Computers & Fluids 76 (2013) 33–49

Contents lists available at SciVerse ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

The independent set perturbation method for efficient computation of sensitivities with applications to data assimilation and a finite element shallow water model

F. Fang [a], C.C. Pain [a], I.M. Navon [b,*], D.G. Cacuci [c], X. Chen [d]

[a] Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London SW7 2BP, UK
[b] Department of Scientific Computing, Florida State University, Tallahassee, FL 32306-4120, USA
[c] Department of Nuclear Engineering, North Carolina State University, Raleigh, NC 27695-7909, USA
[d] Lawrence Livermore National Laboratory Center for Applied Scientific Computing Livermore 94550, USA

ARTICLE INFO

Article history:
Received 24 December 2011
Received in revised form 30 December 2012
Accepted 20 January 2013
Available online 16 February 2013

Keywords:
Variational methods
Shallow water
Optimisation

ABSTRACT

An adjoint model for a 2D Galerkin/Petrov–Galerkin finite element (FE) shallow water (S-W) model is developed using the Independent Set Perturbation (ISP, [40]) sensitivity analysis. Its performance in a full 4-D Var setup with a limited area shallow water equations model is assessed by comparing with the adjoint model derived by the automatic differentiation approach (TAMC), where it is used for optimising the initial conditions. It is shown that the ISP sensitivity analysis provides a very simple approach of forming the adjoint code/gradients/differentiation of discrete forward models (even complex governing equations, discretization methods and non-linear parameterizations) and is realised using a graph colouring approach combined with a perturbation method. Importantly, the adjoint is automatically updated as the forward code continues to be developed. In the test cases, it is shown that the adjoint model using the ISP sensitivity analysis can achieve the accuracy of traditional adjoint models derived by the automatic differentiation method (TAMC) [31]. Further comparison shows that the CPU time required for running the adjoint model using the ISP sensitivity analysis is much less than that required for the automatic differentiation derived adjoint model since the ISP derived adjoint CPU time scales linearly with the problem size.

The ISP sensitivity analysis is further applied to a highly non-linear Petrov–Galerkin FE model. The perturbation size used in deriving the tangent linear model with the ISP sensitivity analysis method is then optimised and the resulting approach used to assimilate both sparse (more realistic) and dense observational data for optimising the initial conditions. A simple first order formula is developed to calculate the perturbation size for each variable, at each node and time level. By applying the ISP sensitivity method to an intermediate complexity model (a shallow water model) this paper outlines steps towards applying the approach to data assimilation (DA) problems involving realistic complex models.

1. Introduction

The Independent Set Perturbation (ISP, [40]) sensitivity analysis is a method of differentiating finite element or control volume models. The method is derived using a graph colouring approach (forming independent sets of variables) combined with a perturbation method to obtain gradients in numerical discretization. Using the ISP sensitivity analysis method the adjoint code becomes simple to derive even with complex governing equations, discretization methods and non-linear parameterizations. Importantly, this greatly reduces the effort required to implement the tangent linear model (differentiation of the discrete forward model) and its adjoint, and maintains the adjoint code as the forward model code continues to evolve, a desirable trait in operational implementations.

Cacuci et al. [1] was the first to originally develop variational/perturbation methods using adjoint functions, based on Frechet-derivatives, for performing systematic and efficient sensitivity analysis of large-scale continuous and/or discrete linear and/or non-linear systems, possibly with bifurcations, critical points and global optimisation. Shortly thereafter, this method was further generalised using Gateaux differential operators [9] for a non-linear operator of the system response along an arbitrary direction. Adjoint sensitivity analysis methods are widely applied to chemical kinetics [2], optimisation [3,8,15], ocean and atmospheric sciences [4–7,31,44–48] and nuclear science [10]. Furthermore, Bischof et al. developed the pseudo-adjoint method for efficient computation of Jacobian matrices using colouring approaches [26,29].

* Corresponding author.
E-mail address: inavon@fsu.edu (I.M. Navon).
URL: http://amcg.ese.imperial.ac.uk (I.M. Navon).

0045-7930/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compfluid.2013.01.025

In Fang et al. [40] the ISP sensitivity analysis approach was tested only for sensitivity purposes while in the present paper the method is applied to a full 4-D Var (optimisation of initial conditions) with a finite element discretization of a shallow water equations model over a limited area domain and its adjoint. This work focuses on comparing the performance of the ISP derived adjoint with Automatic Differentiation (AD) methods as well as optimising the size of perturbations used within the ISP sensitivity analysis.

While the usual AD tool has seen great advances, it remains a challenging task to derive large scale adjoint models [11]. Automatic differentiation tools such as ADOL-C [14], GRESS [42], Odyssée [18], and TAMC [19,20], ADIFOR [26,27], TAPENADE [17] and TAF [16], implement the reverse mode of automatic differentiation in an automated fashion. In addition, for non-linear problems, the implementation of the reverse mode requires one to store or recompute all forward solution variables. This can incur a prohibitively large CPU or memory cost especially for three dimensional problems. A checkpointing scheme proposed by Griewank et al. [12,13], and Charpentier [28] provides a way of overcoming these shortcomings. This approach is being adopted in reverse mode tools but the fact that potentially many values need to be stored and/or recomputed implies that the reverse mode requires considerable storage requirements to achieve the desired low floating point complexity. This apart from the fact that the adjoint code provided by AD tools still requires expert tuning by the user prior to being ready for implementation.

More recently, reduced order approaches have been introduced into 4-D variational data assimilation systems [21–25]. An approximate linear reduced model can be obtained by projecting the forward model onto a reduced subspace. Compared to the classical variational method, the adjoint of the tangent linear model is replaced by the adjoint of a linear reduced forward model [22]. The minimisation process of the related cost function is carried out in reduced subspace and hence reduces the computational cost albeit subject to limitations.

The ISP adjoint method [40] bears some similarity to the pseudo-adjoint method of Bischof et al. [26] in that it efficiently implements exploitation of the Jacobian sparsity associated with the non-linear operator (advancing the system state using graph colouring strategies) in typical stencil-based computations. It also has some similarity to conceptual ideas of Christakopoulos et al. [30]. However it differs in essential algorithmic parts from above approaches as outlined below:

- It is suitable for any forward discrete schemes while pseudo-adjoint methods only tackle explicit schemes e.g. leap frog time differencing scheme.
- Implementation of adjoint codes using the ISP sensitivity analysis method involves only the matrix vector multiplication of the forward model.
- Complex models can be easily differentiated as long as the model can perform matrix vector multiplications involving the forward model.

The accuracy of the ISP sensitivity analysis can be critically dependent on the perturbation size of a number of variables and controls for deriving the tangent linear model [40]. Two approaches are developed for the choice of the perturbation size. These result in a different perturbation size for each finite element solution variable at each time level.

The remainder of this paper is structured as follows: The finite element shallow water model and its discretisations (including both Galerkin and Petrov/Galerkin formulations) are described in Sections 2 and 3. The derivation of the discrete adjoint model using ISP sensitivity approach and the implementation of the adjoint finite element shallow water model are discussed in Sections 4 and 5. The formulae for determining the non-uniform perturbation size used in deriving the tangent linear model is described in Section 6. In Section 7, the above adjoint model is applied for optimising the initial conditions (by data assimilation of observations distributed in space and time) of the shallow water equations model. A comparison of accuracy of the adjoint models derived by the ISP sensitivity analysis and automatic differentiation approach (TAMC) is carried out. Discussion, future perspectives and conclusions are provided in Sections 8 and 9.

2. Shallow water (S-W) equations

The shallow-water equations model is one of the simplest forms of the equations of motion for incompressible fluid for which the depth is relatively small compared to the horizontal dimensions. These equations can be used to help determine the horizontal structure of the atmosphere or oceans. They describe the evolution of an incompressible fluid in response to gravitational and rotational accelerations (see Tan [34], Vreugdenhil [35] and Galewsky et al. [36]). The shallow-water equations in a 2D domain $\Omega = (0,L) \times (0,D)$ can be written as:

$$\frac{\partial \phi}{\partial t} + \frac{\partial (\phi u)}{\partial x} + \frac{\partial (\phi v)}{\partial y} = 0, \tag{1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial \phi}{\partial x} + fv = 0, \tag{2}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{\partial \phi}{\partial y} - fu = 0, \tag{3}$$

where u and v are the velocity components in the x and y axis respectively, $\phi = gh$ is the geopotential height, h is the depth of the fluid and g is the acceleration of gravity. The scalar function f is the Coriolis parameter defined by the β-plane ($\beta = 1.5 \times 10^{-11}$) approximation:

$$f = \hat{f} + \beta\left(y - \frac{D}{2}\right). \tag{4}$$

The Coriolis parameter

$$\hat{f} = 2\omega \sin\theta, \tag{5}$$

is defined at a mean latitude θ_0, here $\hat{f} = 10^{-4}$, where ω is the angular velocity of the earth's rotation and θ is latitude. The shallow-water equations require specification of appropriate initial and boundary conditions. An initial condition is imposed as:

$$w(x,y,0) = \varphi(x,y), \tag{6}$$

where state variables are $w = w(x,y,t) = (u(x,y,t), v(x,y,t), \phi(x,y,t))$ with periodic boundary conditions are assumed in the x-direction:

$$w(0,y,t) = w(L,y,t), \tag{7}$$

while solid wall boundary condition in the y-direction is:

$$v(x,0,t) = v(x,D,t) = 0. \tag{8}$$

The geopotential in $\varphi(x,y)$ will be specified later in the numerical experiments.

3. Discrete system of shallow water equations

The discretisations of the finite element shallow water model are described in this section. Both Galerkin and Petrov/Galerkin finite-element methods (FEMs) are employed. The evolutionary equations of continuity and momentum are coupled at each time step using an extrapolated Crank–Nicolson method to quasi-linearise the nonlinear advective terms.

3.1. The finite-element model formulation at one time level

In the formulation of the Finite-Element model (FEM) variables in (1)–(3), can be written as [41,32]:

$$u=\sum_{k=1}^{N}u_k(t)V_k(x,y),\quad v=\sum_{k=1}^{\mathcal{N}}v_k(t)V_k(x,y),$$
$$\phi=\sum_{k=1}^{N}\phi_k(t)V_k(x,y), \tag{9}$$

where $u_k(t)$, $v_k(t)$ and $\phi_k(t)$ are the time-dependent nodal values of wind fields and geopotential fields respectively, and V_k represents a basis function (interpolation function) defined by the coordinates of the nodes, $\mathcal{N}$ is the number of finite element nodes. The variable vectors at time level n are written:

$$\mathbf{\Phi}^n=(\phi_1^n,\phi_2^n,\cdots,\phi_{\mathcal{N}}^n),\quad \mathbf{u}^n=(u_1^n,u_2^n,\cdots,u_{\mathcal{N}}^n),$$
$$\mathbf{v}^n=(v_1^n,v_2^n,\cdots,v_{\mathcal{N}}^n). \tag{10}$$

A Crank–Nicholson time differencing scheme was applied for temporal integration of the ordinary differential equations resulting from application of the Galerkin FEM [32,33]. The shallow-water equations system was then coupled at every time step so that the equations become quasi-linearised [37,38], since an average is taken at time level $n-1$ and time level n of expressions, while the non-linear advective terms are linearised by estimating them at time level $n+\frac{1}{2}$ using the following second-order approximation in time:

$$w^{\star}=\frac{3}{2}w^n-\frac{1}{2}w^{n-1}+O(\Delta t^2), \tag{11}$$

where the state variables $w=w(x,y,t)=(u,v,\phi)$. Multiplying (1) by V_i and integrating it over the computational domain, the discrete form of the continuity equation which is first to be solved at a given time step $n+1$, is obtained

$$\mathbf{M}(\mathbf{\Phi}^{n+1}-\mathbf{\Phi}^n)-\frac{\triangle t}{2}\mathbf{K}_1^n(\mathbf{\Phi}^{n+1}+\mathbf{\Phi}^n)=0, \tag{12}$$

where $\mathbf{M}$ is the mass matrix at the computational domain Ω,

$$\mathbf{M_{i,j}}=\int\int_{\Omega}V_iV_jd\Omega,\ \mathbf{M_{xi,j}}=\int\int_{\Omega}V_i\frac{\partial V_j}{\partial x}d\Omega,$$
$$\mathbf{M_{yi,j}}=\int\int_{\Omega}V_i\frac{\partial V_j}{\partial y}d\Omega, \tag{13}$$

and

$$K_{1i,j}^n=\int\int_{\Omega}u^{\star n}V_i\frac{\partial V_j}{\partial x}d\Omega+\int\int_{\Omega}V_iv^{\star n}\frac{\partial V_j}{\partial y}d\Omega. \tag{14}$$

By applying Petrov–Galerkin FEM approach into the momentum Eqs. (2) and (3), one obtains:

$$\mathbf{M}(\mathbf{u}^{n+1}-\mathbf{u}^n)+\frac{\triangle t}{2}\mathbf{K}_2^n(\mathbf{u}^{n+1}+\mathbf{u}^n)+\triangle t\mathbf{M}_x\boldsymbol{\phi}^{n+1}-\triangle tf\mathbf{M}\mathbf{v}^{*n}=0, \tag{15}$$
$$\mathbf{M}(\mathbf{v}^{n+1}-\mathbf{v}^n)+\frac{\triangle t}{2}\mathbf{K}_3^n(\mathbf{v}^{n+1}+\mathbf{v}^n)+\triangle t\mathbf{M}_y\boldsymbol{\phi}^{n+1}+\triangle tf\mathbf{M}\mathbf{u}^{n+1}=0, \tag{16}$$

where

$$K_{2i,j}^n=\int\int_{\Omega}u^{*n}V_i\frac{\partial V_j}{\partial x}d\Omega+\int\int_{\Omega}v^{*n}V_i\frac{\partial V_j}{\partial y}d\Omega+\int\int_{\Omega}\nu_u^{eff}\left(\frac{\partial V_i}{\partial x}\frac{\partial V_j}{\partial x}+\frac{\partial V_i}{\partial y}\frac{\partial V_j}{\partial y}\right)d\Omega, \tag{17}$$
$$K_{3i,j}^n=\int\int_{\Omega}u^{n+1}V_i\frac{\partial V_j}{\partial x}d\Omega+\int\int_{\Omega}v^{*n}V_i\frac{\partial V_j}{\partial y}d\Omega+\int\int_{\Omega}\nu_v^{eff}\left(\frac{\partial V_i}{\partial x}\frac{\partial V_j}{\partial x}+\frac{\partial V_i}{\partial y}\frac{\partial V_j}{\partial y}\right)d\Omega. \tag{18}$$

In the v-momentum equation, since the most recent solution for both ϕ^{n+1} and u^{n+1} at the current time step are known, v^{n+1} at the next time step can thus be solved from (u^{n+1},v^n,ϕ^{n+1}).

The effective viscosities ν_u^{eff} and ν_v^{eff} are an application of Petrov–Galerkin discretisation methods modified so as to result in positive viscosities and are given by (see [50]):

$$\nu_u^{eff}=\gamma\frac{r_up_ur_u}{max\left(\epsilon^{eff},\left(\frac{\partial u^{*n}}{\partial x}\right)^2+\left(\frac{\partial u^{*n}}{\partial y}\right)^2\right)},$$
$$\nu_v^{eff}=\gamma\frac{r_vp_vr_v}{max\left(\epsilon^{eff},\left(\frac{\partial v^{*n}}{\partial x}\right)^2+\left(\frac{\partial v^{*n}}{\partial y}\right)^2\right)}, \tag{19}$$

with

$$r_u=\frac{u^n-u^{n-1}}{\Delta t}+u^{*n}\frac{\partial u^{*n}}{\partial x}+v^{*n}\frac{\partial u^{*n}}{\partial y}-fv^{*n},$$
$$r_v=\frac{v^n-v^{n-1}}{\Delta t}+u^{n+1}\frac{\partial v^{*n}}{\partial x}+v^{*n}\frac{\partial v^{*n}}{\partial y}+fu^{n+1}, \tag{20}$$

and

$$p_u=\frac{1}{2}\left(\left(\frac{a_{ux}^{*n}}{\Delta x}\right)^2+\left(\frac{a_{uy}^{*n}}{\Delta y}\right)^2\right)^{-\frac{1}{2}},\quad p_v=\frac{1}{2}\left(\left(\frac{a_{vx}^{n+\frac{1}{2}}}{\Delta x}\right)^2+\left(\frac{a_{vy}^{*n}}{\Delta y}\right)^2\right)^{-\frac{1}{2}}, \tag{21}$$

and

$$\left(a_{ux}^{*n},a_{uy}^{*n}\right)^T=\frac{\left((u^{*n},v^{*n})^T\cdot\nabla_xu^{*n}\right)\nabla_xu^{*n}}{\|\nabla_xu^{*n}\|_2^2},$$
$$\left(a_{vx}^{*n},a_{vy}^{*n}\right)^T=\frac{\left((u^{*n},v^{*n})^T\cdot\nabla_xv^{*n}\right)\nabla_xv^{*n}}{\|\nabla_xv^{*n}\|_2^2}, \tag{22}$$

in which Δx, Δy are the size of the elements in the x- and y-directions and ϵ^{eff} is a tolerance used to avoid a "division by zero" error, here $\epsilon^{eff}=10^{-10}$, with $\gamma=0$ the Galerkin discretisation results while with $\gamma=1$ obtains the Petrov Galerkin, where $\nabla_{xt}u=\left(\frac{u^{n+1}-u^n}{\Delta t},\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}\right)$. Time steps are chosen so as to satisfy the Courant–Friedrichs–Levy criterion for stability.

3.2. Global discrete model of the shallow water equations

The global discrete forward model representation of the governing equations above can be expressed in matrix form:

$$\mathbf{A}(\mathbf{m},\mathbf{\Psi})\mathbf{\Psi}=\mathbf{s}, \tag{23}$$

where $\mathbf{\Psi}^T=(\mathbf{\Psi}^{1^T}\mathbf{\Psi}^{2^T},\ldots,\mathbf{\Psi}^{\mathcal{N}_t{}^T})$ is the vector of state variables; $\mathbf{s}^T=(\mathbf{s}^{1^T}\mathbf{s}^{2^T},\ldots,\mathbf{s}^{\mathcal{N}_t{}^T})$ are the source vectors; $\mathbf{m}^T=(m_1m_2,\ldots,m_{\mathcal{C}})$ ($\mathcal{C}$ is the number of controls) are the controls; and $\mathbf{A}$ is the global matrix making up the discretisation in the forward model at all the time levels, where $\mathcal{N}_t$ is the number of time levels (for steady state problems $\mathcal{N}_t=1$). At time level n, $\mathbf{\Psi}^{nT}=(\mathbf{\Phi}^{nT},\mathbf{u}^{nT},\mathbf{v}^{nT})$ (For $\mathbf{\Phi}^n$, $\mathbf{u}^n$ and $\mathbf{v}^{nT}$, see (10)); $\mathbf{s}^{nT}=\left(\mathbf{s}_{\mathbf{\Psi}}^{n\,T}\mathbf{s}_{\mathbf{u}}^{n\,T}\,\mathbf{s}_{\mathbf{v}}^{n\,T}\right)$ where $\mathbf{s}_{\mathbf{\Psi}}^n$, $\mathbf{s}_{\mathbf{u}}^n$ and $\mathbf{s}_{\mathbf{v}}^n$ are the source vectors associated with $\mathbf{\Phi}^n$, $\mathbf{u}^n$ and $\mathbf{v}^n$ respectively, here $\mathbf{s}_{\mathbf{\Psi}}^n=\left(s_{\mathbf{\Psi}1}^n\ s_{\mathbf{\Psi}2}^n,\ldots,s_{\mathbf{\Psi}\mathcal{N}}^n\right)^T$, $\mathbf{s}_{\mathbf{u}}^n=\left(s_{\mathbf{u}1}^n\ s_{\mathbf{u}2}^n,\ldots,s_{\mathbf{u}\mathcal{N}}^n\right)^T$, and $\mathbf{s}_{\mathbf{v}}^n=\left(s_{\mathbf{v}1}^n\ s_{\mathbf{v}2}^n,\ldots,s_{\mathbf{v}\mathcal{N}}^n\right)^T$.

For the three-level time marching methods used here the global matrix $\mathbf{A}$ has the structure:

$$\mathbf{A}=\begin{pmatrix}\mathbf{P}^2 & & & & \\ \mathbf{H}^3 & \mathbf{P}^3 & & & \\ \mathbf{I}^4 & \mathbf{H}^4 & \mathbf{P}^4 & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{I}^{\mathcal{N}_t} & \mathbf{H}^{\mathcal{N}_t} & \mathbf{P}^{\mathcal{N}_t}\end{pmatrix}, \tag{24}$$

where at time level n

$$\mathbf{P}^n = \begin{pmatrix} \mathbf{M} - \frac{\Delta t}{2}\mathbf{K}_1^n & \mathbf{0} & \mathbf{0} \\ \mathbf{M}_x \Delta t & \mathbf{M} + \frac{\Delta t}{2}\mathbf{K}_2^n & \mathbf{0} \\ \mathbf{M}_y \Delta t & f\mathbf{M}\Delta t & \mathbf{M} + \frac{\Delta t}{2}\mathbf{K}_3^n \end{pmatrix}, \tag{25}$$

$$\mathbf{H}^n = \begin{pmatrix} -\mathbf{M} - \frac{\Delta t}{2}\mathbf{K}_1^n & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M} + \frac{\Delta t}{2}\mathbf{K}_2^n & -1.5f\mathbf{M}\Delta t \\ \mathbf{0} & \mathbf{0} & -\mathbf{M} + \frac{\Delta t}{2}\mathbf{K}_3^n \end{pmatrix}, \tag{26}$$

$$\mathbf{I}^n = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0.5f\mathbf{M}\Delta t \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{27}$$

$$\mathbf{s}^2 = \mathbf{H}^2\boldsymbol{\Psi}^1, \ \mathbf{s}^3 = \mathbf{I}^3\boldsymbol{\Psi}^1, \tag{28}$$

and

$$\mathbf{H}^2 = \begin{pmatrix} -\mathbf{M} - \frac{\Delta t}{2}\mathbf{K}_1^1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{M} + \frac{\Delta t}{2}\mathbf{K}_2^1 & -f\mathbf{M}\Delta t \\ \mathbf{0} & \mathbf{0} & -\mathbf{M} + \frac{\Delta t}{2}\mathbf{K}_3^1 \end{pmatrix}. \tag{29}$$

4. The global discrete adjoint model of the shallow water equations

This section describes in detail how to derive the discrete adjoint model. In general the global matrix $\mathbf{A}$ (see Eq. (23)) is a function of the controls $\mathbf{m}$ and state variables $\boldsymbol{\Psi}$ that is $\mathbf{A} = \mathbf{A}(\mathbf{m}, \boldsymbol{\Psi})$. The cost functional can be expressed as:

$$J = J(\mathbf{m}, \boldsymbol{\Psi}). \tag{30}$$

The "total sensitivity" $\frac{dJ(\mathbf{m},\boldsymbol{\Psi})}{dm_l}$ of $J(\mathbf{m}, \boldsymbol{\Psi})$ with respect to a parameter ("control variable") m_l is obtained by computing its respective Gateaux-differential, which is given by the following expression:

$$\frac{dJ(\mathbf{m},\boldsymbol{\Psi})}{dm_l} = \frac{\partial J}{\partial m_l} + \left(\frac{\partial J}{\partial \boldsymbol{\Psi}}\right)^T \frac{d\boldsymbol{\Psi}}{dm_l}. \tag{31}$$

4.1. The global discrete tangent linear model of the shallow water equations

The differentiation of Eq. (23) with respect to m_l is

$$\frac{d\mathbf{A}}{dm_l}\boldsymbol{\Psi} + \mathbf{A}\frac{\partial \boldsymbol{\Psi}}{\partial m_l} = \frac{d\mathbf{s}}{dm_l}, \tag{32}$$

while

$$\frac{d\mathbf{A}}{dm_l}\boldsymbol{\Psi} = \frac{\partial \mathbf{A}}{\partial m_l}\boldsymbol{\Psi} + \sum_{k=1}^{3\times\mathcal{N}\times\mathcal{N}_t} \frac{\partial \mathbf{A}}{\partial \psi_k}\frac{\partial \psi_k}{\partial m_l}\boldsymbol{\Psi} = \frac{\partial \mathbf{A}}{\partial m_l}\boldsymbol{\Psi} + \mathbf{G}\frac{\partial \boldsymbol{\Psi}}{\partial m_l}, \tag{33}$$

where

$$\sum_{k=1}^{3\times\mathcal{N}\times\mathcal{N}_t} \frac{\partial \mathbf{A}}{\partial \psi_k}\frac{\partial \psi_k}{\partial m_l}\boldsymbol{\Psi} = \left[\frac{\partial \mathbf{A}}{\partial \psi_1}\boldsymbol{\Psi}, \ldots, \frac{\partial \mathbf{A}}{\partial \psi_k}\boldsymbol{\Psi}, \ldots, \frac{\partial \mathbf{A}}{\partial \psi_{\mathcal{N}}}\boldsymbol{\Psi}\right] \times \left[\frac{\partial \psi_1}{\partial m_l}, \ldots, \frac{\partial \psi_k}{\partial m_l}, \ldots, \frac{\partial \psi_{\mathcal{N}}}{\partial m_{\mathcal{N}}}\right]^T = \mathbf{G}\frac{\partial \boldsymbol{\Psi}}{\partial m_l}, \tag{34}$$

$$\mathbf{G} = \left[\frac{\partial \mathbf{A}}{\partial \psi_1}\boldsymbol{\Psi}, \ldots, \frac{\partial \mathbf{A}}{\partial \psi_k}\boldsymbol{\Psi}, \ldots, \frac{\partial \mathbf{A}}{\partial \psi_{\mathcal{N}}}\boldsymbol{\Psi}\right] = (\mathbf{g}_1, , \ldots, \mathbf{g}_k, \ldots, \mathbf{g}_{\mathcal{N}}). \tag{35}$$

The tangent linear model (TLM) (32) can be re-written:

$$\frac{\partial \mathbf{A}}{\partial m_l}\boldsymbol{\Psi} + (\mathbf{A} + \mathbf{G})\frac{\partial \boldsymbol{\Psi}}{\partial m_l} = \frac{d\mathbf{s}}{dm_l}. \tag{36}$$

4.2. The global discrete adjoint model of the shallow water equations

Forming the inner-product of Eq. (36) with an adjoint state variable vector $\boldsymbol{\Psi}^*$ in the Euclidean-space, yields

$$(\boldsymbol{\Psi}^*)^T(\mathbf{A} + \mathbf{G})\frac{\partial \boldsymbol{\Psi}}{\partial m_l} = (\boldsymbol{\Psi}^*)^T\left(\frac{\partial \mathbf{s}}{\partial m_l} - \frac{\partial \mathbf{A}}{\partial m_l}\boldsymbol{\Psi}\right). \tag{37}$$

The adjoint model is defined,

$$(\mathbf{A} + \mathbf{G})^T\boldsymbol{\Psi}^* = \frac{\partial J}{\partial \boldsymbol{\Psi}}. \tag{38}$$

That is,

$$(\mathbf{A}^T + \mathbf{G}^T)\boldsymbol{\Psi}^* = \left(\begin{pmatrix} \mathbf{P}^2 & & & \\ \mathbf{H}^3 & \mathbf{P}^3 & & \\ & \ddots & \ddots & \\ & & \mathbf{I}^{\mathcal{N}_t} \ \mathbf{H}^{\mathcal{N}_t} & \mathbf{P}^{\mathcal{N}_t} \end{pmatrix}^T + \mathbf{G}^T\right)\begin{pmatrix} \boldsymbol{\Psi}^{*1} \\ \boldsymbol{\Psi}^{*2} \\ \vdots \\ \boldsymbol{\Psi}^{*\mathcal{N}_t} \end{pmatrix} = \begin{pmatrix} \frac{\partial J}{\partial \boldsymbol{\Psi}^1} \\ \frac{\partial J}{\partial \boldsymbol{\Psi}^2} \\ \vdots \\ \frac{\partial J}{\partial \boldsymbol{\Psi}^{\mathcal{N}_t}} \end{pmatrix}, \tag{39}$$

in which the adjoint solution is $\boldsymbol{\Psi}^{*n} = (\boldsymbol{\Phi}^{*nT}, \mathbf{u}^{*nT}, \mathbf{v}^{*nT})^T$. The structure of the matrix $\mathbf{G}$ is expressed in (43).

The "total sensitivity" $\frac{dJ(\mathbf{m},\boldsymbol{\Psi})}{dm_l}$ in terms of the adjoint function can be expressed:

$$\frac{dJ}{dm_l} = \frac{\partial J}{\partial m_l} + \left(\frac{\partial \mathbf{s}}{\partial m_l} - \frac{\partial \mathbf{A}}{\partial m_l}\boldsymbol{\Psi}\right)^T\boldsymbol{\Psi}^*. \tag{40}$$

5. The formation of the adjoint model using the independent set graph-colouring methods

This section provides the details of the formation of the adjoint model. The differentiation of discrete forward models here is realised using the ISP sensitivity analysis, i.e., a perturbation method combined with a graph colouring approach. The graph colouring approach (for details, see [40] and Appendix A) is used to construct the matrices in the adjoint model (38) (for example, $\mathbf{G}$) and help accelerate the matrix equation assembly process. The perturbations associated with each variable at each node and time level are grouped in terms of colours and calculated concurrently.

5.1. The formation of the matrix $\mathbf{G}$ for the adjoint shallow water model

Since the matrix $\mathbf{A}$ is known in the adjoint model (38), the remaining part of the calculation consists in the formation of the matrix G in (38) and in particular the formation of the part of $\mathbf{G}$ associated with a given time level n. The matrix $\mathbf{G}$ can be constructed by a perturbation method.

Suppose the perturbation vector is $\boldsymbol{\Delta\Psi}_k = (0, \ldots, 0, \psi_k, 0, \ldots, 0)^T$ for a perturbation $\Delta\psi_k$ of the k^{th} entry in $\boldsymbol{\Psi}$ then the k^{th} column of $\mathbf{G}$ is:

$$\mathbf{G} = (\mathbf{G}' - \overline{\mathbf{G}})\mathbf{E}, \tag{41}$$

where $\mathbf{E}$ is a diagonal matrix with the diagonal entry $\mathbf{E}_{i,i} = \frac{1}{\Delta\psi_i}$, $\overline{\mathbf{G}} = (\mathbf{A}\boldsymbol{\Psi}, \mathbf{A}\boldsymbol{\Psi}, \ldots, \mathbf{A}\boldsymbol{\Psi})$ and $\mathbf{G}' = (\mathbf{g}'_1, \mathbf{g}'_2, \ldots, \mathbf{g}'_{\mathcal{N}}c')$ with

$$\mathbf{g}'_k = \mathbf{A}(\mathbf{m}, \boldsymbol{\Psi} + \boldsymbol{\Delta\Psi}_k)\boldsymbol{\Psi}. \tag{42}$$

In an implementation of the formation of the matrix $\mathbf{G}$ one needs to perform matrix vector multiplications involving $\mathcal{N}_c$ vectors. Thus, the entire matrix $\mathbf{G}'$ can be stored within $\mathcal{N}_c$ vectors of length $3\mathcal{N}\cdot\mathcal{N}_t$.

For the three-level time marching methods in the shallow water model (see the structure of the global matrix in Eq. (24)), the matrix $\mathbf{G}$ in time has the following structure:

$$\mathbf{G}=\begin{pmatrix}\mathbf{L}^2 & & & & & \\ \mathbf{Q}^2 & \mathbf{L}^3 & & & & \\ \mathbf{N}^2 & \mathbf{Q}^3 & \mathbf{L}^4 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \mathbf{N}^{\mathcal{N}_t-2} & \mathbf{Q}^{\mathcal{N}_t-1} & \mathbf{L}^{\mathcal{N}_t}\end{pmatrix}, \tag{43}$$

$$\mathbf{N}^n=\begin{pmatrix}\mathbf{0} & \mathbf{N}^n_{12} & \mathbf{N}^n_{13}\\ \mathbf{0} & \mathbf{N}^n_{22} & \mathbf{N}^n_{23}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix},\quad \mathbf{Q}^n=\begin{pmatrix}\mathbf{0} & \mathbf{Q}^n_{12} & \mathbf{Q}^n_{13}\\ \mathbf{0} & \mathbf{Q}^n_{22} & \mathbf{Q}^n_{23}\\ \mathbf{0} & \mathbf{0} & \mathbf{Q}^n_{33}\end{pmatrix},$$
$$\mathbf{L}^n=\begin{pmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{L}^n_{32} & \mathbf{0}\end{pmatrix}. \tag{44}$$

where $\mathbf{L}_{32}$ is a $\mathcal{N}\times\mathcal{N}$ matrix and $\mathbf{L}_{32i,k}=\int\int_\Omega V_k V_i\sum_{j=1}^{\mathcal{N}}\frac{1}{2}\left(v_j^{n+1}+v_j^n\right)\frac{\partial V_j}{\partial x}d\Omega$ $(k=1,\ldots,\mathcal{N})$.

5.2. The implementation of the adjoint shallow water model

The efficiency of implementation of the discrete adjoint model can be realised because in time marching methods typically all of the adjoint solution operates in an upper triangular structure where each block is associated with a solution variable at a particular time step. This means that one can solve the adjoint equations by marching backwards in time. Thus, matrix vector multiplication, $\mathbf{G}^T\boldsymbol{\Psi}^*$, can be placed on the right hand side of the matrix solution for the adjoint in Eq. (38). In this case the matrix $\mathbf{G}^T$ need not be formed and the matrix vector multiplication $\mathbf{G}^T\boldsymbol{\Psi}^*$ involving $\mathbf{G}'^T\boldsymbol{\Psi}^*$ can be rendered highly efficient for the kth row using:

$$(\mathbf{G}'^T\boldsymbol{\Psi}^*)_k=\mathbf{g}_k'^T\boldsymbol{\Psi}^*=\boldsymbol{\Psi}^{*T}(\mathbf{A}(\mathbf{m},\boldsymbol{\Psi}+\Delta\boldsymbol{\Psi}_k)\boldsymbol{\Psi}). \tag{45}$$

The perturbations associated with each node or variable k say are grouped in terms of colours and in this way a number of these matrix vector multiplications in (45) can be calculated concurrently (for details, see [40] and Appendix A), i.e. form the dot product of $\hat{\mathbf{g}}'_c$ with the part of $\boldsymbol{\Psi}^*$ associated with node k and form $(\mathbf{G}'^T\boldsymbol{\Psi}^*)_k$ for each row k and for colour c in which $\hat{\mathbf{g}}'_c=\sum_{k\in\text{ colour }c}\mathbf{g}'_k$. The graph colouring scheme used for the three co-located variables ψ, u and v is shown in Fig. 1. The distance-two graph associated with the vertices and edges in Fig. 1 is coloured with nine colours and thus there are nine colours for each of the three variables and with coupling for the three time levels resulting a total of $9\times3\times3=81$ colours. This means that a maximum of 81 + 1 matrix vector operations are required in order to form the matrix $\mathbf{G}$. The matrix vector operations of $\mathbf{G}^T\boldsymbol{\Psi}^*$ is undertaken at each time level when running the adjoint model backwards in time.

The discrete adjoint model (39) at each time level, say, $t=n$ can be written as:

$$(\mathbf{P}^{nT}+\mathbf{L}^{nT})\boldsymbol{\Psi}^{*n}=\frac{\partial J}{\partial\boldsymbol{\Psi}^n}-(\mathbf{H}^{n+1T}+\mathbf{Q}^{n-1T})\boldsymbol{\Psi}^{*n+1}-(\mathbf{I}^{n+2T}+\mathbf{N}^{n-2T})\boldsymbol{\Psi}^{*n+2}. \tag{46}$$

The matrices $\mathbf{L}^n$, $\mathbf{Q}^{n-1}$ and $\mathbf{N}^{n-2}$ (a dimensional size of $3\mathcal{N}\times3\mathcal{N}$) in (44) are related to the state solutions and their perturbations at time level n, $n-1$ and $n-2$ respectively. The matrix vector operations of $\mathbf{G}^T\boldsymbol{\Psi}^*$ at this time level is $\mathbf{L}^n\boldsymbol{\Psi}^{*n}$, $\mathbf{Q}^{n-1}\boldsymbol{\Psi}^{*n+1}$ and $\mathbf{N}^{n+2}\boldsymbol{\Psi}^{*n}$ which can be calculated concurrently using the graph colouring scheme. These matrices consist of a 9 point stencil (the graph colouring scheme used for the three variables ψ, u and v is shown in Fig. 1). The terms $\frac{ds}{dm_l}$ and $\frac{dA}{dm_l}$ in (37) can be formed using the same perturbation approach.

6. Determining the perturbation size

Suppose $f=f(\chi)$ is a scalar then the error in $\frac{\partial f(\chi)}{\partial\chi}$ using a perturbation approach can be estimated as:

$$\left(\frac{\partial f(\chi)}{\partial\chi}\right)_E\approx\frac{e|f(\chi)|}{\Delta\chi}, \tag{47}$$

in which e is the accuracy of the computation e.g. $e=10^{-7}$ for single precision and $e=10^{-15}$ for double precision, respectively. Combining the error in the computation with that associated with the perturbation size one obtains:

$$\left(\frac{\partial f(\chi)}{\partial\chi}\right)_E\approx\frac{e|f(\chi)|}{\Delta\chi}+\frac{1}{2}\Delta\chi\left|\frac{\partial^2 f(\chi)}{\partial\chi^2}\right|, \tag{48}$$

or assuming second order terms are evaluated exactly using a third order Taylor series:

$$\left(\frac{\partial f(\chi)}{\partial\chi}\right)_E\approx\frac{e|f(\chi)|}{\Delta\chi}+\frac{1}{4}(\Delta\chi)^2\left|\frac{\partial^3 f(\chi)}{\partial\chi^3}\right|, \tag{49}$$

in which say:

$$\frac{\partial^2 f(\chi)}{\partial\chi^2}\approx\frac{f(\chi+\Delta\chi)-2f(\chi)+f(\chi-\Delta\chi)}{2(\Delta\chi)^2}, \tag{50}$$

and:

$$\frac{\partial^3 f(\chi)}{\partial\chi^3}\approx\frac{f(\chi+2\Delta\chi)-3f(\chi+\Delta\chi)+f(\chi)+f(\chi-\Delta\chi)}{2(\Delta\chi)^3}. \tag{51}$$

Differentiating (48) w.r.t. the perturbation size $\Delta\chi$ and setting the result to zero (which happens at the minimum error) then the step size can be determined from (see [49]):

$$\Delta\chi\approx\sqrt{\frac{e|f(\chi)|}{\left|\frac{\partial^2 f(\chi)}{\partial\chi^2}\right|}}, \tag{52}$$

and using Eq. (48) then to third order accuracy:

$$\Delta\chi\approx\left(\frac{e|f(\chi)|}{\left|\frac{\partial^3 f(\chi)}{\partial\chi^3}\right|}\right)^{\frac{1}{3}}. \tag{53}$$

If one uses the second order perturbation approach to calculate the error then the curvature is exactly represented and thus only the computational error exists and the Taylor series expansion behind the last term in Eq. (48) needs to be truncated at third order which results in a term $\left|\frac{\partial^3 f(\chi)}{\partial\chi^3}\right|$, see Eq. (53). However, this term implies much more demanding computational requirements.

A simple example of the error for the gradient of a scalar function of χ is shown in Fig. 2. Notice that as the theory predicts for a linear function of χ the error is proportional to the magnitude of χ as described by Eqs. (47) and (48) for the quadratic functional, see Fig. 2 (right). A similar graph of the L_2 norm error in the gradient would also be observed when one varies the perturbation size while solving a control volume or finite element discretisation of Burgers equation, see [40].

6.1. A simple method to estimate perturbation size

If normalised with a suitable typical value of the gradient, $\left(\frac{\partial f(\chi)}{\partial\chi}\right)_0$ say, Eq. (47) can be used to form an estimate of the perturbation size. Thus suppose:

$$\frac{e|f(\chi)|}{\Delta\chi\left(\frac{\partial f(\chi)}{\partial\chi}\right)_0}=\alpha, \tag{54}$$

for some scalar α representing the fraction of the normalised error in the gradient required e.g. $\alpha=10^{-4}$ for single precision and $\alpha=10^{-8}$ for double precision. Re-arranging this equation, yields:

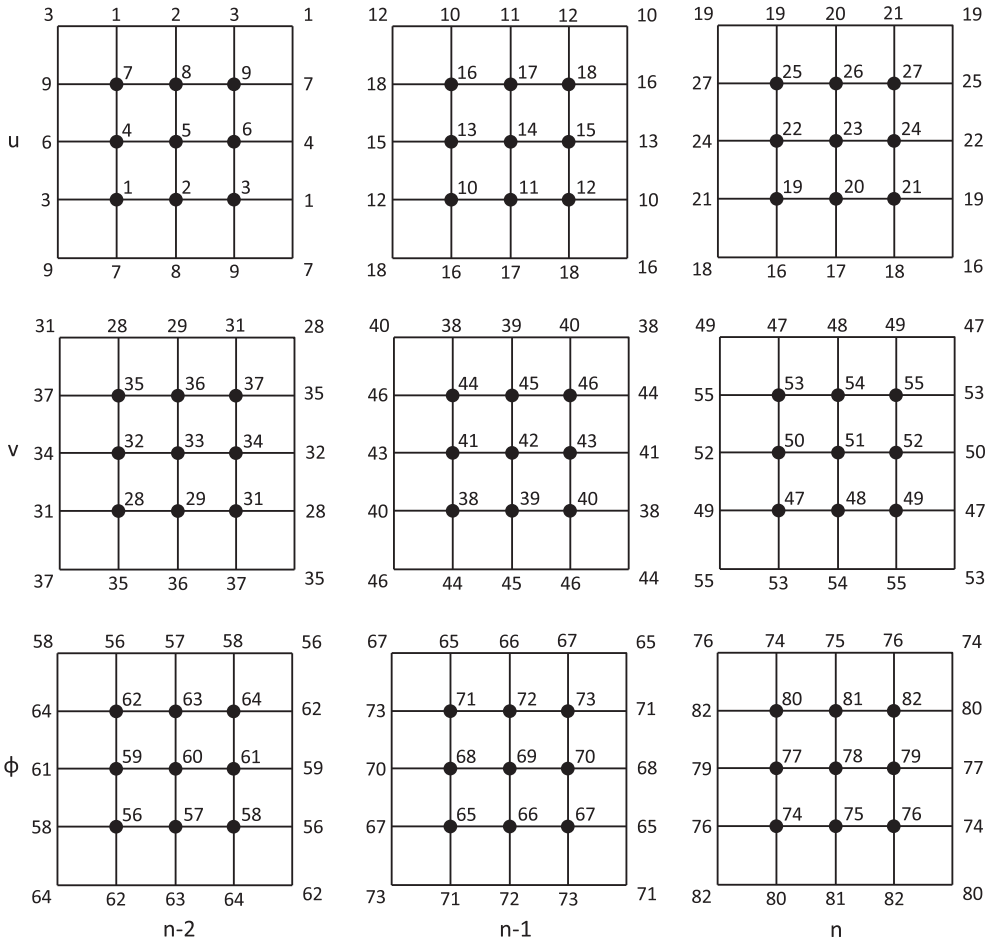


Fig. 1. Distance-two graph colouring associated with 2D linear quadrilateral elements. Nine colours are used for each solution variable (the top horizontal panel: u, the middle horizontal panel: v and bottom horizontal panel: ϕ) at three time levels (the left vertical panel: $t = n - 2$, the middle vertical panel: $t = n - 1$ and the right vertical panel: $t = n$).

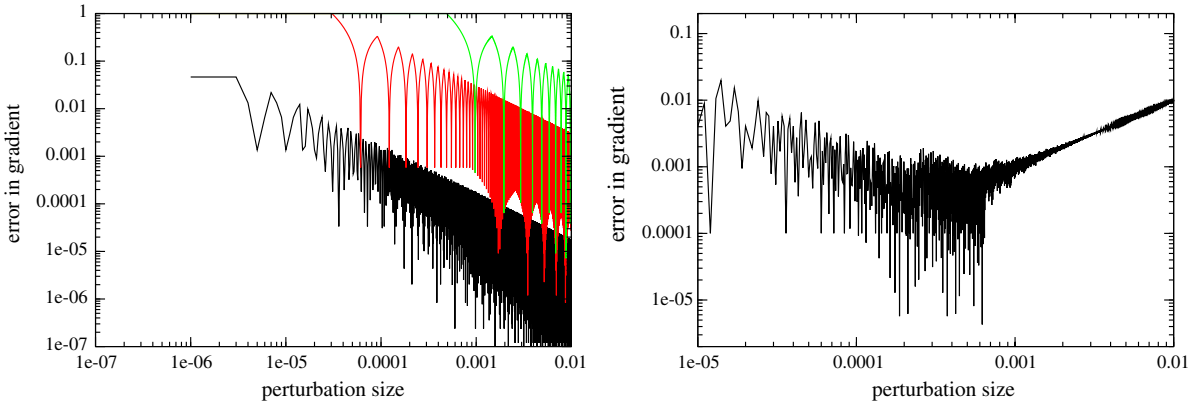


Fig. 2. The error in the first order derivative against perturbation size using a first order accurate perturbation method and single precision arithmetic and $\chi = 1$. (a) the black line: $f = 1 + \chi$, red: $f = 1000 + \chi$, green: $f = 10000 + \chi$. (b) $f = 1 + \chi + \chi^2$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\Delta\chi = \frac{e|f(\chi)|}{\alpha\left(\frac{\partial f(\chi)}{\partial \chi}\right)_0}. \tag{55}$$

When one is marching backwards in time in order to form the adjoint equations then one would use the next/future time level value of the gradient as $\left(\frac{\partial f(\chi)}{\partial \chi}\right)_0$.

6.2. Extension to matrices from discretised equations

The most rigorous extension to matrices $\mathbf{A}(\psi)$ with the matrix elements $A(\psi)_{ij}$ is simply to apply (55) to each matrix element in turn for a perturbation to a scalar ψ_l of the vector $\mathbf{\Psi}$, that is:

$$\Delta\psi_{lij} = \frac{e|A(\psi)_{ij}|}{\alpha\left(\frac{\partial A(\psi)_{ij}}{\partial \psi_l}\right)_0}. \tag{56}$$

However, one may lump the result to the variable ψ_l by considering just row l of the matrix $\mathbf{A}(\psi)$ and using the definition of the matrix $\mathbf{G}$ (Eq. (35)) then:

$$\Delta\psi_l = min\left\{\Delta\psi_{max},\ \frac{e\sum_j|A(\psi)_{lj}\psi_j|}{max\left\{\epsilon_{\Delta\psi},\ \alpha\sum_j|G_{jl}|\right\}}\right\}. \tag{57}$$

The analogous result to the scalar equation result (52) based on the traditional second order Taylor series approach is:

$$\Delta\psi_l = min\left\{\Delta\psi_{max},\ \sqrt{\frac{e\sum_j|A(\psi)_{lj}\psi_j|}{max\left\{\epsilon_{\Delta\psi},\ \sum_j\left|\frac{\partial^2 A(\psi)_{lj}}{\partial^2\psi_l}\psi_j\right|\right\}}}\right\}. \tag{58}$$

Here, $\epsilon_{\Delta\psi}$ is a tolerance used to avoid a "division by zero" error, $\epsilon_{\Delta\psi} = 10^{-10}$ and $\Delta\psi_{max}$ is the maximum value of the perturbation size, $\Delta\psi_{max} = 1$ is used here. The columns associated with all the free surface height variables (geopotential) in matrix $\mathbf{G}$ are zero (see Eq. (35)) because the matrix $\mathbf{A}$ is a linear function of these variables (see Eq. (13)). For the incompressible Navier Stokes equations one would also need to take into account the continuity equation which could be lumped with a row sum of the absolute values of the matrix for example. For robustness the values of the perturbation size $\Delta\psi_l$ should be limited by suitably chosen minimum, $\Delta\psi_{min}$ say. One may also combine the above taking the minimum of $\Delta\psi_l$ given by Eqs. (57) and (58). If one assumes that the non-linearity in $\mathbf{A}(\psi)$ is such that each of the powers (up to 3 say) has roughly equal coefficients then this value of $\alpha = 10^{-4}$ is a reasonable approximation. In fact this is often the case for finite element and control volume models (e.g. running near a Courant number $C = \Delta t\sqrt{\left(\frac{u^n}{\Delta x}\right)^2 + \left(\frac{v^n}{\Delta y}\right)^2}$ of order unity and the model (given by a modified Petrov–Galerkin based viscosity in this case, see Eqs. (17) and (19)) being of similar magnitude to the momentum advection term for the Navier Stokes or SWE's) although there can be important exceptions for the case of geostrophically balanced flow with relatively large values of f.

For time dependent problems, the perturbation size $\Delta\psi_l^n$ on the variable ψ_l^n at time level n is calculated by running the adjoint model backwards in time while the perturbation size $\Delta\psi_l^{n+1}$ obtained at time level $n+1$ is used to calculate $\mathbf{G}_{jl}$ (see Eq. (35)):

$$G_{jl} = \sum_k \frac{\partial A_{jk}}{\partial\psi_l^n}\psi_k \approx \frac{\sum_k\left(A_{jk}(\mathbf{m},\psi_l^n + \Delta\psi_l^{n+1})\psi_k - A_{jk}(\mathbf{m},\psi_l)\psi_k\right)}{\Delta\psi_l^{n+1}}. \tag{59}$$

Here, $\Delta\psi^{n+1}$ will affect the entries of the sub-matrices $\mathbf{H}^n$, $\mathbf{H}^{n+1}$, $\mathbf{H}^{n+2}$, $\mathbf{P}^n$, $\mathbf{P}^{n+1}$, $\mathbf{P}^{n+2}$ in the global matrix $\mathbf{A}$, respectively (see Eq. (24)).

Table 1
Grid schemes used in the test cases.

Grid schemes	Node number	Mesh size
Grid A	16 × 12	400 km × 400 km
Grid B	61 × 23	100 km × 200 km
Grid C	121 × 23	50 km × 200 km
Grid D	241 × 23	25 km × 200 km

7. Numerical results

The ISP sensitivity method has been applied to the 2D shallow water flow model (described previously) in a limited-area domain, of horizontal dimensions 6000 km by 4400 km with the origin $(x,y) = (0,0)$ located at the bottom left corner of the domain. The accuracy and speed of the ISP sensitivity method is compared against AD using TAMC in this section for the shallow water model. The applications also show the results of applying the non-uniform perturbation size as described in Section 6.2. All simulations were performed on a Dell Latitude E4200 in 4G of RAM and with a dual core 1.5 GHz processor. All calculations were performed in double precision.

7.1. Description of the test problem

The test problem used here adopts the initial conditions from the initial height field condition of Grammeltvedt No.1 [39]:

$$h(x,y) = H_0 + H_1\tanh\left(\frac{9(D/2-y)}{2D}\right) + H_2\left(1/\cosh^2\left(\frac{9(D/2-y)}{D}\right)\right)\sin\left(\frac{2\pi x}{L}\right), \tag{60}$$

where this initial condition has energy in wave number one in the x-direction. The initial velocity fields were derived from the initial height field using the geostrophic balance:

$$u = -\left(\frac{g}{\bar{f}}\right)\frac{\partial h}{\partial y}, \quad v = \left(\frac{g}{\bar{f}}\right)\frac{\partial h}{\partial x}. \tag{61}$$

The dimensional constants used here are:

$$\begin{aligned} &D = 4400\ \text{km}, \quad L = 6000\ \text{km}, \quad \bar{f} = 10^{-4}\ \text{s}^{-1}, \\ &\beta = 1.5\times10^{-11}\ \text{s}^{-1}\ \text{m}^{-1}, \\ &g = 10\ \text{ms}^{-1}, \quad H_0 = 2000\ \text{m}, \quad H_1 = 220\ \text{m}, \quad H_2 = 133\ \text{m}, \end{aligned} \tag{62}$$

and the time step size is $\Delta t = 900$ s, and four grid schemes (space increments) used here are shown in Table 1.

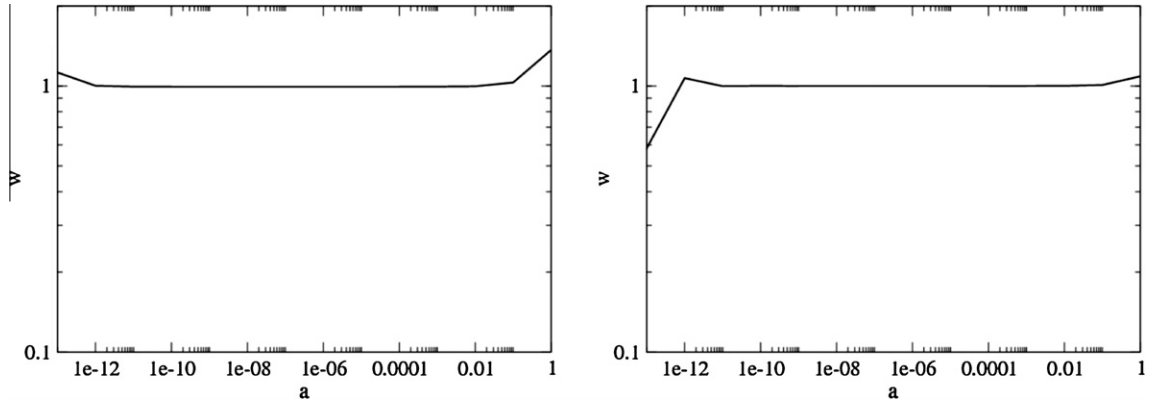
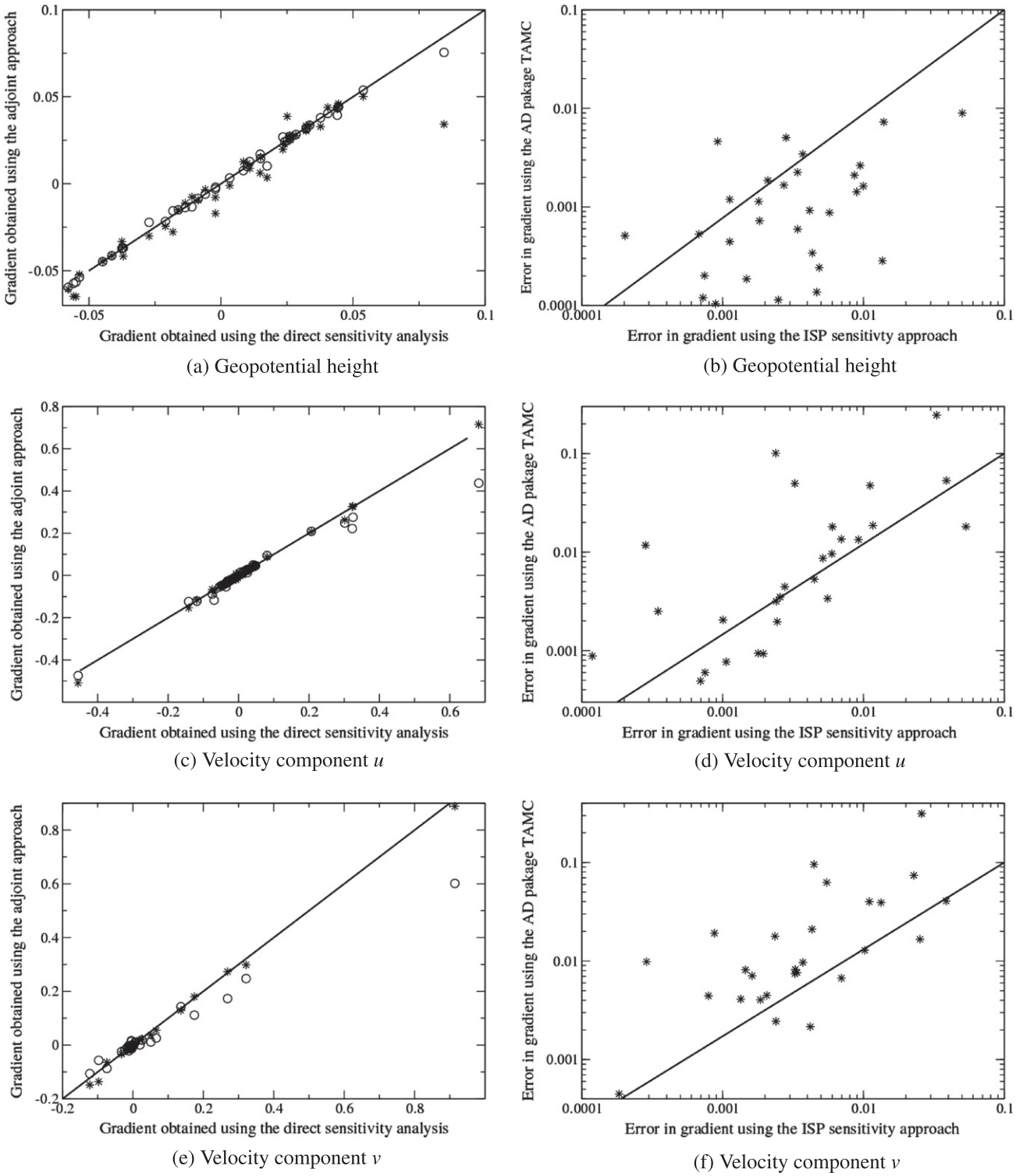


Fig. 3. The accuracy of the adjoint model using a gradient test $\left(w(a) = \frac{\partial J}{\partial\Psi^0} = \frac{(J(\Psi^0+a\delta\Psi^0)-J(\Psi^0))}{(a\delta\Psi^{0T})\nabla J(\Psi^0)}\right)$. The adjoint model is derived using: left panel: ISP sensitivity approach; right panel: Automatic differentiation. The cost functional J is defined as the weighted squared difference between the model solution and observations at the different time levels t_r in the Galerkin FE model where the dense observational data are available (here at every 5 h, see Eq. (63)) and the control variables are the initial conditions $(\mathbf{\Phi}^0, \mathbf{u}^0, \mathbf{v}^0)$.

7.2. Cost function

The aim of 4D-Var is to determine optimal control variables (e.g., initial conditions) by minimising the cost function measuring the lack of fit between model and observations. The cost function J is defined as the weighted squared difference between the initial conditions $\mathbf{\Psi}^0 = (\mathbf{\Phi}^0, \mathbf{u}^0, \mathbf{v}^0)$ and the background state $\mathbf{\Psi}_b = (\mathbf{\Phi}_b, \mathbf{u}_b, \mathbf{v}_b)$, and between the model solution $\mathbf{\Psi}^n$ and observations y_o^n distributed over the computational window of assimilation $[t_0, t_{N_t}]$ (following the notation proposed by Ide et al.) [48]



(a) Geopotential height (b) Geopotential height

(c) Velocity component u (d) Velocity component u

(e) Velocity component v (f) Velocity component v

Fig. 4. The gradient of the cost function J with respect to the initial conditions and gradient error in the Galerkin FE model during the simulation period [0,24] h, where the initial guess is given by applying a 5% uniform random perturbation on the exact initial conditions. In figures (a) (c) and (e), the star marks represent the solutions using the ISP sensitivity analysis while the open cycle marks represent the solutions using the AD package TAMC. The cost functional J is defined as the weighted squared difference between the model solution and observations (available at the time level $t = 24$ h and at every grid point, 400 km) and the control variables are the initial conditions $(\mathbf{\Phi}^0, \mathbf{u}^0, \mathbf{v}^0)$. Grid A (see Table 1) is used.

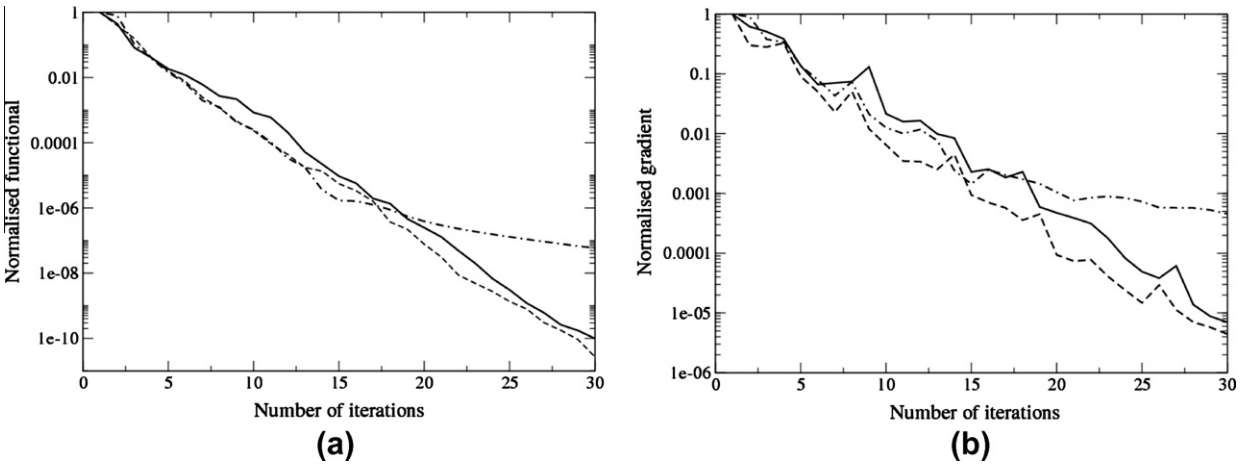


Fig. 5. The normalised cost function and the corresponding normalised gradient with respect to the initial conditions in the Galerkin FE shallow water model during optimisation iteration (the simulation period [0,24] h) using the ISP sensitivity method. The initial guess controls are taken: Case1–Solid line: from the true flow field (pseudo-observations) at t = 1 h; Case 2–dashed-dotted line: from a 'static' status, i.e. $(\mathbf{\Phi}^0, \mathbf{u}^0, \mathbf{v}^0) = 0.0$; Case 3–dashed line: by applying a 5% uniform random perturbation on the exact initial conditions. Left: the normalised cost function; right: the normalised gradient. The cost functional J is defined as the weighted squared difference between the model solution and observations, at the time level t = 24 h (see Eq. (63)) and the control variables are the initial conditions $(\mathbf{\Phi}^0, \mathbf{u}^0, \mathbf{v}^0)$. The normalised cost function: J^i/J^1, where J^i is the cost function at iteration i, and J^1 is the cost function at the first iteration ($J^1 = 1.789 \times 10^3$ for case 1; 1.643×10^4 for case 2 and 2.481 for case 3); The normalised gradient: $\left\|\frac{dJ^i}{dm_l}\right\|_2 \Big/ \left\|\frac{dJ^1}{dm_l}\right\|_2$, where, $\frac{dJ^i}{dm_l}$ is calculated using (40). Grid A (see Table 1) is used.

$$J(\mathbf{\Psi}^0) = \frac{1}{2}(\mathbf{\Psi}^0 - \mathbf{\Psi}_b)^T \mathbf{B}^{-1}(\mathbf{\Psi}^0 - \mathbf{\Psi}_b) + \frac{1}{2}\sum_{n=1}^{N_t}(\mathbf{H}\mathbf{\Psi}^n - y_o^n)^T \mathbf{R}^{-1}(\mathbf{H}\mathbf{\Psi}^n - y_o^n), \tag{63}$$

where $\mathbf{B}$ is the background error covariance matrix (containing information about the magnitude of the background errors and their correlations), $\mathbf{H}$ is the observation operator, $\mathbf{\Psi}^0$ is a vector containing the initial conditions (the control variables $\mathbf{m}_l$ in (30)), $\mathbf{\Psi}^n$ is a vector containing the solution of variables from the model at time level n, and y_o^n is the observation at time level n. $\mathbf{R}^{-1} = \left(R_\phi^{-1}, R_u^{-1}, R_v^{-1}\right)$ are weighting factors that are taken to be the inverse of the statistical root-mean-square observational errors on geopotential height and wind components respectively. In this experiment, the diagonal entries of R^{-1} are taken to be constant values: $R_\phi^{-1} = 10^{-4}$ for geopotential height; and $R_u^{-1} = R_v^{-1} = 10^{-2}$ for velocity components. Also $\mathbf{R}^{-1}$ is only non-zero at the nodes where there are observations. The data that is used in the data assimilation procedure is collected at time level t = 24 h only. This is the default observation scheme in time. However, some of the data assimilation example problems observe the ocean state at time levels t = 4,9,14,19,24 h (e.g. Figs. 3 and 6). There are two options for observations available over the space: (i) at every 400 km, and (ii) at every 2000 km (sparse data) in the x and y directions respectively.

7.3. Accuracy of the adjoint model derived using the ISP sensitivity analysis approach

The accuracy of the adjoint model derived using the ISP sensitivity analysis approach can be determined using a gradient diagnostic [44]. That is the quantity:

$$w(a) = \frac{DJ}{D\mathbf{\Psi}^0} \approx \frac{\left(J(\mathbf{\Psi}^0 + a\delta\mathbf{\Psi}^0) - J(\mathbf{\Psi}^0)\right)}{(a\delta\mathbf{\Psi}^{0T})\nabla J(\mathbf{\Psi}^0)}, \tag{64}$$

which must be of order $1 + O(a)$ in which $\delta\mathbf{\Psi}^0$is an arbitrary unit vector (such as $\delta\mathbf{\Psi}^0 = \nabla J/\|\nabla J\|_2$) (here, the control variables are

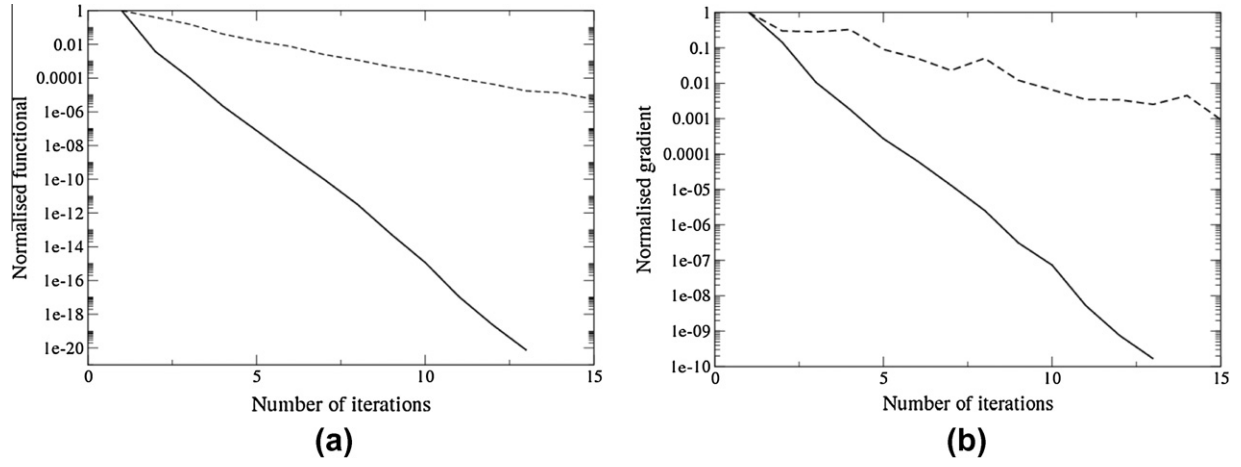


Fig. 6. The effect of the frequency of assimilating the observational data on the convergence of the optimisation. The observational data are assimilated at (a) every 5 h (**solid line**); and only one time level t = 24 h (**dashed line**) during the simulation period [0,24] h. Left: the normalised cost function; right: the normalised gradient with respect to the initial conditions in the Galerkin FE shallow water model during optimisation iteration. The initial guess controls are taken by applying a 5% uniform random perturbation on the exact initial conditions. Grid A (see Table 1) is used.

Table 2
The relative error in the optimised initial condition for the Galerkin FE model (the simulation period is [0,24] h). The optimisation procedure starts from the initial guess controls which are taken from the true flow field at $t = 1$ h. The cost functional J is defined as the weighted squared difference between the model solution and observation data, and the data is observed every 5 h started from $t = 4$ h inclusive. Grid A (see Table 1) is used.

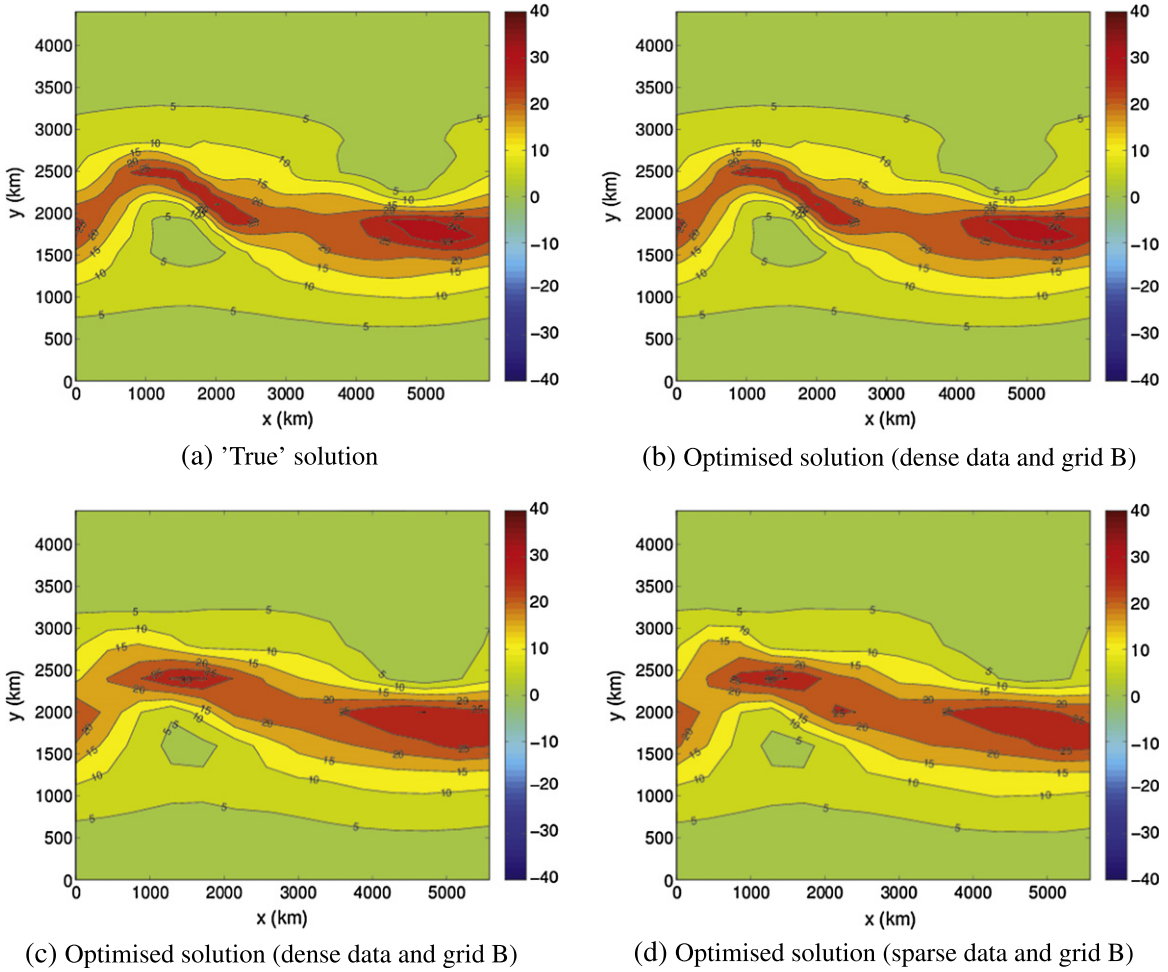
	Automatic differentiation	ISP sensitivity analysis
Maximum error in the geopotential height	0.4236×10^{-6} m	1.4353×10^{-6} m
Maximum error in the velocity component u	4.2960×10^{-6} m/s	1.6303×10^{-6} m/s
Maximum error in the velocity component v	35.814×10^{-6} m/s	24.421×10^{-6} m/s
Mean error in the geopotential height	0.0395×10^{-6} m	0.1613×10^{-6} m
Mean error in the velocity component u	0.1334×10^{-6} m/s	0.1101×10^{-6} m/s
Mean error in the velocity component v	1.5265×10^{-6} m/s	1.1868×10^{-6} m/s

the initial conditions $\mathbf{\Psi}^0 = (\mathbf{\Phi}^0, \mathbf{u}^0, \mathbf{v}^0)$)). Fig. 3 shows the variation of the function w with respect to a. It can be seen that (as required) the function $w(a)$ is close to unity when a varies between 10^{-3} and 10^{-12} which is evidence of the accuracy of the ISP sensitivity method as applied to the shallow water model.

7.4. Comparison of gradients obtained from the ISP sensitivity analysis and the AD package TAMC

7.4.1. Gradient results

The gradient results of the cost function with respect to the initial conditions ($(\mathbf{\Phi}^0, \mathbf{u}^0, \mathbf{v}^0)$) are calculated using both the ISP sensitivity analysis as well as the automatic differentiation package TAMC. The gradient results are given in Fig. 4 and compared with the direct gradient results. The direct gradient is determined by making a small perturbation (10^{-10}) in each of the controls (the initial conditions) and determining the change in the functional and this way calculating the gradient, that is $\frac{\Delta J}{10^{-10}}$. The initial condition for this gradient determination problem is the initial condition described by Eq. (60) with a 5% uniform random perturbation to the initial free surface height at each node. The error in the gradients is defined as the misfit between the adjoint gradient and direct gradient. It can be seen from the numerical results that both the adjoint gradient results are very close to the direct gradient



(a) 'True' solution

(b) Optimised solution (dense data and grid B)

(c) Optimised solution (dense data and grid B)

(d) Optimised solution (sparse data and grid B)

Fig. 7. The Galerkin FE model: the comparison of optimised solutions of speed at $t = 24$ h using the ISP sensitivity analysis approach (the simulation period is [0,24] h). (a) 'True' solution and Optimised solution using; (b) the sparse and grid B; (c) the dense data and grid A; and (d) the sparse data and grid A. **Sparse data** is available at 9 locations: at every 2000 km along the x and y directions; while **dense data** is available at 180 locations: at every 400 km along the x and y directions. The details of the grid schemes are provided in Table 1. The optimisation procedure starts from the initial guess controls which are taken from the true flow field at $t = 1$ h.

Table 3
Comparison of CPU required for running the adjoint model at one time level using the ISP sensitivity analysis and automatic differentiation (TAMC).

Mesh (node number) (control size n)	Running processes	CPU (Automatic differentiation)	CPU (ISP sensitivity analysis)
61×23	Assembling matrix	5	0.015
	Solver	6	0.38
	Transpose	0	0.03
	Matrix multiplication	0	0.18
	Total CPU	11	0.61
121×23	Assembling matrix	12	0.02
	Solver	27	0.79
	Transpose	0	0.06
	Matrix multiplication	0	0.35
	Total CPU	39	1.22
241×23	Sssembling matrix	29	0.05
	Solver	112	1.58
	Transpose	0	0.11
	Matrix multiplication	0	0.7
	Total CPU	114	2.44

results. The error for gradients of the geopotential height (calculated by both the ISP sensitivity analysis and the AD package TAMC) is smaller than 0.06 $m^2\ s^{-2}$. The error for gradients of the velocity components is smaller than 0.05 m/s when obtained by the ISP sensitivity analysis method, while 0.13 m/s by the automatic differentiation. The data is assimilated using observational data available only at $t = 24$ h in this problem.

7.4.2. 4D-Var data assimilation for optimising the initial conditions

The ISP sensitivity analysis derived model is now used to optimise the initial conditions $((\mathbf{\Phi}^0,\mathbf{u}^0,\mathbf{v}^0))$. The data assimilation experiments are designed using an identical twin technique. Pseudo-observations are obtained by running the forward model with a 5% random perturbation on the exact initial conditions, see Eq. (60). To assess the effect of the incomplete observational data on the optimised results, both the dense (available at every 400 km) and sparse (at every 2000 km in the x and y directions respectively) observational data are used. The error between the assimilated (optimised) initial conditions and the exact model variables (obtained from running the model forward in time from conditions (60) is calculated. Optimised results obtained using the ISP sensitivity analysis and those obtained by the automatic differentiation derived adjoint are compared here. The observation for the following three cases are taken at $t = 24$ h only. An initial guess of the initial conditions in the optimisation is taken: case 1 – from the 'true' flow field (pseudo-observations) at $t = 1$ h; case 2 – from a 'static' status, i.e. $(\mathbf{\Phi}^0,\mathbf{u}^0,\mathbf{v}^0) = 0.0$; case 3 – by applying a 5% uniform random perturbation on the exact initial conditions (which are used to produce the pseudo-observations). In this paper, the L-BFGS method [43] is used to minimise the cost functional measuring the misfit between the model solutions and pseudo-observations.

It can be seen from Fig. 5 that the evaluation of the normalised cost function converges to the prescribed tolerance 10^{-10} for cases 1 and 2 and 10^{-7} for case 3 after 30 iterations. The increase in the frequency of assimilating the observational data speeds up the convergence of the normalised cost/gradient function (Fig. 6, where the observational data are assimilated at every 5 h). The normalisation coefficient is the cost function at the start of the first L-BFGS iteration. The speed of convergence for both the ISP sensitivity analysis and automatic differentiation package TAMC is similar. The error in the optimised initial conditions is shown in Table 2. The maximum error in geopotential height is 1.44×10^{-6} (ISP) and 0.42×10^{-6} (TAMC). For velocity components u the maximum error is 1.6×10^{-6} for ISP and 4.3×10^{-6} for TAMC. For velocity components v the maximum error is 2.4×10^{-5} for ISP and 3.6×10^{-5} for TAMC. Thus the ISP sensitivity analysis derived adjoint model achieves a similar level of accuracy as the automatic differentiation model. The optimised results obtained from both the dense and sparse data are close to the 'true' flow solutions (pseudo-observations) (see (c) and (d) in Fig. 7). The accuracy of solutions is further improved (see (b) in Fig. 7) when a high resolution mesh (Grid B, see Table 1) is used.

7.4.3. Comparison of CPU time

For further comparison the CPU time for running the adjoint model at one time level using the ISP sensitivity analysis and

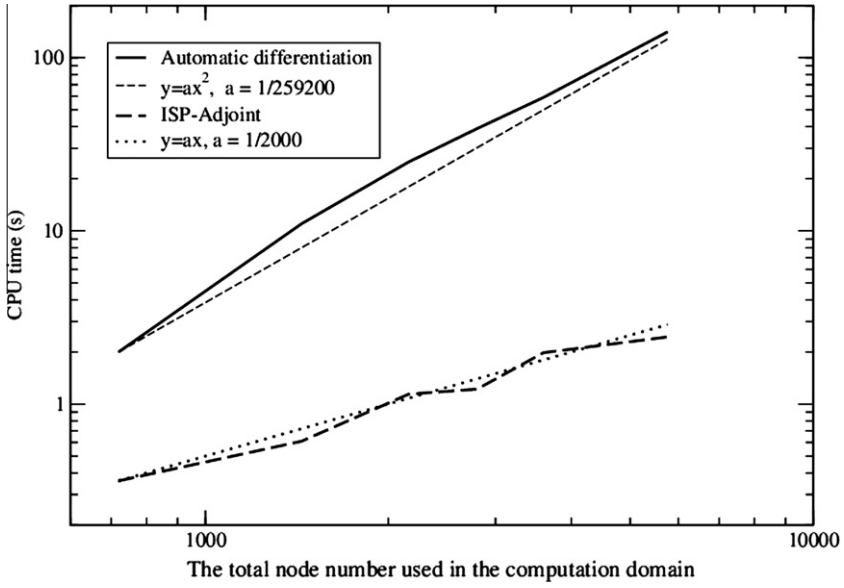
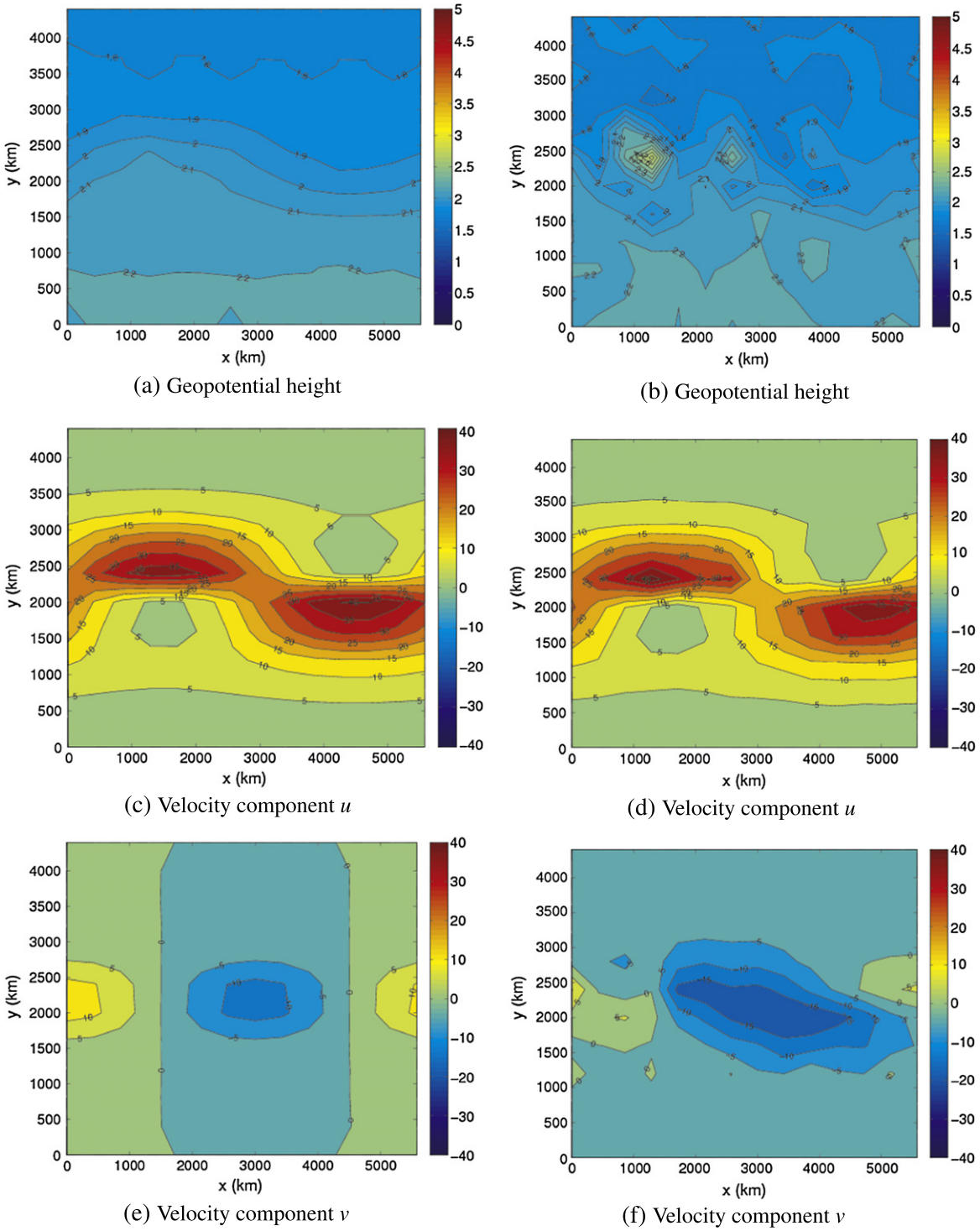


Fig. 8. The comparison of CPU for running the adjoint Galerkin FE model at one time level using the ISP sensitivity analysis and automatic differentiation (TAMC).

automatic differentiation models is listed in Table 3 and plotted in Fig. 8. The CPU time required for running the ISP derived adjoint model is much less than that for running the automatic differentiation derived adjoint model especially for the large number of nodes required in the computational simulation. It is shown that the ISP derived adjoint model requires CPU time proportional to $O(n)$ while CPU time proportional to $O(n^2)$ is required for the automatic differentiation derived adjoint model (where n is the number of nodes used in the simulation). This is similar to results obtained by the pseudo-adjoint method of Bischof et al. [26].

7.5. The non-linear Petrov–Galerkin FE model and variable perturbation size

Up to this point we have used a Galerkin FE discretisation ($\gamma = 0$ in Eqs. (17) and (19)) and thus since the discrete system of



(a) Geopotential height

(b) Geopotential height

(c) Velocity component u

(d) Velocity component u

(e) Velocity component v

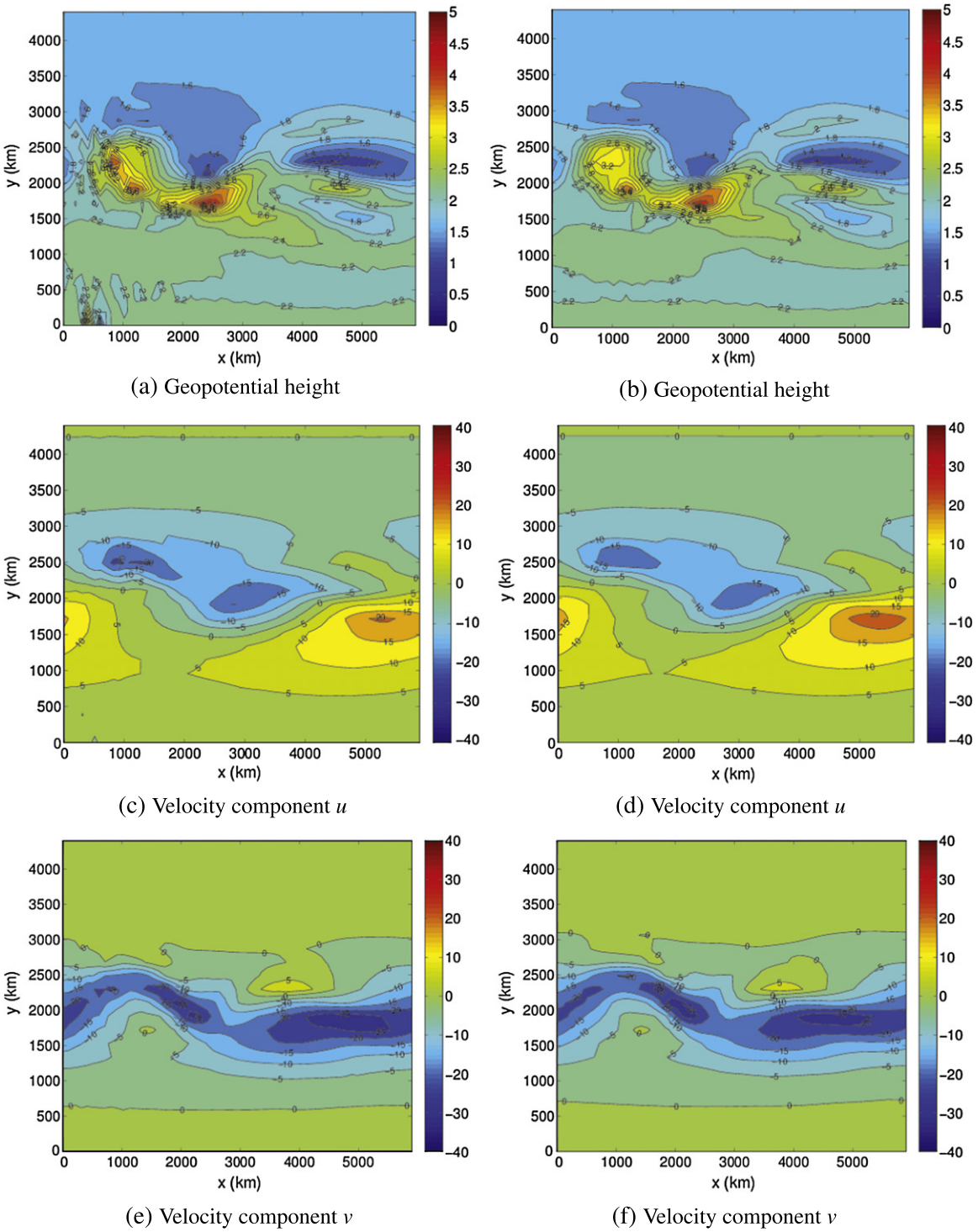
(f) Velocity component v

Fig. 9. The comparison of the optimised initial conditions (using the ISP sensitivity analysis approach) using between **the Petrov–Galerkin FE model** (left panel); and **the Galerkin FE model** (right panel). The simulation period is [0,24] h; and the sparse observational data is available at nine locations (every 2000 km along the x and y directions). The optimisation procedure starts from the initial guess controls which are taken from the true flow field at $t = 1$ h. Grid A (see Table 1) is used.

equations is quadratic the ISP sensitivity method produces exact (to computational round off error) adjoints and sensitivities independent of the perturbation size. In this subsection the consequences of performing simulations using the Petrov–Galerkin approach ($\gamma = 1$) with increased non-linearity are investigated. This model is thought to be representative of sub-grid-scale models which often represent sub-grid-scale and unresolved physics (e.g. turbulence) with highly non-linear functions. This contributes to our efforts to demonstrating that the ISP sensitivity approach can be used in highly complex models by first applying it to an intermediate complexity model (the shallow water equations) with sophisticated sub-grid-scale modelling based on Petrov–Galerkin methods.

7.5.1. Data assimilation using the Petrov–Galerkin FE model

The ISP sensitivity analysis is applied with the Petrov–Galerkin FE model to optimise the initial conditions ($\mathbf{\Phi}^0, \mathbf{u}^0, \mathbf{v}^0$). The optimised results obtained by assimilating the sparse data into the



(a) Geopotential height (b) Geopotential height

(c) Velocity component u (d) Velocity component u

(e) Velocity component v (f) Velocity component v

Fig. 10. The Petrov–Galerkin FE model: the optimised solution at $t = 24$ h using the ISP sensitivity analysis approach (the simulation period is [0,24] h). Left panel: the optimised solution; right panel: the 'true' solution. The sparse data is assimilated at nine locations: at every 2000 km along the x and y directions at time level $t = 24$ h. The optimisation procedure starts from the initial guess controls which are taken from the true flow field at $t = 1$ h. Grid B (see Table 1) is used.

model are smoother (due to the dissipative properties of the Petrov–Galerkin method) and closer to the 'true' solution when compared with those obtained by the Galerkin FE model (Fig. 9). The optimised results at time level $t = 24$ h are very close to the true solution (Fig. 10). The CPU time required for running the Petrov–Galerkin adjoint model is proportional to $O(n)$, which is similar to that required for running Galerkin adjoint model.

7.5.2. Determining the perturbation size

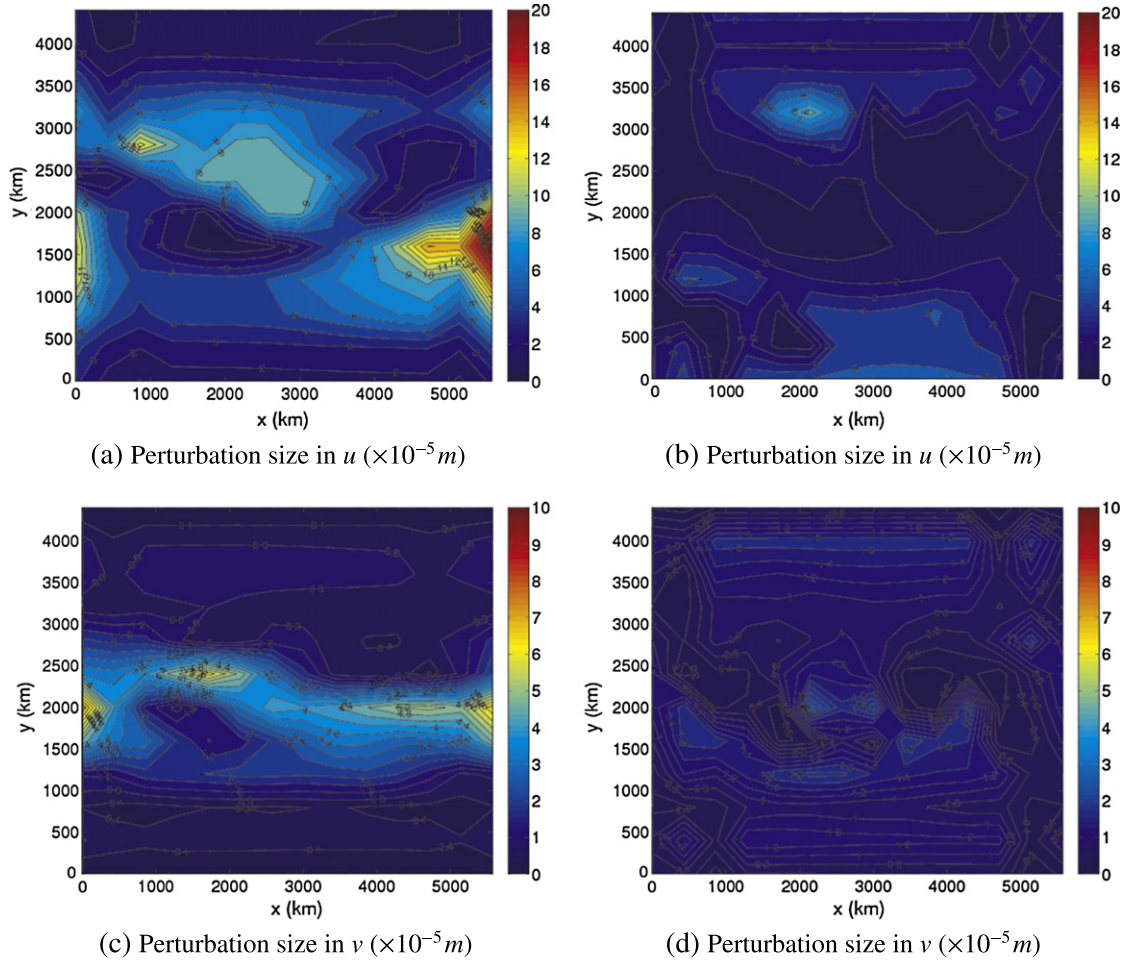
With the extra non-linearity provided by the Petrov–Galerkin method the size of the perturbations now becomes important and so we compare the two approaches developed here to obtain these so that every solution variable now has its own perturbation size at every time level. The first approach is based on traditional Taylor series error analysis (52) and the second approach is relatively simple (55) and is based on the (Figs. 12 and 11) observations that modelling and resolved physics terms in the equations have approximately equal orders of magnitude. Thus the two approaches should yield comparable perturbation size distributions at time levels $t = 0.05$ h (Fig. 12) and $t = 24$ h (Fig. 11). The simpler approach seems to have its largest perturbation sizes in the faster areas of the flow where as for the second order approach such a correlation is not so clear. This is due to the use of the second order term on the denominator of Eq. (58) and may be due to increases in errors in the formation of higher order derivatives – second order derivatives in this case.

The simpler approach is at its most accurate when the linear term in **A** is comparable in magnitude to the non-linear terms in **A** – the discretised system of equations at every time step. For small time step sizes this is clearly not the case as the mass matrices from the time discretisation have the dominant contribution. However, often one chooses time step sizes based on what one can get away with and in this case the linear and non-linear terms may have a similar size and their relative magnitudes may be determined approximately by the grid Courant number C.

8. Further discussion and future perspectives

The application of adjoints is often limited to flow problems where the linearised approximation is valid. One of the critical issues for adjoint based approaches is how to deal with highly non-linear discrete systems, see [51,52]. We show how this can be done by applying the ISP sensitivity analysis to a Petrov–Galerkin method that introduces highly non-linear terms. We show that the ISP sensitivity approach can be used to help assimilate data using a highly non-linear model (see Fig. 10). This should be seen as a step towards applying the method to much more complex problems which typically parametrize (model) unresolved processes using highly non-linear terms e.g. Large Eddy Simulation methods for turbulence modelling.

For the ISP sensitivity approach, the gradient/sensitivity of the cost function J is exact when the discrete forward model is



(a) Perturbation size in u ($\times 10^{-5} m$)

(b) Perturbation size in u ($\times 10^{-5} m$)

(c) Perturbation size in v ($\times 10^{-5} m$)

(d) Perturbation size in v ($\times 10^{-5} m$)

Fig. 11. The Optimised perturbation size at $t = 24$ h (**Sparse data** at nine locations: observed at every 2000 km along the x and y directions). Left panel: the first order approach; Right panel: the second order approach. Grid A (see Table 1) is used.

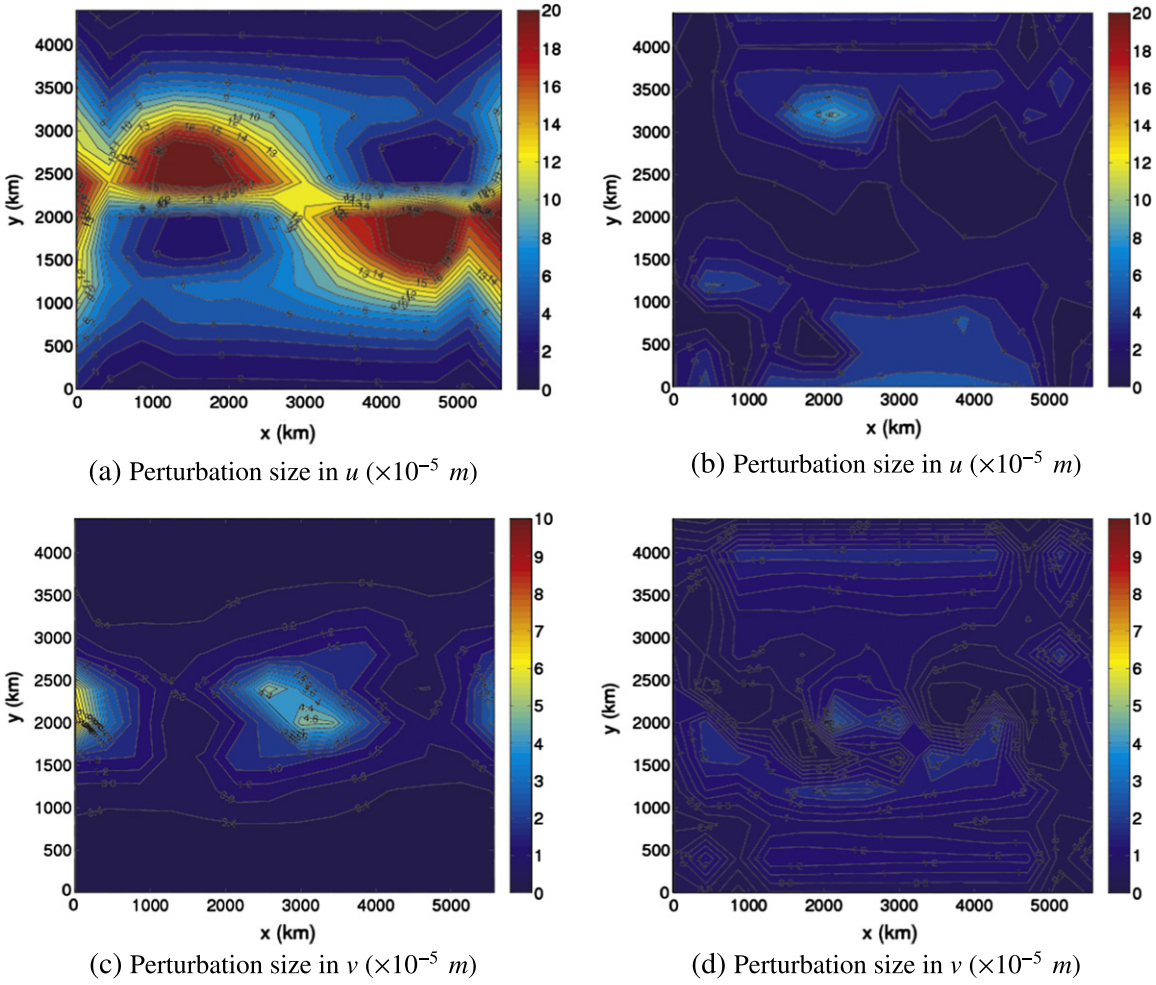
quadratic. To test its accuracy in a highly non-linear model, the ISP sensitivity approach is applied here to a Petrov–Galerkin version of the same shallow water model. The optimal results obtained are promising (see Fig. 10).

For highly non-linear systems (such as produced by the Petrov–Galerkin approach used here), the perturbation size is one of the critical issues affecting the accuracy of sensitivity analysis. A simple formula is developed and shown to work well in practice for the Petrov–Galerkin FE model.

In view of the encouraging results obtained (note: these results are model dependent), we plan to test the ISP sensitivity analysis method on more complex models. Another direction of research for exploration is to test performance of the ISP sensitivity analysis approach vs the performance of other AD packages such as the readily available TAF and TAPENADE. The ISP sensitivity approach is suitable for use in parallel computing as it is based on colouring methods which can easily extract fine grain parallelism.

In addition, the next generation of atmospheric and oceanic models will be even more complex than current generation in for example oceans, atmospheric and multi-phase flows. Thus there is a pressing need to develop methods that can differentiate these models within reasonable CPU times. Often these models are so complex that current AD tools will simply fail or require extensive intervention to produce sensible run times. One of the key reasons for this complexity is the use of unstructured meshes and the associated indirect addressing and additional logic associated with these. Due to the magnitude of effort that goes into compilers AD methods may always lag behind and often unable to be used with the latest enhancements to modern computer languages like C++ and FORTRAN. The ISP sensitivity method as an alternative to AD tools provides a different approach to tackle this pressing problem. In fact it is relatively easy to substitute the formation of the key matrix **G** in Eq. (41) with AD tools and have them operate at this assembly level. This allows the mechanisms developed here to be used but also in conjunction with AD methods and for the method to scale linearly with the problem size, see [26] for examples of this approach. The user intervention sometimes necessary with AD tools is redundant with the ISP sensitivity analysis as it is independent of the forward model. That is any new parameterization or assembly routines will be automatically differentiated. AD tools may also produce code that is much more complex (out of necessity) since the differential of the various terms in the discrete equations can be highly complex after differentiation. The ISP approach circumvents this additional overhead at the cost of introducing perturbations into the approach. However, these perturbations sizes can easily be optimised as demonstrated here.

In addition, there is a possibility of developing a modelling framework that can allow all models to be differentiated within that framework with the proviso that all models be written in such a way that matrix vector multiplication involving the forward model be easily performed. We hope also that in such modelling frameworks that may couple different physics (multi-physics) such



(a) Perturbation size in u ($\times 10^{-5}$ m)

(b) Perturbation size in u ($\times 10^{-5}$ m)

(c) Perturbation size in v ($\times 10^{-5}$ m)

(d) Perturbation size in v ($\times 10^{-5}$ m)

Fig. 12. The Optimised perturbation size at $t = 0.05$ h (**Sparse data** at nine locations: observed at every 2000 km along the x and y directions.). Left panel: the first order approach; right panel: the second order approach. Grid A (see Table 1) is used.

as oceans coupled to an atmosphere and solids coupled to fluids can also be differentiated using the presently discussed approach.

The intent of this work is to make adjoint based methods more accessible with the development of the ISP sensitivity approach in view of its following advantages:

- reduces the CPU time required to differentiate complex models;
- results in differentiated models that scale optimally with problems size (linearly);
- enables the model to automatically keep up with developments of the forward model;
- enables a modelling framework to be written that can allow all models to be differentiated within that framework;
- enables one to extract fine grain parallelism due to the use of colouring methods.

9. Conclusions

The development of the 4D-Var adjoint model for a finite element Galerkin shallow water model is presented here. Using the ISP sensitivity analysis approach, the adjoint model is easily implemented and maintained as the forward model is updated. Differentiation of the discrete forward model is realised by a graph colouring approach combined with a perturbation method to obtain consistent gradients involving discrete equations.

The accuracy of the ISP sensitivity analysis derived adjoint model is validated and the resulting gradients found to have comparable accuracy to automatic differentiation applied through the automatic differentiation package TAMC. The ISP sensitivity analysis derived adjoint model is also applied to optimise the initial conditions of the 2D shallow water model. The rate of convergence, to a local minima, of both the ISP sensitivity analysis and automatic differentiation package TAMC are similar. The CPU time using the ISP sensitivity analysis approach scales linearly with problem size – although this may be achieved with automatic differentiation methods it may require considerable effort. For the test case in this paper, using the ISP sensitivity analysis approach, the CPU time for running the adjoint model is 20 times smaller, for the largest problem considered, than that required by the automatic differentiation approach. Moreover, it is shown that the CPU time required for the ISP derived adjoint model is a linear function of the number of nodes $\mathcal{N}$ used in the simulation, $O(\mathcal{N})$, while $O(\mathcal{N}^2)$ for the automatic differentiation derived adjoint model.

Testing of the ISP sensitivity approach has further been carried out with a highly non-linear Petrov–Galerkin version of the same shallow water model. The optimised initial conditions are very close to the exact ones. The perturbation size is determined by the first/second order formulations and varies over space and time. As far as the ISP sensitivity analysis is concerned the intermediate complexity shallow water model contains all of the complexities associated with a complex flow model (e.g. atmospheric and ocean models): has a highly-non-linear sub-grid-scale model; multiple unknowns at each node requiring multi-dimensional colouring methods; non-uniform and time dependent optimised perturbation sizes. However, using the intermediate complexity model it is much easier to understand and examine the performance of the ISP sensitivity method.

Acknowledgments

This work was carried out under funding from the UK's Natural Environment Research Council (Projects NER/A/S/2003/00595, NE/C52101X/1, NE/C51829X/1 and NE/F012594/1), the Engineering and Physical Sciences Research Council (GR/R60898 and EP/I00405X/1) and the Leverhulme Trust (F/07058/AB), and with support from the NURISP framework seven project, the Imperial College High Performance Computing Service and the Grantham Institute for Climate Change. Prof. I.M. Navon would like to acknowledge support from NSF grant ATM-0931198. Many thanks to Dr. Paul Hovland from Argonne National Laboratory for his advice and Ryder Michelle for help with the technical writing up of this work.

Appendix A. Graph colouring methods

The graph colouring approach, as shown here, can be used to simplify the structure of the matrices (described above) and help accelerate the matrix equation assembly process. The matrices associated with the non-linear terms can be expressed by a set of submatrices. These sub-matrices are time-independent and can be easily differentiated to form adjoint systems of equations. The differentiation of discrete forward models here is realised using the ISP sensitivity analysis i.e., a perturbation method combined with a graph colouring approach. The perturbations associated with each variable at each node and time level are grouped in terms of colours and calculated concurrently (see the section below).

In graph theory, graph colouring is a special case of graph labeling; it is an assignment of labels traditionally called "colours" to elements of a graph subject to certain constraints. In its simplest form, it is a way of colouring the vertices of a graph such that no two adjacent vertices share the same colour; this is called a vertex colouring.

Graph colouring methods are commonly used to model the dependency between different subtasks or data. Here we define the graph $G_r = (V_g, E_g)$, where the vertex set, V_g, are the nodes or cells of the finite element or control volume mesh (i.e. the rows of a discretisation matrix), and the edge set, E_g, is defined by the connectivity under a given stencil, see below. The chromatic number (the smallest number of colours needed to colour the vertices of G_r so that no two adjacent vertices share the same colour), $\chi(G_r)$, is bounded by:

$$\omega(G_r) \leqslant \chi(G_r) \leqslant \Delta(G_r) + 1,$$

where $\omega(G_r)$ is the clique (a subset of its vertices such that every two vertices in the subset are connected by an edge) number and $\Delta(G_r)$ is the maximum vertex degree [53]. $\chi(G_r)$ is the minimum number of colours necessary to colour a graph. However, the number of colours obtained by a colouring algorithm $\mathcal{N}_c$ might exceed this minimum.

An example of a quadratic non-linearity in advection terms can be written in a matrix $\mathbf{Q}_q$ whose element assumes the form:

$$Q_{qij} = \int_\Omega N_i q \frac{\partial N_j}{\partial x} d\Omega, \tag{65}$$

where $q = \sum_{k=1}^{\mathcal{N}} N_k q_k$ and $q = u^{*n}, v^{*n}, u^{n+1}$ in $K_{1\ ij^n}$, $K_{2\ ij^n}$ and $K_{3\ ij^n}$ in Eqs. (14), (17) and (19). When forming non-linear quadratic discretisations, it is important to look at the independent sets of basis functions N_k used in $\int\int_\Omega N_i N_k \frac{\partial N_j}{\partial x} d\Omega$ for any nodes i and j. That is for the ith row and jth column of the matrix associated with $\int\int_\Omega N_i N_k \frac{\partial N_j}{\partial x} d\Omega$ no two basis functions N_k of the same colour contribute a non-zero value to this row and column.

For a node-wise N_k (in Fig. 1) the graph associated with the non-zero's of the matrix $\mathbf{Q}_q^T \mathbf{Q}_q$ can be coloured so as to achieve the required independent sets of $\int\int_\Omega N_i Q_k \frac{\partial N_j}{\partial x} d\Omega$, which is also referred to as a distance-two colouring of the sparsity pattern of matrix $\mathbf{Q}$ [54]. The matrix Q^c_{qij} in (65) can be expressed in the colouring scheme:

$$\mathbf{Q}_q ij^c = \int \int_\Omega N_i b^c N_j d\Omega, \tag{66}$$

where

$$b^c = \sum_{k=1}^{Q} Q_k b_k^c, \quad b_k^c = \begin{cases} 1 & \text{if node } k \text{ is of colour } c; \\ 0 & \text{at other nodes.} \end{cases} \tag{67}$$

The graph colouring approach outlined here for matrix $\mathbf{Q}$ can be applied to efficiently differentiate the matrices $\mathbf{K}_1^n$, $\mathbf{K}_2^n$ and $\mathbf{K}_3^n$ in Eqs. (12), (15) and (16).

Fig. 1 shows the distance-two graph colouring scheme used here for u, v and ϕ at three time levels $t = n-2$, $t = n-1$ and $t = n$. The distance-two graph associated with the vertices and edges is coloured with 9 colours. This graph colouring scheme is applied to all the variables at each time level and thus there are nine colours for each of the three variables (in Fig. 1: the top panel for u, the middle panel for v and the bottom panel for ϕ). Taking into account the three-level time marching methods used here, a total of $9 \times 3 \times 3 = 81$ colours is required in this study.

References

[1] Cacuci DG, Weber CF, Oblow EM, Marable JH. Sensitivity theory for general systems of nonlinear equations. Nucl Sci Eng 1980;75:88–110.
[2] Sandu A, Daescu D, Carmichael GR. Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: I – theory and software tools. Atmos Environ 2003;37:5083–96.
[3] Bennett AF, Chua BS, Leslie LM. Generalized inversion of a global numerical weather prediction model. Meteor Atmos Phys 1996;60:165–78.
[4] Courtier P, Talagrand O. Variational assimilation of meteorological observations with the direct and adjoint shallow-water equations. Tellus 1990;42A:531–49.
[5] Langland RH, Elsberry RL, Errico RM. Evaluation of physical process in an idealized extratropical cyclone using adjoint sensitivity. Quart J Roy Meteor Soc 1995;121:1349–86.
[6] Ghil M, Ide K, Bennett A, Courtier P, Kimoto M, Nagata M, et al. (Eds.), Data assimilation in meteorology and oceanography. Meteorological Society of Japan, 300 pp.
[7] Farrell BF. Optimal excitation of neutral Rossby waves. J Atmos Sci 1988;45:163–72.
[8] Anderson WK, Venkatakrishnan V. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. Comput Fluids 1999;28(4):443–80.
[9] Barhen J, Cacuci DG, Wagschal JJ, Bjerke MA, Mullins CB. Uncertainty analysis of time-dependent nonlinear systems: theory and application to transient thermal-hydraulics. Nucl Sci Eng 1982;81:23–44.
[10] Ionescu-Bujor M, Cacuci DG. Adjoint Sensitivity Analysis of the Two-fluid Model in RELAP5/MOD3.2, Invited Paper, M and C'99, International conference on mathematics and computation, reactor physics and environmental analysis in nuclear applications, vol. 1. Madrid, Spain, September 27–30; 1999. p. 745–54 [Senda Editorial, Madrid, Spain].
[11] Marta AC, Mader CA, Martins JRRA, Van der Weide E, Alonso JJ. A methodology for the development of discrete adjoint solvers using automatic differentiation tools. Int J Comput Fluid Dyn 2007;21(9–10):307–27.
[12] Griewank A, Walther A. Implementation of checkpointing for the reverse or adjoint mode of computational differentiation. ACM Trans Math Software 2000;26(1):19–45.
[13] Griewank A, Walther A. Evaluating derivatives: principles and techniques of algorithmic differentiation. 2nd ed. Philadelphia: SIAM; 2008.
[14] Griewank A, Juedes D, Utke J. ADOL-C, a package for the automatic differentiation of algorithms written in C/C++. ACM Trans Math Software 1996;22:131–67.
[15] Nielsen EJ, Kleb WL. Efficient construction of discrete adjoint operators on unstructured grids using complex variables. AIAA J 2006;44(4):827–36.
[16] Giering R, Kaminski T, Slawig T. Generating efficient derivative code with TAF: adjoint and tangent linear euler flow around an airfoil. Future Gener Comput Syst 2005;21(8):1345–55.
[17] Hascoet L, TAPENADE: a tool for Automatic Differentiation of programs. In: Proceedings of the ECCOMAS conference. Jyvaskyla, Finland, July 2004; 2004. p. 14.
[18] Rostaing N, Dalmas S, Galligo A. Automatic differentiation in Odyssée. Tellus A 1993;45:558–68.
[19] Giering R, Kaminski T. Recipes for adjoint code construction. Trans Math Software 1998;24:437–74.
[20] Gockenbach MS. Understanding code generated by TAMC, IAAA Paper TR00-29 2000. Department of Computational and Applied Mathematics, Rice University: Texas, USA; 2000.
[21] Vermeulen PTM, Heemink AW. Model-reduced variational data assimilation. Mon Weather Rev 2006;134:2888–99.
[22] Altaf MU, Heemink AW, Verlaan M. Inverse shallow-water flow modeling using model reduction. Int J Multiscale Comput Eng 2009;7:577–96.
[23] Cao Y, Zhu J, Navon IM, Luo Z. A reduced order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. Int J Numer Meth Fluids 2007;53(10):1571–83.
[24] Daescu DN, Navon IM. A dual-weighted approach to order reduction in 4D-VAR data assimilation. Mon Weather Review 2008;136(3):1026–41.
[25] Fang F, Pain CC, Navon IM, Piggott MD, Gorman GJ, Farrell PE, et al. POD Reduced order 4D-VAR adaptive mesh ocean modelling. Int J Numer Meth Fluids 2009;60(7):709–32.
[26] Bischof C, Carles A, Corliss G, Griewank A, Hovland P. ADIFOR – generating derivative codes from Fortran programs. Sci Program 1992;1:11–29.
[27] Carle A, Fagan M. ADIFOR 3.0 overview, Technical report CAAM-TR-00-02. Rice University; 2000.
[28] Charpentier I. Checkpointing schemes for adjoint codes: application to the meteorological model Meso-NH. SIAM J Sci Comput 2001;22(6):2135–51.
[29] Bischof CH, Bucker HM, Hovland P, Naumann U, Utke J, editors. Lecture notes in computational science and engineering. Advances in automatic differentiation, vol. 64. Springer-Verlag; 2008. p. 370. doi:10.1007/978-3-540-68942-3.
[30] Christakopoulos F, Jones D, Muller JD. Time-stepping for adjoint CFD codes from automatic differentiation. In: Pereira JC, Sequeira A. editors, ECCOMAS CFD 2010. Lisbon, Portugal; 2010.
[31] Zhu K, Navon IM, Zou X. Variational data assimilation with a variable resolution finite- element shallow-water equations model. Mon Weather Rev 1994;122(5):946–65.
[32] Navon IM. Finite-element simulation of the shallow-water equations model on a limited area domain. Appl Math Modell 1979;3(1):337–48.
[33] Navon IM. FEUDX: a two-stage, high-accuracy, finite-element FORTRAN program for solving shallow-water equations. Comput Geosci 1987;13(3):255–85.
[34] Tan WY. Shallow-water hydrodynamics: mathematical theory and numerical solution for a two-dimensional system of shallow-water equations. Beijing: Elsevier Oceanography Series; 1992. p. 434.
[35] Vreugdenhil CB. Numerical methods for shallow-water flow. Boston: Kluwer Academic Publishers; 1994. p. 276.
[36] Galewsky J, Scott RK, Polvani LM. An initial-value problem for testing numerical models of the global shallow-water equations. Tellus 2004;56(5):429–40.
[37] Wang HH, Halpern P, Douglas J, Dupont T. Numerical solutions of the one-dimensional primitive equations using Galerkin approximation with localized basic functions. Mon Weather Rev 1972;100(10):738–46.
[38] Douglas J, Dupont T. Galerkin method for parabolic problems. SIAM J Numer Anal 1970;7:575–626.
[39] Grammeltvedt A. A survey of finite-difference schemes for the primitive equations for a barotropic fluid. Mon Weather Rev 1969;97:384–404.
[40] Fang F, Pain CC, Navon IM, Gorman GJ, Piggott MD, Allison PA. The independent set perturbation adjoint method: a new method of differentiating mesh-based fluids models. J Numer Meth Fluids 2011;66:976–99.
[41] Zienkiewicz OC, Taylor RL, Nithiarasu P. The finite element method for fluid dynamics. 6th ed. Butterworth-Heinemann; 2005. p. 400.
[42] Horwedel JE. GRESS, a preprocessor for sensitivity analysis of Fortran programs. In: Griewank A, editor. Automatic differentiation of algorithms: theory, implementation, and application. Philadelphia: SIAM; 1991. p. 243–50.
[43] Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization mathematical. Programming 1989;45:503–28.
[44] Navon IM, Zou X, Derber J, Sela J. Variational data assimilation with an adiabatic version of the NMC Spectral model. Mon Weather Rev 1992;120:55–79.
[45] Le Dimet FX, Talagrand O. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. Tellus A 1986;38A(2):97–110.
[46] Daley R. Atmospheric data analysis. Cambridge University Press; 1991.
[47] Miller RN, Zaron EO, Bennett AF. Data assimilation in models with convective adjustment. Mon Weather Rev 1994;122:2607–13.
[48] Ide K, Courtier P, Ghil M, Lorenc AC. Unified notation for data assimilation: operational, sequential and variational. J Meteorologcial Soc Jpn 1997;75(1B):181–9.
[49] Brezillon P, Staub JF, Perault-Staub AM, Milhaud G. Numerical estimation of the first order derivative: approximate evaluation of an optimal step. Comput Maths Appls 1981;7:333–47.
[50] Donea J, Huerta A. Finite element methods for flow problems. John Wiley & Sons, Ltd.; 2003. ISBN: 0-471-49666-9.
[51] Errico RM. What is an adjoint model? Bull Am Meteorol Soc 1997;78(11):2577–91.
[52] Pires C, Vautard R, Talagrand O. On extending the limits of variational assimilation in nonlinear chaotic systems. Tellus A 1996;48:96–121.
[53] Maffray Frédéric. On the coloration of perfect graphs. In: Reed Bruce A, Sales Cláudia L, editors. Recent advances in algorithms and combinatorics. CMS books in mathematics, vol. 11. Springer-Verlag; 2003. p. 65–84. doi:10.1007/0-387-22444-0_3.
[54] Gebremedhin AH, Manne F, Pothen A. What color is your Jacobian? Graph coloring for computing derivatives. SIAM Rev 2005;47(4):629–705.