A data assimilation algorithm for predicting rain

Janjic, Tijana; Ruckstuhl, Yvonne; TOINT, Philippe

Published in:
Quarterly Journal of the Royal Meteorological Society

Publication date:
2020

Document Version
Early version, also known as pre-print

ORIGINAL ARTICLE

A data assimilation algorithm for predicting rain

Tijana Janjić,[1]* | Yvonne Ruckstuhl[1]* | Philippe L. Toint[2]*

[1]Meteorological Institute, Ludwig-Maximilians-University Munich, München, 80333, Germany

[2]Namur Centre for Complex Systems, University of Namur, Namur, Belgium

Correspondence
Tijana Janjić, Meteorological Institute, Ludwig-Maximilians-University Munich, München, 80333, Germany
Email: tijana.pfander@lmu.de

Funding information
Transregional Collaborative Research Center SFB / TRR 165 “Waves to Weather" funded by the German Science Foundation (DFG) through subproject B6: ”Parameter estimation using a data assimilation system for improved representation of clouds"; DFG funding for project ”Conservation laws and ensemble Kalman filter algorithms" (DFG JA1077/3-1) and funding of T. Janjic Heisenberg Award (DFG JA1077/4-1) .

Convective scale data assimilation utilizes high resolution numerical weather prediction models and temporally and spatially dense observations of relevant atmospheric variables. In addition, it requires a data assimilation algorithm, that is able to provide initial conditions of all relevant fields in a way that predicting key forecast phenomena such as convective precipitation and severe wind gusts is possible. There are many challenges in designing data assimilation algorithms for this application. These challenges, for example, include treatment of deficiencies in uncertainty quantification of the model, in particular for representation of intermittent convection. In addition, there are computational challenges, as the state vector is of large size with one third or more of variables containing prognostic hydrometeors variables whose non-negativity needs to be preserved, and as the state vector requires a high updating frequency in order to catch fast changing convection. Finally, often, not only one, but rather an ensemble of predictions is needed in order to correctly specify, for example, the uncertainty of rain at a particular location and initialize the probabilistic forecasts. This even further increases the computational considerations.

Governed by the principle of preservation of physical properties in order to represent features of real atmosphere and the computational efficiency, a specialized algorithm for quadratic optimization (QO, or, formerly, QP) with disjoint linear constraints is presented here. Numerical results

on a simple model designed for testing convective scale data assimilation, show that this algorithm reduces the computational burden compared to more general quadratic programming methods as for example interior point methods, that also impose constraints strictly. In particular, if constraints are disjoint and the rank of the set of linear equality constraints is small, further reduction in computational costs can be achieved.

KEYWORDS
data assimilation, disjoint linear constraints, quadratic optimization, preservation of non-negativity, convective scale predictions

1 | INTRODUCTION

Convection permitting models with single digit horizontal resolution in kilometer partly resolve highly nonlinear dynamics and physics at a wide range of spatial and temporal scales. Advances in data assimilation algorithms are needed to provide initial conditions for these models. In particular there is a need of more frequent assimilation of data in order to catch fast changing convection, better representation of time evolving multivariate covariances, as well as a need for efficient algorithm for updating prognostic hydrometeors values.

In operational practice if incremental three-dimensional variational assimilation (3D-Var, Courtier et al., 1998), four-dimensional variational assimilation (4D-Var, Rabier et al., 2000) or localized ensemble transform Kalman filter (LETKF, Bishop et al., 2001; Hunt et al., 2007) are used then prognostic hydrometeors of cloud microphysics scheme as for example rain, graupel or snow are not updated with these algorithms. Instead, one uses the adjusted state variables to infer hydrometer values through the microphysical scheme or update them using latent heat nudging approach or a cloud analysis scheme (Gustafsson et al., 2018). The alternative to this operational practice would be to estimate them in the state vector, allowing hydrometeors to be modified consistently with other variables. However due to Gaussian assumption, sparse observations and other approximations in these methods, non physical values could be obtained. Therefore, one would need to use physical consistency checks after the estimates are obtained. These checks for example would include setting the negative values of rain, graupel and snow to zero. If in addition an ensemble of simulations is used to construct covariances as in ensemble data assimilation algorithms or hybrid variational algorithms, then consistency checks would be made for each ensemble member, thereby modifying artificially the uncertainty representation. These techniques clearly interfere with the optimality of the computed result and the quality of the resulting prediction, especially given the ill-conditioning of the problem.

The ensemble Kalman filter (EnKF) (Evensen, 1994; Burgers et al. , 1998; Evensen, 2003) algorithm is capable of handling complex and highly nonlinear processes, and for its application the adjoint of the model is not needed. This makes it attractive for convective scale applications. Since the proper estimates of hydrometeors are crucial for prediction on convective scales, the question arises whether the EnKF method can be modified to improve these estimates. If such modification could be found one could as well optimise for example the use of radar observations to initialize numerical weather prediction (NWP) models due to the importance of this data set for prediction of convective storms (Sun et. al, 2014). Various approaches can be taken in the EnKF framework in order to deal with the non-Gaussian

errors. For example, variables can be transformed by assuming that the relevant state variables follow appropriate pre-specified non-Gaussian distributions, such as the lognormal (Cohn, 1997) and truncated Gaussian (Lauvernet et al., 2009) distribution or, more generally, by carrying out a parameterized change of the state variables known as Gaussian anamorphosis (Simon and Bertino, 2009). Lien et al. (2013), for example, apply Gaussian anamorphosis on the assimilation of precipitation with the LETKF.

Instead of the above statistical based approaches, Janjic et al. (2014) modified the EnKF algorithm to include constraints on the state vector. The Janjic et al. (2014) algorithm combines the EnKF and quadratic programming (QP), conserves exactly linear equality and inequality constraints in each of the ensemble members and produces multivariate estimates for all fields. The new method developed outperformed the EnKF as well as the EnKF with a lognormal change of control variables. As argued in the paper, the reason for this is that each of these statistical methods preserves mass (EnKF) or positivity (lognormal EnKF) but not both. Only once both the positivity and mass are preserved in the new algorithm, the good estimates of the fields are obtained. Further, Ruckstuhl and Janjić (2018) extended this work to show that similar conclusions hold for a nonlinear multivariable model designed to test convective scale data assimilation applications. Both of these papers used an active set or interior point quadratic programming algorithm for solving the constrained minimization as implemented in matlab (Gill et al., 1981, 1984) and python (Andersen et al., 2010). Although these results convincingly illustrate the benefits of including the constraints in the minimization, the QP algorithm has turned out to be difficult to implement in practice for applications such as weather forecasting at the convective scale, as explained next.

Let us consider the problem

$$\min_{x,y} \mathcal{J}(x,y) \stackrel{\text{def}}{=} (g_x^T, g_y^T) \begin{pmatrix} x \\ y \end{pmatrix} + \frac{1}{2}(x^T y^T) \begin{pmatrix} P_{xx} & P_{xy} \\ P_{xy}^T & P_{yy} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.1}$$

subject to

$$Ax = b \quad \text{and} \quad y \geq \ell \tag{1.2}$$

where x and g_x belong to R^{n-p}, y and g_y to R^p, P_{xx} is an $n-p \times n-p$ symmetric real matrix, P_{yy} a $p \times p$ symmetric real matrix, P_{xy} a $n-p \times p$ real matrix, A an $m \times n-p$ real matrix with $m \leq n-p$ of rank m, $b \in \mathrm{R}^m$ and the inequality is understood component-wise. It is easy to extend our discussion to more general bound constraints where one requires $\ell \leq y \leq u$ for some $\ell, u \in \mathrm{R}^p$ with $\ell \leq u$, both of them possibly having infinite components. We refer to a problem of the type (1.1)-(1.2) as having *disjoint constraints* in the sense that the two sets of constraints of (1.2) involve disjoint sets of variables. In what follows, we focus on the convex case, and assume that

$$P \stackrel{\text{def}}{=} \begin{pmatrix} P_{xx} & P_{xy} \\ P_{xy}^T & P_{yy} \end{pmatrix}$$

is positive definite.

Typical data assimilation algorithms produce the initial condition of the model by minimizing, perhaps iteratively in a Gauss-Newton framework, an objective function of type (1.1). Minimization is usually performed every hour in convective scale applications using the new measurements of the atmosphere. The result of the minimization is a correction to a prediction of the numerical model for a given time. In this application, the vector $z = (x, y)$ to be estimated consists of variables describing the state of the atmosphere at a given time (such as pressure, temperature, wind direction and speed, ...) in all grid points of the numerical model. Its size n ranges between 10^6 and 10^9, resulting

in truly large-scale problems. The vector y usually describes different water phases such as rain, graupel and snow at all grid points. The physical nature of these variables implies that they have to be non-negative. In this case, p is approximately one third of n, which remains very large, and could be even larger if for example second moment microphysics scheme is used instead of first moment one in convection permitting model.

Both the size of vectors x and y and the frequency of their estimation that is usually an hour or less, significantly limit the number of iterations of the minimization algorithm that can be performed. Further, in ensemble setting, not only one but rather an ensemble of estimates is produced in order to correctly specify uncertainty, even further increasing the computational considerations. The computational cost also depends on the number and the nature of the constraints with inequality constraints are more challenging than equality constraints. Minimization algorithms, such as interior point methods and sequential quadratic programming, handle inequality constraints either by incorporating them in the cost function with a penalty term, or by solving a sequence of equality constrained subproblems instead. It is the purpose of this short paper to show that particular optimization algorithms can be combined in a way that is significantly less expensive in computer time, thereby making a practical application affordable. While these considerations are based on data assimilation for weather forecasting, we note that the methods discussed here are also applicable to similar contexts in a wide variety of problems including chemistry, ecosystems and ocean data assimilation (Bertino et al., 2003; Simon and Bertino, 2009, 2012; Bühner et al., 2013), to mention a few.

In Section 2, we present two algorithms which exploit the fact that the constraints are disjoint. Section 4 discusses the data assimilation results when applying these algorithms on a known convective-scale example due to Würsch and Craig (2014) and their performance. Section 3 describes our experimental setup. A discussion and some perspectives are presented in Section 5.

2 | THE ALGORITHMS

2.1 | QO algorithm for disjoint constraints

For solving inequality constraint problem, it is beneficial to distinguish between so called active constraints and inactive constraints. The constraints that meet the bounds are considered active and define the working set $\mathcal{A}_k$ at iteration k. The other constraints are considered inactive. Subspace spanned by inactive constraints is called working face.

If v is a vector in R^p and $\mathcal{A} \subseteq \{1, \ldots, p\}$, we denote by $v^{\mathcal{A}}$ the vector v reduced to its active components, that is

$$v^{\mathcal{A}} = \begin{cases} v_i & \text{if } \; i \in \mathcal{A} \\ \ell_i & \text{otherwise,} \end{cases}$$

where subscript i denotes the i-th component of a vector. Similarly, $M^{\mathcal{A}}$ is the matrix M reduced to its active columns (and rows, if it is symmetric).

For solving problem (1.1)–(1.2), we propose an active-set algorithm whose feature is to maintain feasibility with respect to the linear equality constraints on x at all iterations, while at the same time using classical projection techniques (Conn et al., 2000, Chapter 12) to enforce feasibility of y. The working set is our prediction or a guess of active set, and the working set corresponding to solution is called active set. Active set algorithms aim to find the active set, by iteratively adding and removing constraints from the working set and solving corresponding equality constrained problems. Our quadratic optimization (QO) algorithm for disjoint constraints is stated as Algorithm 2.1 on the following page.

Algorithm 2.1: QO algorithm for disjoint constraints

Step 0: Initialization. A feasible starting point (x_0, y_0) is given (i.e., $Ax_0 = b$, $y_0 \geq \ell$), as well as an accuracy threshold $\epsilon > 0$. Set $k = 0$.

Step 1: Active-set update.

$$\mathcal{A}_k \stackrel{\text{def}}{=} \{i \in \{1, \ldots, p\} \mid [y_k]_i = \ell_i \text{ and } \nabla_y \mathcal{J}(x_k, y_k) > 0\} \tag{2.1}$$

$$\mathcal{A}_k^c \stackrel{\text{def}}{=} \{i \in \{1, \ldots, p\} \mid i \notin \mathcal{A}_k\} \tag{2.2}$$

Step 2: Termination test. Terminate if $\|[\nabla_y \mathcal{J}(x_k, y_k)]^{\mathcal{A}_k^c}\| \leq \epsilon$.

Step 3: Search direction computation. Solve

$$\begin{pmatrix} P_{xx} & P_{xy}^{\mathcal{A}_k^c} & A^T \\ (P_{xy}^{\mathcal{A}_k^c})^T & P_{yy}^{\mathcal{A}_k^c} & 0 \\ A & 0 & 0 \end{pmatrix} \begin{pmatrix} s_k \\ v_k^{\mathcal{A}_k^c} \\ w_k \end{pmatrix} = - \begin{pmatrix} \nabla_x^{\mathcal{A}_k^c} \mathcal{J}(x_k, y_k) \\ \nabla_y^{\mathcal{A}_k^c} \mathcal{J}(x_k, y_k) \\ 0 \end{pmatrix} \tag{2.3}$$

Step 4: Projected search. Determine $\alpha > 0$ such that (x_{k+1}, y_{k+1}) is the first minimizer of $\mathcal{J}\left(x_k + \alpha s_k, \max\left[y_k + \alpha v_k, \ell\right]\right)$, where v_k is obtained from $v_k^{\mathcal{A}_k^c}$ by setting $[v_k]_i = 0$ for $i \in \mathcal{A}_k$.

To understand the properties of the gradient projected algorithm, let us first consider case of an optimization problem only for variable y. Let y_k be a feasible estimate of y with corresponding working set $\mathcal{A}_k$ consisting of indices i of vector y_k for which $l_i = [y_k]_i$. The next iterate is found in two steps. In first step the minimum is searched along the descent direction, when bound is encountered before the minimum is found the search direction is bent so that it stays feasible. The resulting step size found through Step 4 gives the Cauchy point with a corresponding new working set. In the second step, minimization is performed for obtaining new elements of the vector y that are outside of the working set, i.e. where the bounds are not reached. One would update the working set when a new bound is encountered and restart the procedure in the new (lower dimensional) working face. The resulting feasible estimate defines the next iterate and its active set.

In case of estimation problem for both x and y, at each iteration k a working set is given through (2.1). Working set represents all components of y that are active at iteration k and would leave feasible region if moved along the steepest descent direction. Therefore all of these components of y vector can be temporarily assumed constant and are excluded from search direction. The search direction is found by solving system (2.3) in working face. The step size computation is determined through Step 4 thereby ensuring feasibility of y, while remaining feasibility with respect to the equality constraint. This only holds because the constraints are disjoint. Once all of y components outside of the working set are optimized i.e. $\nabla_y^{\mathcal{A}_k^c} \mathcal{J}(x_k, y_k) = 0$ the solution is found.

If not available on the onset, a feasible point can be computed by solving a linear least-squares problem for x_0 and choosing any $y_0 \geq \ell$. The third line of (2.3) imposes that $As_k = 0$. It is important that this equation be satisfied to high precision if exact feasibility with respect to the linear equality constraint is to be preserved. We refer the reader to Gould and Toint (2002) for a discussion of this point. With this caveat, the system (2.3) can be solved using a Krylov solver like MINRES or GMRES (see Saad, 1996, for a description of these methods), or by a "constrained preconditioned" conjugate gradient method (see Gill et al., 2001; Gould and Toint, 2002). If this is the case, any preconditioner must also be reduced (in its y part) to the subset of currently active variables $\mathcal{A}_k$. If dimension and sparsity of P allows (which is typically not the case in weather forecasting), a stable factorization can also be used to solve (2.3) accurately.

2.2 | The projected algorithm

Often matrix A has a very simple form, for example when representing conservation of mass. In example presented in Section 3 it is of size $1 \times n - p$ and therefore of rank one. We may then easily project the problem into the nullspace of A by defining Z, the projection onto this nullspace, and applying the change of variable $x = Z\tilde{x}$ for $\tilde{x} \in \mathrm{R}^{n-p}$, which leads to the problem

$$\min_{\tilde{x},y} \tilde{\mathcal{J}}(\tilde{x}, y) \stackrel{\text{def}}{=} ((Z^T g_x)^T, g_y^T) \begin{pmatrix} \tilde{x} \\ y \end{pmatrix} + \frac{1}{2} (\tilde{x}^T y^T) \begin{pmatrix} Z^T P_{\tilde{x}x} Z & Z^T P_{xy} \\ P_{xy}^T Z & P_{yy} \end{pmatrix} \begin{pmatrix} \tilde{x} \\ y \end{pmatrix} \tag{2.4}$$

subject to

$$y \geq \ell. \tag{2.5}$$

Problem (2.4)-(2.5) is now a bound-constrained quadratic problem, to which standard techniques can be applied, including for large-scale instances (see Conn et al. (1992); Dostál (1997); Lin and Moré (1999); Hungerlaender and Rendl (2015), for example). In our context, where the number of iterations and storage is often limited, it is important to consider an algorithm whose first iterations are cheap and efficient. In particular, the size of the typical practical prob-

lem precludes methods using matrix factorizations. A well-regarded suitable method applies the Conjugate Gradients (CG) (Hestenes and Stiefel, 1952) to minimize the quadratic in the current face, that is the subspace spanned by the inactive variables at the current iterate, restarting the procedure as needed when new bound constraints become active during the calculation and a new (lower dimensional) face must be explored. Note that the first face contains the negative gradient of the inactive variables, and the first step of CG performs a (in this case projected) line search along this direction, yielding what is known as the generalized Cauchy point. Methods differ essentially by their face changing mechanisms but insist that constraints active at the Cauchy point are not made inactive during the rest of the restarted CG steps. A simple version of the resulting algorithm (based on Conn et al. (1992)) is now stated as Algorithm 2.2 on page ??.

3 | NUMERICAL EXPERIMENTS

In order to illustrate the behaviour of algorithms presented in Section 2 in data assimilation context, we use the modified shallow water model of Würsch and Craig (2014) and twin experiments. In this section we describe the details of our experimental setup.

3.1 | Model

The modified shallow water model has been used for testing different data assimilation algorithms in Haslehner et al. (2016); Ruckstuhl and Janjić (2018). The model is based on the shallow water equations, which have been utilized for a long time in testing both numerical discretization schemes (Sadourny, 1975; Arakawa and Lamb, 1980; Cullen et al., 1997; Sommer and Névir, 2009; Ketefian and Jacobson, 2009) as well as data assimilation algorithms (Cohn and Parrish, 1991; Zeng and Janjić, 2016; Zeng et al., 2017). As the name suggests, in Würsch and Craig (2014) the shallow water equations have been altered in order to mimic key aspects of convection. To that end, a third variable rain r was introduced in addition to the velocity (or wind) u and water height level h fields. The one-dimensional modified shallow water model consists of following equations:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial(\phi + \gamma^2 r)}{\partial x} = \beta_u + D_u\frac{\partial^2 u}{\partial x^2}, \tag{3.1}$$

with

$$\phi = \begin{cases} \phi_c & \text{if } \; h > h_c \\ gh & \text{otherwise,} \end{cases} \tag{3.2}$$

$$\frac{\partial r}{\partial t} + u\frac{\partial r}{\partial x} = D_r\frac{\partial^2 r}{\partial x^2} - \eta r - \begin{cases} \delta\frac{\partial u}{\partial x}, & \text{if } \; h > h_r \;\text{ and }\; \frac{\partial u}{\partial x} < 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

$$\frac{\partial h}{\partial t} + \frac{\partial(uh)}{\partial x} = D_h\frac{\partial^2 h}{\partial x^2}. \tag{3.4}$$

The parameters D_u, D_r, D_h are diffusion constants in equations for momentum, rain mass and continuity equation. Here, g is the gravity constant and $\gamma = \sqrt{gh_0}$ is the gravity wave speed for the absolute fluid layer h_0. To initiate con-

Algorithm 2.2: Projected QO algorithm for disjoint constraints

Step 0: Initialization. A feasible starting point $(\tilde{x}_0, y_0)$ is given (i.e. $y_0 \geq \ell$), as well as an accuracy threshold $\epsilon > 0$. Compute the projection Z onto the null space of A and set $k = 0$.

Step 1: Active-set update. Define

$$\mathcal{A}_k \overset{\text{def}}{=} \{i \in \{1, \ldots, p\} \mid [y_k]_i = \ell_i\} \quad \text{and} \quad \mathcal{A}_k^c \overset{\text{def}}{=} \{1, \ldots, p\} \setminus \mathcal{A}_k. \tag{2.6}$$

Step 2: Termination test. Terminate if the following conditions hold:

- $\|[\nabla_y \tilde{\mathcal{J}}(\tilde{x}_k, y_k)]^{\mathcal{A}_k^c}\| \leq \epsilon$
- $\|\nabla_{\tilde{x}} \tilde{\mathcal{J}}(\tilde{x}_k, y_k)\| \leq \epsilon$
- $\nabla_{y_i} \tilde{\mathcal{J}}(\tilde{x}_k, y_k) \geq 0 \quad \text{for} \quad i \in \mathcal{A}_k$.

Step 3: Find the Cauchy point and determine its active set. Determine $\alpha > 0$ such that $(\tilde{x}_k^c, y_k^c)$ is the first minimizer of

$$\tilde{\mathcal{J}}\left(x_k - \alpha \nabla_{\tilde{x}} \tilde{\mathcal{J}}(\tilde{x}_k, y_k), \max\left[y_k - \alpha \nabla_y \tilde{\mathcal{J}}(\tilde{x}_k, y_k), \ell\right]\right).$$

Set

$$\mathcal{A}_{k,C} \overset{\text{def}}{=} \{i \in \{1, \ldots, p\} \mid [y_k^c]_i = \ell_i\}, \quad \text{and} \quad C_k \overset{\text{def}}{=} \{1, \ldots, p\} \setminus \mathcal{A}_{k,C}. \tag{2.7}$$

Step 4: Minimization beyond the Cauchy point. Apply the CG algorithm to find an approximate minimizer $(\tilde{x}_{k+1}, y_{k+1}^{C_k})$ of

$$((Z^T g_x)^T, g_{y^{C_k}}^T)\begin{pmatrix} \tilde{x} \\ y^{C_k} \end{pmatrix} + \frac{1}{2}(\tilde{x}^T y^{C_k,T})\begin{pmatrix} Z^T P_{xx} Z & Z^T P_{xy^{C_k}} \\ P_{xy^{C_k}}^T Z & P_{y^{C_k} y^{C_k}} \end{pmatrix}\begin{pmatrix} \tilde{x} \\ y^{C_k} \end{pmatrix} \tag{2.8}$$

subject to

$$y^{C_k} \geq \ell^{C_k}. \tag{2.9}$$

Terminate the CG once one (or more) bound(s) of indices $j_1, \ldots, j_s$ are encountered, after a maximum number of iterations or once it has converged. If CG was terminated because bounds were encountered, restart it after redefining $C_k = C_k \setminus \{j_1, \ldots, j_s\}$. Repeat this process until the size of C_k does not decrease anymore.

vection, convective updrafts are modeled by stochastic Gaussian forcing β_u added at random locations in each model time step. These random convergent wind perturbations elevate the fluid surface. Conditional instability is modeled by modifying geopotential ϕ , which is set to low constant value ϕ_c once the height exceeds a threshold representing the level of free convection h_c. Gradient of the geopotential will force fluid into the region of decreased potential. The fluid level will continue to rise until diffusion prevents further growth. When h reaches the rain threshold h_r ($h_r > h_c$) and convergence is positive, rain is 'produced'. The parameter δ is the production rate for rain and η is its removal rate. The value of h_r ensures that production of rain is delayed relative to onset of clouds (height field).

In our numerical implementation of the model, the one dimensional domain, representing 125 km is discretized with 250 points using standard second-order centered differences on a staggered grid. To compute evolution of the dynamical system, the time variable is discretized into time steps of 5 seconds. Periodic boundary conditions are used. The model conserves mass, so the spatial integral over h is constant in time and the rain r cannot be negative. In case rain becomes negative during model intergration, negative values would be set to zero. The other model parameters are given by

$$h_0 = 90\,m, \quad h_c = 90.02\,m, \quad h_r = 90.4\,m, \quad D_u = D_h = 25000\,m^2 s^{-1}, \quad D_r = 200\,m^2 s^{-1},$$

$$\phi_c = 899.77\,m^2 s^{-2}, \quad \eta = 2.5 \cdot 10^{-4}\,s^{-1}, \text{ and } \delta = 1/300.$$

The Gaussian stochastic forcing β_u has a half width of 4 grid points and an amplitude of 0.002 m/s^2. The fields produced by running this model with three random initial conditions are illustrated in Figure 3.1 after 60 model time steps (which is equivalent to five minutes in real time). As illustrated in Figure 3.1 position of clouds (height field) and rain are quite different after only 60 model time steps, mimicking fast changing convective storms whose intermittency is one of the challenges of data assimilation on convective scale.

3.2 | Data assimilation experiments

Data assimilation setup follows (Ruckstuhl and Janjić, 2018), where differences between constrained and unconstrained minimization are documented also in dependence of ensemble size. To illustrate the two algorithms presented in Section 2, we perform a twin experiment, where we consider a model run to be the true state $z = \left(u^T, h^T, r^T\right)^T$, which we call the nature run. At each assimilation time, a vector of synthetic observations $z^{obs} \in \mathbf{R}^o$ is then created by randomly perturbing the nature run such that $z^{obs} = Hz + \epsilon^{obs}$, where H is the $o \times n$ matrix that determines the location and physical nature of the observations, while $\epsilon^{obs} \in \mathbf{R}^o$ is a random noise whose components depend on the observed variable and is computed as follows. For observations of the wind and height field, a Gaussian observation noise is added to the wind u and height h fields with zero mean and standard deviations 0.001 m/s and 0.02 m, respectively. A lognormal noise is added to the rain field with parameters −8 and 1.8, yielding a very small observation bias of 0.000825 and standard deviation of 0.00185. For this choice of parameters, the observation error for each field is approximately 10% of the maximum deviation from the field mean. We chose to observe every 120 model time steps, equivalent to 10 minutes in real time. Location of the observation are chosen to mimic radar data, and are simulated at locations where it rains in the nature run. As illustrated in Figure 3.1 already after five minutes, location of observations in nature run can be quite different compared to location of rain in each ensemble member. In addition for the u field, an extra 25% of observations are simulated at other locations mimicking observations from a different measurement system. The observation error covariance matrix R is taken to be diagonal with values on diagonal corresponding to variances of distributions used for generating observation error vector ϵ^{obs}. Although number of

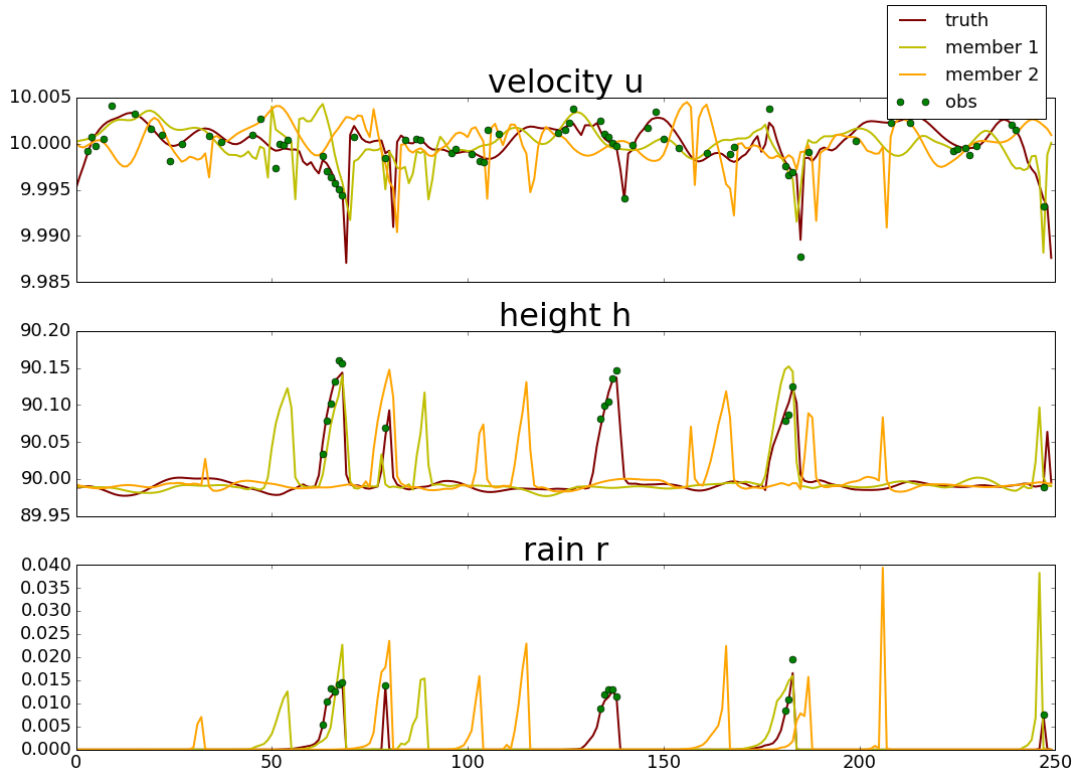

FIGURE 3.1 Fields produced by running the modified shallow water model with three different initial conditions after 60 model time steps. One of the experiments is chosen as nature run (red line). Observations (green circles) are simulated at locations where it rains in the nature run (truth) plus a random noise for all fields and, in addition for the u field, an extra 25% of observations are simulated at other locations. Noise is Gaussian for the h and u fields with zero mean and standard deviations of 0.001 m/s and 0.02 m, respectively. Noise is lognormal for the r field, yielding a very small observation bias of 0.000825 and standard deviation of 0.00185.

observations, and therefore matrix sizes for linear observation operator H, and R will change from one assimilation cycle to next, we omit time index for simplicity.

For data assimilation, Quadratic Programming Ensemble (QPEns) algorithm is used (Janjic et al., 2014). As stochastic EnKF, QPEns uses an ensemble of predictions (backgrounds) $\{z^{b,i}\}_{i=1}^{N}$ to calculate sample background error covariance matrix P^b

$$P^b = \frac{1}{N-1}\sum_{i=1}^{N}\left[z^{b,i} - \bar{z}^b\right]\left[z^{b,i} - \bar{z}^b\right]^T$$

with $\bar{z}^b$ representing the ensemble mean. Since the sample covariance P^b is singular when $N - 1 < n$, it is in our experiments localized with blockwise matrix where each block uses 5th order polynomial function (see Ruckstuhl and Janjić, 2018, for details) with cutoff at eight grid points. From background ensemble, perturbed observation, localized background error covariance and observation error covariance, the QPEns calculates analyses ensemble. However, instead of using Kalman filter formulas to obtain the analyses, it solves the following quadratic programming problem for each member $i = 1, \ldots, N$:

$$z^{a,i} = z^{b,i} + \arg\min_{\delta z^i} \frac{1}{2}[\delta z^{i\,T}(P^b)^{-1}\delta z^i + f^{i\,T}R^{-1}f^i] \tag{3.5}$$

subject to physical constraints. Here $\delta z^i = z^{a,i} - z^{b,i}$ is the analysis increment and $f^i = z^{obs,i} - Hz^{b,i} - H\delta z^i - \bar{\epsilon}^{obs}$. The measurements appearing in f^i are perturbed using same strategy as with generating observations, that is described at the beginning of this section and sample mean $\bar{\epsilon}^{obs}$ is subtracted. The solutions to the N constrained optimization problems form the analysis ensemble $\{z^{a,i}\}_{i=1}^{N}$.

Physical constraints for modified shallow water model are conservation of mass and preservation of non-negativity of rain. In the QPEns framework, these constraints transfer to

$$\delta r^i \geq -r^{b,i} \text{ and } e_h^T \delta h^i = 0, \text{ for } i = 1, \ldots, N \tag{3.6}$$

where $e_h^T = (1, \ldots, 1)$ is of same size as h, and δr, δh are corresponding components of δz. Note that the inequality in (3.6) is equivalent to $r^{a,i} \geq 0$, and equality to $e_h^T h^{a,i} = e_h^T h^{b,i}$.

Therefore, in our setup $n = 750$, $p = 250$, $x = (u^T, h^T)^T$, $A = (0_u^T, e_h^T)$ of size $1xn$ with $0_u^T = (0, \ldots, 0)$ is of same size as u and $y = r$, $N = 50$, $P = (P^b)^{-1} + H^T R^{-1} H$ and for each ensemble member i gradient is calculated as $(g_x^i, g_y^i)^T = -(P^b)^{-1}\bar{z} - H^T R^{-1} z^{obs}$. For Algorithm 2.2, $Z = I - A^T A/h^2$. Data are assimilated every ten minutes and total number of cycles is 35. In order to obtain robust results, twin experiments were repeated many times for different random seeds.

4 | RESULTS

In the following, the results of data assimilation experiments in terms of root mean square errors (RMSE) are discussed. For minimization of the problem (3.5)- (3.6) we will employ algorithms described in Section 2. In addition, benefits and drawbacks of the algorithms are compared in Section 4.2.

4.1 | Data assimilation results

Results for both constrained and unconstrained minimizations are illustrated in Figures 4.2 and 4.3. Constrained minimization (QPEns, red line in Figure 4.2) produces smaller RMSE than unconstrained minimization (EnKF, blue line). Plotted are RMSE values of background and analysis ensemble mean against the nature run, for each variable of the modified shallow water model separately. RMSE values are obtained by averaging over 500 twin experiments. Constrained minimization plot was produced by running both algorithms with $\epsilon = 0.1$. The results during 35 cycles (approximately 6 hours) are indistinguishable between the algorithms and identical to results of interior point method of Python used in Ruckstuhl and Janjić (2018). After 35 cycles of data assimilation, free forecasts are started for additional 6 hours. The differences between constrained and unconstrained minimization persists during six hour forecasts and especially can be seen in the height field. The difference in rain prediction can be only be seen up to 3 hours. For comparison, RMSE of forecasts starting with perfect initial ensemble (black line) generated using observation error statistics is plotted in addition to illustrate intrinsic predictability of this model.

To understand the differences in RMSE, let us consider one snapshot of the results in Figure 4.3, with the truth and observations plotted in black. Errors for constrained and unconstrained minimizationthe value of rain is positive in all grid points. Although differences in RMSE between constrained and unconstrained minimization are small after one assimilation cycle, in Ruckstuhl and Janjić (2018) was shown that data assimilation errors of unconstrained minimization will accumulate over time leading to large errors in total mass and total rain after repeating data assimilation 250 times, i.e. in less than one day.

4.2 | Performance and differences between minimization algorithms

We use an LU decomposition with pivoting to solve (2.3) accurately. Table 4.1 illustrates the performance of the algorithm.

k	$J(z)$	$\lvert\mathcal{A}_k^c\rvert$	$\lVert[\nabla_y \mathcal{J}(x_k, y_k)]^{\mathcal{A}_k^c}\rVert$	α_k	$\lVert z_k - z^*\rVert$
1	-2.562770E+03	107	2.205868E+04	1.	1.434792E-01
2	-2.578068E+03	118	6.314035E+03	1.	4.935165E-02
3	-2.580350E+03	124	4.987211E+03	1.	1.701261E-02
4	-2.582139E+03	129	1.755805E+03	0.6262	1.292911E-02
5	-2.582549E+03	133	1.472659E+02	0.8768	2.030012E-03
k	$J(z)$	$\lvert\mathcal{A}_k^c\rvert$	$\lVert[\nabla_y \mathcal{J}(x_k, y_k)]^{\mathcal{A}_k^c}\rVert$	α_k	$\lVert z_k - z^*\rVert$
1	-6.368068E+02	28	1.946095E+07	1.	5.233164E-02
2	-6.471606E+02	31	1.436930E+06	1.	7.730555E-04
3	-6.472081E+02	33	8.459149E+05	0.3942	5.210342E-04
4	-6.472324E+02	33	1.320271E-07	1.	1.371545E-10

TABLE 4.1 Illustration of performance of the Algorithm 2.1 during first data assimilation cycle and tenth cycle for one of ensemble members.

As illustrated in Table 4.1, Algorithm 2.1 converges in only five iterations on this example. If a more general interior point method like the CVXOPT package (Andersen et al., 2010) is applied for minimization of this problem,

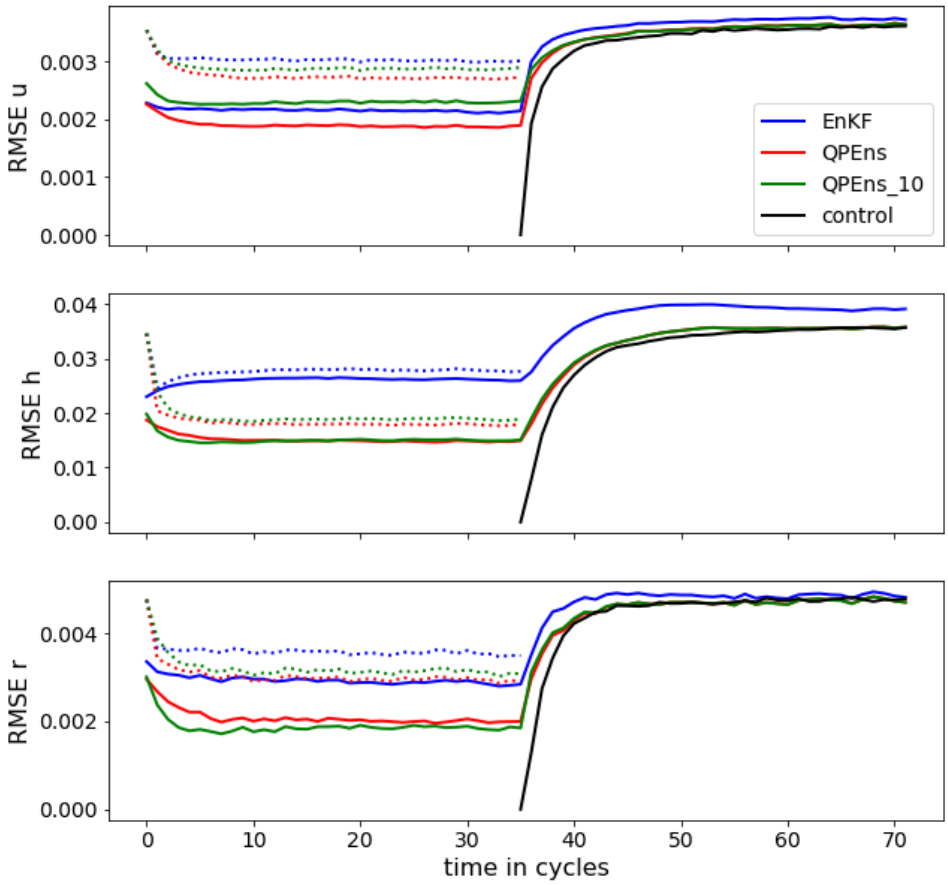

FIGURE 4.2 RMSE of analysis and background for QPEns (red), EnKF (blue) and approximate QPEns that fixes number of iterations to ten (green). After 35 data assimilation cycles, free forecasts starts. To illustrate the predictability of the model, ensemble whose mean and standard deviation are equal to observation statistics (?) is run forward in time (black line) in addition.

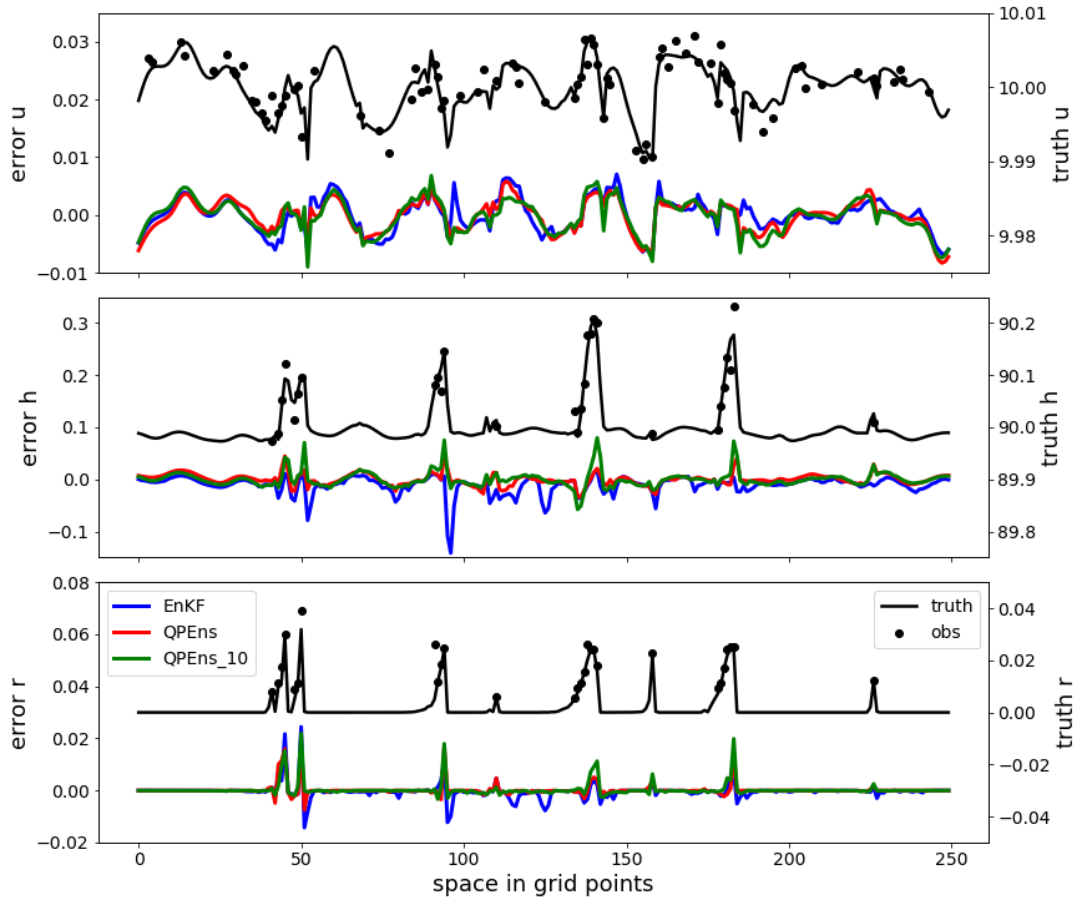

FIGURE 4.3 Results of minimization for unconstrained (red) and constrained (blue) problems based on prior estimate (green line) and observations (green circles).

the number of iterations required is typically between ten and twenty.

In many cases, the efficient application of the CG algorithm requires preconditioning. We refer the reader to Gratton et al. (2013) for a discussion of suitable strategies in the context of data assimilation. It is also known that Algorithm 2.2 could be implemented without Step 4 if mere convergence is wanted, but that performing conjugate gradient iterations as suggested in Conn et al. (1992), very often significantly reduces the number of outer iterations. This was also observed for our test problem. If subproblems in Step 4 are solved accurately, Algorithm 2.2 requires three outer iterations, as illustrated in Table 4.2. If the number of conjugate gradient iterations per outer iteration is fixed a priori (a standard practice in weather forecasting, see for example Tremolet, 2007), the number of outer iterations increases, and could reach twenty, but the cost of each outer iteration decreases. The behavior of the algorithm with the number of CG iterations fixed a priori to 25, 50, 400 and 800 is illustrated in Tables **??** and **??**. For the computational consideration, we also impose an additional stopping criteria to Algorithm 2.2: the algorithm is stopped either when it has converged or when the number of faces reduces to one. Note that the latter criteria is met only if the CG iterations for minimising equation (2.8) did not encounter any bounds prescribed by (2.9), which would suggest that the current guess of the active set is fairly accurate, though not guaranteed to be exact. Fixing the total number of CG iterations per outer iteration limits number of CG restarts during one major iteration and reduces accuracy as well as cost. For example, for a fixed number of 25 CG iterations, the solutions obtained by Algorithm 2.1 and Algorithm 2.2 only coincide to two significant digits, while if 800 CG iterations are allowed, they share eleven significant digits. When allowing the number of CG iterations to increase from 25 to 800, the total number of iterations performed increases from 300 to 2400 and reaches 2472 in case where no limit is set while the cost increases by 75% for no-limit case. While these number are encouraging, they also indicate that more attention must be given to preconditioning.

	$J(z)$	$\lvert\mathcal{A}_k^c\rvert$	$\lVert[\nabla_y \mathcal{J}(x_k, y_k)]^{\mathcal{A}_k^c}\rVert$	α_k	CG its	faces	$\lVert z_k - z^*\rVert$
1	-2.56403E+03	173	1.80947E+02	5.883e-07	84	35	1.03437E-01
2	-2.58203E+03	141	7.26408E+02	8.031e-07	32	5	1.56615E-02
3	-2.58256E+03	133	4.95919E+02	2.819e-07	15	2	7.34838E-03
	$J(z)$	$\lvert\mathcal{A}_k^c\rvert$	$\lVert[\nabla_y \mathcal{J}(x_k, y_k)]^{\mathcal{A}_k^c}\rVert$	α_k	CG its	faces	$\lVert z_k - z^*\rVert$
1	-7.29541E+02	50	4.57538E+05	3.353e-08	78	24	8.85338E-03
2	-7.29892E+02	36	2.82690E+05	4.7e-15	24	3	2.22762E-03
3	-7.29932E+02	34	2.36833E+04	1.991e-14	72	2	1.39713E-04
4	-7.29932E+02	33	6.03305E+03	3.059e-14	40	1	3.65735E-05

TABLE 4.2 Illustration of performance of the Algorithm 2.2. In this table, "CG its" stands for the total number of CG iterations at major iteration k and "faces" is the number of explored faces at iteration k. To illustrate the accuracy, the difference is calculated between result of each major iteration z_k to z^* an end solution of Algorithm 2.1 on page 4. First panel is during first data assimilation cycle and second for tenth cycle for one of ensemble members.

To distinguish between the respective effects of mass conservation eT(ha ? hf) = 0 and positivity constraints ra ? 0, in addition we conducted experiments in which only mass conservation was applied and ones in which only positivity constraints were applied. The results with only mass conservation are very close to the QPEns results, where both type of constraints are imposed (not shown). Experiments where only the positivity of rain is constrained yield slightly higher RMSEs than those corresponding to the EnKF (not shown). This is consistent with the findings of Janji?c et al. (2014), where it was demonstrated that mass conservation is needed to benefit from the positivity constraints.

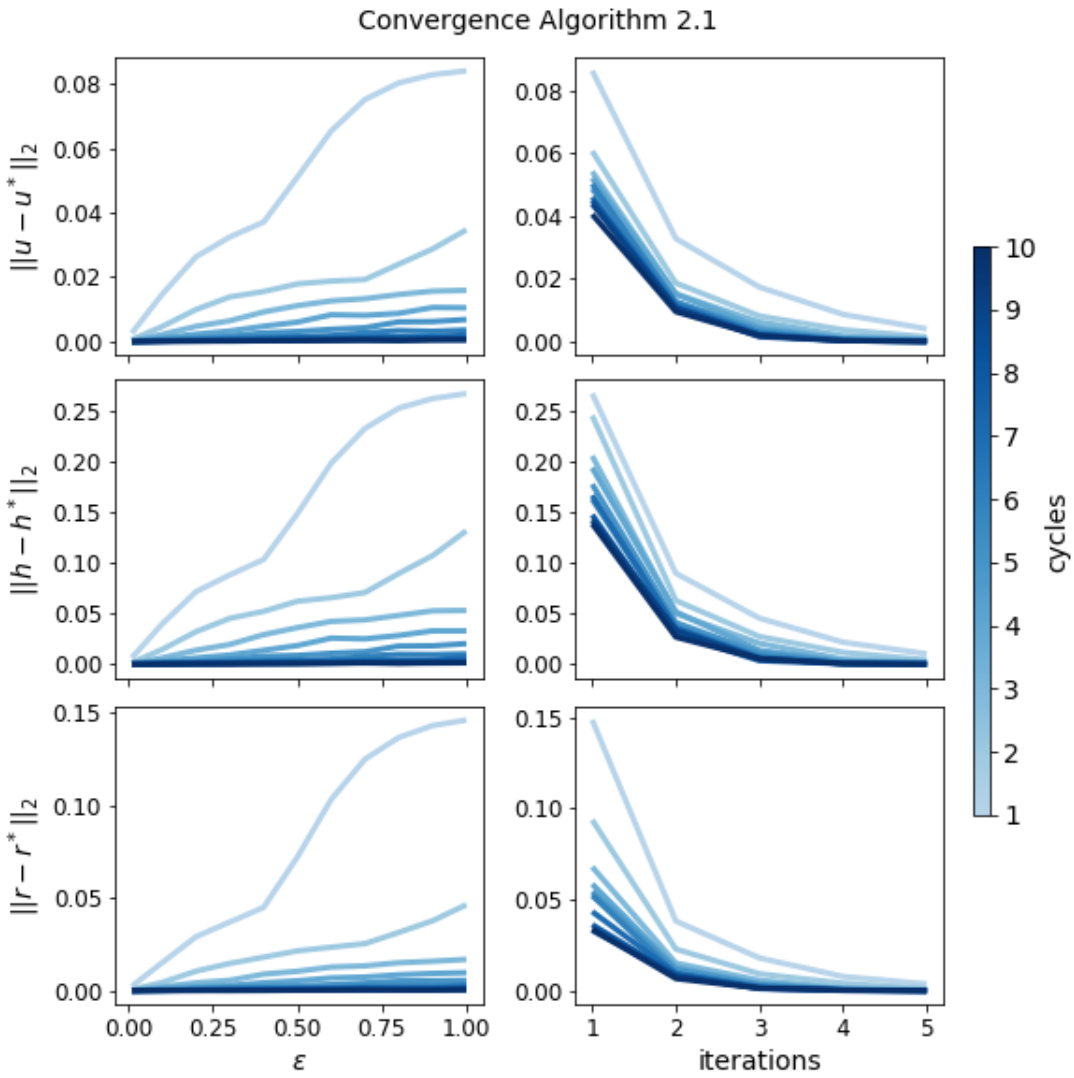

FIGURE 4.4 RMSE to the truth as a function of epsilon (left) and as a function of outer iterations performed (right). Different shades of blue represent results for each of first ten data assimilation cycles performed.

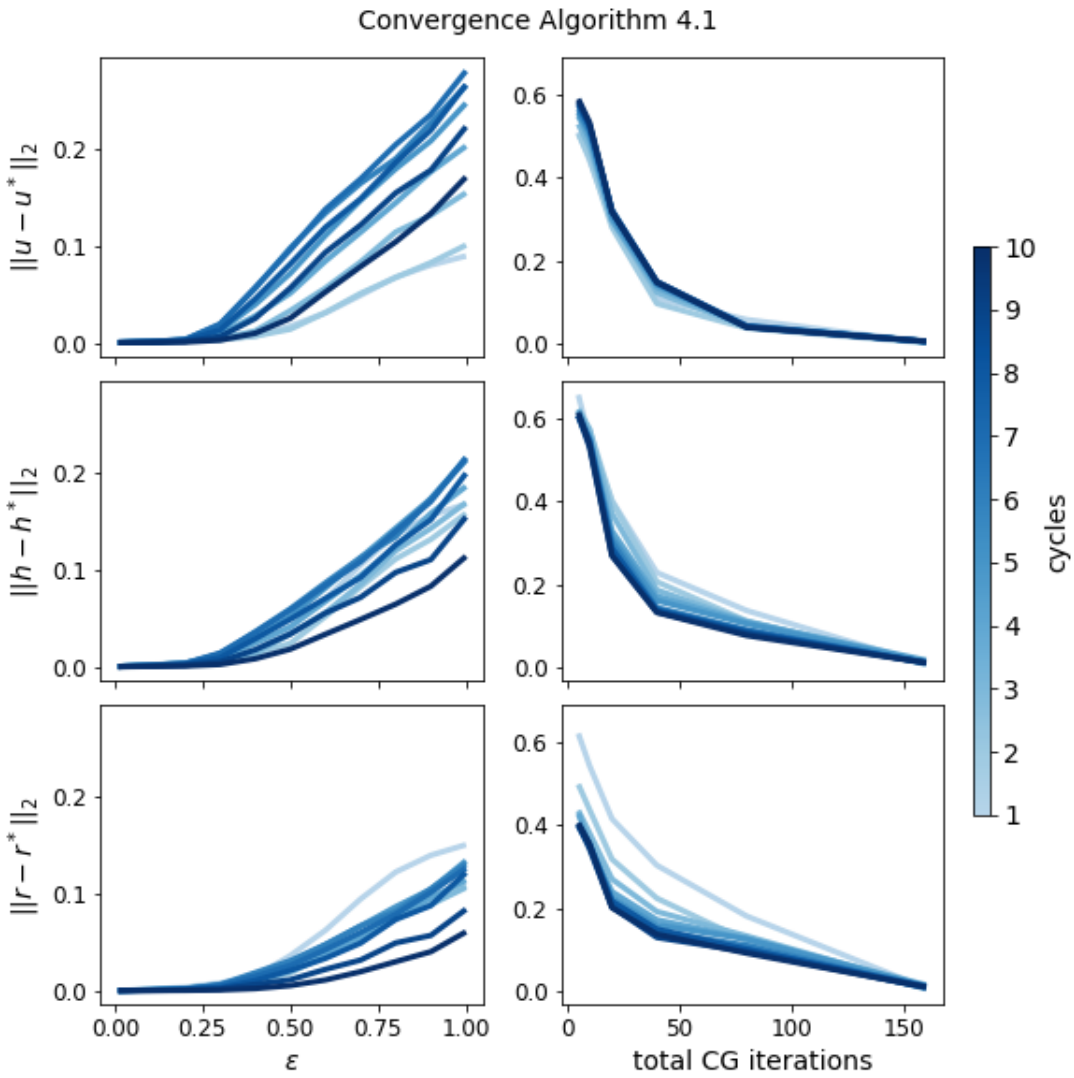

FIGURE 4.5 RMSE to the truth as a function of epsilon (left) and as a function of inner CG iterations performed (right). Different shades of blue represent results for each of first ten data assimilation cycles performed.

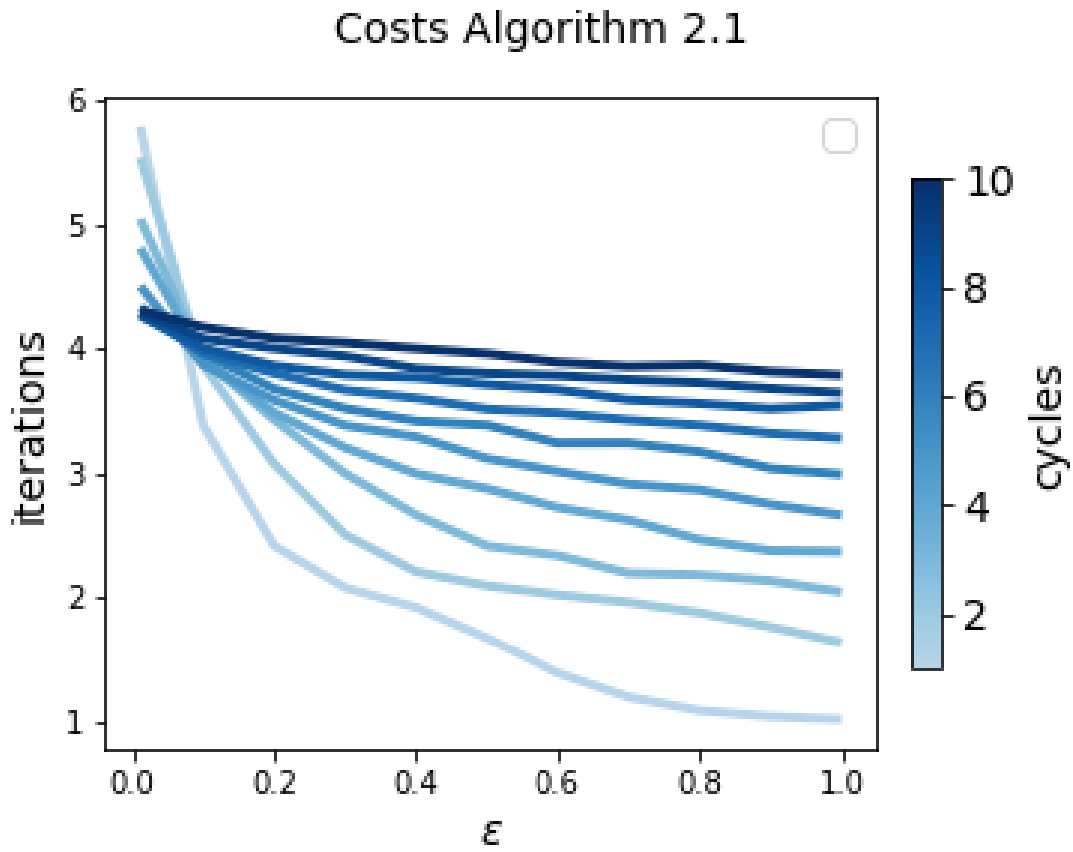

FIGURE 4.6 RMSE to the truth as a function of epsilon (left) and as a function of outer iterations performed (right). Different shades of blue represent results for each of first ten data assimilation cycles performed.

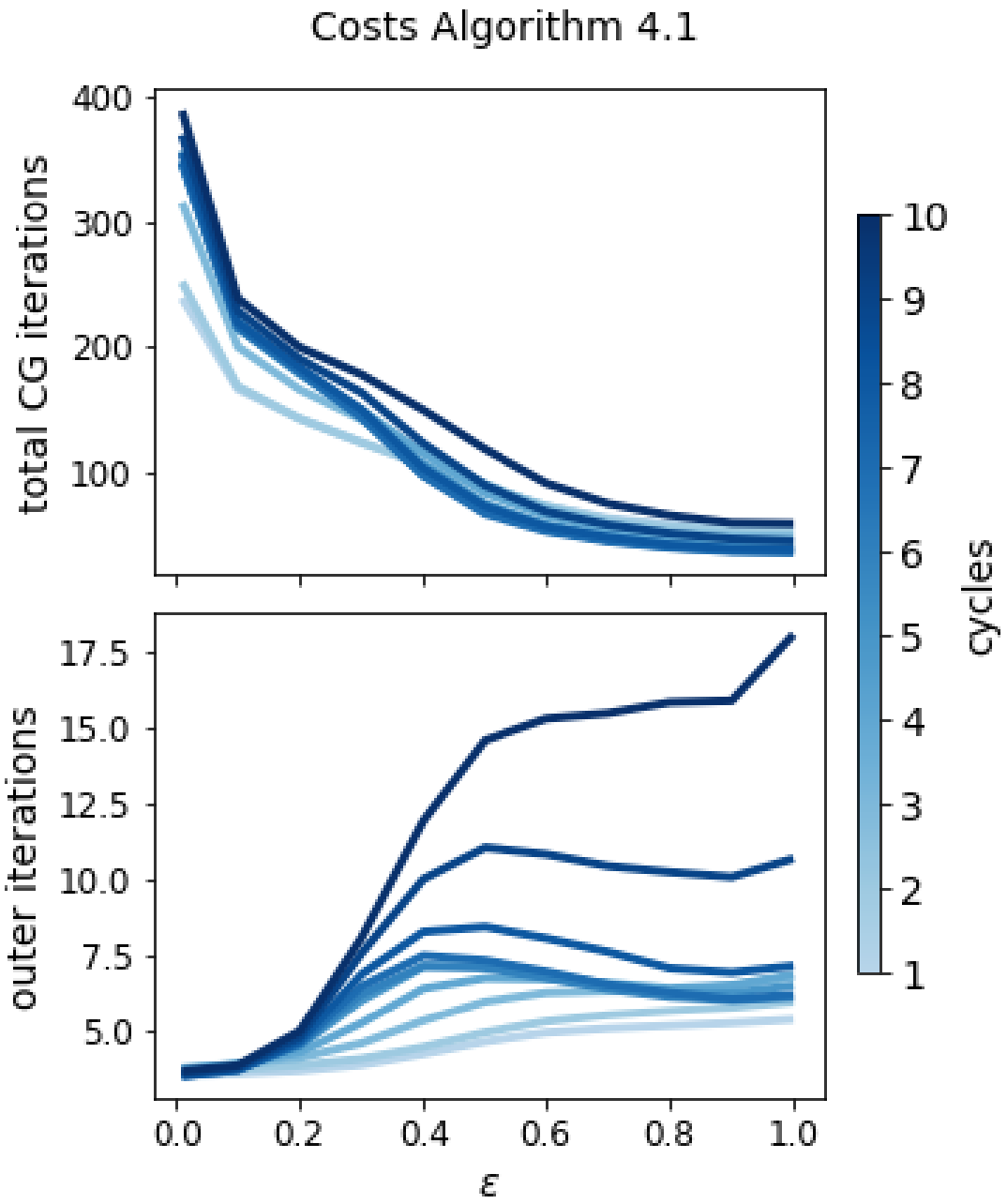

FIGURE 4.7 RMSE to the truth as a function of epsilon (left) and as a function of inner CG iterations performed (right). Different shades of blue represent results for each of first ten data assimilation cycles performed.

We conclude that the effect of mass conservation is both dominant compared with the positivity constraints and necessary for improved

5 | CONCLUSION

In NWP, we can expect in an upcoming decade that even global models will reach the convection permitting resolution. Data assimilation for such geophysical models that resolve many scales of motion and for observations of higher temporal and spatial density would require that we re-evaluate and improve on the methodology. Improved data assimilation systems will therefore require the development of novel approaches that account for non-Gaussianity, model errors, better understanding how different existing methods can represent uncertainty at the data assimilation and forecasting steps of both observations and numerical models (Janjic et al., 2018). But also data assimilation methods would require a better way to estimate (multivariately) prognostic hydrometeors. This is challenging task, due to computational costs of producing non-negative estimates, frequency of their estimation, and a need to initialize probabilistic prediction.

We have presented two projection algorithms which exploit the disjoint nature of constraints typically occurring in weather forecasting applications. While projection methods may be inefficient when the combinatorial aspect of selecting the correct active bounds dominate and many faces need to be explored at each major iteration (in which case the interior-point algorithms often perform better), they do perform well compared to the interior-point algorithms when the gradient quickly provides a good identification of the active constraints. This appears to be the case in our (representative) application.

The first of our methods, Algorithm 2.1, is more efficient than an interior point approach on a representative example, but still requires solving the Karush–Kuhn–Tucker system (2.3), which is impractical in weather forecasting applications due to problem size and frequency of solution. By contrast, Algorithm 2.2 exploits the low rank of the linear equality constraints and uses a well-known iterative approach to compute a possibly approximate solution while ensuring satisfaction of the constraint. If the size of the problem is such that the conjugate gradient algorithm is allowed to converge, the number of outer iterations required by Algorithm 2.2 is smaller or comparable to that required by Algorithm 2.1. If the number of conjugate gradient iterations is limited from the start (as is often the case in weather forecasting applications), the number of outer iterations typically increases and finding the optimal equilibrium between accuracy and cost then depends on the problem at hand. A further advantage of Algorithm 2.2 is that its applies the conjugate gradient to a subproblem whose size is significantly smaller than that of the KKT system (2.3) (remember that $p \approx n/3$).

The observations made in this short paper are encouraging, but the authors are aware that adapting the proposed method(s) to a real environment remains a significant task, as preconditioning and the details of the face changing mechanism will need thought and fine tuning.

acknowledgements

Y. Ruckstuhl and T. Janjic are thankful for funding through Transregional Collaborative Research Center SFB / TRR 165 “Waves to Weather" funded by the German Science Foundation (DFG) through subproject B6: "Parameter estimation using a data assimilation system for improved representation of clouds". T. Janjic is also greatful to the DFG for funding this research under project "Conservation laws and ensemble Kalman filter algorithms" (DFG JA1077/3-1) and for funding of her Heisenberg Award (DFG JA1077/4-1) .

references

M. S. Andersen, J. Dahl, and L. Vandenberghe. Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones. *Mathematical Programming, Series C*, 2:167–201, 2010.

A. Arakawa and V. R. Lamb. A Potential Enstrophy and Energy conserving scheme for the Shallow Water Equations. *Monthly Weather Review*, 109, 18–36, 1980.

L. Bertino, G. Evensen, and H. Wackernagel. Sequential data assimilation techniques in oceanography. *International Statistical Reviews*, 71:223–241, 2003.

Bishop, C. H., B. J. Etherton, and S. Majumdar, 2001: Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Mon. Wea. Rev.*, **129**, 420–436.

Burgers, G. and P. J. van Leeuwen and G. Evensen, 1998: Analysis Scheme in the Ensemble Kalman Filter. *Mon. Wea. Rev.*, **126**, 1719–1724.

M. Bühner, A. Caya, L. Pogson, T. Carrieres, and P. Pestieau. A new environment Canada regional ice analysis system. *Atmosphere-Ocean*, 51(1):18–34, 2013.

Cohn, S. E., 1997: An introduction to estimation theory. *J. Meteor. Soc. Japan*, **75**, 257–288.

S. E. Cohn, and D. F. Parrish. The Behavior of Forecast Error Covariances for a Kalman Filter in Two Dimensions. *Monthly Weather Review*, 119(8): 1757–1785, 1991.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT: *a Fortran package for large-scale nonlinear optimization (Release A)*. Number 17 in Springer Series in Computational Mathematics. Springer Verlag, Heidelberg, Berlin, New York, 1992.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, USA, 2000.

P. Courtier, E. Andersson, W. Heckley, J. Pailleux, D. Vasiljevic, M. Hamrud, A. Hollingsworth, F. Rabier, and M. Fisher. The ECMWF implementation of three-dimensional variational assimilation (3D-Var). I: Formulation. *Quarterly Journal of the Royal Meteorological Society*, 127, 1783–1807, 1998.

M. J. P. Cullen, T. Davies, M. H. Mawson, J. A. James, and S. C. Coulter. An Overview of Numerical Methods for the Next Generation U.K. NWP and Climate Mode1. *Atmosphere-Ocean*, 35, 425–444,1997.

Z. Dostál. Box constrained quadratic programming with proportioning and projections. *SIAM Journal on Optimization*, 7(3):871–887, 1997.

G. Evensen. Sequential data assimilation with a nonliear quasi-gepstrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99, 10143–10162, 1994.

G. Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53, 343–367, 2003.

P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, UK, 1981.

P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright. *Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints*. *ACM Trans. Math. Software*, 10:282–298, 1984.

N. I. M. Gould, M. E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic problems arising in optimization. *SIAM Journal on Scientific Computing*, 23(4):1375–1394, 2001.

N. I. M. Gould and Ph. L. Toint. An iterative working-set method for large-scale non-convex quadratic programming. *Applied Numerical Mathematics*, 43(1–2):109–128, 2002.

S. Gratton, S. Gürol, and Ph. L. Toint. Preconditioning and globalizing conjugate gradients in dual space for quadratically penalized nonlinear-least squares problems. *Computational Optimization and Applications*, 54(1):1–25, 2013.

N. Gustafsson, T. Janjić, C. Schraff, D. Leuenberger, M. Weissman, H. Reich, P. Brousseau, T. Montmerle, E. Wattrelot, A. Bucánek, M. Mile, R. Hamdi, M. Lindskog, J. Barkmeijer, M. Dahlbom, B. Macpherson, S. Ballard, G. Inverarity, J. Carley, C. Alexander, D. Dowell, S. Liu, Y. Ikuta, and T. Fujita. Survey of data assimilation methods for convective-scale numerical weather prediction at operational centres. *Quarterly Journal of the Royal Meteorological Society*, 144(713):1218–1256, 2018.

M. Haslehner, T. Janjić, and G. C. Craig. Testing particle filters on simple convective-scale models. Part 2: A modified shallow-water model. *Quarterly Journal of the Royal Meteorological Society*, 142(697):1628–1646, 2016.

M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of the National Bureau of Standards*, 49:409–436, 1952.

Hungerlaender, P. and F. Rendl A Feasible Active Set Method for Strictly Convex Quadratic Problems with Simple Bounds. *SIAM Journal on Optimization*, 25:3,1633–1659, 2015.

Hunt, B. R., E. J. Kostelich, and I. Szunyogh, 2007: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D*, **230**, 112–126.

T. Janjić, D. McLaughlin, S. E. Cohn, and M. Verlaan. Conservation of mass and preservation of positivity with ensemble-type Kalman filter algorithms. *Monthly Weather Review*, 142:755–773, 2014.

Janjic, T. and Potthast, R. and Van Leeuwen, P.-J. Editorial, *Quarterly Journal of the Royal Meteorological Society*, 144(713): 1189-1190, 2018. doi:10.1002/qj.3382

G. S. Ketefian, and M. Z. Jacobson. A mass, energy, vorticity, and potential enstrophy conserving lateral fluid-land boundary scheme for the shallow water equations. *Journal of Computational Physics*, 228,1–32, 2009.

Lauvernet, C., J.-M. Brankart, F. Castruccio, G. Broquet, P. Brasseur, and J. Verron, 2009: A truncated Gaussian filter for data assimilation with inequality constraints: Application to the hydrostatic stability condition in ocean models. *Ocean Modelling*, **27**, 1–17.

G.-Y. Lien, and E. Kalnay, and T. Miyoshi Effective assimilation of global precipitation: simulation experiments *Tellus A*, 65, 2013.

C. Lin and J. J. Moré. Newton's method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, 9(4):1100–1127, 1999.

E. N. Lorenz. The predictability of a flow which possesses many scales of motion. *Tellus*, XXI/3:289–307, 1969.

Rabier, F., Jaervinen, H., Klinker, E., Mahfouf, J.?F. and Simmons, A. The ECMWF operational implementation of four?dimensional variational assimilation. I: Experimental results with simplified physics. *Q.J.R. Meteorol. Soc.*, 126: 1143–1170, 2000.

Y. Ruckstuhl and T. Janjić. Parameter and state estimation with ensemble Kalman filter based approaches for convective scale data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144(712): 826–841, 2018.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, USA, 1996.

R. Sadourny. The dynamics of finite-difference models of the shallow-water equations. *Journal of the Atmospheric Sciences*, 1, 119–143,1975.

E. Simon and L. Bertino. Application of the Gaussian anamorphosis to assimilation in a 3-D coupled physical-ecosystem model of the North Atlantic with the EnKF: A twin experiment. *Ocean Science*, 5:495–510, 2009.

E. Simon and L. Bertino. Gaussian anamorphosis extension of the DEnKF for combined state and parameter estimation: application to a 1D ocean ecosystem model. *Journal of Marine Systems*, 89:1–18, 2012.

Sommer, M. and Névir, P. A conservative scheme for the shallow-water system on a staggered geodesic grid based on a Nambu representation. *Quarterly Journal of the Royal Meteorological Society*, 135:485-494,2009.

Sun, J. and Xue, M. and Wilson, J. W. and Zawadzki, I. and Ballard, S. P. and Onvlee-Hooimeyer, J. and Joe, P. and Barker, D. M. and Li, Ping-Wah and Golding, B. and Xu, M. and Pinto, J.. Use of NWP for Nowcasting Convective Precipitation: Recent Progress and Challenges *Bulletin of the American Meteorological Society*, 95(3): 409–426, 2014.

Trémolet, Y. Incremental 4D-Var convergence study *Tellus A: Dynamic Meteorology and Oceanography*, 59(5): 706–718, 2007.

M. Würsch and G. C Craig. A simple dynamical model of cumulus convection for data assimilation research. *Meteorologische Zeitschrift*, 23:483–490, 2014.

Y. Zeng, and T. Janjić. Study of conservation laws with the Local Ensemble Transform Kalman Filter. *Quarterly Journal of the Royal Meteorological Society*, 142 (699): 2359–2372, 2016.

Y. Zeng, T. Janjić, Y. Ruckstuhl, and M. Verlaan. Ensemble-type Kalman filter algorithm conserving mass, total energy and enstrophy. *Quarterly Journal of the Royal Meteorological Society*, 143(708): 2902–2914, 2017.