L., Guichard, F., Lions, P.L., Morel, J.M.: Axioms and fundamental equations of image processing. *Arch. Ration. Mech. Anal.* **123**, 200–257 (1993)
- Andersen, O., Knudsen, P., Stenseng, L.: The DTU13 global mean sea surface from 20 years of satellite altimetry. Presented at the IAG Scientific Assembly in Potsdam (2013)
- Bruinsma, S.L., Forste, C., Abrikosov, O., Marty, J.C., Rio, M.H., Mulet, S., Bonvalot, S.: The new ESA satellite-only gravity field model via the direct approach. *Geophys. Res. Lett.* **40**, 3607–3612 (2003)
- Caselles, V., Morel J.M., Sapiro, G., Tannenbaum, A.: Introduction to the special issue on partial differential equations and geometry-driven diffusion in image processing and analysis, *IEEE Trans. Image Process.* **7**(3), 269–373 (1998)
- Catté, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.* **29**, 182–193 (1992)
- Crandall, M.G., Ishii, H., Lions, P.L.: User's guide to viscosity solutions of second order partial differential equations. *Bull. Am. Math. Soc. (NS)* **27**, 1–67 (1992)
- Čunderlík, R., Mikula, K., Tunega, M.: Nonlinear diffusion filtering of data on the Earth's surface. *J. Geodesy* **87**(2), 143–160 (2012)
- Čunderlík, R.: Precise modelling of the static gravity field from GOCE second radial derivatives of the disturbing potential using the method of fundamental solutions. In: International Association of Geodesy Symposia. Springer (2015). doi:[10.1007/1345_2015_211](https://doi.org/10.1007/1345_2015_211)
- DTU, DTU13MDT_5MIN. https://ftp.space.dtu.dk/pub/DTU13/5_MIN/
- Dziuk, G., Elliott, C.M.: Finite elements on evolving surfaces. *IMA J. Numer. Anal.* **27**(2), 262–292 (2007)
- Dziuk, G., Elliott, C.M.: Finite element methods for surface PDEs. *Acta Numer.* **22**, 289396 (2013)
- ESA: GOCE User Toolbox GUT. <https://earth.esa.int/web/guest/software-tools/gut/>
- ESA: Gravity field and steady-state ocean circulation mission. Report for mission selection of the four candidate earth explorer missions, ESA SP-1233(1). ESA Publications Division: ESTEC. Noordwijk, The Netherlands (1999)
- Eymard, R., Gallouet, T., Herbin, R.: Finite Volume Methods, *Handbook of Numerical Analysis*, vol. 7, pp. 713–1020 (2003)
- Faure, E. et al.: A workflow to process 3D+time microscopy images of developing organisms and reconstruct their cell lineage. *Nat. Commun.* **7**. doi:[10.1038/ncomms9674](https://doi.org/10.1038/ncomms9674) (2016) (Article number: 8674)
- Gilbarg, D., Trudinger, N.S.: Elliptic partial differential equations of second order. Springer, Berlin (1988)
- Kichenassamy, S.: The Perona–Malik paradox. *SIAM J. Appl. Math.* **57**(5), 1328–1342 (1997)

- Koenderink, J.: The structure of images. *Biol. Cybern.* **50**, 363–370 (1984)
- Kriva, Z., Mikula, K., Peyrieras, N., Rizzi, B., Sarti, A., Stasova, O.: 3D early embryogenesis image filtering by nonlinear partial differential equations. *Med. Image Anal.* **14**(4), 510–526 (2010)
- Lions, P.L.: Axiomatic derivation of image processing models. *Math. Models Methods Appl. Sci.* **4**, 467–475 (1994)
- Mayer-Grr, T., The GOCO Consortium: The New Combined Satellite Only Model GOCO03s. Presented at the GGHS-2012 in Venice, Italy, October 9–12 (2012)
- Mikula, K., Ramarosy, N.: Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing. *Numer. Math.* **89**(3), 561–590 (2001)
- Nielsen, M., Johansen, P., Olsen, O.F., Weickert, J. (eds.): *Scale-Space Theories in Computer Vision*. Lecture Notes in Computer Science, vol. 1682. Springer, Berlin (1999)
- Nitzberg, M., Shiota, T.: Nonlinear image filtering with edge and corner enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(8), 826–833 (1992)
- Osher, S., Fedkiw, R.: *Level Set Methods and Dynamic Implicit Surfaces*. Springer, Berlin (2003)
- Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. *IEEE Computer Society Workshop on Computer Vision, Proc* (1987)
- Romeny, B.M.T.H. (ed.): *Geometry Driven Diffusion in Computer Vision*. Kluwer, Dodrecht (1994)
- Sapiro, G.: *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, Cambridge (2001)
- Sethian, J.A.: *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science*. Cambridge University Press, New York (1999)
- Weickert, J.: *Anisotropic Diffusion in Computer Vision*. Teubner-Stuttgart, Stuttgart (1998)
- Witkin, A.P.: Scale-space filtering. In: *Proceedings of Eighth International Conference on Artificial Intelligence*, vol. 2, pp. 1019–1022 (1983)
- Young, D.M.: *Iterative Solution of Large Linear Systems*. Academic Press, London (1971)