Latent data assimilation for the quasi-geostrophic (QG) model

Workshop on Machine Learning for Earth Observation and Prediction
ECMWF-ESA, Reading, November 15th 2022

Mathis Peyron (2nd year PhD student)
Supervisors: Selime Gürol, Serge Gratton, Rémi Druilhe

Some data assimilation context

Data assimilation deals with the problem of estimating model parameters or model state variables by combining prior estimates and observations, together with information about their uncertainties, in a statistically optimal manner.

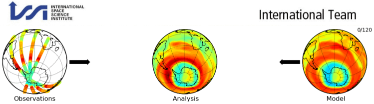


Figure – From the *International Space Science Institute* (Switzerland) website

Context : the state estimation problem

- Let us consider the following stochastic equation which models the system evolution along time (for instance the evolution of atmosphere or ocean) :

$$x_k = \mathcal{M}_k(x_{k-1}) + \eta_k$$
$$y_k = \mathcal{H}_k(x_k) + v_k$$

  - $x_k \in \mathbb{R}^n$ is the system state at time t_k
  - $y_k \in \mathbb{R}^m$ is the observed state value (data) at time t_k
  - $\mathcal{M}_k$ involves the propagation process of the dynamical system
  - $\mathcal{H}_k$ is the observation operator at time t_k
  - η_k is the model error considered as a random variable
  - v_k is the observation error considered as a random variable
  - The errors are assumed to be unbiased and uncorrelated in time

- We wish to estimate x_k given a sequence of observations over time, $Y_k = [y_1^o, \ldots, y_k^o]$. Since the system is stochastic, we cannot expect to obtain the solution exactly. Instead, we consider its conditional probability distribution function, $p(x_k|Y_k)$.

Solution algorithm : Ensemble data assimilation

➤ Ensemble algorithms aim to estimate the mean and the covariance matrix of - supposedly - Gaussian distributions with a few random samples.

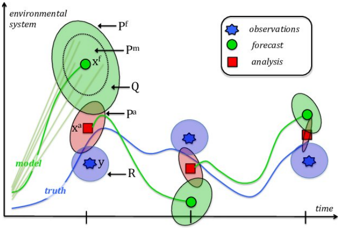


Figure – Ensemble data assimilation, *from Tandeo et al. A Review of Innovation-Based Methods to Jointly Estimate Model and Observation Error Covariance Matrices in Ensemble Data Assimilation*

Propagation step $(p(x_k|Y_{k-1}))$:

$$\boldsymbol{x}_k^f = \mathcal{M}_k(\boldsymbol{x}_{k-1}^a)$$

$$P_k^f = M_k P_{k-1}^a M_k^{\mathrm{T}} + Q_k$$

Correction step $(p(x_k|Y_k))$:

$$K_k = P_k^f H_k^{\mathrm{T}} (R_k + H_k P_k^f H_k^{\mathrm{T}})^{-1}$$

$$\boldsymbol{x}_k^a = \boldsymbol{x}_k^f + K_k(y_k - \mathcal{H}_k(x_k^f))$$

$$P_k^a = (I - K_k H_k) P_k^f$$

Solution algorithm : Ensemble data assimilation

- Ensemble algorithms aim to estimate the mean and the covariance matrix of - supposedly - Gaussian distributions with a few random samples.

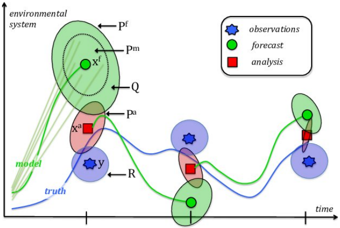


Figure – Ensemble data assimilation, *from Tandeo et al. A Review of Innovation-Based Methods to Jointly Estimate Model and Observation Error Covariance Matrices in Ensemble Data Assimilation*

- In this work, we focus on the Ensemble Transform KF with model error (ETKF-Q) [Fillion et al., 2020].

Ensemble Transform Kalman Filter (ETKF)

- Find a low rank approximation to the error covariance matrix, i.e.

$$P^f \simeq S^f(S^f)^T$$

  where $P^f \in \mathbb{R}^{n\times n}$, $S_f \in \mathbb{R}^{n\times r}$ with $r \ll n$.
- S^f is obtained with random samples.

- Find a low rank approximation to the error covariance matrix, i.e.

$$P^f \simeq S^f (S^f)^T$$

where $P^f \in \mathbb{R}^{n \times n}$, $S_f \in \mathbb{R}^{n \times r}$ with $r \ll n$.

- S^f is obtained with random samples.
- Let us define $d = y - \mathcal{H}(x^f)$. The solution can be rewritten as

$$\boldsymbol{x}^a = x^f + Kd = x^f + P^f H^{\mathrm{T}} (R + H P^f H^{\mathrm{T}})^{-1} d$$

Ensemble Transform Kalman Filter (ETKF)

- Find a low rank approximation to the error covariance matrix, i.e.

$$P^f \simeq S^f (S^f)^T$$

where $P^f \in \mathbb{R}^{n\times n}$, $S_f \in \mathbb{R}^{n\times r}$ with $r \ll n$.

- S^f is obtained with random samples.
- Let us define $d = y - \mathcal{H}(x^f)$. The solution can be rewritten as

$$\begin{aligned}\boldsymbol{x}^a &= x^f + Kd = x^f + P^f H^{\mathrm{T}}(R + HP^f H^{\mathrm{T}})^{-1} d \\ &= \underbrace{x^f}_{\in\mathbb{R}^n} + \underbrace{S^f}_{\in\mathbb{R}^{n\times r}} \underbrace{(S^f)^T H^{\mathrm{T}} \left(R + HS^f (S^f)^T H^{\mathrm{T}}\right)^{-1} d}_{w^a \in \mathbb{R}^r,\ \text{minimization in } \mathbb{R}^m}\end{aligned}$$

- Find a low rank approximation to the error covariance matrix, i.e.

$$P^f \simeq S^f (S^f)^T$$

where $P^f \in \mathbb{R}^{n\times n}$, $S_f \in \mathbb{R}^{n\times r}$ with $r \ll n$.

- S^f is obtained with random samples.
- Let us define $d = y - \mathcal{H}(x^f)$. The solution can be rewritten as

$$\begin{aligned}
\boldsymbol{x}^a &= x^f + Kd = x^f + P^f H^{\mathrm{T}}(R + HP^f H^{\mathrm{T}})^{-1} d \\
&= \underbrace{x^f}_{\in\mathbb{R}^n} + \underbrace{S^f}_{\in\mathbb{R}^{n\times r}} \underbrace{(S^f)^T H^{\mathrm{T}} \left(R + HS^f (S^f)^T H^{\mathrm{T}}\right)^{-1} d}_{w^a \in\mathbb{R}^r, \text{minimization in } \mathbb{R}^m} \\
&= \underbrace{x^f}_{\in\mathbb{R}^n} + \underbrace{S^f}_{\in\mathbb{R}^{n\times r}} \underbrace{\left(I_r + (Y^f)^T R^{-1} Y^f\right)^{-1} (Y^f)^T R^{-1} d}_{w^a \in\mathbb{R}^r, \text{minimization in } \mathbb{R}^r}
\end{aligned}$$

where $Y^f = HS^f$ in the linear case. In the nonlinear case, this operator can be approximated with the use of nonlinear operator $\mathcal{H}$.

Ensemble Transform Kalman Filter (ETKF)

- Find a low rank approximation to the error covariance matrix, i.e.

$$P^f \simeq S^f (S^f)^T$$

where $P^f \in \mathbb{R}^{n\times n}$, $S_f \in \mathbb{R}^{n\times r}$ with $r \ll n$.

- S^f is obtained with random samples.
- Let us define $d = y - \mathcal{H}(x^f)$. The solution can be rewritten as

$$\begin{aligned}
\boldsymbol{x}^a &= x^f + Kd = x^f + P^f H^\mathrm{T}(R + HP^f H^\mathrm{T})^{-1} d \\
&= \underbrace{x^f}_{\in\mathbb{R}^n} + \underbrace{S^f}_{\in\mathbb{R}^{n\times r}} \underbrace{(S^f)^T H^\mathrm{T}\left(R + HS^f(S^f)^T H^\mathrm{T}\right)^{-1} d}_{w^a\in\mathbb{R}^r,\text{minimization in } \mathbb{R}^m} \\
&= \underbrace{x^f}_{\in\mathbb{R}^n} + \underbrace{S^f}_{\in\mathbb{R}^{n\times r}} \underbrace{\left(I_r + (Y^f)^T R^{-1} Y^f\right)^{-1} (Y^f)^T R^{-1} d}_{w^a\in\mathbb{R}^r,\text{minimization in } \mathbb{R}^r}
\end{aligned}$$

where $Y^f = HS^f$ in the linear case. In the nonlinear case, this operator can be approximated with the use of nonlinear operator $\mathcal{H}$.

- The solution is a linear combination of the column vectors of the low-rank matrix S^f.

ETKF-Q

Algorithm 1: ETKF-Q

Initialization :

$E_0^f = [x_0^{f,1}, x_0^{f,2}, \dots, x_0^{f,r}] \in \mathbb{R}^{n\times r}$, independent random samples (members)

Propagation step :

$E_k^f = \mathcal{M}(E_{k-1}^a)$ $\Rightarrow E_k^f = [x_k^{f,1}, x_k^{f,2}, \dots, x_k^{f,r}] \in \mathbb{R}^{n\times r}$

$P_k^f = \Lambda_k^f \left(\Lambda_k^f\right)^T$

Including model error (update P_k^f) : $P_k^f = \Lambda_k^f (\Lambda_k^f)^T + Q$

Calculate S_k^f such that $P_k^f = S_k^f (S_k^f)^T$

Correction step :

Solve $\left(I_r + (Y_k^f)^T R_k^{-1} Y_k^f\right) w_k^a = (Y_k^f)^T R_k^{-1} d_k$ $\Rightarrow [Y_k^f]_i = \frac{\mathcal{H}(x_k^{f,i}) - \bar{y}^f}{\sqrt{r-1}}$

$x_k^a = \bar{x}_k^f + S_k^f w_k^a$ $\Rightarrow \bar{x}_k^f$ is the mean of the members

$S_k^a = S_k^f \left[I_r + (Y_k^f)^T R_k^{-1} Y_k^f\right]^{-1/2}$

$x_k^{a,i} = x_k^a + inflation \times \sqrt{r-1}\, [S_k^a]_i$ $\Rightarrow 1/r \sum_1^r x_k^{a,i} = x_k^a$

Main assumption : we consider a dynamical system which can be represented in a low-dimensional space.

- Then, is it possible to leverage the existence of this low-dimensional space to :
  1. reduce the computational cost
  2. reduce the effect of the noise
  3. extract the main features of the data

- Linear methods : Proper Orthogonal Decomposition ([Lumley, 1967, Holmes et al., 2012]), Krylov-subspace methods ([Gallivan et al., 1994]), Balanced truncation ([Moore, 1981]), etc...

- Nonlinear methods : quadratic manifold ([Geelen et al., 2022]), autoencoders ([Goodfellow et al., 2016]), etc...

⇒ *in this talk, we focus on nonlinear methods based on deep learning techniques.*

A few references

- [Mack et al., 2020] uses convolutional auto-encoders to find a reduced space for 3D-variational data assimilation. No dynamics.
- [Casas, 2018] uses Ensemble KF in the reduced space where the dynamics are propagated by using surrogate model based on deep learning. The dimension reduction is based on linear methods.
- [Amendola et al., 2020] uses a variant of Kalman Filter in a latent space of convolutional AE and propagates the model by using surrogate model based on deep learning. AE and surrogate model are trained separately. Observations are mapped to the latent space.
- More recently, [Zhuang et al., 2022], [Cheng et al.,] and [Mohd Razak et al., 2022] also investigated combining autoencoders with a deep-learning based surrogate.

- [Peyron et al., 2021] performs ETKF-Q in a latent space and propagates the model by using surrogate model based on deep learning. Main contributions :

What we propose ?

- [Peyron et al., 2021] performs ETKF-Q in a latent space and propagates the model by using surrogate model based on deep learning. Main contributions :

  - All-at-once optimization (AE + Surrogate)

What we propose ?

- [Peyron et al., 2021] performs ETKF-Q in a latent space and propagates the model by using surrogate model based on deep learning. Main contributions :
  - All-at-once optimization (AE + Surrogate)
  - Surrogate model loss with several time-steps (increase stability)

- [Peyron et al., 2021] performs ETKF-Q in a latent space and propagates the model by using surrogate model based on deep learning. Main contributions :
  - All-at-once optimization (AE + Surrogate)
  - Surrogate model loss with several time-steps (increase stability)
  - Model error in the latent space

What we propose ?

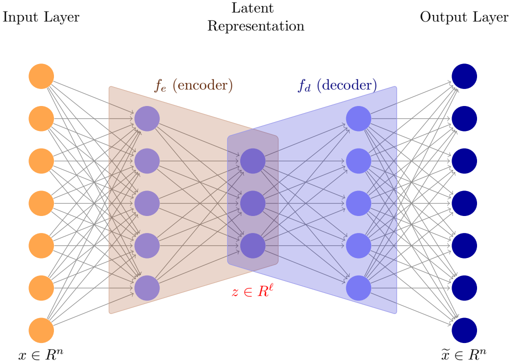
- [Peyron et al., 2021] performs ETKF-Q in a latent space and propagates the model by using surrogate model based on deep learning. Main contributions :
    - All-at-once optimization (AE + Surrogate)
    - Surrogate model loss with several time-steps (increase stability)
    - Model error in the latent space
    - Observations are not projected to the latent space. No additional error from this mapping.

What we propose ?

- [Peyron et al., 2021] performs ETKF-Q in a latent space and propagates the model by using surrogate model based on deep learning. Main contributions :
  - All-at-once optimization (AE + Surrogate)
  - Surrogate model loss with several time-steps (increase stability)
  - Model error in the latent space
  - Observations are not projected to the latent space. No additional error from this mapping.
  - Easy to implement for a general reduced space KFs.

Model reduction with autoencoders

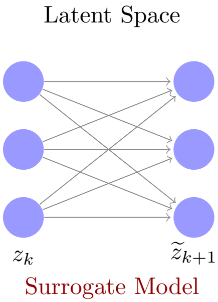
- An autoencoder is a neural network that is trained to reconstruct the input data with the constraint to extract their main features into a reduced space.
- The network consists of two parts : an encoding function $f_e : \mathbb{R}^n \to \mathbb{R}^\ell$ and a decoder $f_d : \mathbb{R}^\ell \to \mathbb{R}^n$



- We learn the functions by minimizing the loss :

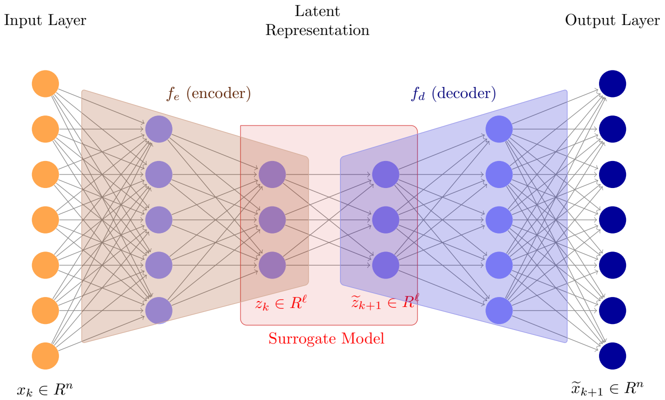
$$\min_{\theta_e, \theta_d} \|x - f_d(\underbrace{f_e(x, \theta_e)}_{z}, \theta_d))\|_2^2$$

- Is it possible to propagate model dynamics in the latent space ?



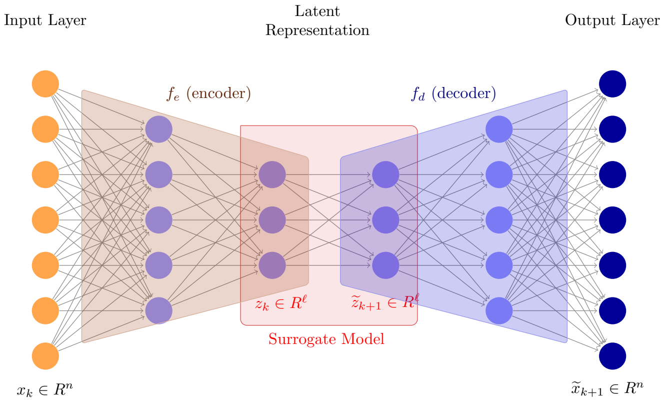
$$\min_{\theta_s} \| \underbrace{f_e(x_{k+1}, \theta_e)}_{z_{k+1}} - f_s(\underbrace{f_e(x_k, \theta_e)}_{z_k}, \theta_s) \|_2^2$$

All-at-once optimization with several time steps



$$\mathcal{L}_{sur}(\theta_e, \theta_d, \theta_s) = \frac{1}{C} \sum_{c=1}^{C} \| x_{k+c} - \underbrace{f_d(\underbrace{f_{s^c}(f_e(x_k, \theta_e), \theta_s)}_{\widetilde{z}_{k+c}}, \theta_d)}_{\widetilde{x}_{k+c}} \|_2^2$$

All-at-once optimization with several time steps



$$\mathcal{L} = \mathcal{L}_{AE} + \lambda \mathcal{L}_{sur}$$

$$\mathcal{L}_{AE}(\theta_e, \theta_d) = \frac{1}{C} \sum_{c=1}^{C} \| x_{k+c} - \underbrace{f_d(f_e(x_{k+c}))}_{\widetilde{x}_{k+c}} \|_2^2$$

- Remember that

$$\boldsymbol{x}^a = \underbrace{x^f}_{\in\mathbb{R}^n} + \underbrace{S^f}_{\in\mathbb{R}^{n\times r}} \underbrace{\left(I_r + (Y^f)^T R^{-1} Y^f\right)^{-1} (Y^f)^T R^{-1} d}_{w^a \in\mathbb{R}^r,\text{minimization in } \mathbb{R}^r}$$

- Remember that

$$\boldsymbol{x}^a = \underbrace{x^f}_{\in\mathbb{R}^n} + \underbrace{S^f}_{\in\mathbb{R}^{n\times r}} \underbrace{\left(I_r + (Y^f)^T R^{-1} Y^f\right)^{-1} (Y^f)^T R^{-1} d}_{w^a \in \mathbb{R}^r,\text{minimization in } \mathbb{R}^r}$$

- Once we have nonlinear functions, f_e and f_d, we can obtain initial random samples in the latent space, i.e. $Z_0^f = f_e(E_0^f)$ and rewrite the solution as :

$$x^a = f_d\left(\underbrace{z^f}_{\in\mathbb{R}^\ell} + \underbrace{L^f}_{\in\mathbb{R}^{\ell\times r}} \underbrace{w^a}_{\in\mathbb{R}^r}\right)$$

- Remember that

$$\boldsymbol{x}^a = \underbrace{x^f}_{\in\mathbb{R}^n} + \underbrace{S^f}_{\in\mathbb{R}^{n\times r}} \underbrace{\left(I_r + (Y^f)^T R^{-1} Y^f\right)^{-1} (Y^f)^T R^{-1} d}_{w^a \in\mathbb{R}^r,\text{minimization in } \mathbb{R}^r}$$

- Once we have nonlinear functions, f_e and f_d, we can obtain initial random samples in the latent space, i.e. $Z_0^f = f_e(E_0^f)$ and rewrite the solution as :

$$x^a = f_d\left(\underbrace{z^f}_{\in\mathbb{R}^\ell} + \underbrace{L^f}_{\in\mathbb{R}^{\ell\times r}} \underbrace{w^a}_{\in\mathbb{R}^r}\right)$$

- Y_k^f is calculated as follows :

$$[Y_k^f]_i = \frac{\mathcal{H}(f_d(z_k^{f,i})) - \bar{y}^f}{\sqrt{r-1}} \text{ with } \bar{y}^f = 1/r \sum_1^r \mathcal{H}(f_d(z_k^{f,i}))$$

Latent space DA

- Remember that

$$\boldsymbol{x}^a = \underbrace{x^f}_{\in\mathbb{R}^n} + \underbrace{S^f}_{\in\mathbb{R}^{n\times r}} \underbrace{\left(I_r + (Y^f)^T R^{-1} Y^f\right)^{-1} (Y^f)^T R^{-1} d}_{w^a \in\mathbb{R}^r,\text{minimization in } \mathbb{R}^r}$$

- Once we have nonlinear functions, f_e and f_d, we can obtain initial random samples in the latent space, i.e. $Z_0^f = f_e(E_0^f)$ and rewrite the solution as :

$$x^a = f_d\left(\underbrace{z^f}_{\in\mathbb{R}^\ell} + \underbrace{L^f}_{\in\mathbb{R}^{\ell\times r}} \underbrace{w^a}_{\in\mathbb{R}^r}\right)$$

- Y_k^f is calculated as follows :

$$[Y_k^f]_i = \frac{\mathcal{H}(f_d(z_k^{f,i})) - \bar{y}^f}{\sqrt{r-1}} \text{ with } \bar{y}^f = 1/r\sum_1^r \mathcal{H}(f_d(z_k^{f,i}))$$

- We work with random samples in the latent space.
- The solution is obtained by using nonlinear transformation from latent space to the model space.

Latent space data assimilation

Algorithm 2: Latent-ETKF-Q

Initialization :

$Z_0^f = [f_e(x_0^{f,1}), f_e(x_0^{f,2}), \ldots, f_e(x_0^{f,r})] \in \mathbb{R}^{\ell \times r}$, members in the latent space

Propagation step :

$Z_k^f = f_s(Z_{k-1}^a) \qquad \Rightarrow Z_k^f \in \mathbb{R}^{\ell \times r}$

$F_k^f = \Gamma_k^f \left(\Gamma_k^f\right)^T$

(update) $F_k^f = \Gamma_k^f (\Gamma_k^f)^T + Q_\ell \quad \Rightarrow F_k^f \in \mathbb{R}^{\ell \times \ell}$, model error in the latent space

Calculate L_k^f such that $F_k^f = L_k^f (L_k^f)^T$

Correction step :

Solve $\left(I_r + (Y_k^f)^T R_k^{-1} Y_k^f\right) w_k^a = (Y_k^f)^T R_k^{-1} d_k$

$z_k^a = \bar{z}_k^f + L_k^f w_k^a$

$L_k^a = L_k^f \left[I_r + (Y_k^f)^T R_k^{-1} Y_k^f\right]^{-1/2}$

$z_k^{a,i} = z_k^a + inflation \times \sqrt{r-1}\, [L_k^a]_i$

Solution : $x_k^a = f_d(z_k^a)$

- We may reduce the computational cost and memory when $\ell \ll n$: ☺

  the algorithm is performed in the latent space

- We may reduce the computational cost and memory when $\ell \ll n$: ☺

  the algorithm is performed in the latent space

- We may increase the overall accuracy : ☺
  - The solution is not a linear combination of the ensembles like in the standard ETK-F. The solution is an output of a nonlinear transformation learnt through physical state trajectories.
  - The error covariance matrices are propagated in the latent space. We may reduce the noise.

- We may reduce the computational cost and memory when $\ell \ll n$: ☺

  the algorithm is performed in the latent space

- We may increase the overall accuracy : ☺
  - The solution is not a linear combination of the ensembles like in the standard ETK-F. The solution is an output of a nonlinear transformation learnt through physical state trajectories.
  - The error covariance matrices are propagated in the latent space. We may reduce the noise.

- We need to define the latent space dimension ! 😐
- We need to estimate the model error in the latent space ! 😐

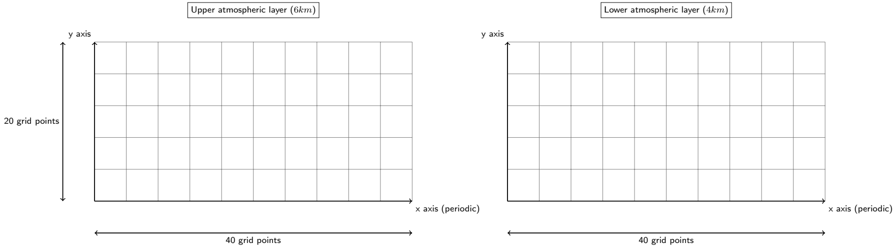
The quasi-geostrophic model is described by two variables, namely the streamfunction ψ and the potential vorticity q which are related through the following equations :

The quasi-geostrophic model is described by two variables, namely the streamfunction $\boldsymbol{\psi}$ and the potential vorticity $\boldsymbol{q}$ which are related through the following equations :

$$\begin{cases} \boldsymbol{q}_1 = \nabla^2 \boldsymbol{\psi}_1 - F_1 \left(\boldsymbol{\psi}_1 - \boldsymbol{\psi}_2 \right) + \beta y \\ \boldsymbol{q}_2 = \nabla^2 \boldsymbol{\psi}_2 - F_2 \left(\boldsymbol{\psi}_2 - \boldsymbol{\psi}_1 \right) + \beta y + R_s \end{cases} \tag{1}$$

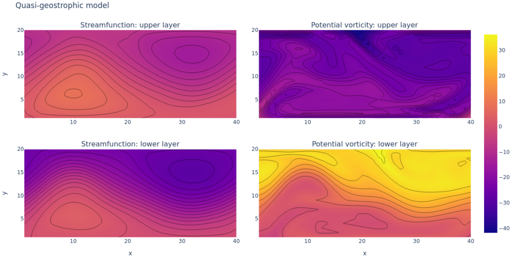
The quasi-geostrophic model is described by two variables, namely the streamfunction ψ and the potential vorticity q which are related through the following equations :

$$\begin{cases} q_1 = \nabla^2 \psi_1 - F_1 (\psi_1 - \psi_2) + \beta y \\ q_2 = \nabla^2 \psi_2 - F_2 (\psi_2 - \psi_1) + \beta y + R_s \end{cases} \quad (1)$$



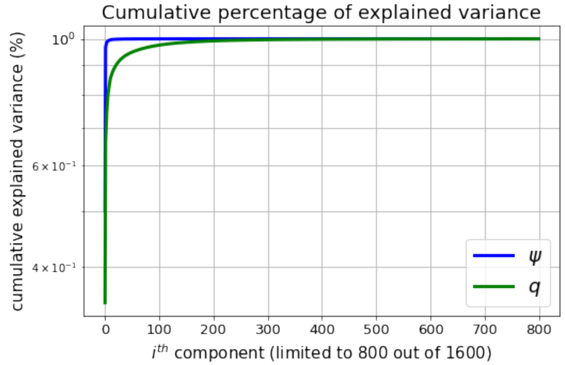
The quasi-geostrophic model is described by two variables, namely the streamfunction $\boldsymbol{\psi}$ and the potential vorticity $\boldsymbol{q}$ which are related through the following equations :

$$\begin{cases} \boldsymbol{q}_1 = \nabla^2 \boldsymbol{\psi}_1 - F_1 (\boldsymbol{\psi}_1 - \boldsymbol{\psi}_2) + \beta y \\ \boldsymbol{q}_2 = \nabla^2 \boldsymbol{\psi}_2 - F_2 (\boldsymbol{\psi}_2 - \boldsymbol{\psi}_1) + \beta y + R_s \end{cases} \tag{1}$$



The quasi-geostrophic model is described by two variables, namely the streamfunction ψ and the potential vorticity q which are related through the following equations :

$$\begin{cases} q_1 = \nabla^2 \psi_1 - F_1 (\psi_1 - \psi_2) + \beta y \\ q_2 = \nabla^2 \psi_2 - F_2 (\psi_2 - \psi_1) + \beta y + R_s \end{cases} \quad (1)$$



- Data assimilation parameters :
  - latent dimension $\ell \in \{10, 30, 100, 200\}$.
  - physical field (ψ) dimension $n = 1600$.
  - as many members as the latent dimension (*i.e.* $m = 10, 30, 100, 200$).
  - Observation error covariance matrix $R = \sigma_R I_n$ with $\sigma_R = 0.4$
  - 48 iterations/cycles (analogous to a 4-day forecast).
  - Model error covariance matrix $Q = \sigma_Q I_n$

- Data assimilation parameters :
  - latent dimension $\ell \in \{10, 30, 100, 200\}$.
  - physical field (ψ) dimension $n = 1600$.
  - as many members as the latent dimension (*i.e.* $m = 10, 30, 100, 200$).
  - Observation error covariance matrix $R = \sigma_R I_n$ with $\sigma_R = 0.4$
  - 48 iterations/cycles (analogous to a 4-day forecast).
  - Model error covariance matrix $Q = \sigma_Q I_n$

- Data assimilation grid search :
  - *inflation* ranges from 0.8 to 2.0 (to control sampling noise in ETKF-Q)
  - σ_{Q_ℓ} ranges from 10^{-6} to 10 (model error in the latent space)

QG model : Numerical Results

- Data assimilation parameters :
  - latent dimension $\ell \in \{10, 30, 100, 200\}$.
  - physical field (ψ) dimension $n = 1600$.
  - as many members as the latent dimension (*i.e.* $m = 10, 30, 100, 200$).
  - Observation error covariance matrix $R = \sigma_R I_n$ with $\sigma_R = 0.4$
  - 48 iterations/cycles (analogous to a 4-day forecast).
  - Model error covariance matrix $Q = \sigma_Q I_n$
- Data assimilation grid search :
  - *inflation* ranges from 0.8 to 2.0 (to control sampling noise in ETKF-Q)
  - σ_{Q_ℓ} ranges from 10^{-6} to 10 (model error in the latent space)
- Deep learning parameters :
  - dataset made of 400 simulations (generated with OOPS).
  - optimizer : Adam
  - batch size : 16
  - the autoencoder and the surrogate model are Multilayer Perceptrons (MLP) networks.
  - encoding mapping $f_e : \left(\mathbb{R}^{1600} \to \mathbb{R}^{800}\right) \to LReLU(0.5) \to \left(\mathbb{R}^{800} \to \mathbb{R}^{\ell}\right) \to Id$
  - decoding mapping $f_d : \left(\mathbb{R}^{\ell} \to \mathbb{R}^{800}\right) \to LReLU(0.5) \to \left(\mathbb{R}^{800} \to \mathbb{R}^{1600}\right) \to Id$
  - the surrogate consists of four *tanh* activations followed by a *LReLU(0.5)* and an *Identity* function with skip connections.

♦ Reference ETKF-Q experiment with $\sigma_Q = 0.1$:

$m = 10$		$m = 30$		$m = 100$		$m = 200$		$m = 1600$	
RMSE	Inflation	RMSE	Inflation	RMSE	Inflation	RMSE	Inflation	RMSE	Inflation
0.482	1.0	**0.264**	1.3	**0.203**	1.05	**0.175**	0.9	**0.172**	**1.0**

Table – Numerical results (mean RMSE) for the reference ETKF-Q experiment with five different ensemble sizes and a $\sigma_Q = 0.1$ model error. Only inflation is tuned.

♦ Reference ETKF-Q experiment with $\sigma_Q = 0.1$:

$m = 10$		$m = 30$		$m = 100$		$m = 200$		$m = 1600$	
RMSE	Inflation	RMSE	Inflation	RMSE	Inflation	RMSE	Inflation	RMSE	Inflation
0.482	1.0	**0.264**	1.3	**0.203**	1.05	**0.175**	0.9	**0.172**	**1.0**

Table – Numerical results (mean RMSE) for the reference ETKF-Q experiment with five different ensemble sizes and a $\sigma_Q = 0.1$ model error. Only inflation is tuned.

♦ Latent ETKF-Q experiments with $\sigma_Q = 0$ for autoencoders and PCA :

Name	$\ell = 10$			$\ell = 30$			$\ell = 100$			$\ell = 200$		
	RMSE	Infl.	$\boldsymbol{\sigma}_{Q_\ell}$	RMSE	Infl.	$\boldsymbol{\sigma}_{Q_\ell}$	RMSE	Infl.	$\boldsymbol{\sigma}_{Q_\ell}$	RMSE	Infl.	$\boldsymbol{\sigma}_{Q_\ell}$
AE+Sur	**0.312**	0.8	0.1	**0.117**	0.8	0.01	**0.072**	0.8	0.01	**0.0697**	1.1	$1e$-3
PCA+Sur	**0.396**	1.05	10	**0.226**	0.8	1	**0.0972**	1.05	0.1	**0.0805**	1.1	0.01

Table – Numerical results (mean RMSE) for **latent** data assimilation algorithms. Two parameters are tuned : *inflation* and $\boldsymbol{\sigma}_{Q_\ell}$.

QG model : Numerical Results

♦ Reference ETKF-Q experiment with $\sigma_Q = 0.1$:

$m = 10$		$m = 30$		$m = 100$		$m = 200$		$m = 1600$	
RMSE	Inflation	RMSE	Inflation	RMSE	Inflation	RMSE	Inflation	RMSE	Inflation
0.482	1.0	**0.264**	1.3	**0.203**	1.05	**0.175**	0.9	**0.172**	**1.0**

Table – Numerical results (mean RMSE) for the reference ETKF-Q experiment with five different ensemble sizes and a $\sigma_Q = 0.1$ model error. Only inflation is tuned.

♦ Latent ETKF-Q experiments with $\sigma_Q = 0$ for autoencoders and PCA :

Name	$\ell = 10$			$\ell = 30$			$\ell = 100$			$\ell = 200$		
	RMSE	Infl.	$\boldsymbol{\sigma}_{Q_\ell}$	RMSE	Infl.	$\boldsymbol{\sigma}_{Q_\ell}$	RMSE	Infl.	$\boldsymbol{\sigma}_{Q_\ell}$	RMSE	Infl.	$\boldsymbol{\sigma}_{Q_\ell}$
AE+Sur	**0.312**	0.8	0.1	**0.117**	0.8	0.01	**0.072**	0.8	0.01	**0.0697**	1.1	1e-3
PCA+Sur	**0.396**	1.05	10	**0.226**	0.8	1	**0.0972**	1.05	0.1	**0.0805**	1.1	0.01

Table – Numerical results (mean RMSE) for **latent** data assimilation algorithms. Two parameters are tuned : *inflation* and $\boldsymbol{\sigma}_{Q_\ell}$.

- ➥ with both space reduction techniques (AE or PCA), we achieve better RMSE scores than our reference ETKF-Q algorithm with model error.
- ➥ the piece-wise activation functions of the autoencoder have a stronger compression capabality than PCA.

- We propose a new latent-space DA methodology :
  - explores the ability of deep learning to create a reduced space
  - defines a stable surrogate network within the latent space to perform model propagation
  - train NN for dimension reduction and surrogate model together
  - implements the ensemble DA within the learned reduced space.
- The latent space algorithm may improve the accuracy since the decoder is a nonlinear transformation that fits the manifold where the state trajectory statistically belong, when such a structure exists.
- The latent space algorithm may reduce the computational cost by performing the minimization in a reduced space
- The proposed proof of concept is encouraging and quite general that can be adapted to other DA methods.

Thank you for your attention !

Amendola, M., Arcucci, R., Mottet, L., Casas, C. Q., Fan, S., Pain, C. C., Linden, P., and Guo, Y. (2020).
Data assimilation in the latent space of a neural network.
CoRR, abs/2012.12056.

Casas, C. A. Q. (2018).
Fast ocean data assimilation and forecasting using a neural-network reduced-space regional ocean model of the north Brazil current.
PhD thesis, Imperial College London.

Cheng, S., Chen, J., Anastasiou, C., Angeli, P., Matar, O., Guo, Y.-K., Pain, C., and Arcucci, R.
Generalised latent assimilation in heterogeneous reduced spaces with machine learning surrogate models.

Fillion, A., Bocquet, M., Gratton, S., Gürol, S., and Sakov, P. (2020).
An iterative ensemble kalman smoother in presence of additive model error.

SIAM/ASA Journal on Uncertainty Quantification, 8(1) :198–228.

Gallivan, K., Grimme, E., and Van Dooren, P. (1994).
Asymptotic waveform evaluation via a lanczos method.
Applied Mathematics Letters, 7(5) :75–80.

Geelen, R., Wright, S., and Willcox, K. (2022).
Operator inference for non-intrusive model reduction with quadratic manifolds.

Goodfellow, I., Bengio, Y., and Courville, A. (2016).
Deep learning.
MIT press.

Holmes, P., Lumley, J., Berkooz, G., and Rowley, C. (2012).
Turbulence, coherent structures, dynamical systems and symmetry.
Turbulence, Coherent Structures, Dynamical Systems and Symmetry, by Philip Holmes , John L. Lumley , Gahl Berkooz ,

Clarence W. Rowley, Cambridge, UK : Cambridge University Press, 2012.

Lumley, J. (1967).
Atmospheric turbulence and wave propagation. the structure of inhomogeneous turbulence.

Mack, J., Arcucci, R., Molina-Solana, M., and Guo, Y.-K. (2020).
Attention-based convolutional autoencoders for 3d-variational data assimilation.
Computer Methods in Applied Mechanics and Engineering, 372 :113291.

Mohd Razak, S., Jahandideh, A., Djuraev, U., and Jafarpour, B. (2022).
Deep Learning for Latent Space Data Assimilation in Subsurface Flow Systems.
SPE Journal, 27(05) :2820–2840.

Moore, B. (1981).

Principal component analysis in linear systems : Controllability, observability, and model reduction.
IEEE Transactions on Automatic Control, 26(1) :17–32.

Peyron, M., Fillion, A., Gürol, S., Marchais, V., Gratton, S., Boudier, P., and Goret, G. (2021).
Latent space data assimilation by using deep learning.
Quarterly Journal of the Royal Meteorological Society, 147(740) :3759–3777.

Zhuang, Y., Cheng, S., Kovalchuk, N., Simmons, M., Matar, O., Guo, Y., and Arcucci, R. (2022).
Ensemble latent assimilation with deep learning surrogate model : application to drop interaction in microfluidics device.
Lab on a Chip.