

RESEARCH ARTICLE

A data assimilation algorithm for predicting rain

Tijana Janjić¹ | Yvonne Ruckstuhl¹ | Philippe L. Toint²¹Meteorological Institute,
Ludwig-Maximilians-University Munich,
München, Germany²Namur Centre for Complex Systems,
University of Namur, Namur, Belgium

Correspondence

T. Janjić, Meteorological Institute,
Ludwig-Maximilians-University Munich,
Theresienstrasse 37, München 80333,
Germany.
Email: tijana.pfander@lmu.de

Funding information

Transregional Collaborative Research
Center German Science Foundation
(DFG), Grant/Award Numbers:
SFB/TRR 165, DFG JA1077/3-1,
DFG JA1077/4-1

Abstract

Convective-scale data assimilation uses 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 for a state vector of large size with one third or more of its components containing prognostic hydrometeor variables whose non-negativity needs to be preserved. The algorithm also needs to be fast as the state vector requires a high updating frequency in order to catch fast-changing convection. A computationally efficient algorithm for quadratic optimization (QO, or formerly QP) is presented here, which preserves physical properties in order to represent features of the real atmosphere. Crucially for its performance, it exploits the fact that the resulting linear constraints may be disjoint. Numerical results on a simple model designed for testing convective-scale data assimilation show accurate results and promising computational cost. In particular, if constraints on physical quantities are disjoint and their rank is small, further reduction in computational costs can be achieved.

KEYWORDS

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

1 | INTRODUCTION

Convection-permitting models with single digit horizontal resolution in kilometres partly resolve highly nonlinear dynamics and physics at a wide range of spatial and temporal scales. However, the quality of their prediction also depends on new advances in data assimilation algorithms which provide suitable initial conditions. In particular, there is a need for 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 algorithms for updating prognostic hydrometeor values such as rain, graupel or snow.

Unfortunately, prognostic hydrometeors are not updated within a number of algorithms used in operational practice, such as incremental three-dimensional variational assimilation (3D-Var; Courtier *et al.*, 1998), four-dimensional variational assimilation (4D-Var; Rabier *et al.*, 2000) or the localized ensemble transform Kalman filter (LETKF; Bishop *et al.*, 2001; Hunt *et al.*, 2007). Instead, the adjusted state variables are used to infer hydrometeor values through the microphysical scheme,

or to update them using the latent heat nudging approach or a cloud analysis technique (Gustafsson *et al.*, 2018). A more coherent approach would be to estimate these values directly in the state vector, allowing hydrometeors to be modified consistently with other variables. However, due to the underlying Gaussian assumption, sparse observations and other approximations in these methods, non-physical values could appear and *a posteriori* physical consistency checks are still needed. These checks include, for example, setting the negative values of rain, graupel and snow to zero. Moreover, if an ensemble of simulations is used to construct covariances, as in ensemble data assimilation algorithms or hybrid variational algorithms, then consistency checks must be made for each ensemble member, thereby artificially modifying the uncertainty representation. Importantly, 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, 2003; Burgers *et al.*, 1998) is capable of handling complex and highly nonlinear processes, and the adjoint of the model is not needed for its application. This makes it attractive for convective-scale applications. Since 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 a modification could be found, one could for example optimize the use of radar observations to initialize numerical weather prediction (NWP) models, a real advantage given the importance of this dataset for prediction of convective storms (Sun *et al.*, 2014). Various approaches can be taken within the EnKF framework in order to deal with 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 parametrized 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.

As an alternative to the above statistical-based approaches, Janjić *et al.* (2014) modified the EnKF algorithm to include constraints directly on the state vector. The Janjić *et al.* (2014) algorithm, Quadratic Programming Ensemble (QPEs), combines the EnKF and quadratic optimization (QO), exactly preserves feasibility with respect to linear equality and inequality constraints in each of the ensemble members, and produces multivariate estimates for all fields. This method was shown

to outperform the EnKF as well as the EnKF with a lognormal change of control variables. As argued in the paper, this is caused by the fact that each of these statistical methods preserves mass (EnKF) or positivity (lognormal EnKF) but not both. This approach was extended by Ruckstuhl and Janjić (2018), showing that similar conclusions hold for a nonlinear multivariable model designed to test convective-scale data assimilation applications. Both 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 directly in the minimization, the QO 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

$$\min_{\mathbf{x}, \mathbf{y}} \mathcal{J}(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} (\mathbf{g}_{\mathbf{x}}^T \quad \mathbf{g}_{\mathbf{y}}^T) \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} + \frac{1}{2} (\mathbf{x}^T \quad \mathbf{y}^T) \begin{pmatrix} \mathbf{P}_{\mathbf{xx}} & \mathbf{P}_{\mathbf{xy}} \\ \mathbf{P}_{\mathbf{xy}}^T & \mathbf{P}_{\mathbf{yy}} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \quad (1)$$

subject to

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \text{and} \quad \mathbf{y} \geq \boldsymbol{\ell}, \quad (2)$$

where $\mathbf{x}$ and $\mathbf{g}_{\mathbf{x}}$ belong to $\mathcal{R}^{n-p}$, $\mathbf{y}$ and $\mathbf{g}_{\mathbf{y}}$ to $\mathcal{R}^p$, $\mathbf{P}_{\mathbf{xx}}$ is an $(n-p) \times (n-p)$ symmetric real matrix, $\mathbf{P}_{\mathbf{yy}}$ a $p \times p$ symmetric real matrix, $\mathbf{P}_{\mathbf{xy}}$ a $(n-p) \times p$ real matrix, $\mathbf{A}$ an $m \times (n-p)$ real matrix with $m \leq (n-p)$ of rank m , $\mathbf{b} \in \mathcal{R}^m$ and the inequality is understood component-wise. It is easy to extend our discussion to more general bound constraints where one requires $\boldsymbol{\ell} \leq \mathbf{y} \leq \mathbf{u}$ for some $\boldsymbol{\ell}$, $\mathbf{u} \in \mathcal{R}^p$ with $\boldsymbol{\ell} \leq \mathbf{u}$, both of them possibly having infinite components. We refer to a problem of the type (1)–(2) as having *disjoint constraints* in the sense that the two sets of constraints of (2) involve disjoint sets of variables. In what follows, we focus on the convex case, and assume that

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

is positive definite.

Typical data assimilation algorithms produce initial condition of the model by minimizing, perhaps iteratively in a Gauss–Newton framework, an objective function of type (1). Minimization is usually performed every hour in convective-scale applications using 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 $\mathbf{z} = (\mathbf{x}^T, \mathbf{y}^T)^T$ 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 $\mathbf{y}$ 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 a second-moment microphysics scheme is used instead of a first-moment scheme in the convection-permitting model.

Both the size of vectors $\mathbf{x}$ and $\mathbf{y}$ and the frequency of their estimation (which is usually an hour or less), significantly limit the number of iterations of the minimization algorithm that can be performed. Moreover, in an ensemble setting, not only one but an ensemble of estimates is produced in order to correctly specify uncertainty, further increasing the computational burden. The computational cost also depends on the number and the nature of the constraints, with inequality constraints being often more challenging than equality constraints. It is the purpose of this short paper to show that particular optimization algorithms can be combined in a way that is less expensive in computer time, thereby making a practical application affordable. While these considerations are based on data assimilation for weather forecasting, 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).

The paper is organized as follows. Section 2 introduces two algorithms which exploit the fact that the constraints are disjoint to solve problems of type (1)–(2). The application of these optimization algorithms is explored using a known convective-scale example first introduced by Würsch and Craig (2014). The model and the data assimilation framework are described in Section 3. Section 4 first analyses the convergence of both algorithms for the minimization problems appearing in the QPEns. This is followed by an investigation on how the performance of the QPEns is affected when the proposed minimization algorithms are terminated early. Here, the data assimilation results are compared against the EnKF as a benchmark, since the EnKF solution would be the same as the QPEns solution in case no constraints are imposed. A final discussion and some perspectives are presented in Section 5.

2 | THE ALGORITHMS

2.1 | A QO algorithm for disjoint constraints

For solving problem (1)–(2), we propose an ‘active-set’ algorithm whose feature is to maintain feasibility with respect to the linear equality constraints on $\mathbf{x}$ at each iteration k , while at the same time using classical projection techniques (Conn *et al.*, 2000, chapter 12) to enforce feasibility of $\mathbf{y}$. Active set algorithms aim at iteratively finding the set of constraints satisfied as equalities at the solution by temporarily choosing some ‘active constraints’ at a given iteration and solving the corresponding equality constrained problem. This set of active constraints is then updated as the iterations proceed. This type of method is in contrast with ‘interior point’ methods where an adaptive logarithmic barrier prevents iterates to satisfy constraints exactly, keeping them ‘interior’ to the feasible domain. Chapters 12 and 13 of Conn *et al.* (2000) provide a description of these method classes.

Interior point methods perform well with respect to active set methods when the combinatorial aspect of selecting the correct active bounds dominate. However, active set methods are superior when identification of the active constraints is efficient. Our active-set QO algorithm for disjoint constraints uses efficient classical projection techniques suitable for bound constrained optimization problems to identify the active constraints. These techniques are in general not applicable when the optimization problem is subject to other constraints as well, in which case more generic but less efficient techniques have to be used. However, here we exploit during minimization that moving along the projected search direction on $\mathbf{y}$ does not influence the constraints on $\mathbf{x}$, thereby allowing the use of these efficient projection techniques to find the active set.

The active-set QO algorithm for disjoint constraints is stated as Algorithm 1. In this description, we used the notation that, if $\mathcal{I}$ is a subset of $\{1, \dots, p\}$ and $\mathbf{v}$ a vector in $\mathbb{R}^p$, then the vector $\mathbf{v}^{\mathcal{I}}$ of dimension $|\mathcal{I}|$ (the number of elements in $\mathcal{I}$) is the vector $\mathbf{v}$ reduced to its components in $\mathcal{I}$. Analogously, $M^{\mathcal{I}}$ is the matrix M reduced to its columns (and rows, if it is symmetric) whose index is in $\mathcal{I}$.

We now discuss the successive steps of this algorithm. If not available at the onset, a feasible point can be computed in *Step 0* by solving a linear least-squares problem for $\mathbf{x}_0$ and choosing any $\mathbf{y}_0 \geq \ell$. The set $\mathcal{A}_k$ defined in (3), *Step 1*, is the ‘active set’ for iteration k and contains the indices of all variables that are exactly at their bound and that would leave the feasible region if moved along the steepest-descent direction. The algorithm fixes them

for the duration of the k th iteration, making the corresponding bound constraints ‘active’. The variables indexed by the complementary set $\mathcal{I}_k$ are called ‘free variables’ and are allowed to vary within their bounds at iteration k . Since, as we observe below, (4) ensures optimality of the $\mathbf{x}$ variables, termination of the algorithm occurs in *Step 2* at an approximate solution when all the free components of $\mathbf{y}$ are optimized to a certain accuracy, that is, when $\nabla_{\mathbf{y}}^{\mathcal{I}_k} \mathcal{J}(\mathbf{x}_k, \mathbf{y}_k)$ is sufficiently small. A search direction is found in *Step 3* by solving (in the free variables) the standard Karush–Kuhn–Tucker (KKT) optimality conditions for problem (1) restricted to $(\mathbf{x}, \mathbf{y}^{\mathcal{I}_k})$, which are given by the system (4). Note that the vector $\mathbf{w}_k$ then contains the Lagrange multipliers associated with the equality constraints. The third line of (4) imposes that $\mathbf{A}\mathbf{s}_k = \mathbf{0}$, and it is important that this equation is satisfied with high accuracy if exact feasibility with respect to the linear equality constraint is to be preserved. We refer the reader to Gould and Toint (2002) for an in-depth discussion of this point. With this caveat, the system (4) can be solved using a Krylov solver like MINRES or GMRES (Saad, 1996 gives a description of these methods), or by a ‘constrained preconditioned’ conjugate gradient method (Gould *et al.*, 2001; Gould and Toint, 2002). If this is the case, any preconditioner must also be reduced (in its $\mathbf{y}$ part) to the subset of currently free variables indexed by $\mathcal{I}_k$. If the dimension and sparsity of $\mathbf{P}$ allows (which is typically not the case in weather forecasting), a stable factorization can also be used to solve (4) accurately. Because the equality and bound constraints are disjoint, the step size computation in *Step 4* ensures feasibility of $\mathbf{y}$, without altering feasibility of $\mathbf{x}$ with respect to the linear equality constraints (ensured by the third line of (4)). Observe that the trajectory of the vector $\max[\mathbf{y}_k + \alpha \mathbf{v}_k, \mathcal{L}]$ for $\alpha \geq 0$ is the projection of the half line $\mathbf{y}_k + \alpha \mathbf{v}_k$ on the feasible domain for the bound constraints. Thus *Step 4* simply finds the first minimizer of a quadratic function along a piecewise linear path. The next iteration $k + 1$, starts in *Step 1* where a new active set $\mathcal{A}_{k+1}$ and new complementary set $\mathcal{I}_{k+1}$ is defined, possibly freeing variables and activating new ones.

2.2 | A projected algorithmic variant

Often the matrix $\mathbf{A}$ has a very simple form, for example when representing conservation of mass. In the 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 $\mathbf{A}$ by choosing $\mathbf{Z}$ to be the symmetric orthogonal projection matrix onto that nullspace and applying the change of variable $\mathbf{x} = \mathbf{Z}\tilde{\mathbf{x}}$ which leads to the

Algorithm 1. QO algorithm for disjoint constraints

Step 0: Initialization. A feasible starting point $(\mathbf{x}_0, \mathbf{y}_0)$ is given (i.e., $\mathbf{A}\mathbf{x}_0 = \mathbf{b}$, $\mathbf{y}_0 \geq \mathcal{L}$), as well as an accuracy threshold $\epsilon > 0$. Set $k = 0$.

Step 1: Active-set update.

$$\mathcal{A}_k \stackrel{\text{def}}{=} \{i \in \{1, \dots, p\} \mid [\mathbf{y}_k]_i = \mathcal{L}_i \text{ and } \nabla_{\mathbf{y}} \mathcal{J}(\mathbf{x}_k, \mathbf{y}_k) > 0\}$$

$$\text{and } \mathcal{I}_k \stackrel{\text{def}}{=} \{1, \dots, p\} \setminus \mathcal{A}_k. \quad (3)$$

Step 2: Termination test.

Terminate if $\|\nabla_{\mathbf{y}} \mathcal{J}(\mathbf{x}_k, \mathbf{y}_k)^{\mathcal{I}_k}\| \leq \epsilon \|\nabla_{\mathbf{y}} \mathcal{J}(\mathbf{x}_0, \mathbf{y}_0)^{\mathcal{I}_0}\|$.

Step 3: Search direction computation. Solve

$$\begin{pmatrix} \mathbf{P}_{\mathbf{xx}} & \mathbf{P}_{\mathbf{xy}}^{\mathcal{I}_k} & \mathbf{A}^T \\ (\mathbf{P}_{\mathbf{xy}}^{\mathcal{I}_k})^T & \mathbf{P}_{\mathbf{yy}}^{\mathcal{I}_k} & \mathbf{0} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{s}_k \\ \mathbf{v}_k^{\mathcal{I}_k} \\ \mathbf{w}_k \end{pmatrix} = - \begin{pmatrix} \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{x}_k, \mathbf{y}_k) \\ \nabla_{\mathbf{y}}^{\mathcal{I}_k} \mathcal{J}(\mathbf{x}_k, \mathbf{y}_k) \\ \mathbf{0} \end{pmatrix} \quad (4)$$

Step 4: Projected search. Determine $\alpha > 0$ such that $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$ is the first minimizer of

$\mathcal{J}(\mathbf{x}_k + \alpha \mathbf{s}_k, \max[\mathbf{y}_k + \alpha \mathbf{v}_k, \mathcal{L}])$, where $\mathbf{v}_k$ is an expansion of $\mathbf{v}_k^{\mathcal{I}_k}$ to dimension p obtained by inserting zero components for $i \in \mathcal{A}_k$.

problem

$$\min_{\tilde{\mathbf{x}}, \mathbf{y}} \tilde{\mathcal{J}}(\tilde{\mathbf{x}}, \mathbf{y}) \stackrel{\text{def}}{=} ((\mathbf{Z}\mathbf{g}_{\mathbf{x}})^T \mathbf{g}_{\mathbf{y}}^T) \begin{pmatrix} \tilde{\mathbf{x}} \\ \mathbf{y} \end{pmatrix} + \frac{1}{2} (\tilde{\mathbf{x}}^T \mathbf{y}^T) \begin{pmatrix} \mathbf{Z}\mathbf{P}_{\mathbf{xx}}\mathbf{Z} & \mathbf{Z}\mathbf{P}_{\mathbf{xy}} \\ \mathbf{P}_{\mathbf{xy}}^T\mathbf{Z} & \mathbf{P}_{\mathbf{yy}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}} \\ \mathbf{y} \end{pmatrix} \quad (5)$$

subject to

$$\mathbf{y} \geq \mathcal{L}. \quad (6)$$

Problem (5)–(6) is now a bound-constrained quadratic problem. Note that, without the disjoint constraints assumption, inequality constraint (6) would not be simple bounds and therefore more complex algorithms than classical projection techniques would be required. It should be mentioned that problem (5)–(6) is no longer strictly convex. However, it remains strictly convex in the nullspace of $\mathbf{A}$, which is the range space of $\mathbf{Z}$ and thus the range space of the problem’s Hessian matrix. As a consequence, standard Krylov methods such as Conjugate Gradients (CG; Hestenes and Stiefel 1952) may still be applied, including for large-scale instances (e.g. Conn *et al.*, 1992; Dostál 1997; Lin and Moré 1999). Indeed, such methods are known to operate only in the range space of the problem’s

Algorithm 2. Projected QO algorithm for disjoint constraints

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

Step 1: Active-set update. Define

$$\begin{aligned} \mathcal{A}_k &\stackrel{\text{def}}{=} \{i \in \{1, \dots, p\} \mid [\mathbf{y}_k]_i = \mathcal{C}_i\} \quad \text{and} \\ \mathcal{I}_k &\stackrel{\text{def}}{=} \{1, \dots, p\} \setminus \mathcal{A}_k. \end{aligned} \quad (7)$$

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

- $\|\nabla_{\mathbf{y}} \tilde{\mathcal{J}}(\tilde{\mathbf{x}}_k, \mathbf{y}_k)\|^{L_k} \leq \epsilon \|\nabla_{\mathbf{y}} \tilde{\mathcal{J}}(\tilde{\mathbf{x}}_0, \mathbf{y}_0)\|^{L_0}$
- $\|\nabla_{\tilde{\mathbf{x}}} \tilde{\mathcal{J}}(\tilde{\mathbf{x}}_k, \mathbf{y}_k)\| \leq \epsilon \|\nabla_{\tilde{\mathbf{x}}} \tilde{\mathcal{J}}(\tilde{\mathbf{x}}_0, \mathbf{y}_0)\|^{L_0}$
- $\nabla_{\mathbf{y}_i} \tilde{\mathcal{J}}(\tilde{\mathbf{x}}_k, \mathbf{y}_k) \geq 0$ for $i \in \mathcal{A}_k$.

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

$$\tilde{\mathcal{J}}\left(\mathbf{x}_k - \alpha \nabla_{\tilde{\mathbf{x}}} \tilde{\mathcal{J}}(\tilde{\mathbf{x}}_k, \mathbf{y}_k), \max\left[\mathbf{y}_k - \alpha \nabla_{\mathbf{y}} \tilde{\mathcal{J}}(\tilde{\mathbf{x}}_k, \mathbf{y}_k), \mathcal{C}\right]\right).$$

Set

$$\begin{aligned} \mathcal{A}_{k,C} &\stackrel{\text{def}}{=} \{i \in \{1, \dots, p\} \mid [\mathbf{y}_k^c]_i = \mathcal{C}_i\}, \quad \text{and} \\ \mathcal{C}_k &\stackrel{\text{def}}{=} \{1, \dots, p\} \setminus \mathcal{A}_{k,C}. \end{aligned} \quad (8)$$

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

$$\begin{aligned} & \left((\mathbf{Z}\mathbf{g}_x)^T, \mathbf{g}_{\mathbf{y}^{C_k}}^T \right) \begin{pmatrix} \tilde{\mathbf{x}} \\ \mathbf{y}^{C_k} \end{pmatrix} \\ & + \frac{1}{2} \begin{pmatrix} \tilde{\mathbf{x}}^T \\ \mathbf{y}^{C_k,T} \end{pmatrix} \begin{pmatrix} \mathbf{Z}\mathbf{P}_{xx}\mathbf{Z} & \mathbf{Z}\mathbf{P}_{xy^{C_k}} \\ \mathbf{P}_{xy^{C_k}}^T \mathbf{Z} & \mathbf{P}_{y^{C_k}y^{C_k}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}} \\ \mathbf{y}^{C_k} \end{pmatrix} \end{aligned} \quad (9)$$

subject to

$$\mathbf{y}^{C_k} \geq \mathcal{C}^{C_k}. \quad (10)$$

Terminate the CG iterations when one (or more) bound(s) of indices $j_1, \dots, j_s$ are encountered, or if convergence occurred within prescribed accuracy. If CG was terminated because bounds were encountered, perform a line search similar to Step 3 along the search direction to update the iterate and repeat Step 4 after redefining $\mathcal{C}_k = \mathcal{C}_k \setminus \{j_1, \dots, j_s\}$ and using the new iterate as a starting point.

Hessian when started in this range space (with $\mathbf{Z}\mathbf{g}_x$ in our case). The use of the CG method also presents further advantages in our context. The first is that, when suitably preconditioned, it is well-known to be efficient for large problems whose size precludes methods using matrix factorizations and where the number of iterations is often limited by computational considerations. The second is that it only requires products of the Hessian matrix with vectors, which can be cheaply computed without explicitly forming $\mathbf{Z}\mathbf{P}_{xx}\mathbf{Z}$. The third is that it can be ‘warm-started’ in the sense that it is able to exploit a good guess of the active set – which is expected when convergence occurs – instead of returning to the full space at each major iteration. We therefore choose to apply the CG method with diagonal preconditioning to minimize the quadratic cost function in the current *face* of the feasible domain, that is the subspace spanned by the $\tilde{\mathbf{x}}$ and the free $\mathbf{y}$ 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 free variables, and the first step of CG performs a (in this case projected) line search along this direction. The resulting minimizer is known as the generalized Cauchy point and the decrease of the objective function obtained at this point is known to guarantee overall convergence of the algorithm (Conn *et al.*, 2000, chapter 12). Methods differ essentially by their face changing mechanisms but require 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 stated as Algorithm 2.

3 | NUMERICAL EXPERIMENTS

In order to illustrate the behaviour of algorithms presented in Section 2 in a 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 set-up.

3.1 | Model

The modified shallow-water model has been used for testing different data assimilation algorithms in Haslehner *et al.* (2016) and 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) and data assimilation algorithms

(Cohn and Parrish, 1991; Zeng and Janjić, 2016; Zeng *et al.*, 2017). As the name suggests, the shallow-water equations have been altered in Würsch and Craig (2014) 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 fluid height level h fields. The one-dimensional modified shallow-water model consists of following equations:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{\partial(\phi + \gamma^2 r)}{\partial x} = \beta_u + D_u \frac{\partial^2 u}{\partial x^2}, \quad (11)$$

with

$$\phi = \begin{cases} \phi_c & \text{if } h > h_c \\ gh & \text{otherwise,} \end{cases} \quad (12)$$

$$\frac{\partial r}{\partial t} + u \frac{\partial r}{\partial x} = D_r \frac{\partial^2 r}{\partial x^2} - \eta r - \begin{cases} \delta \frac{\partial u}{\partial x}, & \text{if } h > h_r \text{ and } \frac{\partial u}{\partial x} < 0 \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

$$\frac{\partial h}{\partial t} + \frac{\partial(uh)}{\partial x} = D_h \frac{\partial^2 h}{\partial x^2}. \quad (14)$$

The parameters D_u , D_r , D_h are diffusion coefficients in the equations for momentum, rain mass and the 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 convection, convective updrafts are modelled by stochastic Gaussian forcing β_u added at random locations in each model time step. These random wind perturbations, when they create convergence, elevate the fluid surface. Conditional instability is modelled by modifying the geopotential ϕ , which is set to a low constant value ϕ_c once the height exceeds a threshold representing the level of free convection h_c . The gradient of the geopotential will force the fluid into the region of the decreased potential. The fluid level will then 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 the 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 centred differences on a staggered grid. To compute evolution of the dynamical system, the time variable is discretized into time steps of 5 s. 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. If rain becomes negative during model integration, negative values are set to zero.

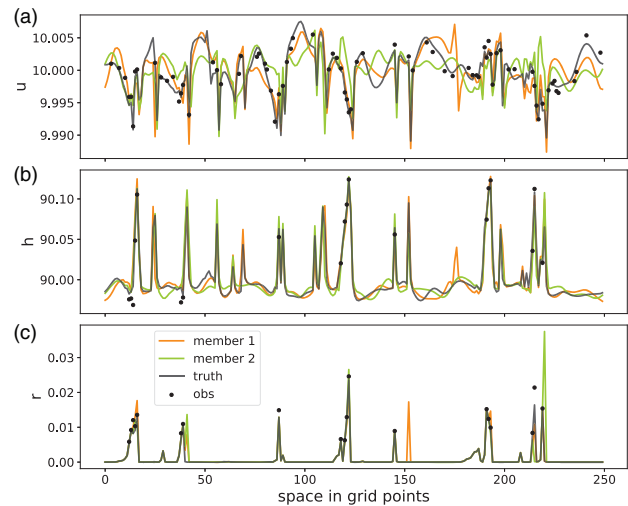


FIGURE 1 Fields of (a) u , (b) h and (c) r produced by running the modified shallow-water model starting from identical initial conditions after 120 model time steps. One of the experiments is chosen as the nature run (black line). Observations (black circles) are simulated at locations where it rains in the nature run (truth) plus a random noise for all fields and an extra 25% observations are additionally simulated for the u field at other locations. Noise is Gaussian for the h and u fields with zero mean and standard deviations of $0.001 \text{ m} \cdot \text{s}^{-1}$ 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

The other model parameters are given by

$$\begin{aligned} h_0 &= 90 \text{ m}, & h_c &= 90.02 \text{ m}, & h_r &= 90.4 \text{ m}, \\ D_u &= D_h = 25000 \text{ m}^2 \cdot \text{s}^{-1}, & D_r &= 200 \text{ m}^2 \cdot \text{s}^{-1}, \\ \phi_c &= 899.77 \text{ m}^2 \cdot \text{s}^{-2}, & \eta &= 2.5 \times 10^{-4} \text{ s}^{-1}, \\ & & \text{and } \delta &= 1/300. \end{aligned}$$

The Gaussian stochastic forcing β_u has a half-width of four grid points and an amplitude of $0.002 \text{ m} \cdot \text{s}^{-2}$. The fields produced by running this model starting from identical initial conditions are illustrated in Figure 1 after 120 model time steps (which is equivalent to 10 min in real time). As illustrated in Figure 1, starting with exactly the same initial conditions, the position of clouds (height field) and rain can be different after 120 model time steps, mimicking fast-changing convective storms whose intermittency is one of the challenges of data assimilation on the convective scale.

3.2 | Data assimilation experiments

The data assimilation set-up follows Ruckstuhl and Janjić (2018). To investigate the performance of the two algorithms presented in Section 2, we conduct a twin

experiment, where we consider a model run to be the true state $\mathbf{z} = (\mathbf{u}^T, \mathbf{h}^T, \mathbf{r}^T)^T$, which we call the nature run. At each assimilation time, a vector of synthetic observations $\mathbf{z}^{\text{obs}} \in \mathfrak{R}$ is created by randomly perturbing the nature run such that $\mathbf{z}^{\text{obs}} = \mathbf{H}\mathbf{z} + \epsilon^{\text{obs}}$, where $\mathbf{H}$ is the $o \times n$ matrix that determines the location and physical nature of the observations, while $\epsilon^{\text{obs}} \in \mathfrak{R}^o$ 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 $0.001 \text{ m}\cdot\text{s}^{-1}$ 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, which is equivalent to 10 min in real time. The location of the observations are chosen to mimic radar data, and are simulated at locations where it rains in the nature run. As illustrated in Figure 1, it only takes 120 time steps (10 min) for the location of the observations in the nature run to differ from that of rain in each ensemble member. In addition, an extra 25% of observations are assimilated for the u field at other locations, mimicking observations from a different measurement system. The observation-error covariance matrix $\mathbf{R}$ is taken to be diagonal with values on the diagonal corresponding to the variance of the distributions used for generating the observation error vector ϵ^{obs} . Although the number of observations, and therefore the matrix sizes for the linear observation operator $\mathbf{H}$, and $\mathbf{R}$ will change from one assimilation cycle to next, we omit the time index for notational simplicity.

We employ the Quadratic Programming Ensemble (QPEnS) algorithm (Janjić *et al.*, 2014) for data assimilation. As the stochastic EnKF, the QPEnS uses an ensemble of predictions (backgrounds) $\{\mathbf{z}^{b,i}\}_{i=1}^N$ to calculate sample background-error covariance matrix $\mathbf{P}^b$

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

with $\bar{\mathbf{z}}^b$ representing the ensemble mean. Since the sample covariance $\mathbf{P}^b$ is singular when $(N-1) < n$, it is in our experiments localized with a blockwise matrix where each block uses fifth-order polynomial function (Ruckstuhl and Janjić, 2018 give details) with cut-off at eight grid points. The QPEnS algorithm then calculates the analysis ensemble from the background ensemble, the perturbed observations, the localized background-error covariance matrix and the observation-error covariance matrix. However,

instead of using the Kalman filter formulae to obtain the analyses, it solves the following quadratic programming problem for each member $i = 1, \dots, N$:

$$\mathbf{z}^{a,i} = \mathbf{z}^{b,i} + \arg \min_{\delta \mathbf{z}^i} \frac{1}{2} \left[\delta \mathbf{z}^{i,T} (\mathbf{P}^b)^{-1} \delta \mathbf{z}^i + \mathbf{f}^{i,T} \mathbf{R}^{-1} \mathbf{f}^i \right] \quad (15)$$

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

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

$$\delta \mathbf{r}^i \geq -\mathbf{r}^{b,i} \quad \text{and} \quad \mathbf{e}_h^T \delta \mathbf{h}^i = 0, \quad \text{for } i = 1, \dots, N, \quad (16)$$

where $\mathbf{e}_h^T = (1, \dots, 1)$ is of the same size as $\mathbf{h}$, and $\delta \mathbf{r}$, $\delta \mathbf{h}$ are the corresponding components of $\delta \mathbf{z}$. Note that the inequality in (16) is equivalent to $\mathbf{r}^{a,i} \geq 0$, and the equality to $\mathbf{e}_h^T \mathbf{h}^{a,i} = \mathbf{e}_h^T \mathbf{h}^{b,i}$.

Therefore, in our set-up $n = 750$, $p = 250$, $\mathbf{x} = (\mathbf{u}^T, \mathbf{h}^T)^T$, $\mathbf{A} = (\mathbf{0}_u^T, \mathbf{e}_h^T)$ of size $1 \times (n-p)$ with $\mathbf{0}_u^T = (0, \dots, 0)$ of same size as $\mathbf{u}$ and $\mathbf{y} = \mathbf{r}$, $N = 50$, $\mathbf{P} = (\mathbf{P}^b)^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$ and for each ensemble member i the gradient is calculated as $(\mathbf{g}_x^i, \mathbf{g}_y^i)^T = -\mathbf{H}^T \mathbf{R}^{-1} (\mathbf{z}^{\text{obs},i} - \mathbf{H}\mathbf{z}^{b,i})$. Data are assimilated every 10 min and for a total of 36 cycles. In order to obtain robust results, twin experiments were repeated many times for different random seeds.

4 | RESULTS

In the data assimilation experiments discussed in this section, the minimization of the problem (15)–(16) is carried out using the algorithms described in Section 2. First, the convergence of both algorithms is evaluated in Section 4.1. Next, the effect on the data assimilation results of only partially solving the minimization problems in the QPEnS is investigated in Section 4.2.

4.1 | Convergence of the minimization algorithms

To apply Algorithm 1 we use an LU decomposition with pivoting to solve (4) accurately. The lower block of this system only contains one equation in this case. Table 1 illustrates the performance of the algorithm for one ensemble

k	$J(\mathbf{z})$	$ I_k $	$\frac{\ [\nabla_y J(\mathbf{x}_k, \mathbf{y}_k)]^{I_k}\ }{\ [\nabla_y J(\mathbf{x}_0, \mathbf{y}_0)]^{I_0}\ }$	α_k	$\ \mathbf{z}_k - \mathbf{z}^*\ $
1	-2.562770×10^3	107	0.5458	1.0	1.434792×10^{-1}
2	-2.578068×10^3	118	0.1562	1.0	4.935165×10^{-2}
3	-2.580350×10^3	124	0.1234	1.0	1.701261×10^{-2}
4	-2.582139×10^3	129	0.0434	0.6262	1.292911×10^{-2}
5	-2.582549×10^3	133	0.0036	0.8768	2.030012×10^{-3}
1	-6.368068×10^2	28	645.312	1.0	5.233164×10^{-2}
2	-6.471606×10^2	31	47.648	1.0	7.730555×10^{-4}
3	-6.472081×10^2	33	28.050	0.3942	5.210342×10^{-4}
4	-6.472324×10^2	33	4.4×10^{-12}	1.0	1.371545×10^{-10}

Note: To illustrate the accuracy, the difference is calculated between result of each major iteration $\mathbf{z}_k$ and the solution $\mathbf{z}^*$. The upper (lower) rows report the first (tenth) data assimilation cycle for one of the ensemble members.

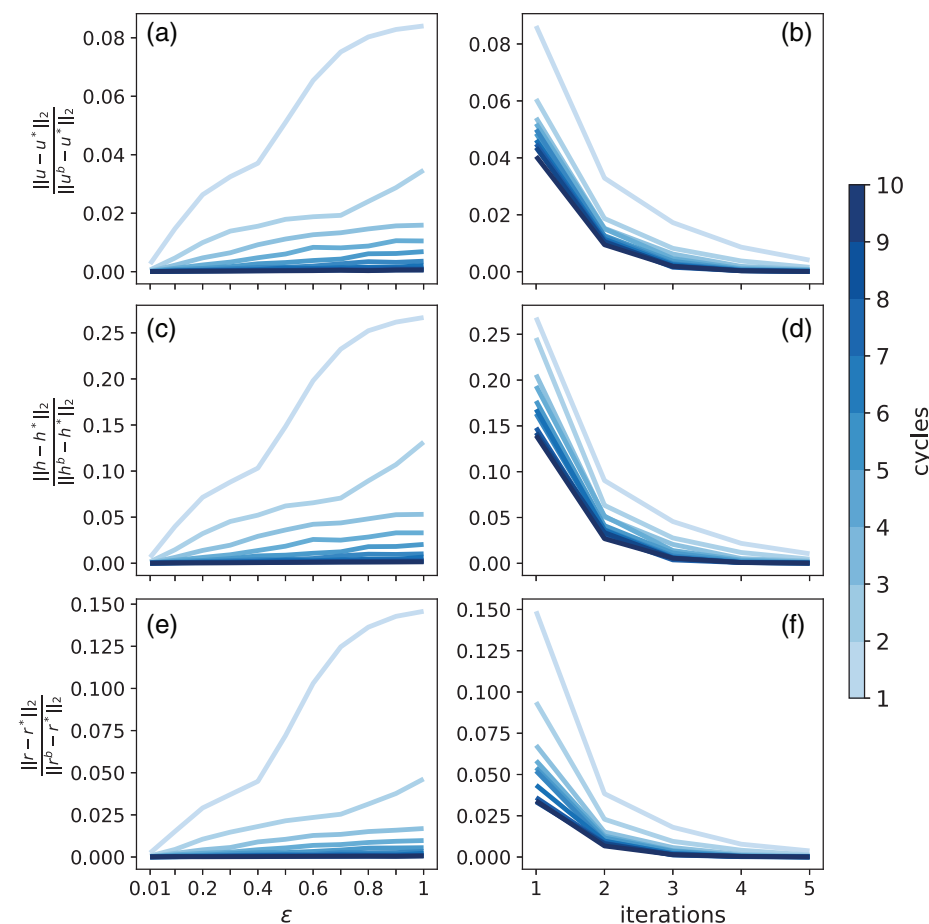


FIGURE 2 Scaled RMSE with respect to the solution as a function of (a, c, e) ϵ and (b, d, f) iterations k performed for Algorithm 1. The scaling is done using the RMSE of the background state (which is used as starting point $(\mathbf{x}_0, \mathbf{y}_0)$ of the algorithm) with respect to the solution, so that the scaled RMSE values range from 0 (fully solved problem) to 1 (0 iterations performed). Different shades of blue represent results for each of the first ten data assimilation cycles. The smallest ϵ value used in the experiments is 0.01, then values 0.1 to 1 in steps of 0.1. Results for (a, c, e) were obtained by averaging over 374 experiments and for (b, d, f) over 100 experiments

member during a first data assimilation cycle and during a tenth cycle. In this example we set $\epsilon = 0.1$, and we see (in the last column of Table 1) that the RMSE compared to the solution decreases at each iteration, reaching an accuracy of $\sim 10^{-3}$ after five iterations in the first cycle example and a tenth digit accuracy after four iterations in the tenth cycle example. The minimization stops when the ratio of

the gradient to the initial gradient reaches ϵ . The third column in the Table 1 shows the number of inactive constraints, that is, the k th guess of the set of grid points where it rains, which determines the free variables in the computation of the search direction (4). After the search direction is calculated, the size of the allowed step is computed in Step 4 of Algorithm 1, and reported in the fifth column in

Table 1. For completeness, the value of the cost function (1) in each iteration is included in Table 1 (second column). As illustrated in Table 1, Algorithm 1 converges in only five iterations on this example, while a more general interior point method like the CVXOPT package (Andersen *et al.*, 2010) typically requires more iterations to solve the same problem (not shown).

Figure 2 shows the dependence of the RMSE with respect to the solution on the specified accuracy and on the number of iterations performed for the first ten cycles of data assimilation. By the tenth cycle, the background is relatively close to the solution and high accuracy is reached within a small number of iterations. Note that the stopping criterion of the minimization algorithm is relative to the gradient of the cost function at the starting point ($\mathbf{x}_0, \mathbf{y}_0$), which in our case is the background. That this gradient decreases as the background becomes more skilful in further cycles makes sense, leading to a higher accuracy requirement for a given ϵ , as is clear from Figure 2a,c,e. To illustrate the cost of this algorithm, the number of iterations is plotted as a function of the specified relative accuracy in Figure 3. The number of iterations increases to six only for very small ϵ .

In Algorithm 2 minimization is done in the projected space with $\mathbf{Z} = \mathbf{I} - \mathbf{A}^T \mathbf{A} / \mathbf{e}_h^T \mathbf{e}_h$. We found that a suitable preconditioner for the CG is an important ingredient for efficiency. In our case, the diagonal of the matrix entering the CG algorithm is used. We refer the reader to Gratton *et al.* (2013) for a discussion of suitable precondition strategies in the context of data assimilation. It is also known that Algorithm 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. The CG subproblems are terminated when the residual is smaller than a fraction $\tilde{\epsilon}$ of the residual computed from the initial

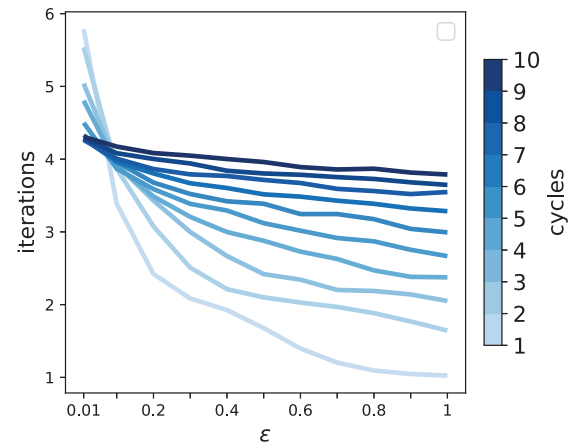


FIGURE 3 Number of iterations k required for Algorithm 1 as a function of the specified relative accuracy ϵ . Different shades of blue represent results for each of the first ten data assimilation cycles. The smallest ϵ value used in the experiments is 0.01, then values 0.1 to 1 in steps of 0.1. Results were obtained by averaging over 374 experiments

quantities entering the CG algorithm. We somewhat arbitrarily fixed $\tilde{\epsilon} = \epsilon$ for all the experiments presented here. As in Table 1, Table 2 illustrates the performance of the algorithm for one ensemble member during the first and tenth data assimilation cycle. As before, we choose $\epsilon = 0.1$ in this example, and we see that in this case the error with respect to the solution decreases with each iteration, reaching an accuracy of 10^{-3} after three outer iterations and a total of 131 CG iterations in the first cycle. In the tenth data assimilation cycle, four digits of accuracy are achieved after four outer iterations and a total of 214 CG iterations. Note that the minimization terminates only once all conditions in Step 2 of Algorithm 2 are satisfied. The third column of Table 2 shows the number of free variables at each iteration. This number decreases, contrary to the increase over iterations in Table 1. Although this

TABLE 2 Illustration of the performance of Algorithm 2

k	$J(\mathbf{z})$	$ \mathcal{I}_k $	$\frac{\ \nabla_{\mathbf{y}} J(\mathbf{x}_k, \mathbf{y}_k)\ _{\mathcal{I}_k}}{\ \nabla_{\mathbf{y}} J(\mathbf{x}_0, \mathbf{y}_0)\ _{\mathcal{I}_0}}$	α_k	CG its	faces	$\ \mathbf{z}_k - \mathbf{z}^*\ $
1	-2.56403×10^3	173	4.0178×10^{-4}	5.883×10^{-7}	84	35	1.03437×10^{-1}
2	-2.58203×10^3	141	0.0016	8.031×10^{-7}	32	5	1.56615×10^{-2}
3	-2.58256×10^3	133	0.0011	2.819×10^{-7}	15	2	7.34838×10^{-3}
1	-7.29541×10^2	50	1.86	3.353×10^{-8}	78	24	8.85338×10^{-3}
2	-7.29892×10^2	36	1.15	4.700×10^{-15}	24	3	2.22762×10^{-3}
3	-7.29932×10^2	34	0.10	1.991×10^{-14}	72	2	1.39713×10^{-4}
4	-7.29932×10^2	33	0.02	3.059×10^{-14}	40	1	3.65735×10^{-5}

Note: 'CG its' denotes the 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 $\mathbf{z}_k$ and the solution $\mathbf{z}^*$. The upper (lower) rows report the first (tenth) data assimilation cycle for one of the ensemble members.

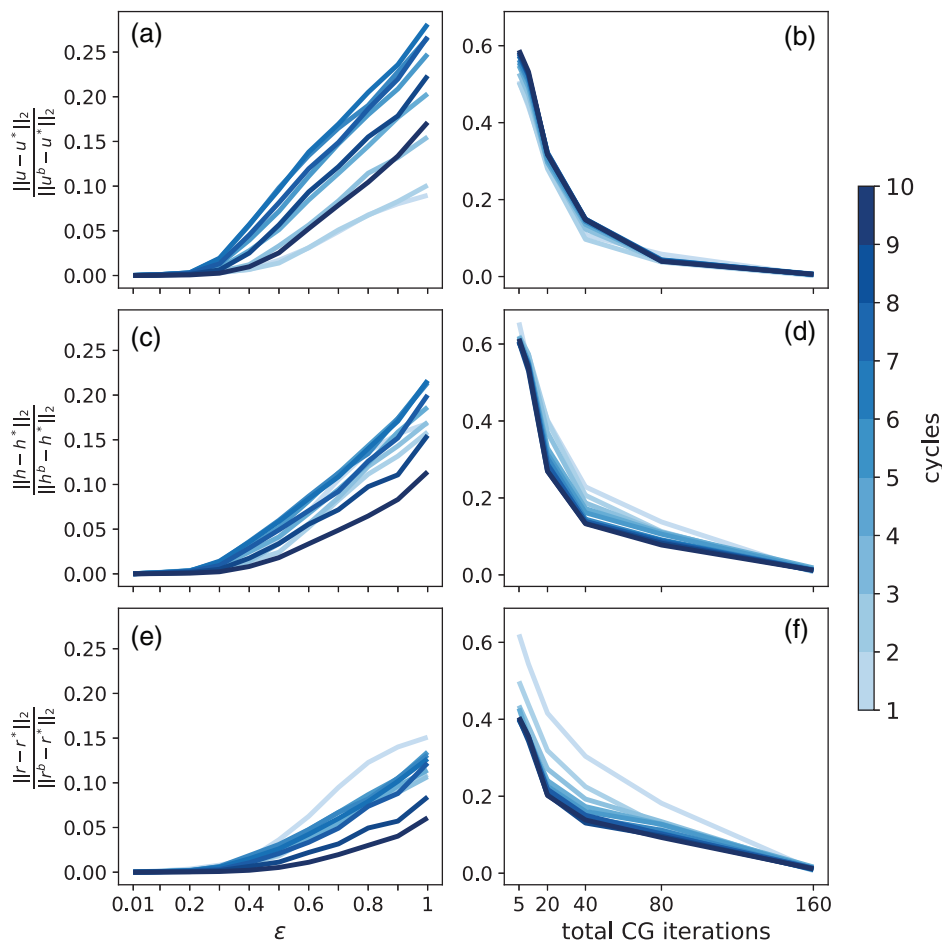


FIGURE 4 Scaled RMSE with respect to the solution as a function of (a, c, e) ϵ and (b, d, f) the accumulated number of inner CG iterations performed for Algorithm 2. The scaling is done as in Figure 4.2. Different shades of blue represent results for each of the first ten data assimilation cycles. The smallest ϵ value used in the experiments is 0.01, then values 0.1 to 1 in steps of 0.1. Results were obtained by averaging over 500 experiments

happens in most of the experiments, opposite cases occur as well. Most likely this is a problem-specific feature and not a general property of the two algorithms. The fifth column reports the step-size to the Cauchy point computed in Step 3. The fact that this step-size is very small illustrates the poor conditioning of the problem. We also note that the number of successive progressively smaller subspaces in which the CG is applied (listed under the ‘faces’ heading) is quite small and decreases as the guess of the optimal active set improves (and the associated number of CG iterations – under the heading ‘CG its’ – decreases). For completeness, the value of the cost function in each iteration is also included in the Table.

The dependence of the RMSE on the specified accuracy and the number of total CG iterations performed is shown in Figure 4 for the first ten cycles of data assimilation. The total CG iterations refers to the total number of CG iterations accumulated on all outer iterations. In contrast to Algorithm 1, a clear correlation between the cycle number and the accuracy as a function of ϵ (Figure 4a,c,e) is not observed. We believe this is due to the stopping criterion of the CG subproblems, which depends on the latest iterate. In addition preconditioning is involved, which affects the stopping criterion as well.

Figure 5a shows the number of accumulated CG iterations as a function of ϵ . Note that the CG iterations dominate the costs of the algorithm, thereby allowing a natural cost-comparison between Algorithm 2 and any other variational data assimilation algorithms that apply the CG for minimization. Figure 5b is a direct result of our termination choice for the CG. Generally, when the subproblem in Step 4 is solved to a high accuracy, few outer iterations are needed and vice versa. We found that the number of CG iterations accumulated over all outer iterations is fairly insensitive to the exact choice of termination test for the CG, as long as it is not too lenient (not shown). For $\tilde{\epsilon} = \epsilon$, Figure 5b indicates that the average ratio of outer iterations to CG iterations per outer iteration is larger for lower accuracy of the solution.

4.2 | Data assimilation results

When the minimization problems of type (15)–(16) are solved to high accuracy, both algorithms yield the same data assimilation results as for example obtained with the interior point method of Python used in Ruckstuhl and Janjić (2018). However it is interesting to investigate the

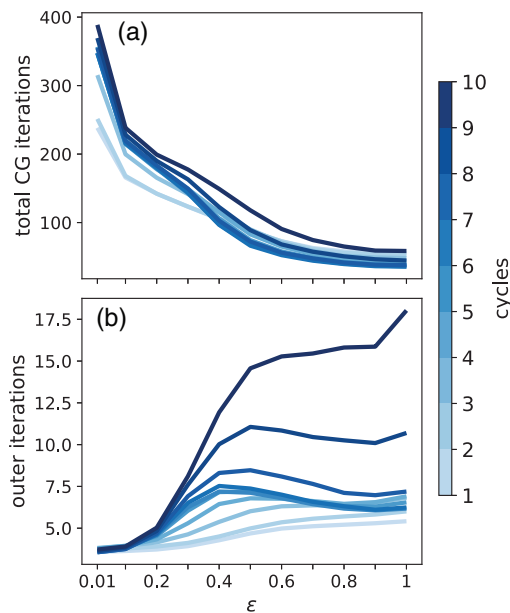


FIGURE 5 Total number of (a) CG iterations and (b) outer iterations required for Algorithm 2 as a function of the specified relative accuracy ϵ . Different shades of blue represent results for each of the first ten data assimilation cycles. The smallest ϵ value used in the experiments is 0.01, then values 0.1 to 1 in steps of 0.1. Results were obtained by averaging over 500 experiments

effect of terminating the algorithms early, as this might be necessary due to the size of the minimization problem encountered in practice. An advantage of the presented algorithms is that they guarantee that the constraints are satisfied at each iteration. Interior point methods, on the other hand, either violate the constraints throughout the minimization process or only allow approaching the bounds near the solution (depending on the precise formulation of the algorithm), making them unsuitable for early termination in our context. We performed an experiment where Algorithm 1 is terminated after one iteration and Algorithm 2 once a total of ten CG iterations is reached. The results are presented in Figures 6 and 7. As reference, both the QPEns with fully solved minimization problems (red line) and the EnKF (blue line), which is equivalent to the QPEns without constraints, are plotted as well. As thoroughly discussed in Ruckstuhl and Janjić (2018), the QPEns produces smaller RMSE than the EnKF. The differences between the QPEns and the EnKF persist during 6 hr forecasts and may especially be seen in the height field. The difference in the rain prediction is most pronounced during the first 3 hr. The approximate QPEns for Algorithm 1 (orange line) is very close to the exact QPEns, indicating that performing one iteration for each minimization problem already yields desired results. The approximate QPEns for Algorithm 2 (green line) yields worse analyses for wind variable u than the

EnKF, although this is corrected throughout the forecast. The other variables clearly benefit from the imposed constraints, even when the minimization problems are not fully solved. For the rain variable r , the RMSE is even lower than for the exact QPEns. The RMSE of forecasts starting with a perfect initial ensemble (black line) is also plotted for comparison, in order to illustrate the intrinsic predictability of this model. Although all ensemble members initially start from the same true state in the perfect ensemble, they differ during the forecast due to stochastic perturbations in the wind field (11).

To understand the differences in RMSE, let us consider one snapshot of the results in Figure 7, with the truth plotted in black. One notices that, due to the small localization radius and the observations that mimic radar data, differences in the results are primarily the spurious convection that arises for the EnKF. Although differences in the solution of the constrained minimization problems (corresponding to the QPEns) and unconstrained minimization problems (corresponding to the EnKF) are generally small, Ruckstuhl and Janjić (2018) show that data assimilation errors of the EnKF will accumulate over time leading to large errors in total mass and total rain after repeating data assimilation 250 times, that is, in less than one day. We refer the interested reader to Ruckstuhl and Janjić (2018) for the investigation of the differences between the QPEns and the EnKF analysis for different data assimilation settings, including ensemble size. The approximate QPEns for Algorithm 1 closely follows the exact QPEns. This does not hold for the approximate QPEns of Algorithm 2, where large differences are detected in the wind (u) variables. Also in the other variables the differences from the exact solution are more pronounced than for the approximate QPEns of Algorithm 1.

5 | CONCLUSION

In NWP, we can expect that even global models will reach the convection-permitting resolution in the forthcoming decade. 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 of how different existing methods can represent uncertainty in data assimilation and forecasting steps of both observations and numerical models (Janjić *et al.*, 2018). But also, data assimilation methods would require a better way to estimate (multivariately) prognostic hydrometeors. This is a challenging task, due to the computational

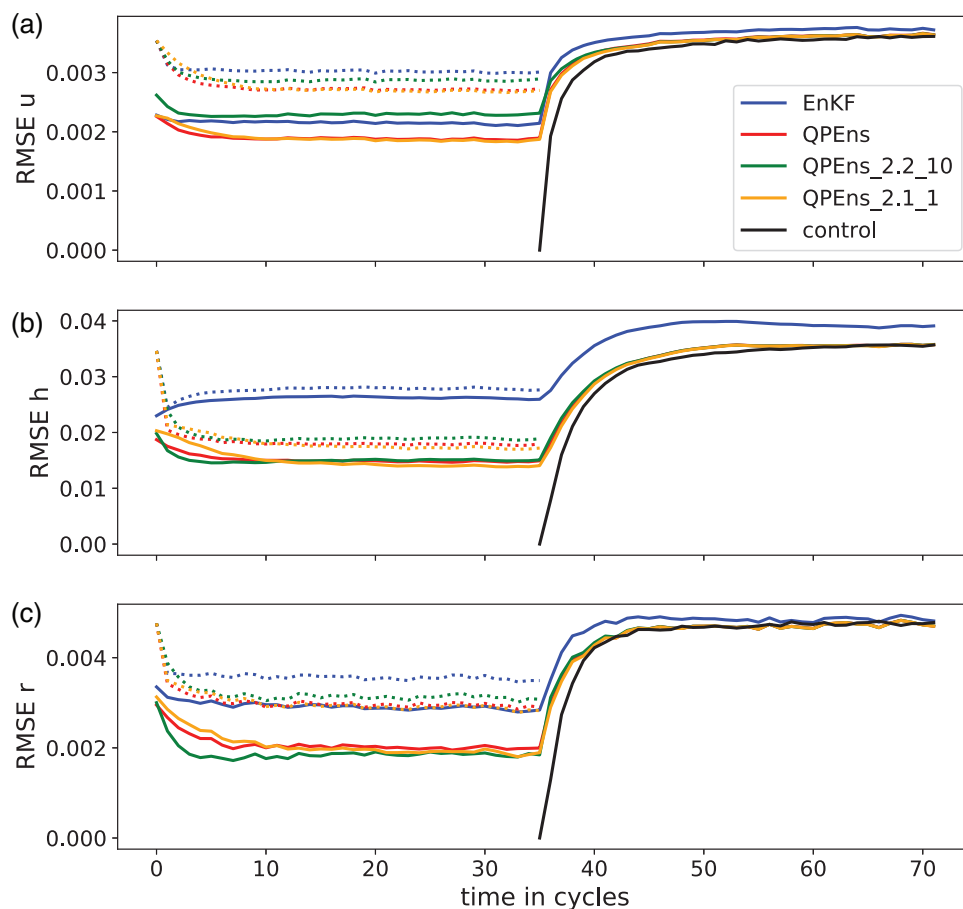


FIGURE 6 RMSE for (a) u , (a) h and (c) r against the nature run of the analysis ensemble mean (solid) and background ensemble mean (dotted) for the QPEns (red), the EnKF (blue) and the approximate QPEns of Algorithm 1 (orange) and Algorithm 2 (green). After 36 data assimilation cycles, free forecasts start. To illustrate the predictability of the model, a perfect ensemble of 50 members is run forward in time (black line) with identical initial conditions. RMSE values are obtained by averaging over 500 twin experiments.

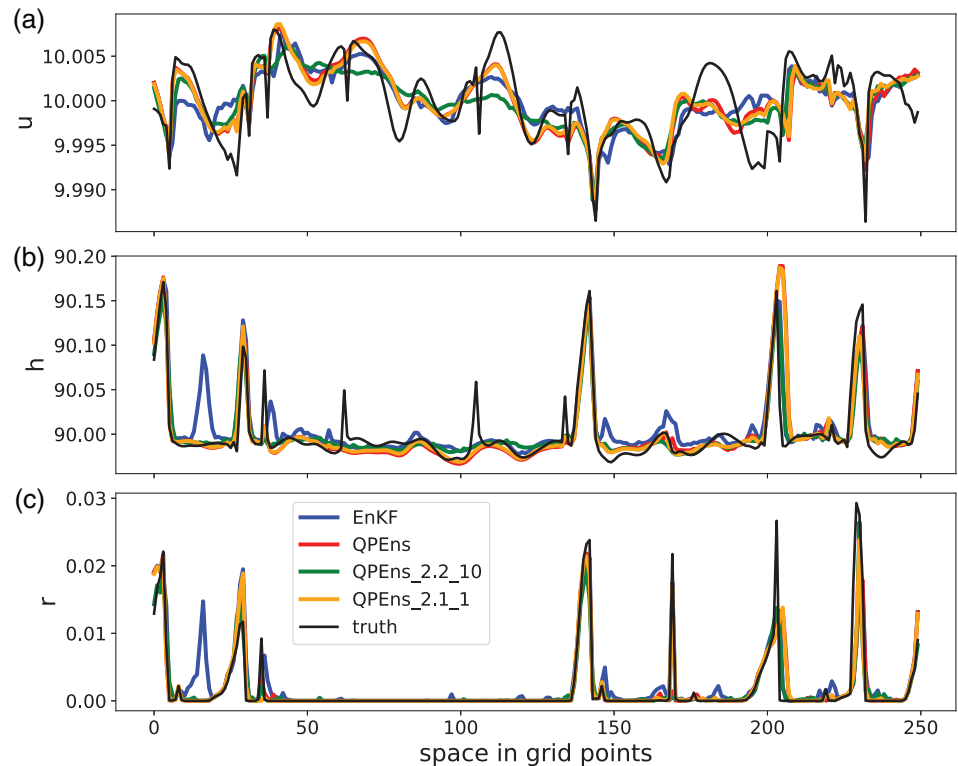
costs of producing non-negative estimates, the frequency of their estimation, and a need to initialize probabilistic prediction. We have presented two projection algorithms which exploit the disjoint nature of constraints typically occurring in weather forecasting applications. While projection methods 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. In addition, in contrast to interior point methods, the presented algorithms live in the feasible region, thereby allowing early termination of the minimization process without violation of constraints. We showed that, after imposing a termination test of a total of ten CG iterations for Algorithm 2, the QPEns still significantly outperformed the EnKF.

The first of our methods, Algorithm 1, is more efficient than an interior point approach on a representative example, but still requires solving the Karush–Kuhn–Tucker system (4), which is impractical in weather forecasting applications due to the problem

frequency and size. By contrast, Algorithm 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 and conditioning of the problem are such that the conjugate gradient algorithm is allowed to converge, the number of outer iterations required by Algorithm 2 is smaller or comparable to that required by Algorithm 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.

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. Both algorithms could be applied in pure 3D-Var or ensemble/hybrid setting, and for Algorithm 2 already existing implementations of the CG could be modified. However, our current implementation is in physical space. Further computational savings would be possible by revising the constrained problem in observational space and assimilating data in batches (similar to that done for the unconstrained problem in Cohn *et al.*,

FIGURE 7 Snapshot for modified shallow-water variables (a) u , (b) h and (c) r after 36 cycles of data assimilation. Depicted are true fields (black), background ensemble mean of the EnKF (blue), QPEns (red), approximate QPEns for Algorithm 1 (orange) and Algorithm 2 (green)



1998), combined with algorithms that draw inspiration from Gratton and Tshimanga 2009 and Gratton *et al.*, 2011,2012). The alternative would be a more elaborate preconditioning, which would allow transformation of the data assimilation problem into a smaller control space. For example, in an ensemble setting, the square root of the inverse of the Hessian is easy to calculate and could be used for preconditioning (Zupanski, 2005). However, the current challenge with this approach is that inequality constraints are then more difficult to handle. Finally, more thought should be given to possible improvements of the details of the face-changing mechanism, such as exploiting suggestions in Dostál (1997).

ACKNOWLEDGEMENTS

YR and TJ are grateful 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*. TJ is also grateful 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).

ORCID

Tijana Janjić  <https://orcid.org/0000-0002-8837-0879>

REFERENCES

- Andersen, M.S., Dahl, J. and Vandenberghe, L. (2010) Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones. *Mathematical Programming, Series C*, 2, 167–201.
- Arakawa, A. and Lamb, V.R. (1980) A potential enstrophy and energy conserving scheme for the shallow-water equations. *Monthly Weather Review*, 109, 18–36.
- Bertino, L., Evensen, G. and Wackernagel, H. (2003) Sequential data assimilation techniques in oceanography. *International Statistical Reviews*, 71, 223–241.
- Bishop, C.H., Etherton, B.J. and Majumdar, S. (2001) Adaptive sampling with the ensemble transform Kalman filter. Part I: theoretical aspects. *Monthly Weather Review*, 129, 420–436.
- Burgers, G., van Leeuwen, P.J. and Evensen, G. (1998) Analysis scheme in the Ensemble Kalman Filter. *Monthly Weather Review*, 126, 1719–1724.
- Bühner, M., Caya, A., Pogson, L., Carrieres, T. and Pestieau, P. (2013) A new Environment Canada regional ice analysis system. *Atmosphere–Ocean*, 51(1), 18–34.
- Cohn, S.E., da Silva, A., Guo, J., Sienkiewicz, M. and Lamich, D. (1998) Assessing the effects of data selection with the DAO physical-space statistical analysis system. *Monthly Weather Review*, 126, 2913–2926.
- Cohn, S.E. (1997) An introduction to estimation theory. *Journal of the Meteorological Society of Japan*, 75, 257–288.
- Cohn, S.E. and Parrish, D.F. (1991) The behavior of forecast error covariances for a Kalman Filter in two dimensions. *Monthly Weather Review*, 119(8), 1757–1785.
- Conn, A.R., Gould, N.I.M. and Toint, P.L. (1992) *LANCELOT: A Fortran Package for Large-scale Nonlinear Optimization (Release A)*, (330. p). Berlin: Springer.

- Conn, A.R., Gould, N.I.M. and Toint, P.L. (2000) *Trust-region Methods*, (978. p). Philadelphia, PA: SIAM. MPS-SIAM Series on Optimization.
- Courtier, P., Andersson, E., Heckley, W., Pailleux, J., Vasiljevic, D., Hamrud, M., Hollingsworth, A., Rabier, F. and Fisher, M. (1998) The ECMWF implementation of three-dimensional variational assimilation (3D-Var). I: formulation. *Quarterly Journal of the Royal Meteorological Society*, 127, 1783–1807.
- Cullen, M.J.P., Davies, T., Mawson, M.H., James, J.A. and Coulter, S.C. (1997) An overview of numerical methods for the next generation UK NWP and climate model. *Atmosphere–Ocean*, 35, 425–444.
- Dostál, Z. (1997) Box constrained quadratic programming with proportioning and projections. *SIAM Journal on Optimization*, 7(3), 871–887.
- Evensen, G. (1994) Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99, 10143–10162.
- Evensen, G. (2003) The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53, 343–367.
- Gill, P.E., Murray, W. and Wright, M.H. (1981) *Practical Optimization*. Cambridge, MA: Academic Press.
- Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H. (1984) Procedures for optimization problems with a mixture of bounds and general linear constraints. *ACM Transactions on Mathematical Software*, 10, 282–298.
- Gould, N.I.M., Hribar, M.E. and Nocedal, J. (2001) On the solution of equality constrained quadratic problems arising in optimization. *SIAM Journal on Scientific Computing*, 23(4), 1375–1394.
- Gould, N.I.M. and Toint, P.L. (2002) An iterative working-set method for large-scale non-convex quadratic programming. *Applied Numerical Mathematics*, 43(1–2), 109–128.
- Gratton, S. and Tshimanga, J. (2009) An observation-space formulation of variational assimilation using a restricted preconditioned conjugate-gradient algorithm. *Quarterly Journal of the Royal Meteorological Society*, 135, 1573–1585.
- Gratton, S., Toint, P.L. and Tshimanga, J. (2011) Range-space variants and inexact matrix-vector products in Krylov solvers for linear systems arising from inverse problems. *SIAM Journal on Matrix Analysis and Applications*, 32(3), 969–986. <https://doi.org/10.1137/090780493>
- Gratton, S., Gürol, S., Toint, P.L., Tshimanga, J. and Weaver, A. (2012). Krylov methods in the observation space for data assimilation. Report No. 58/2012, Mathematisches Forschungsinstitut Oberwolfach, Germany.
- Gratton, S., Gürol, S. and Toint, P.L. (2013) Preconditioning and globalizing conjugate gradients in dual space for quadratically penalized nonlinear-least squares problems. *Computational Optimization and Applications*, 54(1), 1–25.
- Gustafsson, N., Janjić, T., Schraff, C., Leuenberger, D., Weissman, M., Reich, H., Brousseau, P., Montmerle, T., Wattrelot, E., Bucánek, A., Mile, M., Hamdi, R., Lindskog, M., Barkmeijer, J., Dahlbom, M., Macpherson, B., Ballard, S., Inverarity, G., Carley, J., Alexander, C., Dowell, D., Liu, S., Ikuta, Y. and Fujita, T. (2018) Survey of data assimilation methods for convective-scale numerical weather prediction at operational centres. *Quarterly Journal of the Royal Meteorological Society*, 144, 1218–1256.
- Haslehner, M., Janjić, T. and Craig, G.C. (2016) Testing particle filters on simple convective-scale models. Part 2: a modified shallow-water model. *Quarterly Journal of the Royal Meteorological Society*, 142, 1628–1646.
- Hestenes, M.R. and Stiefel, E. (1952) Methods of conjugate gradients for solving linear systems. *Journal of the National Bureau of Standards*, 49, 409–436.
- Hunt, B.R., Kostelich, E.J. and Szunyogh, I. (2007) Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D*, 230, 112–126.
- Janjić, T., McLaughlin, D., Cohn, S.E. and Verlaan, M. (2014) Conservation of mass and preservation of positivity with ensemble-type Kalman filter algorithms. *Monthly Weather Review*, 142, 755–773.
- Janjić, T., Potthast, R. and Van Leeuwen, P.J. (2018) Editorial. *Quarterly Journal of the Royal Meteorological Society*, 144, 1189–1190. <https://doi.org/10.1002/qj.3382>.
- Ketefian, G.S. and Jacobson, M.Z. (2009) 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.
- Lauvernet, C., Brankart, J.-M., Castruccio, F., Broquet, G., Brasseur, P. and Verron, J. (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.
- Lien, G.-Y., Kalnay, E. and Miyoshi, T. (2013) Effective assimilation of global precipitation: simulation experiments. *Tellus A*, 65(1). <https://doi.org/10.3402/tellusa.v65i0.19915>.
- Lin, C. and Moré, J.J. (1999) Newton's method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, 9(4), 1100–1127.
- Lorenz, E.N. (1969) The predictability of a flow which possesses many scales of motion. *Tellus*, XXI(3), 289–307.
- Rabier, F., Jaervinen, H., Klinker, E., Mahfouf, J.-F. and Simmons, A. (2000) The ECMWF operational implementation of four-dimensional variational assimilation. I: experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, 126, 1143–1170.
- Ruckstuhl, Y. and Janjić, T. (2018) Parameter and state estimation with ensemble Kalman filter-based approaches for convective-scale data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144, 826–841.
- Saad, Y. (1996) *Iterative Methods for Sparse Linear Systems*. Boston, MA: PWS Publishing Company.
- Sadourny, R. (1975) The dynamics of finite-difference models of the shallow-water equations. *Journal of the Atmospheric Sciences*, 32, 680–689.
- Simon, E. and Bertino, L. (2009) Application of the Gaussian anamorphosis to assimilation in a 3D coupled physical-ecosystem model of the North Atlantic with the EnKF: a twin experiment. *Ocean Science*, 5, 495–510.
- Simon, E. and Bertino, L. (2012) 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.
- Sommer, M. and Névir, P. (2009) 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.
- Sun, J., Xue, M., Wilson, J.W., Zawadzki, I., Ballard, S.P., Onville-Hoomeyer, J., Joe, P., Barker, D.M., Li, P.-W., Golding, B., Xu, M. and Pinto, J. (2014) Use of NWP for nowcasting

- convective precipitation: recent progress and challenges. *Bulletin of the American Meteorological Society*, 95(3), 409–426.
- Trémolet, Y. (2007) Incremental 4D-Var convergence study. *Tellus A: Dynamic Meteorology and Oceanography*, 59(5), 706–718.
- Würsch, M. and Craig, G.C. (2014) A simple dynamical model of cumulus convection for data assimilation research. *Meteorologische Zeitschrift*, 23, 483–490.
- Zeng, Y. and Janjić, T. (2016) Study of conservation laws with the Local Ensemble Transform Kalman Filter. *Quarterly Journal of the Royal Meteorological Society*, 142, 2359–2372.
- Zeng, Y., Janjić, T., Ruckstuhl, Y. and Verlaan, M. (2017) Ensemble-type Kalman filter algorithm conserving mass, total energy and enstrophy. *Quarterly Journal of the Royal Meteorological Society*, 143, 2902–2914.
- Zupanski, M. (2005) Maximum Likelihood Ensemble Filter: theoretical aspects. *Monthly Weather Review*, 133, 1710–1726.

How to cite this article: Janjić T, Ruckstuhl Y, Toint PL. A data assimilation algorithm for predicting rain. *QJR Meteorol. Soc.* 2021;147:1949–1963. <https://doi.org/10.1002/qj.4004>