

New data assimilation system DNDAS for high-dimensional models*

Qun-bo Huang(皇群博)^{1,2,3,†}, Xiao-qun Cao(曹小群)^{1,3}, Meng-bin Zhu(朱孟斌)^{1,3},
Wei-min Zhang(张卫民)^{1,3}, and Bai-nian Liu(刘柏年)^{1,3}

¹ College of Computer, National University of Defense Technology, Changsha 410073, China

² Weather Center of PLA Air Force, Beijing 100843, China

³ Academy of Ocean Science and Engineering, National University of Defense Technology, Changsha 410073, China

(Received 17 November 2015; revised manuscript received 7 January 2016; published online 1 April 2016)

The tangent linear (TL) models and adjoint (AD) models have brought great difficulties for the development of variational data assimilation system. It might be impossible to develop them perfectly without great efforts, either by hand, or by automatic differentiation tools. In order to break these limitations, a new data assimilation system, dual-number data assimilation system (DNDAS), is designed based on the dual-number automatic differentiation principles. We investigate the performance of DNDAS with two different optimization schemes and subsequently give a discussion on whether DNDAS is appropriate for high-dimensional forecast models. The new data assimilation system can avoid the complicated reverse integration of the adjoint model, and it only needs the forward integration in the dual-number space to obtain the cost function and its gradient vector concurrently. To verify the correctness and effectiveness of DNDAS, we implemented DNDAS on a simple ordinary differential model and the Lorenz-63 model with different optimization methods. We then concentrate on the adaptability of DNDAS to the Lorenz-96 model with high-dimensional state variables. The results indicate that whether the system is simple or nonlinear, DNDAS can accurately reconstruct the initial condition for the forecast model and has a strong anti-noise characteristic. Given adequate computing resource, the quasi-Newton optimization method performs better than the conjugate gradient method in DNDAS.

Keywords: data assimilation, dual-number, optimization, dual-number data assimilation system

PACS: 05.45.-a, 02.60.-x, 02.30.Zz, 02.60.Pn

DOI: 10.1088/1674-1056/25/5/050502

1. Introduction

Clifford proposed the dual-number theory in 1873,^[1] Yang *et al.*^[2] and Denavit^[3] extended the theory. Currently, the dual-number theory is mainly applied in the spatial mechanisms, robot design, matrix mechanics, and spacecraft pilot. Li^[4] pointed out the importance of the theory in the spatial mechanisms and robot design. Zhang^[5] described the displacement relationship in the spatial mechanisms using the dual-number conception. Wang *et al.*^[6] estimated the rotational motion and the translational motion of the chaser spacecraft relative to the target spacecraft based on the dual-number. Hrdina *et al.*^[7] showed that the dual-number is a proper tool for directly classifying the geometric errors and integrating the multi-axis machines' error models. The dual-number method is similar to the complex-step approximation; Cao *et al.*^[8] applied the complex-variable method to the data assimilation problems of general nonlinear dynamical systems, and also verified the availability for non-differentiable forecast models. However, the complex-step approximation can only be valid for one variable once, which would lead to very expensive cost for large-scale calculation problems. Yu *et al.*^[9] implemented a general-purpose tool to automatically differentiate the forward model using the arithmetic of dual-numbers, which is more accurate and efficient than the popular complex-step ap-

proximation, and the computed derivatives are the same as the analytical results up to machine precision. Andrews^[10] utilized dual-number automatic differentiation to analyze the local sensitivities of three physical models. The dual-number may serve as a means to simplify a model or focus future model developments and associated experiments. Fike^[11] developed the mathematical properties of hyper-dual numbers and then applied the hyper-dual numbers to a multi-objective optimization method. On this basis, Tanaka *et al.*^[12] proposed a method referred to as hyper-dual step derivative (HDS), which is based on the numerical calculation of strain energy derivatives using hyper-dual numbers, does not suffer from either roundoff errors or truncation errors, and is thereby a highly accurate method with high stability. Recently, Cao *et al.*^[13] formally introduced the dual-number theory into the community of data assimilation in numerical weather prediction (NWP), and proposed a new data assimilation method.

Variational principles for various differential equations have been investigated for centuries.^[14] Currently, the most popular data assimilation methods are 3/4D-Var^[15–22] and Kalman filter.^[23–25] The aim of data assimilation is to provide accurate initial state information to the discrete partial differential equations controlling the motions of atmosphere or ocean. A data assimilation system optimizes the combination of the observation data and the prior information, and hence

*Project supported by the National Natural Science Foundation of China (Grant Nos. 41475094 and 41375113).

†Corresponding author. E-mail: hqb09@163.com

provides an optimum estimation of the initial atmospheric state to the NWP model.^[26] Taking the incremental 4D-Var as an example, it measures the distances between the analysis states and observations and also the background states. The cost function can be minimized in terms of increments by an optimization algorithm if the first-order derivatives or gradient of the cost function are calculated and provided.^[27] Due to the high non-linearity of the atmospheric models and observation operators in NWP, the tangent linear and adjoint techniques are widely applied in the field of data assimilation. The TL model is the first-order linearized approximation of the forecast model with respect to the background trajectory, which is used for computing the incremental updates with perturbations to the initial state variables. The TL model approximately calculates the perturbed model trajectory, then the cost function can be attained.^[28] The gradient of the cost function with respect to the initial condition variable is computed by a backward integration of the AD model. Our ultimate goal is to find out the optimal initial condition by an optimization algorithm, of which the gradient of the cost function is very necessary and important input information.

Constructing the TL and the AD models manually is quite complicated and time-consuming, we should first write out the TL model corresponding to the forward forecast model, then construct the AD model. In order to reduce the difficulty of development, the whole forecast model is usually divided into many small segments, and then we write out the TL and the AD models of each segment line by line. Although using this approach makes the development process much simpler, it is still very easy to make mistakes. Furthermore, the actual atmospheric motion equations are very complicated and need some smoothing processes, such as temporal, spatial, and boundary smoothing. These processes will bring great difficulties to us in developing the AD model codes, and sometimes the constructed AD model cannot be completely reversible for the forward integration model.

Because of the drawbacks described above, now a lot of automatic differentiation tools have been developed, such as TAMC^[29] and TAPENADE.^[30] Automatic differentiation is possibly implemented by the strict rules of tangent linear and adjoint coding. But there are still some unresolved issues. (i) It is still not a black box tool. (ii) It cannot handle non-differentiable instructions. (iii) It is necessary to create huge arrays to store the trajectory. (iv) The codes often need to be cleaned-up and optimized.^[27] (v) It has a worse behavior to second- or higher-order derivatives. At present, generating the AD model is not yet fully automatic in reverse mode.^[31] It seems that in the foreseeable future, human intervention will be required to finish the most work of AD in reverse mode.^[32]

The optimization algorithms play a very important role in the variational data assimilation systems. There are two se-

ries of classical optimizing algorithms: gradient-type method and Newton-type method.^[33] The gradient-type method (such as conjugate gradient method, CG) requires only the first-order derivative information, which can be directly employed to large-scale unconstrained optimization problems. But for a non-quadratic cost function, the performance of the CG method deteriorates sharply accompanied by a slow convergence speed. The Newton-type method minimizes the cost function by using the information of the second-order Hessian matrix, and the convergence speed is generally faster than the gradient-type method. In practice, the Hessian of most cost functions are not positive definite symmetric matrices which cannot be represented by an analytical solution, leading to using only the first-order derivative of the cost function (which is called as quasi-Newton method). So far, the BFGS is generally thought to be the best quasi-Newton method which can keep the algorithmic global convergence even if it only uses the one-dimensional inexact searching.

At each iteration of the optimization calculation, it inevitably involves solving and saving the first-order and the second-order Hessian matrix information of the cost function. Furthermore, the dimension of the atmospheric state variables can be up to 10^8 – 10^9 in the data assimilation system. Generally, high-dimensional models have the natural attribute of high nonlinearity, and the majority of the geoscientific systems exhibit strongly non-linear characteristics which are also known as “the curse of dimensionality”.^[34] Currently, the linearization numerical approximation schemes are frequently used in NWP for high-dimensional data assimilation, these approaches would give rise to high resolution regional forecast departure occurring owing to the convergence problems and the calculation accuracy problems. There is a corresponding relationship between the adjoint space and the dual-number space derived from the original space in mathematics. They are equivalent to the adjoint state vector and the model state vector. The dual-number operational rules applied in this paper are strictly without any assumptions and approximations, the convergence and precision of the dual-number method can meet the computing requirements of high-dimensional models. How to quickly and accurately calculate the cost function and its gradient information by optimization algorithms is an important question. Nonlinear differential equations generally do not have analytical solutions, which could be replaced by numerical approximate solutions. However, certain characteristics of the equations tend to be ignored because of the machine truncation error.^[35] The most attractive feature of DNDAS is that it can efficiently and accurately calculate the first-order derivative information without TL and AD models.

In this paper, we design an attractive new data assimilation system, dual-number data assimilation system (DNDAS), based on the dual-number automatic differentiation principles.

DNDAS can accurately determine the cost function and its gradient without developing TL and AD models. The structure of this paper is organized as follows. The following section introduces the dual-number automatic differentiation theory. Section 3 presents the designed architecture of the DNDAS. Section 4 discusses two optimization algorithms. Section 5 analyzes the results of the simulated data assimilation experiments. Finally, conclusions and plans for future work are given in Section 6.

2. Dual-number automatic differentiation principles

A dual-number \hat{x} can be expressed as a pair of ordered real numbers x and x'

$$\hat{x} = x + \varepsilon x', \quad (1)$$

where x is the real unit, x' is the dual unit, and ε is the dual mark which is defined as $\varepsilon^2 = \varepsilon^3 = \dots = 0$, $\varepsilon \neq 0$. To facilitate the representation, equation (1) can be written as $\hat{x} = \langle x, x' \rangle$. The basic algorithms of dual-number are detailed in Ref. [13]. Here we focus on the principles of dual-number automatic differentiation.

The dual-number function is expanded into Taylor series

$$f(\hat{x}) = f(x + \varepsilon x') = f(x) + \varepsilon x' \cdot df(x)/dx, \quad (2)$$

thus the function argument can be extended to the dual-number vector

$$\hat{\mathbf{x}} = \mathbf{x} + \varepsilon \mathbf{x}',$$

then

$$f(\hat{\mathbf{x}}) = \langle f(\mathbf{x}), \nabla f(\mathbf{x}) \cdot \mathbf{x}' \rangle, \quad (3)$$

where $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n)$ and \mathbf{x}, \mathbf{x}' are n -dimensional control variables.

To obtain the derivation of arbitrary function $f(\mathbf{x})$ at point \mathbf{x}_0 , we calculate the function value $f(\langle \mathbf{x}_0, \mathbf{x}' \rangle)$ at the dual-number point \mathbf{x}_0 in the dual-number space, which equals to evaluating $\langle f(\mathbf{x}), \nabla f(\mathbf{x}) \cdot \mathbf{x}' \rangle$.

In order to compute cost function $J(\mathbf{x})$ and its gradient in DNDAS, we construct the corresponding dual-number vector $\hat{\mathbf{x}} = \langle \mathbf{x}_i, \mathbf{e}_i \rangle$ ($i = 1, \dots, n$, \mathbf{e}_i is the unit vector whose i -th element equals to 1). We then substitute the dual-number vector $\hat{\mathbf{x}}$ into the cost function, and the cost function $J(\mathbf{x})$ becomes the dual-number function $\hat{J}(\hat{\mathbf{x}})$. After doing the Taylor expansion, $\hat{J}(\hat{\mathbf{x}})$ becomes

$$\begin{aligned} \hat{J}(\hat{\mathbf{x}}) &= J(\mathbf{x}) + \varepsilon \mathbf{x}' J'(\mathbf{x}) + \varepsilon^2 \mathbf{x}'^2 \frac{J''(\mathbf{x})}{2} + \dots + (\varepsilon \mathbf{x}')^n \frac{J^n(\mathbf{x})}{n!} \\ &= \langle J(\mathbf{x}), \nabla_{\mathbf{x}} J \rangle, \end{aligned} \quad (4)$$

where $\varepsilon^n = 0$ ($n > 1$). We eliminate the component whose order is higher than two. The real and the dual units of $\hat{J}(\hat{\mathbf{x}})$

represent the cost function and its gradient, respectively. As shown in Eq. (4), when calculating the first-order derivative of the cost function with respect to the control variables, we just need to perform forward integration once in the dual-number space.

3. DNDAS

In variational data assimilation, the cost function is usually defined as

$$\begin{aligned} J(\mathbf{x}) &= J_b(\mathbf{x}) + J_o(\mathbf{x}) + J^c \\ &= \frac{1}{2} (\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}_b) \\ &\quad + \frac{1}{2} [\mathbf{y}^o - \mathbf{H}(\mathbf{x})]^T \mathbf{R}^{-1} [\mathbf{y}^o - \mathbf{H}(\mathbf{x})] + J^c, \end{aligned} \quad (5)$$

where $J_b(\mathbf{x})$ is the background term, $J_o(\mathbf{x})$ is the observation term, J^c describes the physical and dynamic constraint conditions which is optional, \mathbf{x} is the control variable vector, \mathbf{x}_b is the background vector (the prior information), \mathbf{y}^o is the observation vector, and \mathbf{B} and \mathbf{R} are the background error covariance matrix and the observation error covariance matrix, respectively. It is seen that the smaller the cost function is, the closer the numerical model solution is to the true atmospheric state. Figures 1 and 2 show the frame of the DNADS and the process of minimization, respectively.

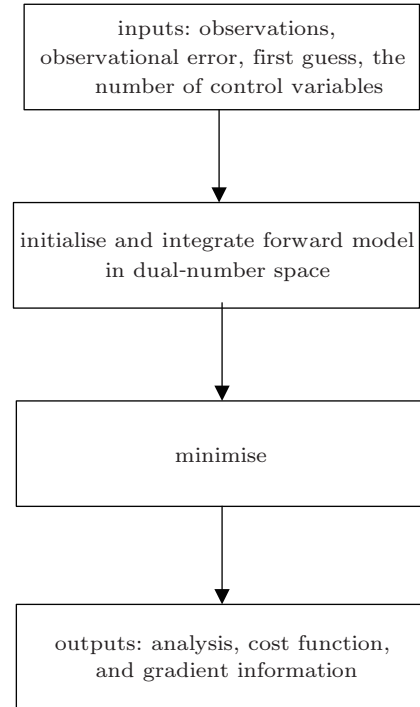


Fig. 1. Frame of DNADS.

Note that, the convergence criterion $\|\nabla_{\mathbf{x}_i} J\| - \|\nabla_{\mathbf{x}_{i+1}} J\| < 10^{-6}$ (where $\|\cdot\|$ gives the L2 norm) is used to reduce the amount of calculation, and to guarantee that the

calculation error is smaller than the machine error. Due to the fact that DNDAS is still in testing phase, we consider the forecast model to be perfect and do not impose the background error in assimilation experiments (Section 5).

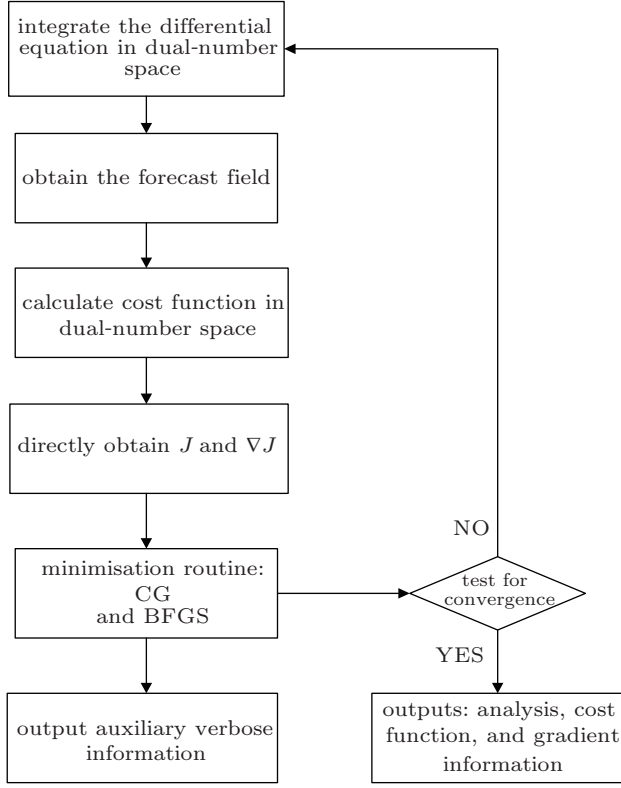


Fig. 2. Minimise and output.

4. Optimization algorithms

After obtaining the cost function and its gradient vector, a descent algorithm is chosen to determine the optimal initial state of the forecast model. Optimization or minimization problems have the general form: $\min f(x)$, $x = (x^1, \dots, x^n) \in \mathbf{R}^n$, x represents the control variables, and n is the number of control variables. $f: \mathbf{R}^n \rightarrow \mathbf{R}$ is the cost function which is required to be continuously differentiable. The general process of optimization calculation is defined as follows.

Step 1 Set an initial value $x_0 \in \mathbf{R}^n$.

Step 2 Calculate the search direction d_k of cost function f at point x_k .

Step 3 Calculate the step length factor α_k by linear search. If $f(x_{k+1}) \leq f(x_k)$, generate a sequence $\{x_k\}$ of control variables x , namely, $x_{k+1} = x_k + \alpha_k d_k$.

Step 4 If the convergence criterion $\|\nabla f(x_{k+1})\| \leq \varepsilon$ is satisfied, then stop. Otherwise, go back to Step 2.

In this paper, the conjugate gradient algorithm is the Beale restarted memoryless variable metric algorithm.^[36] The quasi-Newton algorithm is the BFGS variable metric method.^[37] For convenience, the conjugate gradient method is called CG, the BFGS variable metric method is called BFGS.

CG and BFGS both use the Davidon cubic interpolation^[38] as the basic linear search method to find the best step length. During the calculation process, the step length α must obey the Wolfe criteria,^[39] which are

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k, \quad (6)$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \quad 0 < \rho < \sigma < 1, \quad (7)$$

where $\rho = 0.0001$ and $\sigma = 0.9$ in this paper. The convergence criterion is $\|g\| < \varepsilon \max(1, \|x\|)$, $\varepsilon = 10^{-6}$.

4.1. Conjugate gradient method

The steepest descent direction selected by CG has the feature of conjugacy, so only the first-order derivative information is required during the calculation. The algorithm is described as follows.^[40]

Step 1 Set the initial point x_0 and parameters ρ , σ , ε .

Step 2 Set $k = 0$, and then calculate $f(x_0)$ and $g_0 = g(x_0) = \nabla f(x_0)$. Meanwhile, set $d_0 = -g_0$ and $\alpha_0 = 1/\|g_0\|$.

Step 3 After the initial state is calculated, we need to check whether the minimum is achieved at this point. If $\|g_0\| < \varepsilon \max(1, \|x_0\|)$, then stop the iteration, output $x^* = x_0$. Otherwise, continue.

Step 4 Calculate the initial α_k when non-restart, $\alpha_k = \alpha_{k-1} \|d_{k-1}\|_2 / \|d_k\|_2$. On this basis, make the cubic interpolation to find α_k , the termination condition of linear search is the Wolfe criterion, so that $f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k)$.

Step 5 Set $x_{k+1} = x_k + \alpha_k d_k$.

Step 6 Compute $f(x_{k+1})$, $g(x_{k+1})$, $s_k = x_{k+1} - x_k$, and $y_k = g(x_{k+1}) - g(x_k)$.

Step 7 If $\|g\| < \varepsilon \max(1, \|x\|)$, then stop the iteration. Otherwise, continue.

Step 8 When the iteration $k = 1000$ or $|g_{k+1}^T g_k| \geq 0.2 \|g_{k+1}\|^2$, perform Beale restart, obtain a new search direction d_{k+1} , $k := k + 1$, go back to Step 4. Otherwise, continue.

Step 9 Compute d_{k+1} .

Step 10 Self-adjustment search direction d_{k+1} , $d_{k+1} = [2(f(x_{k+1}) - f(x_k))/d_{k+1}^T g_{k+1}] d_{k+1}$, $k := k + 1$, go back to Step 4.

4.2. BFGS variable metric algorithm method

The BFGS method requires the cost function to be convex and twice continuously differentiable. The iterative solution can be written as^[41]

$$x_{k+1} = x_k + \alpha_k G_k^{-1} d_k, \quad (8)$$

where $G_k^{-1} = \nabla^2 f(x)^{-1}$. We have

$$G_{k+1}^{-1} y_k \approx s_k, \quad (9)$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. $H_{k+1} \approx G_{k+1}^{-1}$ is the approximately inverse Hessian matrix of f at x_{k+1} . It can be found from quasi-Newton

equation $\mathbf{H}_{k+1}\mathbf{y}_k = \mathbf{s}_k$ that \mathbf{H}_{k+1} is updated accompanied by $\{\mathbf{s}_k, \mathbf{y}_k\}$, and avoids direct calculation of the second-order Hessian matrix. We obtain the equation $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{H}_k \mathbf{d}_k$. The BFGS algorithm is described as follows.

Step 1 Start with an estimate \mathbf{x}_0 and the estimate of the initial inverse Hessian matrix \mathbf{H}_0 which is usually the identity matrix \mathbf{I} .

Step 2 If $\|\mathbf{g}_k\| \leq \varepsilon$, then stop. Otherwise evaluate $\mathbf{d}_{k+1} = -\mathbf{H}_{k+1}\mathbf{g}_{k+1}$, and scale the search direction $\|\mathbf{d}_{k+1}\| = \|\mathbf{s}_{k-1}\|$, $k = 1, \dots, n$ which are the previous n iterations.

Step 3 Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$.

Step 4 Evaluate $f(\mathbf{x}_{k+1})$, $\mathbf{g}(\mathbf{x}_{k+1})$, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, and $\mathbf{y}_k = \mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k)$.

Step 5 If $\mathbf{s}_k^T \mathbf{g}_{k+1} > \mathbf{s}_k^T \mathbf{g}_k$ and $\delta f = f(\mathbf{x}_{k+1}) - (f(\mathbf{x}_k) + 0.0001 \mathbf{d}_k^T \mathbf{g}_k) < 0$, then stop the linear search. Go back to Step 2.

Step 6 Perform the Shanno and Kettler's extrapolation^[42] or Davidon's cubic interpolation to estimate a new step length α_k .

Step 7 Update \mathbf{H}_{k+1} by the following formula:

$$\mathbf{H}_{k+1} = (\mathbf{I} - \mathbf{s}_k \mathbf{y}_k^T / \mathbf{s}_k^T \mathbf{y}_k) \mathbf{H}_k (\mathbf{I} - \mathbf{s}_k \mathbf{y}_k^T / \mathbf{s}_k^T \mathbf{y}_k)^T + \mathbf{s}_k \mathbf{s}_k^T / \mathbf{s}_k^T \mathbf{y}_k, \quad k := k + 1,$$

go back to Step 2 and repeat the whole process.

The calculation process of BFGS is similar to that of CG, but there are some differences between them. BFGS considers the impact of the scale factor and updates matrix \mathbf{H} during the linear search process. It requires more memory resources during the calculation process, i.e., $n(n+7)/2$ and $5n+2$, where n is the number of the control variables.^[37]

5. Assimilation experiments

5.1. Simple model

Carbon-11 is a radioactive element with a decay rate of 3.46%/min. The attenuation constant is $\lambda = 2.076 \text{ s}^{-1}$. The atomic number density is A_0 at $t = 0$. The atomic number density differential equation is

$$-\frac{dA}{dt} = \lambda A(t). \quad (10)$$

The analytical solution is $A = A_0 e^{-\lambda t}$.

This first-order ordinary differential equation is treated as a simple dynamic system. Our goal is to assimilate the accurate initial state of the atomic density by using DNDAS.

In the experiment, we integrate Eq. (10) forward by the fourth-order Runge–Kutta method in the real number space, obtaining a numerical integral sequence with the true initial condition $A(0) = A_0 = 100$. The time step length $\Delta t = 0.01$, the integral interval is $[0, 6]$, and the assimilation window length is $[0, 0.06]$. The observations are obtained from the

sequence by imposing Gaussian random error $N(0, \delta)$, where the error mean is zero and the standard deviation is δ . $\delta = 0$ represents that the observation sequence is noiseless.

The first guess of the atomic density is set as $A_b = 8.5$. Considering that the observation should be sparse in reality, three observations at 2nd, 4th, and 6th time steps are selected.

Without considering the background information in the experiment (merely considering the observation influence), the discrete form of the cost function is

$$J(\hat{A}_0) = \frac{1}{2} \sum_{i=1}^I (A_i - A_i^{\text{obs}})^2, \quad (11)$$

where I represents the number of observations, and A_i and A_i^{obs} are the numerical solution and the observation, respectively.

The dual-number variable corresponding to the initial background variable A_b is \hat{A}_0 , $\hat{A}_0 = A_b + \varepsilon$. Then, we perform forward integration in the dual-number space to obtain $\hat{f}(\hat{A})$, whose real unit is the original cost function J and dual unit is the gradient $\nabla_A J$. Finally, we update the initial state by the optimization algorithm.

Figures 3(a) and 3(b) show the cost function and the gradient norm changing with iterations, respectively. It is seen that the cost function and its gradient norm reach convergence after three iterations, and the accuracy of the gradient norm nearly reaches $O(10^{-8})$.

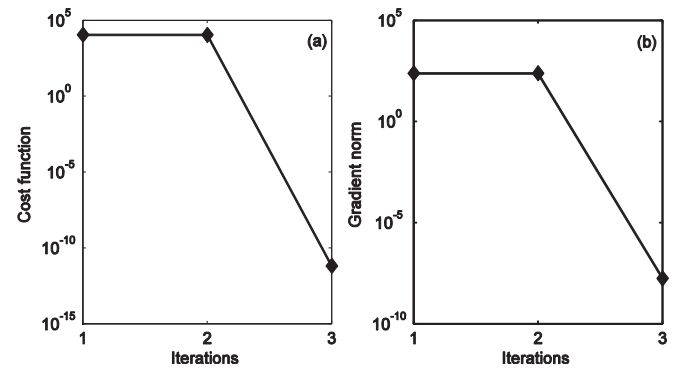


Fig. 3. Trends of (a) the cost function and (b) its gradient norm.

Figure 4 shows that the initial state converges to the truth with the increase of iterations, indicating that we obtain an accurate estimation of the unknown initial state by DNDAS. Figure 5(a) shows the comparison between the assimilated forecast trajectory and observations (without observation error). The two trajectories are very close, which illustrates that the assimilated initial value is very accurate for the forecast model. In the meantime, figure 5(b) shows the comparison between the model forecast trajectory (numerical solution) and the true trajectory (analytical solution), which is consistent with the result in Fig. 5(a).

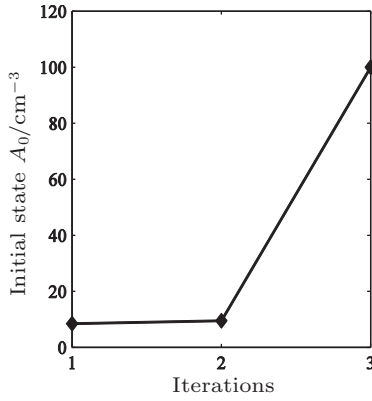
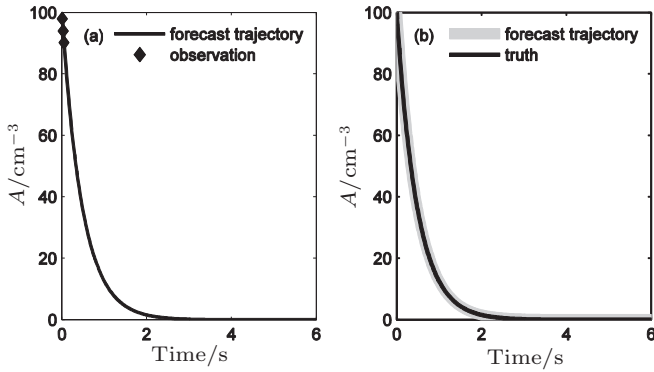

 Fig. 4. Trend of initial atomic density A_0 .


Fig. 5. (a) Comparison between the assimilated forecast trajectory and observations, and (b) comparison between the assimilated forecast trajectory and the true trajectory.

However, there are always errors in the observation data, including observation operator errors and so on. Therefore, we consider the validity of DNDAS with Gaussian observational errors $\delta = 0.1$ and $\delta = 0.2$ in the observational sequences, respectively. Table 1 shows that even if there are observational errors, DNDAS still converges to the true value with high precision, indicating that the new assimilation system has a good anti-noise performance.

Table 1. The assimilation results at different observational error levels.

	Initial state A_0	Cost function J	Iteration
Truth	100.0	/	/
$\delta = 0.0$	100.000064	6.529×10^{-12}	3
$\delta = 0.1$	99.9997799	3.940×10^{-7}	3
$\delta = 0.2$	99.9978279	2.124×10^{-8}	3

5.2. Lorenz-63 chaotic model

Lorenz-63 chaotic system is the most famous nonlinear dynamic system, which is the first example of chaotic dissipative system.^[43] The basis of optimal control to the chaotic system is to control the highly sensitive initial state.^[44] Therefore, how to obtain the accurate initial conditions is crucial. The Lorenz-63 model equations are

$$\begin{aligned} dx/dt &= 10(-x + y), \\ dy/dt &= 28x - y - xz, \end{aligned}$$

$$dz/dt = xy - 8z/3. \quad (12)$$

We integrate the Lorenz-63 equations forward by the fourth-order Runge-Kutta method in the real number space, with the true initial condition of the model $x_t = (x_0, y_0, z_0) = (2.0, 3.0, 2.5)$, time step length $\Delta t = 0.01$, integral interval $[0, 1]$, and assimilation window $[0, 0.2]$. The observations are obtained by adding Gaussian random error $N(0, \delta)$ to the integral sequence of the numerical model.

Set the first guess of the model as $x_b = (12.0, 2.0, 9.0)$. To ensure the effectiveness of the DNDAS, we only select a few observations at 4th, 8th, 12th, 16th, and 20th time steps.

The cost function is defined as follows (without considering the background information):

$$\hat{J}(\hat{x}) = \frac{1}{2} \sum_{i=1}^I \|x_i - x_i^o\|, \quad (13)$$

where I represents the number of observations.

The dual-number vector corresponding to the initial background vector x_b is \hat{x}_0 , $\hat{x}_0 = x_b + \varepsilon x'_b = (\langle x_1, e_1 \rangle, \langle x_2, e_2 \rangle, \langle x_3, e_3 \rangle)$, where e_i ($i = 1, 2, 3$) is the unit vector with the i -th element being 1. We integrate the cost function $\hat{J}(\hat{x})$ in the dual-number space to obtain the original cost function and its gradient (corresponding to the real and the dual units of $\hat{J}(\hat{x})$).

The calculation process of the Lorenz-63 model is similar to that discussed in Subsection 5.1, and we will investigate the impact of two different optimization methods on the DNDAS.

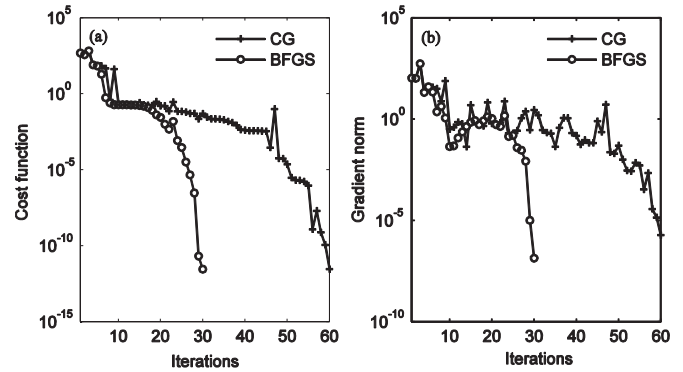
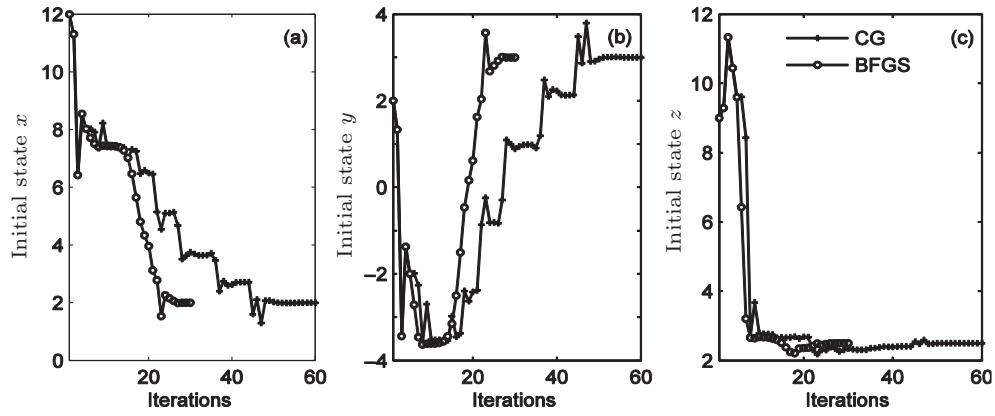


Fig. 6. Trends of (a) the cost function and (b) its gradient norm.

For the cost function and its gradient, without observation error ($\delta = 0$), CG and BFGS can quickly reach convergence after 60 iterations and 30 iterations, respectively, and obtain the same accuracy (Fig. 6). The trends of the cost function and its gradient norm are shown in Figs. 6(a) and 6(b). The trends of control variables x , y , and z changing along the iterations are plotted in Figs. 7(a)–7(c). We can observe from the plots that the three control variables all converge to the truth by using CG and BFGS, however, the iteration number of BFGS is about half less than that of CG in DNDAS.


 Fig. 7. Trends of variables (a) x , (b) y , and (c) z .

To verify the validity of the DNDAS under the existence of observation error, we have done four groups of experiments (experiment groups 1–4) with $\delta = 0.0$, $\delta = 0.1$, $\delta = 0.2$, and $\delta = 0.5$. In each group of experiment, we evaluate and compare the performances of DNDAS (with BFGS and CG methods), 3D-Var, and 4D-Var, wherein 3D-Var and 4D-Var utilize the identical observation conditions and background conditions as used by DNDAS. Table 2 illustrates the results from the DNDAS, it is obvious that no matter of BFGS or CG, the converged values are all close to the true ones at the end of the iterations in DNDAS. Even with $\delta = 0.5$, the accuracy is still

high, which indicates that the DNDAS has an anti-noise feature for the nonlinear chaotic model. Departures of the analysis from the truth in experiment groups 2 and 3 are presented in Fig. 8. Because of the limited space, we only present the departures of variables y (Fig. 8(a)) and z (Fig. 8(b)), the trends of variable x is similar (not shown). In experiment group 2, the departures of variables y and z are a little bigger in DNDAS than those in 3D-Var and 4D-Var when the observation error is small. However, the performance of DNDAS is better at higher observation error levels.

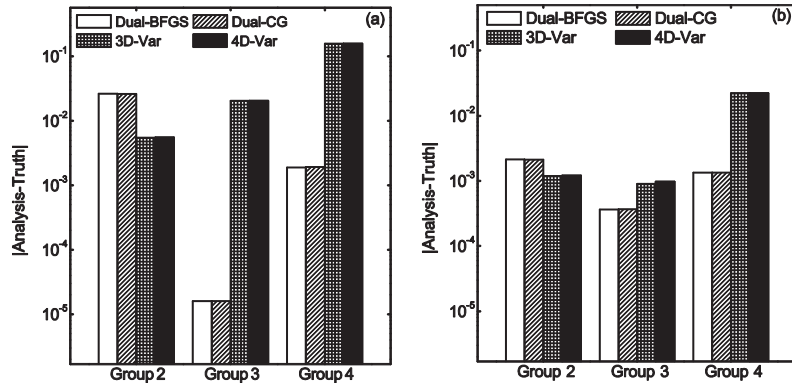
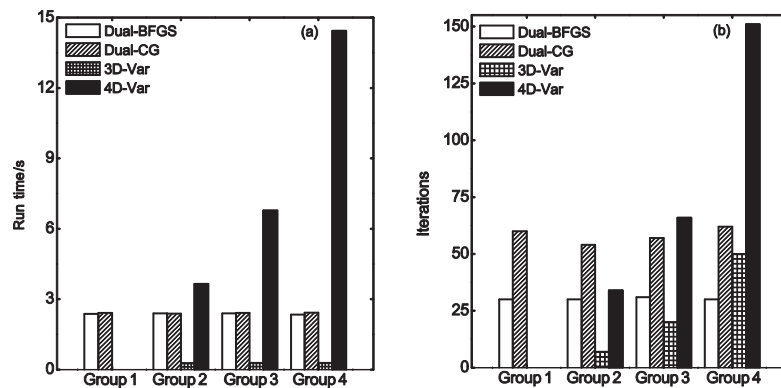

 Fig. 8. The absolute values of departures (analysis minus truth) of variables (a) y and (b) z in the experiments.


Fig. 9. (a) Running time and (b) iterations in the experiments.

Finally, it is found that, for small-scale problems, the running times of CG and BFGS are nearly the same (see Fig. 9(a)), but the iteration number of BFGS is half less than that of CG (see Fig. 9(b)). The main reason is that CG needs to test two step length factor α cases satisfying the Wolfe criterion during the linear searching, whereas BFGS only needs to test one case. Consequently, CG is suitable for large-scale problems, while BFGS is suitable for small-scale problems which can improve the convergence performance in DNDAS. In addition, with the increase of the observation errors, especially for the 4D-Var method, the running time and the iteration number increase significantly, which further evidences the good anti-noise performance of the DNDAS.

Table 2. Assimilation results at different observation error levels.

		x_0	y_0	z_0
Truth		2.0	3.0	2.5
Group 1	CG	2.000004	2.999996	2.500001
	BFGS	2.000001	2.999992	2.500000
Group 2	CG	1.978332	3.026054	2.502130
	BFGS	1.978307	3.026236	2.502133
Group 3	CG	1.999674	2.999984	2.499634
	BFGS	1.999675	2.999984	2.499635
Group 4	CG	2.001449	2.998074	2.498656
	BFGS	2.001410	2.998122	2.498660

5.3. High-dimensional model (Lorenz-96)

The Lorenz-96 model is a dynamical system which was introduced by Lorenz in 1996.^[45] The model equations contain quadratic, linear, and constant terms representing advection, dissipation, and external forcing. Numerical integration indicates that small errors (differences between solutions) tend to double in about 2 days.^[46] It is defined as follows:

$$\frac{dx_j}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + F, \quad (14)$$

where x_j ($j = 1, \dots, N$) represents the model state which is the values of some atmospheric parameters (such as temperature), and F is a forcing constant which is a common value known to cause a chaotic behavior (we usually set $F = 8.0$ for maintaining the unbiased feature of the model). There are N control variables.^[47] The boundary conditions are set to be $x_{-1} = x_{N-1}$, $x_0 = x_N$, $x_{N+1} = x_1$, so the variables x_j form a cyclic chain.

We conduct four groups of experiments, the numbers of control variables are 4, 40, 100, and 400 respectively, and two optimization algorithms (CG and BFGS) are both applied in each group; these experiments are named Exp4-CG/BFGS, Exp40-CG/BFGS, Exp100-CG/BFGS, and Exp400-CG/BFGS.

We integrate the Lorenz-96 equations forward 1000 steps with time step length $\Delta t = 0.001$ by the fourth-order Runge-Kutta method in the real number space. The true initial condi-

tion of the Lorenz-96 model is

$$F = 8.0, \quad x_j^{00} = \begin{cases} F - 0.5, & j = N/2 - 1, \\ F + 0.5, & j = N/2, \\ F + 1.5, & j = N/2 + 1, \\ F, & \text{else.} \end{cases} \quad (15)$$

The previous 100 integral values are divided into 10 intervals in order. The observations are obtained by taking the last value in each interval. Due to the high precision of the fourth-order Runge-Kutta method, we take observations as unbiased.

The first guess of the forecast model is

$$F = 7.0, \quad x_j^{b0} = \begin{cases} F - 3.5, & j = N/2 - 1, \\ F + 3.5, & j = N/2, \\ F + 4.5, & j = N/2 + 1, \\ F, & \text{else.} \end{cases} \quad (16)$$

The cost function is defined as follows (without considering the background information):

$$\hat{J}(\hat{x}) = \frac{1}{2} \sum_{i=1}^I \sum_{j=1}^N \|x_{ij} - x_{ij}^{\text{obs}}\|, \quad (17)$$

where I represents the number of observations.

Subsequently, we construct the dual-number vector as

$$\hat{x}^0 = (\langle x_1, e_1 \rangle, \langle x_2, e_2 \rangle, \langle x_3, e_3 \rangle, \dots, \langle x_N, e_N \rangle).$$

The whole calculation process is similar to that described for the Lorenz-63 experiment. Next we provide the results of our assimilation experiments.

5.3.1. The convergence

Figures 10 and 11 show that the cost function and the gradient norm both reach convergence rapidly with the same precision, and the iteration number of the BFGS is about half less than that of the CG in each experiment except for Exp4-CG/BFGS. It can be concluded that, for the forecast model

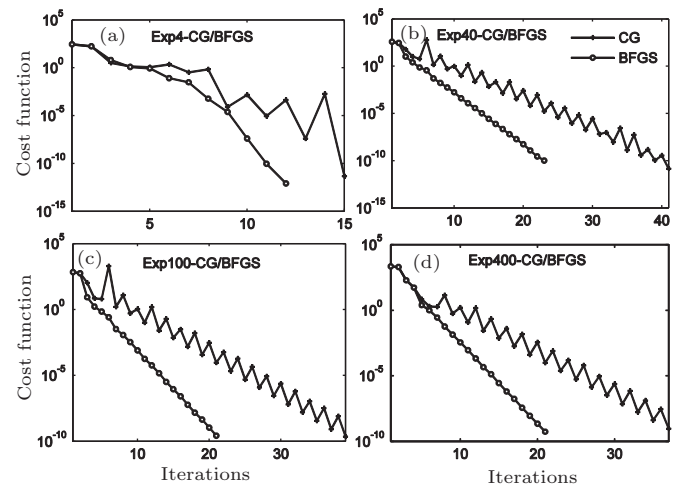


Fig. 10. Trends of the cost function: (a) Exp4-CG/BFGS, (b) Exp40-CG/BFGS, (c) Exp100-CG/BFGS, and (d) Exp400-CG/BFGS.

with middle-high dimensional control variables, the convergence performance of the quasi-Newton method is still generally better than the conjugate gradient method in DADAS. Furthermore, there are only small amounts of change and reduction in iterations and calculation accuracy of the cost function and its gradient norm with the increase of the number of control variables. The above results have demonstrated that DNDAS works well, and it is applicable to the high-dimensional variables data assimilation with stability and reliability.

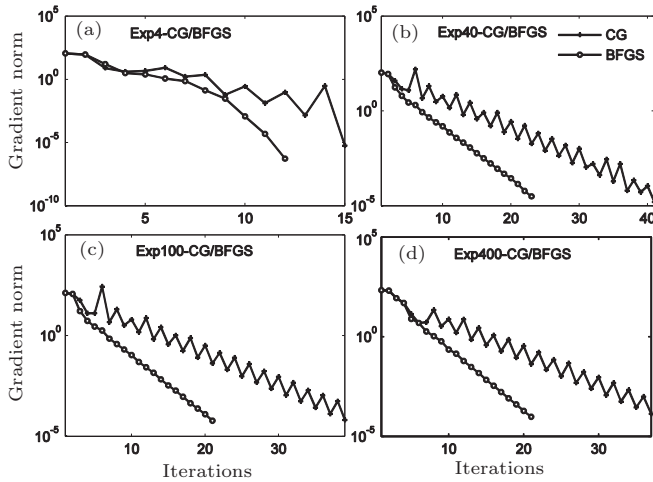


Fig. 11. Trends of the gradient norm: (a) Exp4-CG/BFGS, (b) Exp40-CG/BFGS, (c) Exp100-CG/BFGS, and (d) Exp400-CG/BFGS.

5.3.2. The quality of assimilated results

Figure 12(a) shows the analysis increment (analysis minus observation) and observation increment (observation minus background) in the Exp400-BFGS experiment. Compar-

ing with the observation increment (Fig. 12(b)), the magnitude of the analysis increment in Fig. 12(a) reaches a very small scale $O(10^{-6})$, indicating that the analysis field assimilated by DNDAS is quite close to the truth. Fluctuations of analysis increments in the intermediate zone in Fig. 12(a) also show the correlation among Lorenz-96 model's control variables because we provide different initial values for these variables. The Exp400-CG experiment has a very similar pattern (not shown).

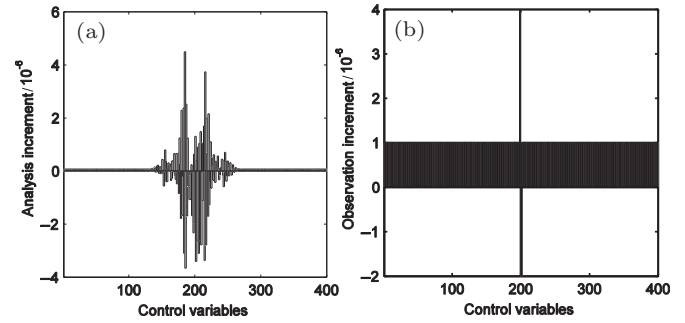


Fig. 12. (a) Analysis increment and (b) observation increment of each control variable in Exp400-BF experiment.

In addition, we restart the forecast model with the assimilated analysis field. Figure 13(a) suggests that the forecast trajectory (broad line) is consistent with the true trajectory (thin line) of the 199th, 200th and 201st control variables. Figure 13(b) shows the consistency between the background trajectory (blue line), the assimilated forecast trajectory (black line), the true trajectory (red line), and the observation (green dot) of the 199th variable. These results also verify the availability for data assimilation by DNDAS.

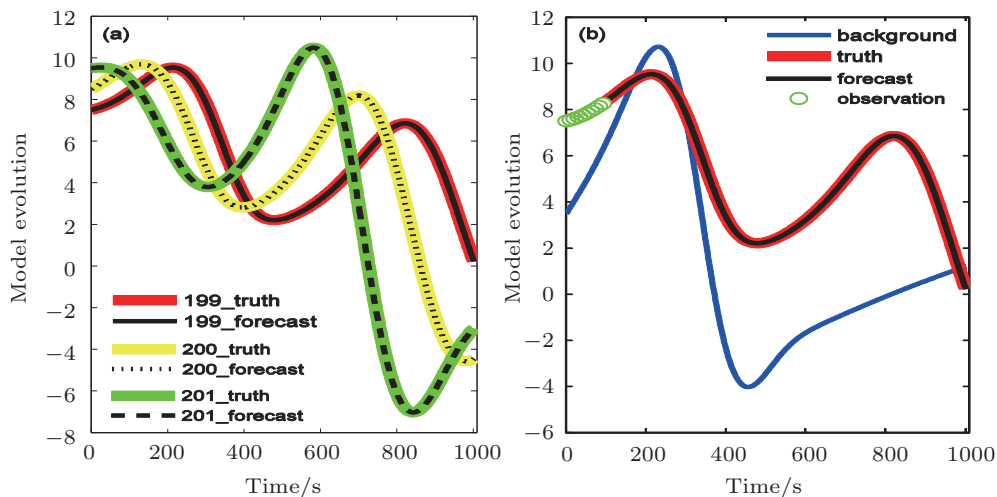


Fig. 13. (color online) (a) Comparison between the assimilated forecast trajectory and the truth of the 199th, 200th, and 201st control variables. (b) Comparison between the assimilated forecast trajectory, the truth, the observation, and the background trajectory of the 199th control variable.

6. Conclusion and perspectives

The results of this study suggest that the DNDAS works well. Whether it is a simple or a nonlinear chaotic model, the

DNDAS can quickly converge to the desired accuracy, indicating that the new assimilation system has a certain universality. The fact that the DNDAS is still competent under the condi-

tion of imposing observation error and not introducing background information indicates its robustness. The framework of DNDAS is so simple that it can be coded and transplanted to other computing platforms flexibly.

For DNDAS, the optimization methods (such as CG and BFGS) have a slight influence on the running time. The iteration number of BFGS is about half less than that of CG, however, BFGS needs more memory storage space and can be too expensive for large scale applications. We should do more experiments to determine the most efficient optimization algorithm.

Results in Subsection 5.3 show that the DNDAS is suited for forecast models with middle-high dimensional control variables. Moreover, the convergence speed and the calculation accuracy will not increase with the expansion of the dimension in the control variables.

The DNDAS in the paper is only applicable to calculate the first-order derivatives. If you wish to get the higher-order derivatives, a more advanced theory needs to be explored, such as the hyperdual number theory. The innovative approach might be more suitable for computing the second-order Hessian matrices information in minimization algorithms, bringing about more exact second-order derivatives than the traditional methods. Further benefits generated from the advanced theory may still have to be verified.

In summary, the DNDAS system performs well in high-dimensional forecast models and has potential to become a new data assimilation system. We aim to move to practical application of the DNDAS in the future. It is expected that more advanced optimization algorithms and the hyper-dual number theory will be the next promising tools to improve the DNDAS.

References

- [1] Clifford W K 1871 *Proceedings of the London Mathematical Society* (London: Oxford University Press) p. 381
- [2] Yang A T and Freudenstein F 1964 *J. Appl. Mech.* **31** 300
- [3] Denavit J 1956 *Description and Displacement Analysis of Mechanisms Based on (2x2) Dual Matrices* (Ph.D. Dissertation) (Evanston: Northwestern University)
- [4] Li D W 1992 *J. South China Agr. Univ.* **13** 88 (in Chinese)
- [5] Zhao T Z 1994 *J. B. Inst. Techno.* **14** 50 (in Chinese)
- [6] Wang J Y, Liang H C, Sun Z W and Zhang S J 2012 *Acta Aeronaut. Astronaut. Sin.* **33** 1881 (in Chinese)
- [7] Hrdina J and Vasik P 2014 *J. Appl. Math.* **91** 688
- [8] Cao X Q, Song J Q, Zhang W M, Zhao Y L and Liu B N 2013 *Acta Phys. Sin.* **62** 170504 (in Chinese)
- [9] Yu W B and Blair M 2013 *Comput. Phys. Commun.* **184** 1446
- [10] Andrews J M 2013 *J. Fluids Eng.* **135** 0612061
- [11] Fike J A 2013 *Multi-objective Optimization using Hyper-dual Numbers* (Ph. D. Dissertation) (Palo Alto: Stanford University)
- [12] Tanaka M, Sasagawa T, Omote R, Fujikawa M, Balzani D and Schroder J 2015 *Comput. Methods Appl. Mech. Engrg.* **283** 22
- [13] Cao X Q, Huang Q B, Liu B N, Zhu M B and Yu Y 2015 *Acta Phys. Sin.* **64** 130502 (in Chinese)
- [14] Cao X Q, Song J Q, Zhang W M and Zhao J 2011 *Chin. Phys. B* **20** 090401
- [15] Zhang L, Huang S X, Shen C and Shi W L 2011 *Chin. Phys. B* **20** 0129201
- [16] Zhang L, Huang S X, Shen C and Shi W L 2011 *Chin. Phys. B* **20** 0119201
- [17] Zhao G Z, Yu X J, Xu Y and Zhu J 2010 *Chin. Phys. B* **19** 070203
- [18] Zhong J, Fei J F, Huang S X and Du H D 2012 *Acta Phys. Sin.* **14** 149203 (in Chinese)
- [19] Zhang W M, Cao X Q and Song J Q 2012 *Acta Phys. Sin.* **24** 249202 (in Chinese)
- [20] Zhu M B, Zhang W M, Cao X Q and Yu Y 2014 *Chin. Phys. B* **23** 069202
- [21] Zhu M B, Zhang W M, Cao X Q 2013 *Acta Phys. Sin.* **63** 189203 (in Chinese)
- [22] Cao X Q, Song J Q, Zhu X Q, Zhang L L, Zhang W M and Zhao J 2012 *Chin. Phys. B* **21** 020203
- [23] Leng H Z, Song J Q, Cao X Q and Yang J H 2012 *Acta Phys. Sin.* **61** 070501 (in Chinese)
- [24] Hu G G, Gao S S, Zhong Y M and Gao B B 2015 *Chin. Phys. B* **24** 070202
- [25] Leng H Z and Song J Q 2013 *Chin. Phys. B* **22** 030505
- [26] Liu B N, Zhang W M, Cao X Q, Zhao Y L, Huang Q B and Luo Y 2015 *Chin. J. Geophys. Ch.* **58** 1526 (in Chinese)
- [27] Benedetti A 2014 *ECMWF Training Course*
- [28] Errico M R, Vukicevic T and Raeder K 1993 *Tellus* **45** 462
- [29] Giering R and Kaminski T 1998 *Trans. Math. Software* **24** 437
- [30] Hascoet L 2004 *Proceedings of the ECCOMAS Conference*, July 24–28, 2004, Jyväskylä, Finland
- [31] Griewank A 1989 *Mathematical Programming: Recent Developments and Applications* (Tanabe: Kluwer Academic) p. 83
- [32] Sambridge M, Rickwood P, Rawlinson N and Sommacal S 2007 *Geophys. J. Int.* **170** 1
- [33] Zhu X Q 2008 (Ph. D. Dissertation) (Changsha: National University of Defense Technology) (in Chinese)
- [34] Snyder C, Bengtsson T, Bickel P and Anderson J 2008 *Mon. Wea. Rev.* **136** 4629
- [35] Wu Q K 2012 *Acta Phys. Sin.* **61** 020203 (in Chinese)
- [36] Shanno D F and Phua K H 1978 *Math. Program* **14** 149
- [37] Shanno D F and Phua K H 1980 *ACM Trans. On Math. Software* **6** 618
- [38] Davidon W C 1991 *SIAM J. Optimiz.* **1** 1
- [39] Wolfe P 1971 *SIAM Rev.* **23** 054203
- [40] Andrei N 2008 *ICI Technical Report 13/08*
- [41] Shanno D F and Phua K H 1976 *ACM Trans. Math. Software* **2** 87
- [42] Shanno D F and Kettler P C 1970 *Math. Comput.* **24** 657
- [43] Lorenz E N 1963 *J. Atmos. Sci.* **20** 130
- [44] Cao X Q 2013 *Acta Phys. Sin.* **62** 230505 (in Chinese)
- [45] Lorenz E N 1996 *ECMWF Proc. Seminar* **1**
- [46] Lorenz E N and Emanuel K A 1998 *J. Atmos. Sci.* **55** 399
- [47] Ott E, Hunt B R, Szunyogh I, Zimin A V, Kostelich E, Corazza M, Kalnay E, Patil D J and Yorke J A 2004 *Tellus A* **56** 415