# A parametric Kalman filter (PKF) tour of data assimilation practical and theoretical data assimilation

O. Pannekoucke[1,2,3], R. Menard[4], M. Bocquet[5], R. Fablet[6], S. Ricci[3], A. Perrot[1], O. Thual[3,7], M. Sabathier[8], V. Maget[8]

[1] CNRM UMR3589, CNRS, Météo-France, France. [2] INPT-ENM, France. [3] CERFACS, France. [4] Environment and Climate Change Canada, Canada. ARQI/Air Quality Research Division Environment and Climate Change Canada, Dorval (Québec), Canada. [5] CEREA, joint lab École des Ponts ParisTech and EdF R&D, Université Paris-Est, France. [6] IMT-Atlantic, UMR CNRS Lab-STICC, Brest, France. [7] Université de Toulouse, INPT, CNRS, IMFT, France. [8] ONERA, Toulouse, France.

Mathematical Approaches of Atmospheric Constituents Data Assimilation and Inverse Modeling | BIRS | 19-24 March 2023

Under linear assumptions [Kalman, 1960] filter details the dynamics of Gaussian uncertainty along the analysis and forecast cycles. Analysis update writes

$$\begin{cases} \mathbf{K} = \mathbf{P}^f \mathbf{H}^T (\mathbf{H}\mathbf{P}^f \mathbf{H}^T + \mathbf{R})^{-1}, \\ \mathcal{X}^a = \mathcal{X}^f + \mathbf{K}(\mathcal{Y}^o - \mathbf{H}\mathcal{X}^f), \\ \mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^f, \end{cases} \tag{1}$$

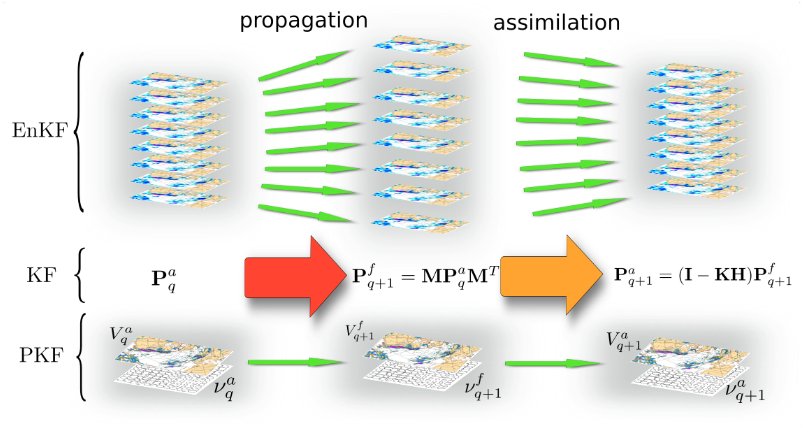
where $\mathbf{P}^f = \mathbb{E}\left[e^f {e^f}^T\right]$ and $\mathbf{P}^a = \mathbb{E}\left[e^a {e^a}^T\right]$, with the forecast evolution

$$\begin{cases} \mathcal{X}^f = \mathbf{M}\mathcal{X}^a, \\ \mathbf{P}^f = \mathbf{M}\mathbf{P}^a\mathbf{M}^T. \end{cases} \tag{2}$$

This is a simple algorithm. But update of forecast covariance matrix $\mathbf{P}^f = \mathbf{M}\mathbf{P}^a\mathbf{M}^T$ is numerically costly.

**KF needs approximations for practical implementation in large systems!**

# Parametric Kalman Filter




What are the PKF equations for the forecast and analysis steps ?

# Table of contents

# Table of contents

# VLAT covariance matrices

In this talk we consider covariance models parameterized by the variance and the local anistropy tensor fields – the VLATcov model [Pannekoucke, 2021]. For an error field $e(t, x)$,

- the variance is defined as $V(t, x) = \mathbb{E}\left[e^2\right]$

In this talk we consider covariance models parameterized by the variance and the local anistropy tensor fields – the VLATcov model [Pannekoucke, 2021]. For an error field $e(t, x)$,

- the variance is defined as $V(t, x) = \mathbb{E}\left[e^2\right]$
- the local anisotropy tensor is given either by the metric tensor, $\boldsymbol{g}(t, x)$, which measures the anisotropy of the correlation function

$$\rho(t, x, x+\delta x) = \frac{\mathbb{E}\left[e(t, x)e(t, x+\delta x)\right]}{\sqrt{V_x V_{x+\delta x}}} \underset{\delta x \to 0}{=} 1 - \frac{1}{2}||\delta x||^2_{\boldsymbol{g}_x} + \mathcal{O}(\delta x^2),$$

# VLAT covariance matrices

In this talk we consider covariance models parameterized by the variance and the local anistropy tensor fields – the VLATcov model [Pannekoucke, 2021]. For an error field $e(t,x)$,

- the variance is defined as $V(t,x) = \mathbb{E}\left[e^2\right]$
- the local anisotropy tensor is given either by the metric tensor, $\boldsymbol{g}(t,x)$, which measures the anisotropy of the correlation function

$$\rho(t,x,x+\delta x) = \frac{\mathbb{E}\left[e(t,x)e(t,x+\delta x)\right]}{\sqrt{V_x V_{x+\delta x}}} \underset{\delta x \to 0}{=} 1 - \frac{1}{2}||\delta x||^2_{\boldsymbol{g}_x} + \mathcal{O}(\delta x^2),$$

  or the aspect tensor [Purser et al., 2003], $\boldsymbol{s}(t,x)$, which is the matrix inverse of the metric tensor

$$\boldsymbol{s}_x = \boldsymbol{g}_x^{-1},$$

  and extends the correlation length-scale of [Daley, 1991].

# VLAT covariance matrices

In this talk we consider covariance models parameterized by the variance and the local anistropy tensor fields – the VLATcov model [Pannekoucke, 2021]. For an error field $e(t, x)$,

- the variance is defined as $V(t, x) = \mathbb{E}\left[e^2\right]$
- the local anisotropy tensor is given either by the metric tensor, $\boldsymbol{g}(t, x)$, which measures the anisotropy of the correlation function

$$\rho(t, x, x + \delta x) = \frac{\mathbb{E}\left[e(t, x)e(t, x + \delta x)\right]}{\sqrt{V_x V_{x+\delta x}}} \underset{\delta x \to 0}{=} 1 - \frac{1}{2}||\delta x||^2_{\boldsymbol{g}_x} + \mathcal{O}(\delta x^2),$$

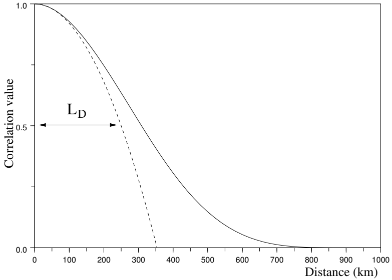
  or the aspect tensor [Purser et al., 2003], $\boldsymbol{s}(t, x)$, which is the matrix inverse of the metric tensor

$$\boldsymbol{s}_x = \boldsymbol{g}_x^{-1},$$

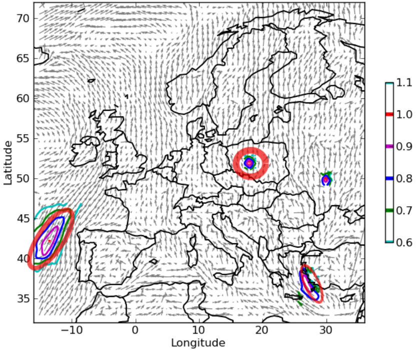
  and extends the correlation length-scale of [Daley, 1991].
  Note that $(\boldsymbol{g}_x)_{ij} = \mathbb{E}\left[\partial_i \left(\frac{e}{\sqrt{V}}\right) \partial_j \left(\frac{e}{\sqrt{V}}\right)\right] = \mathbb{E}\left[\partial_i \varepsilon \partial_j \varepsilon\right]$ where $\varepsilon = e/\sqrt{V}$ is the normalized error [Berre, 2000, Weaver and Mirouze, 2013].

$$\rho(\mathbf{x}, \mathbf{x}+\delta\mathbf{x}) = 1 - \frac{1}{2}||\delta\mathbf{x}||^2_{\boldsymbol{g}_\mathbf{x}} + \mathcal{O}(||\delta\mathbf{x}||^3) \equiv 1 - \frac{1}{2}||\delta\mathbf{x}||^2_{\boldsymbol{s}_\mathbf{x}^{-1}} + \mathcal{O}(||\delta\mathbf{x}||^3), \tag{3}$$

the local aspect tensor $\boldsymbol{s}_\mathbf{x}$ characterized the local anisotropy of the local correlation function at $\mathbf{x}$




Daley lenth-scale in 1D, $L_D = \sqrt{s}$




Mean flow and Anisotropy for few correlation functions [Jaumouillé et al., 2013]

# Table of contents

**Algorithm 1** Iterated process building the analysis state and its error covariance matrix for the first-order PKF (PKFO1) for VLATcov models where the local anisotropy is parametrized by the local metric tensors $\boldsymbol{g}$.

**Require:** Fields of $\boldsymbol{g}^f$ and $V^f$, $V^o$ and locations $\boldsymbol{x}_l$ of the $p$ observations to assimilate

**for** $l = 1 : p$ **do**

*0 - Initialization of the intermediate quantities*

$\mathcal{Y}^o{}_l = \mathcal{Y}^o(\boldsymbol{x}_l)$, $\mathcal{X}^f_l = \mathcal{X}^f(\boldsymbol{x}_l)$

$V^f_l = V^f_{\boldsymbol{x}_l}$, $V^o_l = V^o_{\boldsymbol{x}_l}$

*1- Set the correlation function from the VLATcov model*

$\rho_l(\boldsymbol{x}) = \rho(\boldsymbol{g}^f)(\boldsymbol{x}_l, \boldsymbol{x})$

*2 - Computation of the analysis state and its error statistics*

$\mathcal{X}^a_{\boldsymbol{x}} = \mathcal{X}^f_{\boldsymbol{x}} + \sigma^f_{\boldsymbol{x}}\rho_l(\boldsymbol{x})\frac{\sigma^f_l}{V^f_l+V^o_l}\left(\mathcal{Y}^o{}_l - \mathcal{X}^f_l\right),$

$V^a_{\boldsymbol{x}} = V^f_{\boldsymbol{x}}\left(1 - [\rho_l(\boldsymbol{x})]^2\frac{V^f_l}{V^f_l+V^o_l}\right)$

$\boldsymbol{g}^a_{\boldsymbol{x}} = \frac{V^f_{\boldsymbol{x}}}{V^a_{\boldsymbol{x}}}\boldsymbol{g}^f_{\boldsymbol{x}}$

*3 - Update of the forecast state and its error statistics*

$\mathcal{X}^f_{\boldsymbol{x}} \leftarrow \mathcal{X}^a_{\boldsymbol{x}}$

$V^f_{\boldsymbol{x}} \leftarrow V^a_{\boldsymbol{x}}$

$\boldsymbol{g}^f_{\boldsymbol{x}} \leftarrow \boldsymbol{g}^a_{\boldsymbol{x}}$

**end for**

Return fields $\mathcal{X}^a$, $\boldsymbol{g}^a$ and $V^a$

**Algorithm 1** Iterated process building the analysis state and its error covariance matrix for the first-order PKF (PKFO1) for VLATcov models where the local anisotropy is parametrized by the local metric tensors $\boldsymbol{g}$.

**Require:** Fields of $\boldsymbol{g}^f$ and $V^f$, $V^o$ and locations $\boldsymbol{x}_l$ of the $p$ observations to assimilate

**for** $l = 1 : p$ **do**

*0 - Initialization of the intermediate quantities*

$\mathcal{Y}^o{}_l = \mathcal{Y}^o(\boldsymbol{x}_l)$, $\mathcal{X}^f_l = \mathcal{X}^f(\boldsymbol{x}_l)$

$V^f_l = V^f_{\boldsymbol{x}_l}$, $V^o_l = V^o_{\boldsymbol{x}_l}$

*1- Set the correlation function from the VLATcov model*

$\rho_l(\boldsymbol{x}) = \rho(\boldsymbol{g}^f)(\boldsymbol{x}_l, \boldsymbol{x})$

*2 - Computation of the analysis state and its error statistics*

$$\mathcal{X}^a_{\boldsymbol{x}} = \mathcal{X}^f_{\boldsymbol{x}} + \sigma^f_{\boldsymbol{x}}\rho_l(\boldsymbol{x})\frac{\sigma^f_l}{V^f_l+V^o_l}\left(\mathcal{Y}^o{}_l - \mathcal{X}^f_l\right),$$

$$V^a_{\boldsymbol{x}} = V^f_{\boldsymbol{x}}\left(1 - [\rho_l(\boldsymbol{x})]^2\frac{V^f_l}{V^f_l+V^o_l}\right)$$

$$\boldsymbol{g}^a_{\boldsymbol{x}} = \frac{V^f_{\boldsymbol{x}}}{V^a_{\boldsymbol{x}}}\boldsymbol{g}^f_{\boldsymbol{x}}$$

*3 - Update of the forecast state and its error statistics*

$\mathcal{X}^f_{\boldsymbol{x}} \leftarrow \mathcal{X}^a_{\boldsymbol{x}}$

$V^f_{\boldsymbol{x}} \leftarrow V^a_{\boldsymbol{x}}$

$\boldsymbol{g}^f_{\boldsymbol{x}} \leftarrow \boldsymbol{g}^a_{\boldsymbol{x}}$

**end for**

Return fields $\mathcal{X}^a$, $\boldsymbol{g}^a$ and $V^a$

**Algorithm 2** Iterated process building the analysis state and its error covariance matrix for the second-order PKF (PKFO2) for VLATcov models where the local anisotropy is parametrized by the local metric tensors $\boldsymbol{g}$.

**Require:** Fields of $\boldsymbol{g}^f$ and $V^f$, $V^o$ and locations $\boldsymbol{x}_l$ of the $p$ observations to assimilate

**for** $l = 1 : p$ **do**

*0 - Initialization of the intermediate quantities*

$\mathcal{Y}^o{}_l = \mathcal{Y}^o(\boldsymbol{x}_l)$, $\mathcal{X}^f_l = \mathcal{X}^f(\boldsymbol{x}_l)$

$V^f_l = V^f_{\boldsymbol{x}_l}$, $V^o_l = V^o_{\boldsymbol{x}_l}$

*1- Set the correlation function from the VLATcov model*

$\rho_l(\boldsymbol{x}) = \rho(\boldsymbol{g}^f)(\boldsymbol{x}_l, \boldsymbol{x})$

*2- Computation of the analysis state and its error statistics*

$$\mathcal{X}^a_{\boldsymbol{x}} = \mathcal{X}^f_{\boldsymbol{x}} + \sigma^f_{\boldsymbol{x}}\rho_l(\boldsymbol{x})\frac{\sigma^f_l}{V^f_l+V^o_l}\left(\mathcal{Y}^o{}_l - \mathcal{X}^f_l\right),$$

$$V^a_{\boldsymbol{x}} = V^f_{\boldsymbol{x}}\left(1 - [\rho_l(\boldsymbol{x})]^2\frac{V^f_l}{V^f_l+V^o_l}\right)$$

$$g^a_{ij}(\boldsymbol{x}) = \frac{V^f_{\boldsymbol{x}}}{V^a_{\boldsymbol{x}}}g^f_{ij}(\boldsymbol{x}) + \frac{1}{4V^f_{\boldsymbol{x}}V^a_{\boldsymbol{x}}}\left(\partial_i V^f_{\boldsymbol{x}}\right)\left(\partial_j V^f_{\boldsymbol{x}}\right) - \frac{1}{V^a_{\boldsymbol{x}}}\partial_i(\rho_l(\boldsymbol{x})\sigma^f_{\boldsymbol{x}})\partial_j(\rho_l(\boldsymbol{x})\sigma^f_{\boldsymbol{x}})\frac{V^f_l}{V^f_l+V^o_l} - \frac{1}{4(V^a_{\boldsymbol{x}})^2}\left(\partial_i V^a_{\boldsymbol{x}}\right)\left(\partial_j V^a_{\boldsymbol{x}}\right)$$

*3 - Update of the forecast state and its error statistics*

$\mathcal{X}^f_{\boldsymbol{x}} \leftarrow \mathcal{X}^a_{\boldsymbol{x}}$

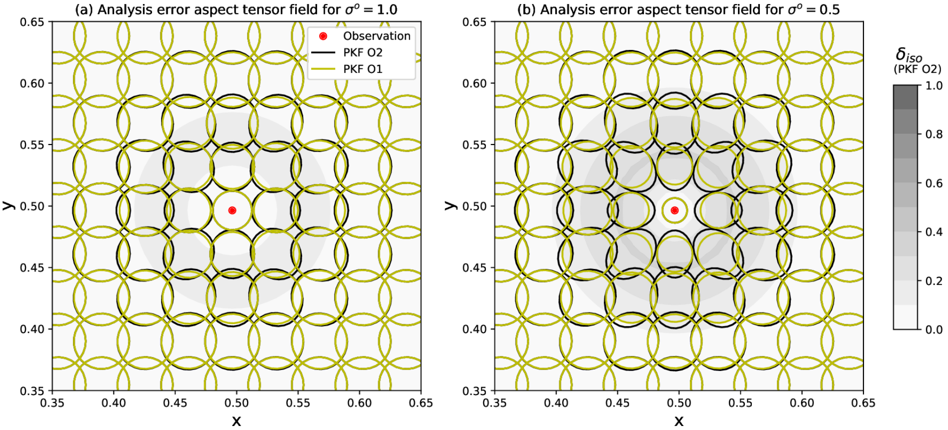
$V^f_{\boldsymbol{x}} \leftarrow V^a_{\boldsymbol{x}}$

$\boldsymbol{g}^f_{\boldsymbol{x}} \leftarrow \boldsymbol{g}^a_{\boldsymbol{x}}$

**end for**

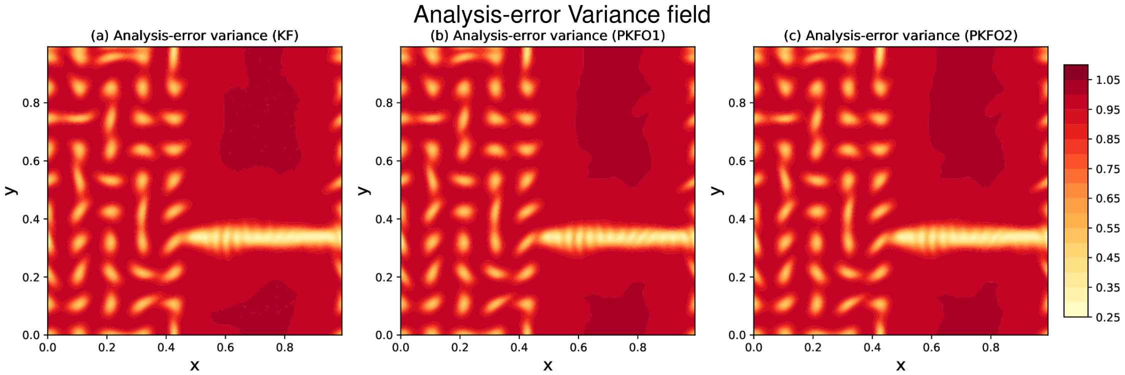
Return fields $\mathcal{X}^a$, $\boldsymbol{g}^a$ and $V^a$

# Ex. assimilation of a single obs. in a 2D domain



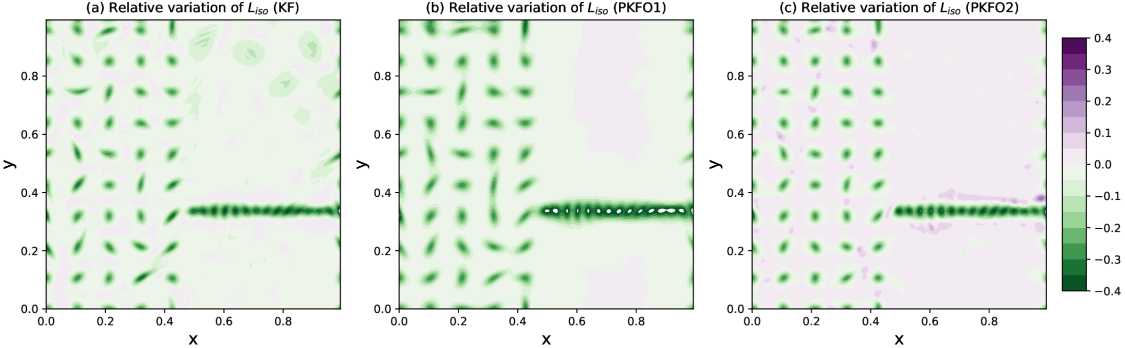

Here, the KF solution coincides with the PKFO2.

# Ex. assimilation of an obs. network in a 2D domain



# Ex. assimilation of an obs. network in a 2D domain

Relative variation of isotropic length scale, $r = \frac{L^a_{iso} - L^f_{iso}}{L^f_{iso}}$, where $L_{iso} = \sqrt{\mathrm{Tr}(\mathbf{s})/2}$



# Table of contents

$$\partial_t \chi = \mathcal{M}(\partial \chi), \tag{4}$$

the Reynolds decomposition $\mathcal{X}(t, \boldsymbol{x}, \omega) = \mathbb{E}\left[\mathcal{X}\right](t, \boldsymbol{x}) + e(t, \boldsymbol{x}, \omega)$ leads to

$$\text{PKF forecast step dynamics} \left\{ \begin{array}{l} \partial_t \mathbb{E}\left[\mathcal{X}\right] = \mathcal{M}(t, \partial \mathbb{E}\left[\mathcal{X}\right]) + \mathcal{M}''(t, \partial \mathbb{E}\left[\mathcal{X}\right])(\mathbb{E}\left[\partial \boldsymbol{e} \otimes \partial \boldsymbol{e}\right]), \\ \partial_t V = 2\mathbb{E}\left[e \partial_t e\right], \\ \partial_t \boldsymbol{g} = \partial_t \mathbb{E}\left[\partial_i \left(\frac{e}{\sqrt{V}}\right) \partial_j \left(\frac{e}{\sqrt{V}}\right)\right] \equiv \partial_t \mathbb{E}\left[\partial_i \varepsilon \partial_j \varepsilon\right], \end{array} \right. \tag{5}$$

[Pannekoucke et al., 2016, Pannekoucke et al., 2018, Pannekoucke and Arbogast, 2021], and extends the seminal work of [Cohn, 1993].

$$\partial_t \chi = \mathcal{M}(\partial \chi), \tag{4}$$

the Reynolds decomposition $\mathcal{X}(t, \boldsymbol{x}, \omega) = \mathbb{E}[\mathcal{X}](t, \boldsymbol{x}) + e(t, \boldsymbol{x}, \omega)$ leads to

PKF forecast step dynamics
$$\begin{cases} \partial_t \mathbb{E}[\mathcal{X}] = \mathcal{M}(t, \partial\mathbb{E}[\mathcal{X}]) + \mathcal{M}''(t, \partial\mathbb{E}[\mathcal{X}])(\mathbb{E}[\partial \boldsymbol{e} \otimes \partial \boldsymbol{e}]), \\ \partial_t V = 2\mathbb{E}[e\partial_t e], \\ \partial_t \boldsymbol{g} = \partial_t \mathbb{E}\left[\partial_i \left(\frac{e}{\sqrt{V}}\right) \partial_j \left(\frac{e}{\sqrt{V}}\right)\right] \equiv \partial_t \mathbb{E}[\partial_i \varepsilon \partial_j \varepsilon], \end{cases} \tag{5}$$

[Pannekoucke et al., 2016, Pannekoucke et al., 2018, Pannekoucke and Arbogast, 2021], and extends the seminal work of [Cohn, 1993].

The PKF dynamics can be computed by using a computer algebra system.

SymPKF performs the symbolic computation of the PKF for VLATcov model and can also automatically generate codes (finite difference) for the theoretical and numerical exploration [Pannekoucke and Arbogast, 2021].

see `https://github.com/opannekoucke/sympkf`

# Illustration: SymPKF on the Burgers' equation

```python
# Import of libraries
from sympy import symbols, Function, Derivative, Eq
from sympkf import PDESystem, SymbolicPKF, t

# Set the spatial coordinate system
x = symbols('x')
# Set the constants
kappa = symbols('kappa')
# Define the spatio-temporal scalar field
u = Function('u')(t,x)
```

```python
# Definition of the Burgers dynamics
burgers_equation = Eq(Derivative(u,t),
    -u*Derivative(u,x)+kappa*Derivative(u,x,2))
burgers_equation
```

$$\frac{\partial}{\partial t}u(t,x) = \kappa\frac{\partial^2}{\partial x^2}u(t,x) - u(t,x)\frac{\partial}{\partial x}u(t,x)$$

```python
# Processing of the PDE system
burgers = PDESystem( burgers_equation )
burgers
```

```
PDE System :
        prognostic functions : u(t, x)
        constant functions   :
        exogeneous functions :
        constants            : kappa
```

```python
# Define the PKF system
pkf_burgers = SymbolicPKF(burgers)
```

```python
# Compute the PKF system rndeered in metric tensor form (the computatation is only␣
↪performed at the first call)
for equation in pkf_burgers.in_metric: display(equation)
```

$$\frac{\partial}{\partial t}u(t,x) = \kappa\frac{\partial^2}{\partial x^2}u(t,x) - u(t,x)\frac{\partial}{\partial x}u(t,x) - \frac{\frac{\partial}{\partial x}\mathrm{V_u}(t,x)}{2}$$

$$\frac{\partial}{\partial t}\mathrm{V_u}(t,x) = -2\kappa\,\mathrm{V_u}(t,x)\,\mathrm{g_{u,xx}}(t,x) + \kappa\frac{\partial^2}{\partial x^2}\mathrm{V_u}(t,x) - \frac{\kappa\left(\frac{\partial}{\partial x}\mathrm{V_u}(t,x)\right)^2}{2\,\mathrm{V_u}(t,x)} - u(t,x)\frac{\partial}{\partial x}\mathrm{V_u}(t,x) - 2\,\mathrm{V_u}(t,x)\frac{\partial}{\partial x}u(t,x)$$

$$\frac{\partial}{\partial t}\mathrm{g_{u,xx}}(t,x) = 2\kappa\,\mathrm{g_{u,xx}}^2(t,x) - 2\kappa\mathbb{E}\left(\varepsilon_u(t,x,\omega)\frac{\partial^4}{\partial x^4}\varepsilon_u(t,x,\omega)\right) - 3\kappa\frac{\partial^2}{\partial x^2}\mathrm{g_{u,xx}}(t,x) + \frac{2\kappa\,\mathrm{g_{u,xx}}(t,x)\frac{\partial^2}{\partial x^2}\mathrm{V_u}(t,x)}{\mathrm{V_u}(t,x)} + \frac{\kappa\frac{\partial}{\partial x}\mathrm{V_u}(t,x)\frac{\partial}{\partial x}\mathrm{g_{u,xx}}(t,x)}{\mathrm{V_u}(t,x)} - \frac{2\kappa\,\mathrm{g_{u,xx}}(t,x)\left(\frac{\partial}{\partial x}\mathrm{V_u}(t,x)\right)^2}{\mathrm{V_u}^2(t,x)} - u(t,x)\frac{\partial}{\partial x}\mathrm{g_{u,xx}}(t,x) - 2\,\mathrm{g_{u,xx}}(t,x)\frac{\partial}{\partial x}u(t,x)$$

For $u \leftarrow \mathbb{E}[u]$,

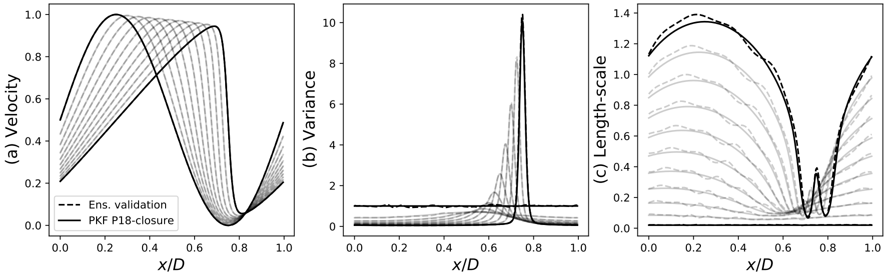
$$
\begin{aligned}
\frac{\partial}{\partial t} u &= \kappa \frac{\partial^2}{\partial x^2} u - u \frac{\partial}{\partial x} u - \frac{\frac{\partial}{\partial x} \mathrm{V_u}}{2} \\
\frac{\partial}{\partial t} \mathrm{V_u} &= -\frac{2\kappa \mathrm{V_u}}{\nu_{\mathrm{u,xx}}} + \kappa \frac{\partial^2}{\partial x^2} \mathrm{V_u} - \frac{\kappa \left(\frac{\partial}{\partial x} \mathrm{V_u}\right)^2}{2 \mathrm{V_u}} - u \frac{\partial}{\partial x} \mathrm{V_u} - 2 \mathrm{V_u} \frac{\partial}{\partial x} u \\
\frac{\partial}{\partial t} \mathrm{s_{u,xx}} &= 2\kappa {\mathrm{s_{u,xx}}}^2 \mathbb{E}\left(\varepsilon_\mathrm{u} \frac{\partial^4}{\partial x^4} \varepsilon_\mathrm{u}\right) - 3\kappa \frac{\partial^2}{\partial x^2} \mathrm{s_{u,xx}} \\
&\quad -2\kappa + \frac{6\kappa \left(\frac{\partial}{\partial x} \mathrm{s_{u,xx}}\right)^2}{\mathrm{s_{u,xx}}} - \frac{2\kappa \mathrm{s_{u,xx}} \frac{\partial^2}{\partial x^2} \mathrm{V_u}}{\mathrm{V_u}} + \frac{\kappa \frac{\partial}{\partial x} \mathrm{V_u} \frac{\partial}{\partial x} \mathrm{s_{u,xx}}}{\mathrm{V_u}} + \\
&\quad \frac{2\kappa \mathrm{s_{u,xx}} \left(\frac{\partial}{\partial x} \mathrm{V_u}\right)^2}{{\mathrm{V_u}}^2} - u \frac{\partial}{\partial x} \mathrm{s_{u,xx}} + 2 \mathrm{s_{u,xx}} \frac{\partial}{\partial x} u
\end{aligned}
$$

is a coupled system, where the term $\mathbb{E}\left(\varepsilon_\mathrm{u} \frac{\partial^4}{\partial x^4} \varepsilon_\mathrm{u}\right)$ is unclosed, and is due to the diffusion

# Example of Analytical Closure

[Pannekoucke et al., 2018] proposed the local Gaussian closure

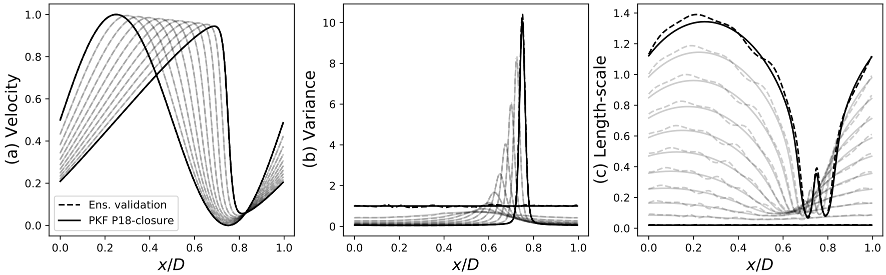
$$\mathbb{E}\left(\varepsilon_u \frac{\partial^4}{\partial x^4} \varepsilon_u\right) \sim 3g_u^2 - 2\partial_x^2 g_u = 2\frac{\partial_x^2 s_u}{s_u^2} + 3\frac{1}{s_u^2} - 4\frac{(\partial_x s_u)^2}{s_u^3}$$



# Example of Analytical Closure

[Pannekoucke et al., 2018] proposed the local Gaussian closure

$$\mathbb{E}\left(\varepsilon_u \frac{\partial^4}{\partial x^4} \varepsilon_u\right) \sim 3g_u^2 - 2\partial_x^2 g_u = 2\frac{\partial_x^2 s_u}{s_u^2} + 3\frac{1}{s_u^2} - 4\frac{(\partial_x s_u)^2}{s_u^3}$$




The design of analytical closure can be difficult, but can be done using IA: PDE-NetGen [Pannekoucke and Fablet, 2020]

see https://github.com/opannekoucke/pdenetgen

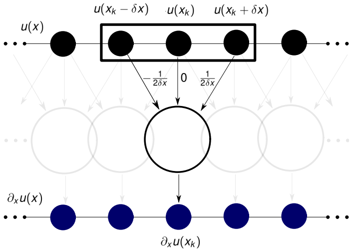
# Hybridation physics-IA: CNN as differential operators

For a function $u(x)$, a finite difference approximation of $\partial_x u$ on a regular grid is for instance

$$\partial_x u(x_k) \approx \frac{u(x_k+\delta x) - u(x_k-\delta x)}{2\delta x}$$

that can be computed as

$$\partial_x u = \sigma(au+b),$$



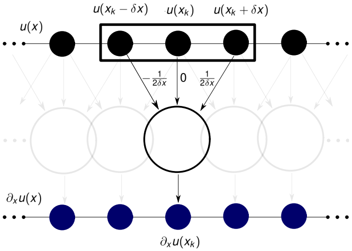

That is a convolutional neural network (CNN)

For a function $u(x)$, a finite difference approximation of $\partial_x u$ on a regular grid is for instance

$$\partial_x u(x_k) \approx \frac{u(x_k + \delta x) - u(x_k - \delta x)}{2\delta x}$$

that can be computed as

$$\partial_x u = \sigma(au + b),$$




That is a convolutional neural network (CNN)

PDE-NetGen implements a finite difference operator $\mathcal{F}$ such that for any multi-index $\alpha$,

$$\mathcal{F}^\alpha u(x) \approx \partial^\alpha u(x) + \mathcal{O}(|\delta x|^2)$$

For instance:

$$\mathcal{F}_x^3 u(x,y) = \partial_x^3 u(x,y) + \mathcal{O}(\delta x^2),$$

$$\mathcal{F}_{xy}^2 u(x,y) = \partial_{xy}^2 u(x,y) + \mathcal{O}(\delta x^2, \delta x \delta y, \delta y^2).$$

[Pannekoucke and Fablet, 2020] proposed to find a closure by the design of an automatic generation of neural network that translates PDE in NN. $\mathbb{E}\left(\varepsilon_u \frac{\partial^4}{\partial x^4}\varepsilon_u\right) \sim a_0 \frac{\partial_x^2 s_u}{s_u^2} + a_1 \frac{1}{s_u^2} + a_2 \frac{(\partial_x s_u)^2}{s_u^3}$

```
# Compute the PKF system rendered in aspect tensor form (the computatation is only
performed at the first call)
for equation in pkf_burgers.in_aspect: display(equation)
```

$$\frac{\partial}{\partial t}u(t,x) = \kappa\frac{\partial^2}{\partial x^2}u(t,x) - u(t,x)\frac{\partial}{\partial x}u(t,x) - \frac{\frac{\partial}{\partial x}\mathrm{V_u}(t,x)}{2}$$

$$\frac{\partial}{\partial t}\mathrm{V_u}(t,x) = -\frac{2\kappa\,\mathrm{V_u}(t,x)}{\mathrm{s_{u,xx}}(t,x)} + \kappa\frac{\partial^2}{\partial x^2}\mathrm{V_u}(t,x) - \frac{\kappa\left(\frac{\partial}{\partial x}\mathrm{V_u}(t,x)\right)^2}{2\,\mathrm{V_u}(t,x)} - u(t,x)\frac{\partial}{\partial x}\mathrm{V_u}(t,x) - 2\,\mathrm{V_u}(t,x)\frac{\partial}{\partial x}u(t,x)$$

$$\frac{\partial}{\partial t}\mathrm{s_{u,xx}}(t,x) = 2\kappa\,\mathrm{s_{u,xx}}^2(t,x)\mathbb{E}\left(\varepsilon_u(t,x,\omega)\frac{\partial^4}{\partial x^4}\varepsilon_u(t,x,\omega)\right) - 3\kappa\frac{\partial^2}{\partial x^2}\mathrm{s_{u,xx}}(t,x) - 2\kappa + \frac{6\kappa\left(\frac{\partial}{\partial x}\mathrm{s_{u,xx}}(t,x)\right)^2}{\mathrm{s_{u,xx}}(t,x)} - \frac{2\kappa\,\mathrm{s_{u,xx}}(t,x)\frac{\partial^2}{\partial x^2}\mathrm{V_u}(t,x)}{\mathrm{V_u}(t,x)} + \frac{\kappa\frac{\partial}{\partial x}\mathrm{V_u}(t,x)\frac{\partial}{\partial x}\mathrm{s_{u,xx}}(t,x)}{\mathrm{V_u}(t,x)} + \frac{2\kappa\,\mathrm{s_{u,xx}}(t,x)\left(\frac{\partial}{\partial x}\mathrm{V_u}(t,x)\right)^2}{\mathrm{V_u}^2(t,x)} - u(t,x)\frac{\partial}{\partial x}\mathrm{s_{u,xx}}(t,x) + 2\,\mathrm{s_{u,xx}}(t,x)\frac{\partial}{\partial x}u(t,x)$$

**Introduction of the closure ine the PKF dynamics**

```
from pdenetgen import TrainableScalar

# Set the closure by using TrainableScalar
a, b, c = [TrainableScalar(l) for l in 'abc']
closure_proposal = a*Derivative(nu,x,2)/nu**Integer(2)+b*1/nu**Integer(2)+\
          c*Derivative(nu,x)**2/nu**Integer(3)
display(closure_proposal)
```

$$\frac{a\frac{\partial^2}{\partial x^2}\nu_{\mathrm{u,xx}}(t,x)}{\nu_{\mathrm{u,xx}}^2(t,x)} + \frac{b}{\nu_{\mathrm{u,xx}}^2(t,x)} + \frac{c\left(\frac{\partial}{\partial x}\nu_{\mathrm{u,xx}}(t,x)\right)^2}{\nu_{\mathrm{u,xx}}^3(t,x)}$$

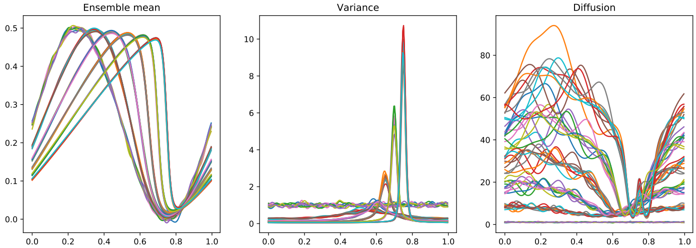
```
# Replace the closure(t,x) by the proposed closure
pkf_dynamics[2] = pkf_dynamics[2].subs(Function('closure')(t,x),closure_proposal)

# Generate the NN code leading to the ClosedPKFBurgers class.
exec(NNModelBuilder(pkf_dynamics,'ClosedPKFBurgers').code)
```

**Sample of code generated to define the ClosedPKFBurgers class**

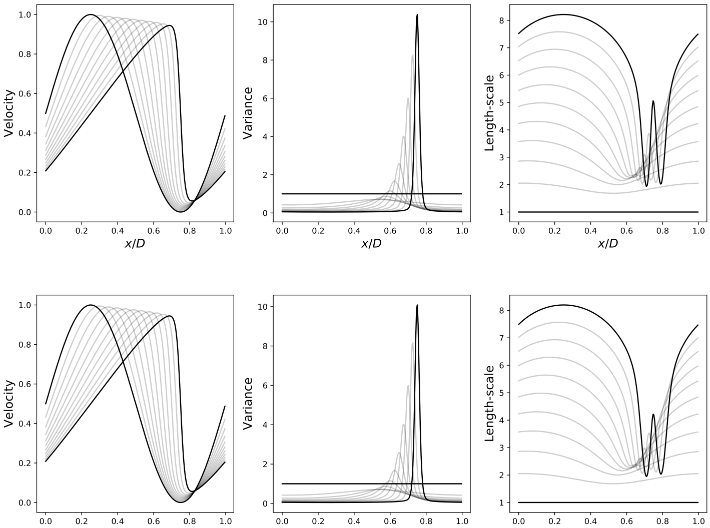
```
[..]
pow_21 = keras.layers.multiply([div_17,div_17,] ,name='PowLayer_21')
mul_28 = keras.layers.multiply([pow_21,Dnu_u_xx_x_o2],name='MulLayer_28')
train_scalar_9 = TrainableScalarLayerFactory(input_shape=mul_28.shape, name='TrainableScalar_a'
            init_value=0,use_bias=False,mean=0.0,stddev=1.0,seed=None,wl2=None)(mul_28)
            #TrainableScalar name: 'a'
add_8 = keras.layers.add([train_scalar_7,train_scalar_8,train_scalar_9],name='AddLayer_8')
mul_26 = keras.layers.multiply([pow_17,add_8],name='MulLayer_26')
[..]
```

Compute numerous ensemble forecasting (here 400)



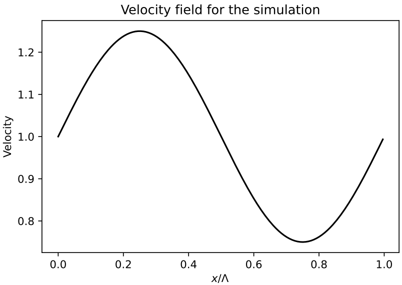
# Machine learning estimation of $a_0$,$a_1$ and $a_2$

$a_0 = 1.864$ , $a_1 = 3.004$, $a_2 = -3.604$ Trained-NN (top) vs. Proposed closure (bottom) ($a_0 = 2$, $a_1 = 3$, $a_2 = -4$)



# Table of contents

$$\partial_t c + u\partial_x c = 0. \tag{6}$$

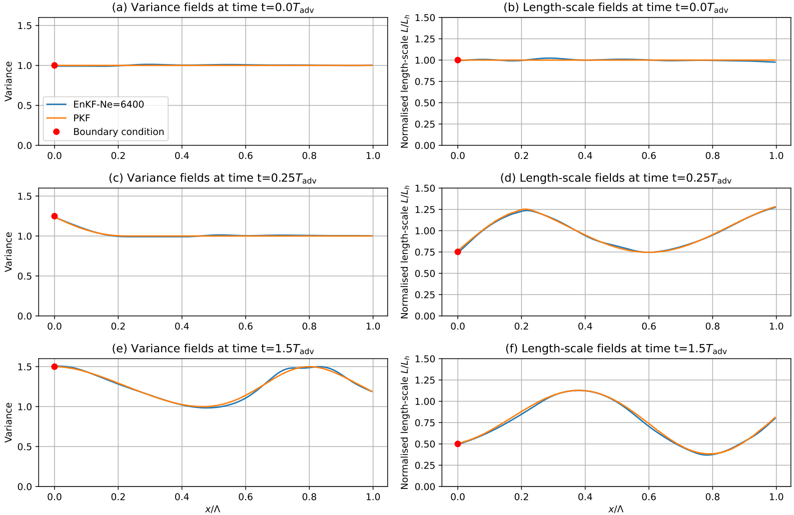



the PKF dynamics reads as (alternative to $\mathrm{P}^f = \mathrm{M}\mathrm{P}^a\mathrm{M}^\mathrm{T}$ for VLATcov.)

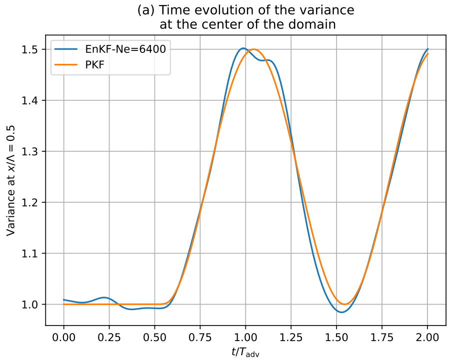
$$\partial_t c = -u\partial_x c, \tag{7a}$$
$$\partial_t V_c = -u\partial_x V_c, \tag{7b}$$
$$\partial_t s_{c,xx} = -u\partial_x s_{c,xx} + 2s_{c,xx}\partial_x u, \tag{7c}$$

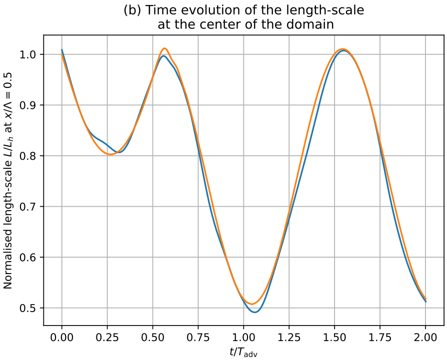
# Advection & PKF : Dirichlet at $x = 0$, open channel in $\Lambda$

## PKF validated by ensemble estimation



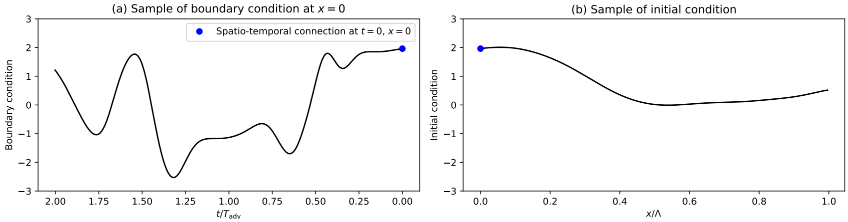
PKF validated by ensemble estimation



# Advection & PKF : Dirichlet at $x = 0$, open channel in $\Lambda$

Ensemble of forecast generated for the ensemble validation of the PKF.



# Advection & PKF: details of the EnKF setting

For a smooth random error (in time) $\eta(t)$, the error variance is defined as

$$V_\eta(t) = \mathbb{E}\left[\eta(t)^2\right],$$

and the time auto-correlation is characterized from

$$\mathbf{g}_{tt}(t) = \mathbb{E}\left[\partial_t\left(\frac{\eta(t)}{V_\eta(t)}\right)\partial_t\left(\frac{\eta(t)}{V_\eta(t)}\right)\right]. \tag{8}$$

If the error at $x = 0$ stands as $e(t, x = 0) = \eta(t)$, then $V_\eta(t) = V(t, x = 0)$, and the temporal metric tensor reads as

$$\mathbf{g}_{tt,x=0}(t) = \mathbb{E}\left[\partial_t\varepsilon(t, \mathbf{x} = 0)\partial_t\varepsilon(t, \mathbf{x} = 0)\right], \tag{9}$$

where $\varepsilon = e/\sqrt{V}$ is the normalized error associated with the spatial error $e$.

For the advection where $\partial_t e_c = -u\partial_x e_c$, then

$$g_{c,tt} \underset{x=0}{=} u^2 g_{c,xx} + \frac{u^2\left(\partial_x V_c\right)^2}{4V_c^2} + \frac{u\partial_t V_c \partial_x V_c}{2V_c^2} + \frac{\left(\partial_t V_c\right)^2}{4V_c^2},$$

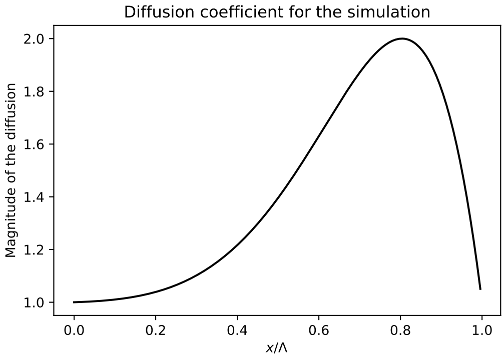
or

$$g_{c,tt} \underset{x=0}{=} u^2 g_{c,xx},$$

under local homogeneous and stationary assumptions.

$$\partial_t f = \partial_x(D\partial_x f). \tag{8}$$

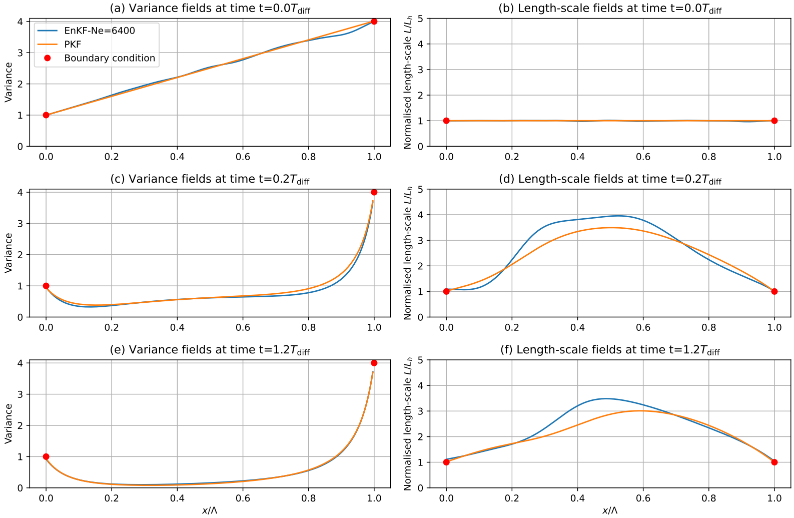
here $f$ stands for e.g. the density of a plasma (Fokker-Planck Eq.)



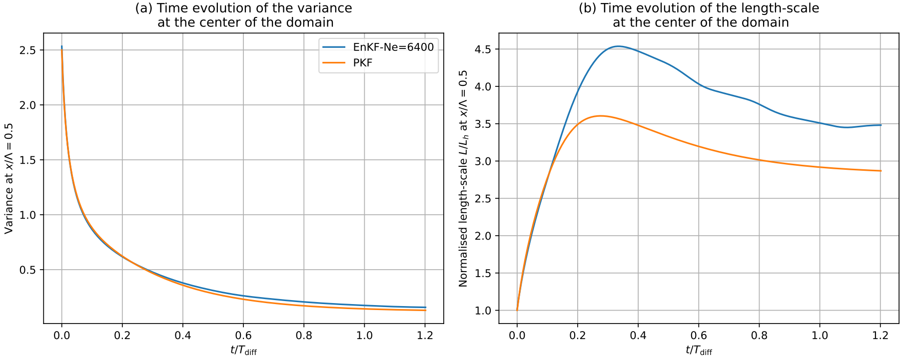

Diff. coef. similar to those encountered in radiation belt simulations.

# Heterogeneous Diff. Eq. & Dirichlet BC & PKF

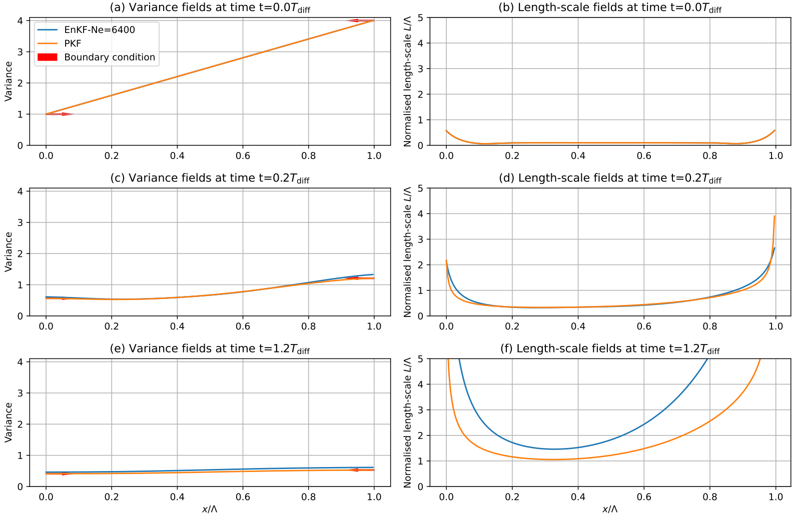
## PKF validated by ensemble estimation (EnKF: $g_{f,tt}(t,x) \approx 3D(x)^2 g_{f,xx}(t,x)$)



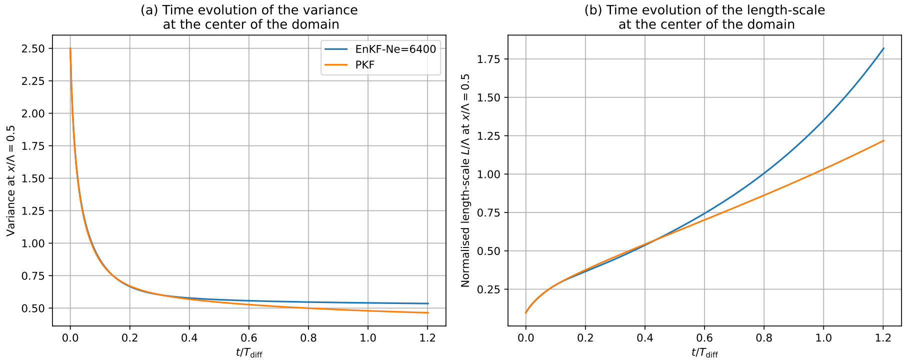
PKF validated by ensemble estimation (EnKF: $g_{f,tt}(t,x) \approx 3D(x)^2 g_{f,xx}(t,x)$)



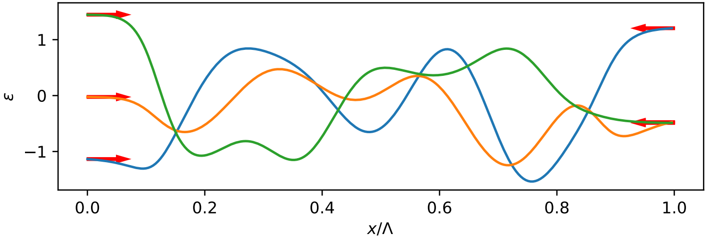
## PKF validated by ensemble estimation



## PKF validated by ensemble estimation



Samples for the ensemble validation



# Table of contents

# Assimilation cycles applied to transport of a passive scalar

For the linear transport

$$\partial_t c + \mathbf{u}\nabla c = 0, \tag{9}$$

SymPKF gives the PKF dynamics: (with $c \leftarrow \mathbb{E}[c]$)

$$\begin{aligned}
\frac{\partial}{\partial t} c &= -u\frac{\partial}{\partial x} c - v\frac{\partial}{\partial y} c \\
\frac{\partial}{\partial t} \mathrm{V_c} &= -u\frac{\partial}{\partial x} \mathrm{V_c} - v\frac{\partial}{\partial y} \mathrm{V_c} \\
\frac{\partial}{\partial t} \mathrm{s_{c,xx}} &= -u\frac{\partial}{\partial x} \mathrm{s_{c,xx}} - v\frac{\partial}{\partial y} \mathrm{s_{c,xx}} + 2\,\mathrm{s_{c,xx}} \frac{\partial}{\partial x} u + 2\,\mathrm{s_{c,xy}} \frac{\partial}{\partial y} u \\
\frac{\partial}{\partial t} \mathrm{s_{c,xy}} &= -u\frac{\partial}{\partial x} \mathrm{s_{c,xy}} - v\frac{\partial}{\partial y} \mathrm{s_{c,xy}} + \mathrm{s_{c,xx}} \frac{\partial}{\partial x} v + \\
& \quad \mathrm{s_{c,xy}} \frac{\partial}{\partial x} u + \mathrm{s_{c,xy}} \frac{\partial}{\partial y} v + \mathrm{s_{c,yy}} \frac{\partial}{\partial y} u \\
\frac{\partial}{\partial t} \mathrm{s_{c,yy}} &= -u\frac{\partial}{\partial x} \mathrm{s_{c,yy}} - v\frac{\partial}{\partial y} \mathrm{s_{c,yy}} + 2\,\mathrm{s_{c,xy}} \frac{\partial}{\partial x} v + 2\,\mathrm{s_{c,yy}} \frac{\partial}{\partial y} v
\end{aligned}$$

Assimilation cycles starting from an isotropic forecast-error covariance at t=0.

**PKF forecast steps** are computed with

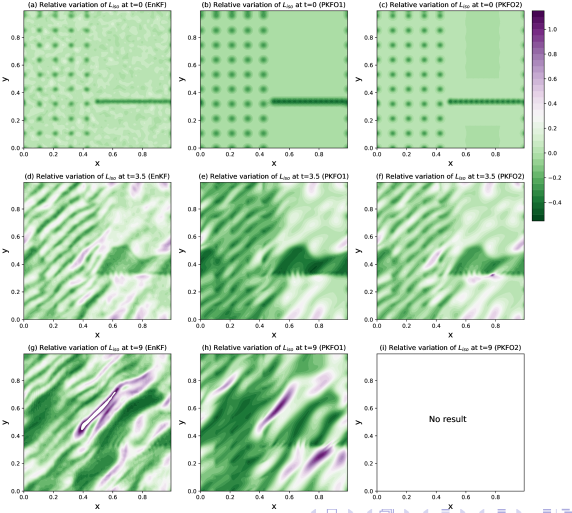
$$
\begin{aligned}
\partial_t c + \mathbf{u}\nabla c &= 0,\\
\partial_t V_c + \mathbf{u}\nabla V_c &= 0,\\
\partial_t \mathbf{s}_c + \mathbf{u}\nabla \mathbf{s}_c &= (\nabla \mathbf{u})\,\mathbf{s}_c + \mathbf{s}_c\,(\nabla \mathbf{u})^T + \eta\nabla^2 \mathbf{s}_c.
\end{aligned}
$$

**PKF analysis steps** are performed using Algo 1 (PKF01) & 2 (PKO2).

Validation of the PKF based on EnKF using 1000 members.

see [Pannekoucke, 2021], see also GOSAT assim in Sina's work [Voshtani et al., 2022a, Voshtani et al., 2022b]

Relative variation of isotropic length scale, $r = \frac{L^a_{iso} - L_h}{L_h}$



Assimilation cycles starting from an isotropic forecast-error covariance at t=0.

**PKF forecast steps** are computed with

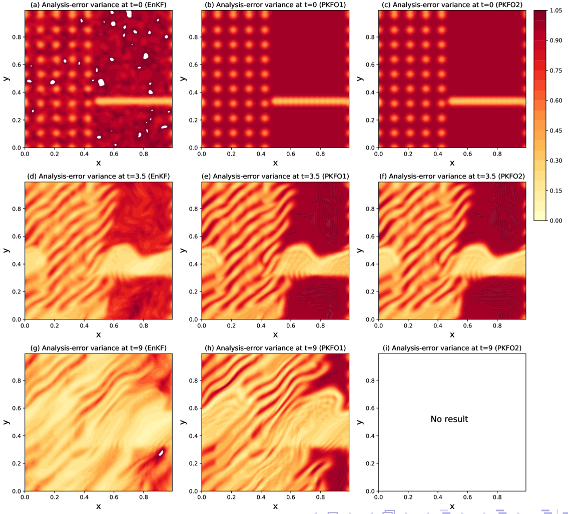
$$\begin{aligned}\partial_t c + \mathbf{u}\nabla c &= 0,\\ \partial_t V_c + \mathbf{u}\nabla V_c &= 0,\\ \partial_t \mathbf{s}_c + \mathbf{u}\nabla \mathbf{s}_c &= (\nabla\mathbf{u})\,\mathbf{s}_c + \mathbf{s}_c\,(\nabla\mathbf{u})^T + \eta\nabla^2\mathbf{s}_c.\end{aligned}$$

**PKF analysis steps** are performed using Algo 1 (PKF01) & 2 (PKO2).

Validation of the PKF based on EnKF using 1000 members.

see [Pannekoucke, 2021], see also GOSAT assim in Sina's work [Voshtani et al., 2022a, Voshtani et al., 2022b]

Analysis-error variance fields



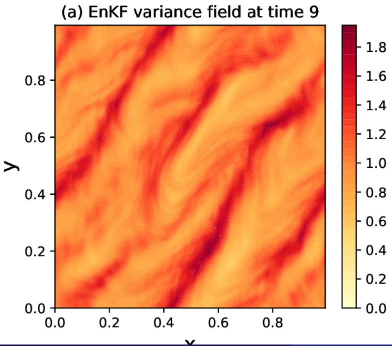
# Table of contents

But for the EnKF

$$\partial_t V_c + \mathbf{u}\nabla V_c \neq 0$$

because discretization leads to solve

$$\partial_t c + \mathbf{u}\nabla c = -\frac{\delta x^2 u}{6}\partial_x^3 c - \frac{\delta y^2 v}{6}\partial_y^3 c,$$



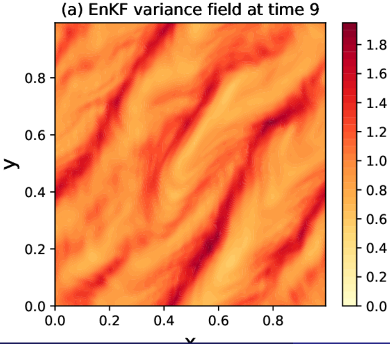
# PKF and model-error covariance

But for the EnKF

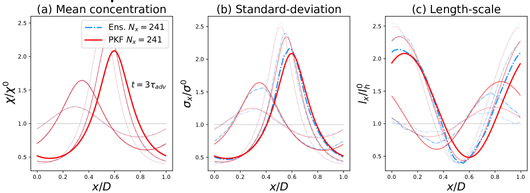
$$\partial_t V_c + \mathbf{u}\nabla V_c \neq 0$$

because discretization leads to solve

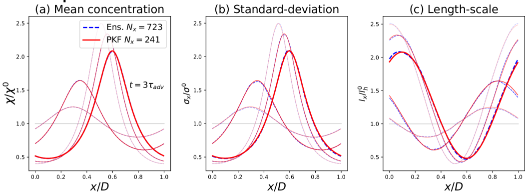
$$\partial_t c + \mathbf{u}\nabla c = -\frac{\delta x^2 u}{6}\partial_x^3 c - \frac{\delta y^2 v}{6}\partial_y^3 c,$$




Comp. in 1D at $n = 241$ PKF vs. EnKF



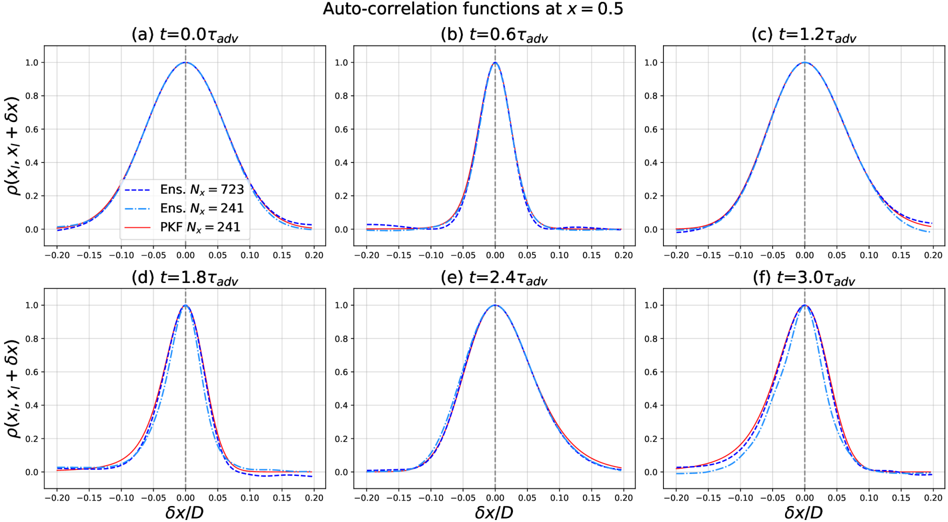

Comp. in 1D at $n = 241$ PKF vs. $n = 723$ EnKF




see [Perrot et al., 2023]

# PKF and model-error covariance

Some correlation functions in 1D exp. for transport (2nd order spatial derivative)




see [Perrot et al., 2023]

When solving the adection equation

$$\partial_t c + \mathbf{u}\partial_x c = 0, \tag{10}$$

where $\mathbf{u}(t,x) > 0$ is an heterogeneous wind field and $c(t,x)$ a passive scalar field. The modified equation associated with the Euler-upwind scheme

$$\frac{c_i^{q+1} - c_i^q}{\delta t} = -u_i \frac{c_i^q - c_{i-1}^q}{\delta x}, \tag{11}$$

reads as

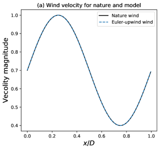
$$\partial_t C + U\partial_x C = \kappa \partial_x^2 C, \tag{12}$$

where

$$\begin{cases} U(t,x) = u - \frac{\delta t}{2}\partial_t u + \frac{\delta t}{2} u\partial_x u, \\ \kappa(t,x) = \frac{u}{2}\left(\delta x - u\delta t\right). \end{cases} \tag{13}$$

which shows that the num. model is suffering from dispersion and dissipation.

Note that similar expressions are obtained for semi-Lagrangian discretization as used in NWP and air quality.

$\mathbf{u}(x) \rightarrow$ 

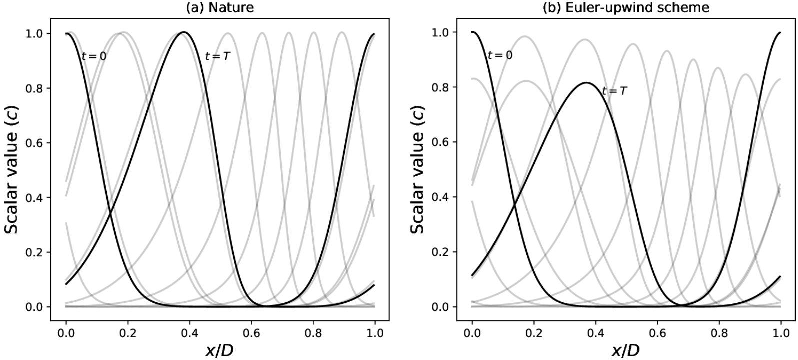




Figure: Nature versus numerical dynamics

Transport with conservation for the nature
but heterogeneous damping for the num. model == model error.

# Predictability-error covariance dynamics: the model

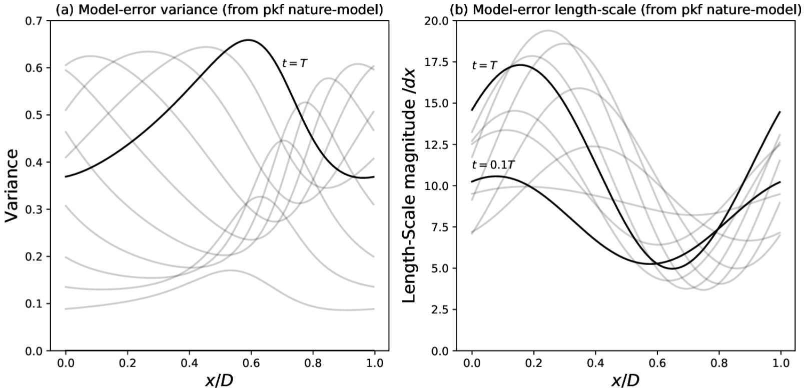
With the local Gaussian closure (op. cit.) the predictability-error covariance dynamics for

$$\partial_t C + U(t,x)\partial_x C = \kappa(t,x)\partial_x^2 C, \tag{14}$$

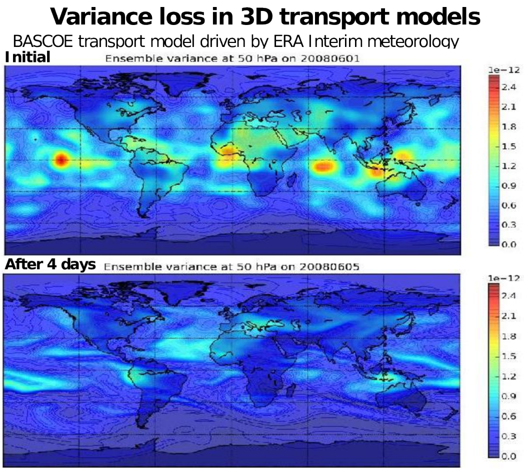
reads as

$$\begin{aligned}
\partial_t C &= -U\partial_x C + \kappa\partial_x^2 C,\\
\partial_t V^p &= U\partial_x V^p - \frac{2V^p\kappa}{s^p} + \kappa\partial_x^2 V^p - \frac{\kappa\left(\partial_x V^p\right)^2}{2V^p}\\
\partial_t s^p &= -U\partial_x s^p + (2\partial_x U)s^p +\\
&\quad \kappa\partial_x^2 s^p + 4\kappa - \frac{2\left(\partial_x s^p\right)^2}{s^p}\kappa + \partial_x\kappa\partial_x s^p - \frac{2\partial_x^2 V^p}{V^p}\kappa s^p +\\
&\quad \frac{\partial_x V^p}{V}\kappa\partial_x s^p - \frac{2\partial_x V^p}{V^p}s^p\partial_x\kappa + \frac{2\left(\partial_x V^p\right)^2}{V^{p2}}\kappa s^p,
\end{aligned}$$

# Time evolution of the low-dependent part of $\mathbf{P}^m$




Evolution of the flow-dependent part of the model-error covariance [Pannekoucke et al., 2021]




see [Ménard et al., 2021]

the high-order time scheme version of the modified equation that predict variance time evolution

$$\partial_t V^p + u\partial_x V^p = U\partial_x V^p - \frac{2V^p\kappa}{(L^p)^2} + \kappa\partial_x^2 V^p - \frac{\kappa\left(\partial_x V^p\right)^2}{2V^p}$$

$$\begin{cases} U(t,x) = -\frac{\Delta t}{2}\partial_t u + \frac{\Delta t}{2} u\partial_x u, \\ \kappa(t,x) = \frac{u}{2}\left(\Delta x - u\Delta t\right). \end{cases} \tag{15}$$

reads as, when corrected to force transpart of variance

$$\partial_t V^p + u\partial_x V^p = I - \frac{2V^p\kappa}{(L^p)^2} + \kappa\partial_x^2 V^p - \frac{\kappa\left(\partial_x V^p\right)^2}{2V^p}$$

with this time $\kappa = \frac{u\Delta x}{2}$. See [Ménard et al., 2021] who proposed a flow dependent inflation for the EnKF $I$ to ensure the true transport of $V^p$. Connexion with Shay's presentation of monday.

# Table of contents

# Simple multivariate chemical transport model: LV-1D

Lotka-Voltera interaction for species $A$ and $B$

$$X + A \xrightarrow{k_1} 2A,$$
$$A + B \xrightarrow{k_2} 2B$$
$$B \xrightarrow{k_3} Y.$$

leads to the dynamics in 1D domain

$$\begin{cases} \partial_t A + u\partial_x A = -A\partial_x u + k_1 A - k_2 AB \\ \partial_t B + u\partial_x B = -B\partial_x u + k_2 AB - k_3 B \end{cases}$$

This offers a minimal framework to explore multivariate assimilation in chemical transport model (CTM)

- ☒ Multivariate (2 species)
- ☒ Non-linear dynamics (as often the case CTM)
- ☒ Continuous fields so to take advantage of the PKF

# Multivariate PKF dynamics for LV in 1D domain

$$\partial_t A + u\partial_x A = -A\partial_x u + k_1 A - k_2 AB - k_2 V_{AB} \tag{16a}$$

$$\partial_t B + u\partial_x B = -B\partial_x u - k_3 B + k_2 AB + k_2 V_{AB} \tag{16b}$$

$$\partial_t V_{AB} + u\partial_x V_{AB} = -2V_{AB}\partial_x u + V_{AB}(k_1 - k_2 B - k_3 + k_2 A) + k_2 V_A B - k_2 V_B A \tag{16c}$$

$$\partial_t V_A + u\partial_x V_A = -2V_A\partial_x u + 2[V_A(k_1 - k_2 B) - k_2 A V_{AB}] \tag{16d}$$

$$\partial_t V_B + u\partial_x V_B = -2V_B\partial_x u + 2[V_B(-k_3 + k_2 A) + k_2 B V_{AB}] \tag{16e}$$

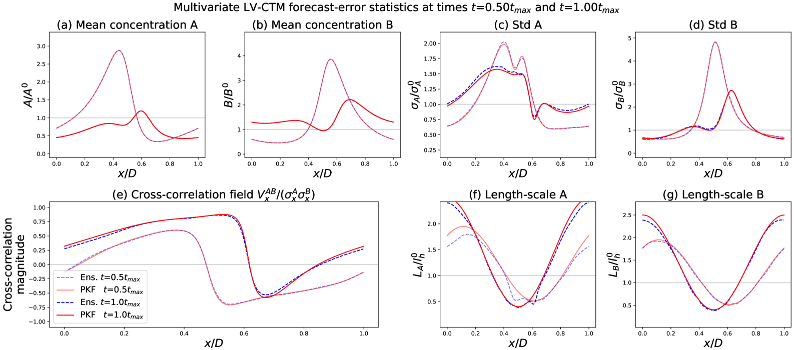
$$\partial_t s_A + \underbrace{u\partial_x s_A}_{T_{A,adv-1}} = \underbrace{2s_A\partial_x u}_{T_{A,adv-2}} - \underbrace{\frac{2k_2 A V_{AB} s_A}{V_A}}_{T_{A,chem-1}} + \underbrace{\frac{2k_2 A\sigma_B s_A^2 \overline{\partial_x\tilde{\varepsilon}_A\partial_x\tilde{\varepsilon}_B}}{\sigma_A}}_{T_{A,chem-2}} .. \tag{16f}$$

$$\partial_t s_B + \underbrace{u\partial_x s_B}_{T_{B,adv-1}} = \ldots \tag{16g}$$

with cross-correlation approx.

$$r_{AB}(\mathbf{x},\mathbf{y}) = \frac{1}{2}\left(\frac{V_{AB}(\mathbf{x})}{\sigma_A(\mathbf{x})\sigma_B(\mathbf{x})} + \frac{V_{AB}(\mathbf{y})}{\sigma_A(\mathbf{y})\sigma_B(\mathbf{y})}\right)\exp\left(-||\mathbf{x}-\mathbf{y}||^2_{[\frac{1}{4}(s_A(\mathbf{x})+s_B(\mathbf{x})+s_A(\mathbf{y})+s_B(\mathbf{y}))]^{-1}}\right), \tag{17}$$

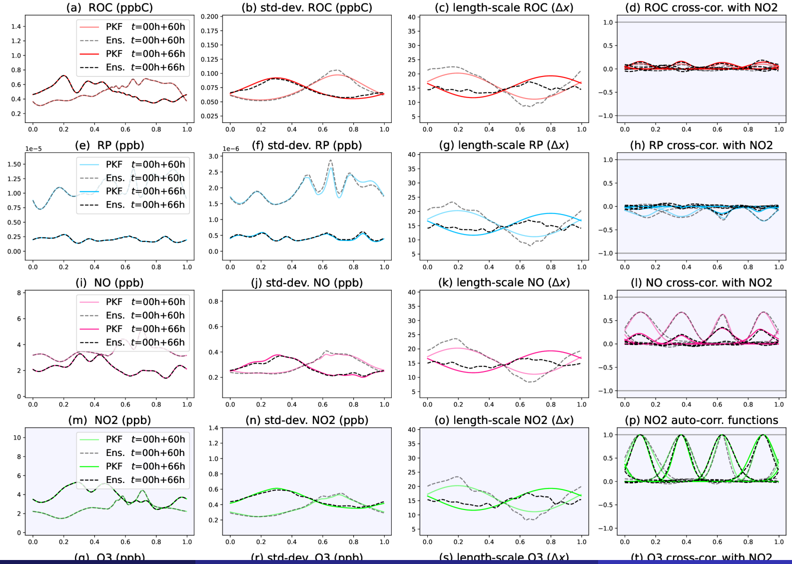
# Multivariate PKF dynamics for LV in 1D domain




[Perrot et al., 2023]

# Multivariate PKF dynamics for GRS (6 chem. species) in 1D domain

Multivariate forecast statistics for GRS: Ens. estimation ($N_e$=1600, black dashed lines) and PKF (colored lines)



# Table of contents

# Conclusion & Perspectives

- In the PKF error-covariance matrices are approximated by some covariance model
- The Assimilation cycle described for univariate assimilation
- The PKF is a pratical tool that approximates the KF (or its non-linear second-order extension)
- The dynamics of the parameters approximates the real error-covariance matrix.
- Symbolic tools have been designed to facilitate the computation of the PKF dynamics (SymPKF)
- PKF often needs a closures
- IA tools have been designed to replaced unkown terms by NN parameterizations or to discover analytical closures (PDE-NetGen)
- The PKF dynamics gives access to the physics of uncertainty, and appears as a theoretical tool
- Which has been explored for understanding the model-error covariance due to the discretization of PDEs
- Multivariate PKF assimilation – some preliminary results for air quality !

# Conclusion & Perspectives

- In the PKF error-covariance matrices are approximated by some covariance model
- The Assimilation cycle described for univariate assimilation
- The PKF is a pratical tool that approximates the KF (or its non-linear second-order extension)
- The dynamics of the parameters approximates the real error-covariance matrix.
- Symbolic tools have been designed to facilitate the computation of the PKF dynamics (SymPKF)
- PKF often needs a closures
- IA tools have been designed to replaced unkown terms by NN parameterizations or to discover analytical closures (PDE-NetGen)
- The PKF dynamics gives access to the physics of uncertainty, and appears as a theoretical tool
- Which has been explored for understanding the model-error covariance due to the discretization of PDEs
- Multivariate PKF assimilation – some preliminary results for air quality !

Perspectives

- Accounting for 2D/3D bounded domains (– interesting results for EnKF ?)
- Accounting for the meteorology / parameter uncertainty in the PKF dynamics
- Multivariate extension application to geophysical dynamics (SW eq.)
- Application in targeting and sensivity analysis

Berre, L. (2000).
Estimation of synoptic and mesoscale forecast error covariances in a limited-area model.
Monthly Weather Review, 128:644–667.

Cohn, S. (1993).
Dynamics of short-term univariate forecast error covariances.
Monthly Weather Review, 121(11):3123–3149.

Daley, R. (1991).
Atmospheric Data Analysis.
Cambridge University Press, New York, USA.

Jaumouillé, E., Emili, E., Pannekoucke, O., Massart, M., and Piacentini, A. (2013).
Modelisation dynamique de la matrice des covariances d'erreur d'ebauche avec valentina-ensemble.
ACHILLE Newsletter, 11:4–8.

Kalman, R. E. (1960).
A new approach to linear filtering and prediction problems.
Journal Basic Engineering, 82:35–45.

Ménard, R., Skachko, S., and Pannekoucke, O. (2021).
Numerical discretization causing error variance loss and the need for inflation.
Quarterly Journal of the Royal Meteorological Society.

Pannekoucke, O. (2021).
An anisotropic formulation of the parametric kalman filter assimilation.
Tellus A: Dynamic Meteorology and Oceanography, 73(1):1–27.

Pannekoucke, O. and Arbogast, P. (2021).
SymPKF (v1.0): a symbolic and computational toolbox for the design of parametric kalman filter dynamics.
Geoscientific Model Development, 14(10):5957–5976.

Pannekoucke, O., Bocquet, M., and Ménard, R. (2018).
Parametric covariance dynamics for the nonlinear diffusive burgers' equation.
Nonlinear Processes in Geophysics, 2018:1–21.

Pannekoucke, O. and Fablet, R. (2020).
PDE-NetGen 1.0: from symbolic partial differential equation (PDE) representations of physical processes to trainable neural network representations.
Geoscientific Model Development, 13(7):3373–3382.

Pannekoucke, O., Ménard, R., El Aabaribaoune, M., and Plu, M. (2021).
A methodology to obtain model-error covariances due to the discretization scheme from the parametric kalman filter perspective.
Nonlinear Processes in Geophysics, 28(1):1–22.

Pannekoucke, O., Ricci, S., Barthelemy, S., Ménard, R., and Thual, O. (2016).
Parametric kalman filter for chemical transport model.
Tellus, 68:31547.

Perrot, A., Pannekoucke, O., and Guidard, V. (2023).
Toward a multivariate formulation of the pkf dynamics : application to a simplified chemical transport model.
Nonlinear Processes in Geophysics.

Purser, R., Wu, W.-S., D.Parrish, and Roberts, N. (2003).
Numerical aspects of the application of recursive filters to variational statistical analysis. part ii: Spatially inhomogeneous and anisotropic general covariances.
Monthly Weather Review, 131:1536–1548.

Voshtani, S., Ménard, R., Walker, T. W., and Hakami, A. (2022a).
Assimilation of GOSAT methane in the hemispheric CMAQ; part i: Design of the assimilation system.
Remote Sensing, 14(2):371.

Voshtani, S., Ménard, R., Walker, T. W., and Hakami, A. (2022b).
Assimilation of GOSAT methane in the hemispheric CMAQ; part II: Results using optimal error statistics.
Remote Sensing, 14(2):375.

Weaver, A. T. and Mirouze, I. (2013).
On the diffusion equation and its application to isotropic and anisotropic correlation modelling in variational assimilation.
Quarterly Journal of the Royal Meteorological Society, 139(670):242–260.