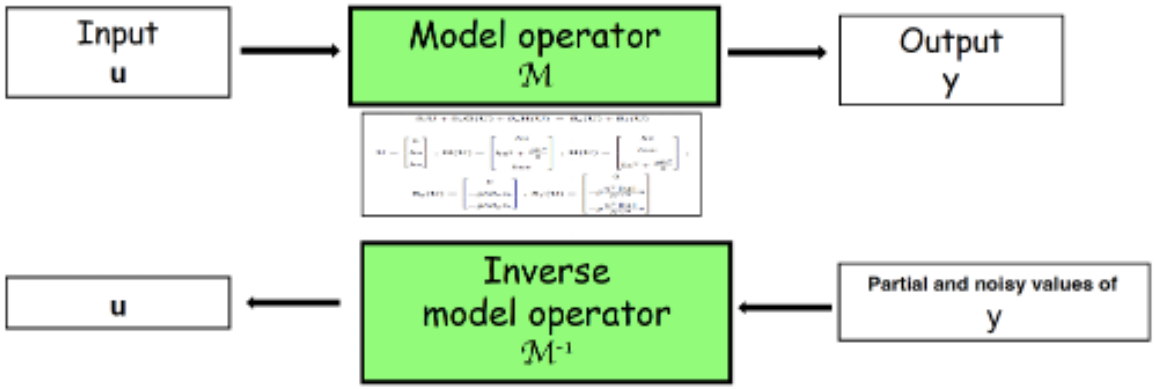
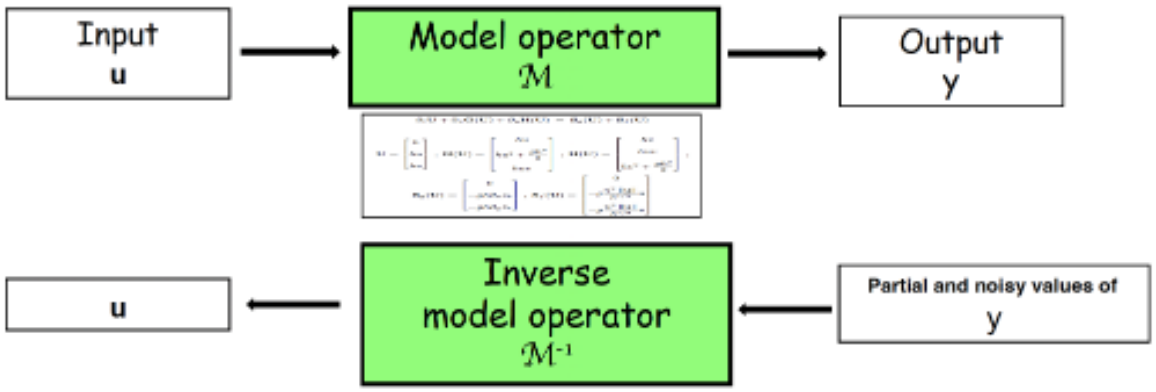
Generalities on inverse problems



In the "Hadamard sense", a model is well-posed if the following conditions hold :
i) it admits an unique solution y (given u),
ii) the solution $y(u)$ depends continuously on the input parameter(s) u
$\rightarrow$ stability concept.
Otherwise, it is said to be *ill-posed*.

▷Direct problems are usually the way that can be solved easily (compared to the inverse problem).
▷If the direct model operator maps causes to effects, the inverse operator maps the effects to the causes.

Generalities on inverse problems



▷Naïvely solving the inverse problem as $u = \mathcal{M}^{-1}(y^{obs})$ generally does not work even for linear operators $\mathcal{M}$.
Indeed,
- observations y^{obs} are not numerous enough or partial only : undetermined problem,
- the inverse operator $\mathcal{M}^{-1}$ is generally ill-conditioned : small variations of y implies large variations of u.

▷Indeed,
the direct models generally represent the lowest frequencies of the modelled phenomena = the energetic modes.
$\Rightarrow$ the highest frequencies of $y(u)$ are the lowest frequencies of $u(y)$.
$\Rightarrow$ separating noise from the inverse model output is a difficult task....

On Data Assimilation (DA)

▷DA combines in an "optimal" way the different knowledge : mathematical-physical model / the observations - datasets / the statistical errors (observations, models) and probabilistic priors (PDFs) (uncertainties modeling).

▷DA processes enable to :

- identify uncertain - unknown physical quantity of interest(s) (model parameters),
- calibrate the model,
- correct the model prediction.

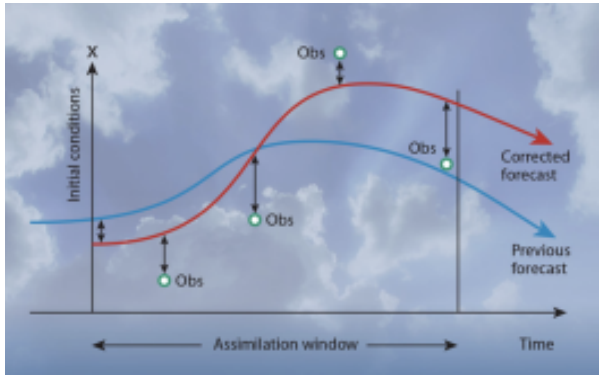


Fig. (L) Time-dependent model : assimilation window + prediction. Image source : ECMWF.

▷Once calibrated the model may be used as :

- a physically-based interpolator between sparse observations - data,
- a predictor (case of time-dependent models).

The Best Linear Unbiased Estimator (BLUE)

The simplest statistic estimator

Consider m measurements z_i^{obs} of a quantity u :

$$\boxed{z_i^{obs} = u + \varepsilon_i^{obs},\ i = 1, \ldots, m} \tag{1}$$

Hyp. : the obs errors ε_i^{obs} are unbiased and uncorrelated.

The BLUE $u^* \underset{def}{=}$ a linear function of z^{obs}, unbiased ($E(u^*) = u$), with the smallest variance.

▷In the scalar case ($u \in \mathbb{R}$),

$$u^* = \frac{1}{Var(u^*)}\left(\sum_{i=1}^{m} \frac{1}{\sigma_i^2} z_i^{obs}\right) \text{ with } \frac{1}{Var(u^*)} = \sum_{i=1}^{m} \frac{1}{\sigma_i^2}$$

Therefore the BLUE u^* minimizes the cost function : $\boxed{j(u) = \frac{1}{2}\sum_{i=1}^{m} \frac{1}{\sigma_i^2}(u - z_i^{obs})^2}$.

▷The Hessian, here $j''(u)$, measures the "analysis accuracy" / estimation conditioning, $\boxed{j''(u) = \frac{1}{Var(u^*)}}$.

Kalman Filter (KF) like formulation

▷Basic scalar case ($u \in \mathbb{R}$) with $m = 2$ obs. The BLUE re-writes as :

$$u^* = \frac{\sigma_2^2 z_1^{obs} + \sigma_1^2 z_2^{obs}}{\sigma_1^2 + \sigma_2^2} = z_1^{obs} + \left(\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \right) (z_2^{obs} - z_1^{obs})$$

Considering that :
a) z_1^{obs} is a first estimation (the *background-first guess* value) : $u_b \equiv z_1^{obs}$.
b) z_2^{obs} is an independent observation, newly obtained ($z \equiv z_2^{obs}$),
the BLUE u^* re-reads as :

$$\boxed{u^* = u_b + \underbrace{\left(\frac{\sigma_u^2}{\sigma_u^2 + \sigma_0^2} \right)}_{gain} \underbrace{(z - u_b)}_{innovation}} \tag{2}$$

▷In the general case, the KF estimate reads as :

$$\boxed{u^* = u_b + \text{Gain} \times \text{Innovation}}$$

The KF for a linear dynamic model

Linear dynamic model : $\boxed{u_n = M\ u_{n-1} + \varepsilon_{n-1}^{mod}}$ n the time index, $n \geq 1$.

The observations z^{obs} (available at the same time instants) : $\boxed{z_n^{obs} = Zu_n + \varepsilon_n^{obs}}$.

▷At each newly acquired data z_n^{obs}, the *KF estimation of the new value* u_n *is defined as being the BLUE* :

$$\boxed{u_n^{estim} = u_n^{model} + \underbrace{K}_{gain}\underbrace{(z_u^{obs} - Zu_n^{model})}_{innovation}} \tag{3}$$

▷If the operators M and Z are both **linear**, if the **errors are Gaussian**, **KF is the optimal sequential method**. See e.g. [Evensen's book 09].

▷For "weakly" non-linear problems, the Extended KF (ExKF) may be a good option...
▷For large dimensional problems, adopt the **Ensemble Kalman Filter (EnKF)** which is a recursive filter estimating the state u as the basic KF.
EnKF consists to perform a Monte-Carlo algorithm to estimate the covariance errors matrices.

Bayesian analysis

Probabilistic analysis

▷The estimation problem :

$$\boxed{\mathcal{M}(u) = z_{obs} + \varepsilon_{obs}} \tag{4}$$

The operator $\mathcal{M}(\cdot)$ is here a-priori non-linear.

▷Let $p(u)$ be the *prior distribution* of u, $p(z_{obs}|u)$ be the probability of z_{obs} given u (= the likelihood resulting from the direct model $\mathcal{M}(\cdot)$).

Following the Bayes law, the *posterior distribution* $p(u|z_{obs})$ satisfies :

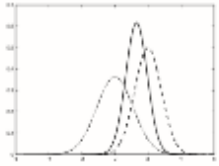
$$\boxed{p(u|z_{obs}) \propto p(z_{obs}|u) \cdot p(u)}$$

The most probable $u(x)$ (= the MAP), given the observations z_{obs} and the background (first estimation) u_b satisfies :

$$u^* = arg \max_u (p(u|z_{obs} \text{ and } u_b))$$

By considering the negative log-likelihood function

$\boxed{j(u) = -\log(p(u|z_{obs} \text{ and } u_b)) + cst}$, we have : $\boxed{u^* = arg \min_{u(x)} j(u)}$.



Bayesian analysis

If Prior(s) are Gaussian...

▷Thanks to the Bayes law, $j(u) = -\log p(z^{obs}|u) - \log p(u) + \text{ c}$.

▷**Prior PDFs are here assumed to be Gaussian**.

- The prior $p(u) \sim \mathcal{N}(u_b, B^{-1})$ B invertible covariance matrix ($\sigma_u^2 = det(B)$).

$$p(u) = \frac{1}{(2\pi)^{n/2}\sigma_u} \exp\left(-\frac{1}{2}\|u - u_b\|^2_{B^{-1}}\right)$$

- Likelihood $p(z^{obs}|u)$ is supposed to be Gaussian too (...) : $p(z_{obs}|u) \sim \mathcal{N}(\overline{z_{obs}}, R^{-1})$.

R an invertible covariance matrix. R is often chosen diagonal...

The M observations are supposed to all independent. Then,

$$p(z_{obs}|u) \propto \exp\left(\sum_{m=1}^{M} \|z_{obs,m} - \overline{z_{obs}}\|^2_{R^{-1}}\right) \tag{5}$$

▷NB. If the model is linear, then the likelihood is Gaussian too ; otherwise it is not.

▷**The posterior** $p(u|z_{obs})$ is Gaussian as a product of two Gaussians :

$$\boxed{p(u|z_{obs}) \propto \exp\left(-\frac{1}{2}\|u - \overline{u_p}\|^2_{P^{-1}}\right)} \tag{6}$$

with : $\overline{u_p} = P^{-1}(B^{-1}u_b + Z^T R^{-1} z_{obs})$, $P^{-1} = (B^{-1} + Z^T R^{-1} Z)$.

Bayesian analysis

Equivalence with a variational approach

▷Moreover, we get the most probable (= the Maximum A-Posteriori MAP) as :

$\boxed{u^* = arg\min_u j(u)}$ with

$$\boxed{j(u) \sim \frac{1}{2}\|\mathcal{M}(u) - z^{obs}\|^2_{R^{-1}} + \frac{1}{2}\|u - u_b\|^2_{B^{-1}}} \quad (7)$$

▷The Tykhonov-like regularization term $\|u - u_b\|^2_{B^{-1}}$
$\Leftrightarrow$ Gaussian hypothesis on u with expected minimal variance...

▷**Numerical computations.**
The posterior $p(u|z_{obs})$ (therefore the MAP, quantiles etc) can be computed by using the Metropolis-Hastings algorithm, a Markov Chain Monte Carlo (MCMC) algorithm [1].

$\oplus$ Pro : provides the finest expected information : $p(u|z_{obs})$.

$\ominus$ Con : requires extremely high numbers of model outputs ($O(10^p)$ with $p \approx 4$ and more).

$\Rightarrow$ hardly feasible for CPU-time consuming models and/or high dimension ($\dim(u) >>$).

1. MCMC are Markov Chains to perform Monte Carlo estimations.

Variational approach

▷This consists to formulate the estimation problem as an optimization problem : minimize $j(u)$ as (7) in the LQG case :

$$\boxed{j(u) = \frac{1}{2}\|Zu - z^{obs}\|^2_{R^{-1}} + \frac{1}{2}\|u - u_b\|^2_{B^{-1}}} \tag{8}$$

⇒ Differentiable optimization problem.
⇒ Gradient-based algorithms (BFGS) or Gauss-Newton method if affordable.

▷However, real-world estimation problems are not simple linear estimation problem as $Zu = z_{obs}$...
In what follows, the inverse problem consists at estimating $u(x)$ such that :

$$\boxed{\begin{array}{c} A(u(x); y(x)) = B(u(x)) \quad x \in \Omega \\ \text{with the observations } z_{obs} \text{ connected to } y.(z_{obs} \equiv y_{obs} \text{ in the most simple case}). \end{array}} \tag{9}$$

$A(u; y)$: a non-linear PDE-based operator.

Example : $A(u; y) = -div(\lambda(u)\nabla y) + y^3$ *with* $\lambda(\cdot)$ *a given function, and the RHS* $B(u) = g(\|\nabla u\|)$ *with* $g(\cdot)$ *a given function.*

▷The unknown parameter u is related to the measurements z_{obs} through the mathematical model $\boxed{\mathcal{M} : u \mapsto y(u)}$, $y(u)$ solution of (9).

Variational Data Assimilation (VDA)

▷**Direct model** (here a stationary non-linear PDE) :

$$\left\{ \begin{array}{l} \text{Given } u(x), \text{ find } y(x) \text{ such that :} \\ A(u(x); y(x)) = F(u(x)) \text{ in } \Omega \\ \oplus \text{ B.C. on } \partial\Omega \end{array} \right. \tag{10}$$

The parameter-to-state operator ("model operator") : $\mathcal{M}(u) = y^u$.

▷The **observation function** $J(u; y)$ is decomposed as :

$$\boxed{J(u; y) \;=\; J_{obs}(y) \;+ \alpha_{reg}\; J_{reg}(u)} \tag{11}$$

$J_{obs}(y) \;=\; \sum_m \|Z(y)(t_m) - z_m^{obs}\|^2_{\square}$ and $J_{reg}(u) \sim \|k - k_b\|^2_{\square}$ or $\sim \|D^p k\|^2_0$.
Z : the observation operator, potentially non-linear.

▷The **cost function** $j(u)$ is defined from the observation function $J(u; y)$ as :

$$\boxed{j(u) = J(u; y^u)} \tag{12}$$

with $y^u(x)$ the unique solution of the direct model, given $u(x)$.

▷The **optimization problem** :

$$\boxed{\left\{ \begin{array}{l} \text{Minimize } j(u) \text{ in } \mathcal{U}_{ad} \text{ under the "model constraint"} \\ \text{since } j(u) = J(u; y^u) \text{ with } \; y^u = \mathcal{M}(u). \end{array} \right.} \tag{13}$$

This is an **optimal control problem**.

▷Minimization by **gradient-based algorithm** $\Rightarrow$ compute the gradient $\nabla j(u)$.

On the importance of the norms in the cost function $j(u)$

Back to a basic 1D example
Let us consider 2 measurements of a scalar quantity u : $z_1 = 1$ and $z_2 = 2$.

$\Rightarrow$ Cost function to minimize : $j(u) = (u - z_1)^2 + (u - z_2)^2 = \|\mathrm{u} - \mathrm{z}_{obs}\|_2^2$.
$\hookrightarrow$ solution $u^* = \frac{3}{2}$.

$\triangleright$Hypothesis : z_2 is a measurement of the quantity $2u$ instead of u. Footnote [2].
$\Rightarrow$ Cost function to minimize : $j(u) = (u - z_1)^2 + (2u - z_2)^2 = \|\mathrm{u} - \mathrm{z}_{obs}\|_W^2$
with $W = diag(1, 2)$ $\hookrightarrow$ solution $u^* = \frac{9}{5}$!...

$\triangleright$The least-square solution depends on the fixed measurements norm $\|\cdot\|_\square^2$...

$\triangleright$**In the LQG case**, the BLUE (time-indep. estim), KF (time-dep. estim) and the Bayesian analysis provide **the same estimation** which **all equal the VDA solution if considering the norms R^{-1} and B^{-1}** defined from

$$\boxed{R = (Cov(\varepsilon_{obs})) \text{ and } B = (Cov(\varepsilon_b))} \tag{14}$$

2. Equivalently : we have more confidence on the 2nd measurement, therefore we assign a weight 2 to it.

Linear Quadratic Gaussian (LQG) case : equivalences

▷**In the LQG case**, the BLUE (time-indep. estim), KF (time-dep. estim) and the Bayesian analysis provide **the same estimation** which **all equal the VDA solution if considering the norms R^{-1} and B^{-1}** defined from

$$\boxed{R = (Cov(\varepsilon_{obs})) \text{ and } B = (Cov(\varepsilon_b))}$$

▷Each approach (KF / Bayesian-MCMC / VDA) has its pros and cons.

▷For large dimensional problems ($\dim(u) >>$) and non linear models :
VDA or combination of VDA-EnKF are the most suitable approach, with the R^{-1} and B^{-1} norms defined as best as possible...

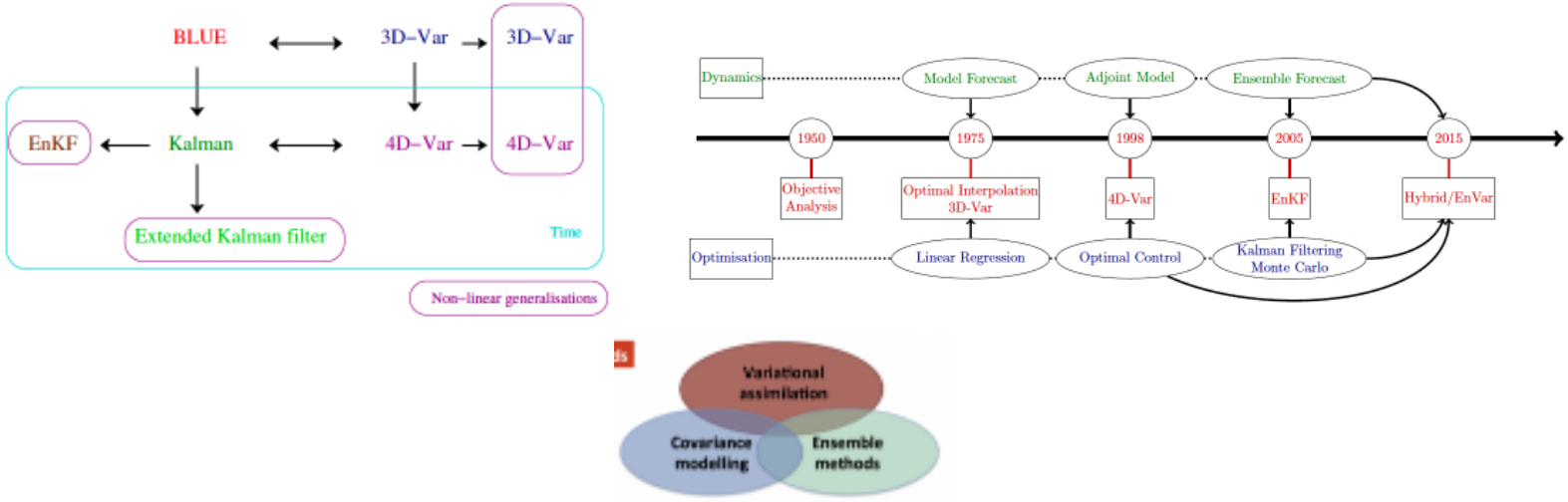
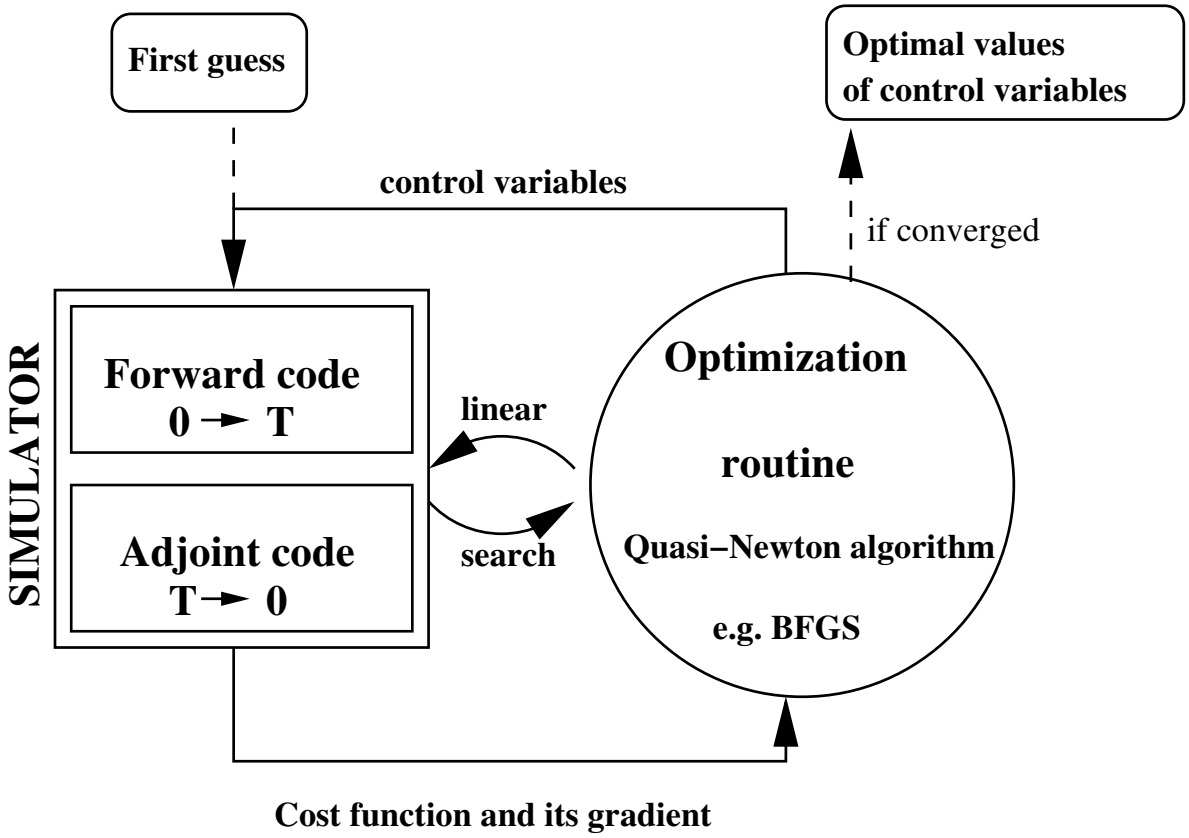


Figure – Images source : M. Bocquet's course.

Variational DA algorithms

3D-Var / 4D-Var



▷Cost(Inverse Problem)$\approx \mathcal{O}(10^2 \times$Cost(Direct Problem)).
▷Non-linear problems : the obtained $u^* = arg\,\min_u j(u)$ depends on the priors u_b, the norms R^{-1} and B^{-1}...

VDA vs KF in terms of algorithms

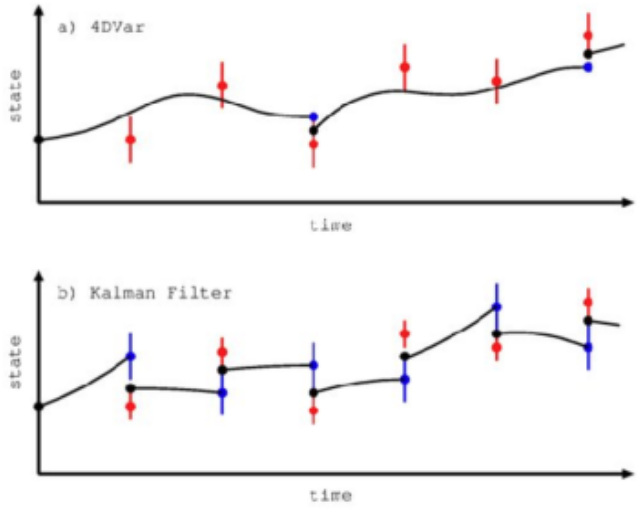


Figure – 4D-Var algorithm vs KF algorithm. In red : the obs ; in blue : the new estimation.

VDA vs KF in terms of algorithms :
- At each iteration n, the KF estimation depends on the previous time step state u_{n-1} and the current measurement z_n only : no additional past information is required like in VDA.
- KF is *not* affordable for large systems,
on contrary to VDA and EnKF which are well adapted to large dimensional systems.
- For non-linear models, basic KF does not apply. For "weakly" non linear problems, the Extended KF (ExKF) based on linearizations is an option. ExKF is then not optimal anymore.

Variational Data Assimilation

Computing the gradient $\nabla j(u)$ (steady-state case)

▷We have : $\boxed{J(u\,;y) \;=\; J_{obs}(y) \;+\alpha_{reg}\; J_{reg}(u)}$ with $\boxed{j(u) = J(u\,;y^u)}$.
Therefore the gradient :

$$\boxed{< \nabla j(u_0), \delta u > = < \nabla J_{obs}(y(u_0)), w^{\delta u} > + \alpha_{reg} < \nabla J_{reg}(u_0), \delta u >} \tag{15}$$

where $w^{\delta u}$ denotes the derivative of the state y with respect to u in the direction δu :
$\boxed{w^{\delta u} =< \nabla_u y(u_0), \delta u >}$.

▷**Option feasible for dim**$(u) = O(1)$ **: the linear tangent model**.

$$(\mathcal{LT}) \begin{cases} \text{Given } u \text{ and } y(u) \text{ the unique solution of the direct problem } (\mathcal{D}), \\ \textbf{given } \delta u, \text{ find } w^{\delta u} \text{ such that :} \\ \partial_y A(u;y) \cdot w^{\delta u}(x) \;=\; [-\partial_u A(u;y) + L'(u)] \cdot \delta u(x,t) \end{cases} \tag{16}$$

▷In time-dependent problems, the control / uncertain parameter is : $\boxed{c(x,t) = (y_{\mathbf{0}}(x), u(x,t))}$,
and the gradient components are : $\nabla j(c) = (\partial_{y_{\mathbf{0}}} j(c), \partial_u j(c))^T$.

Variational Data Assimilation

Computing the gradient $\nabla j(u)$ (steady-state case)

▷**For large dimensional problem (dim(u) >>) : introduce the adjoint model**.

$$(\mathcal{A}) \begin{cases} \text{Given } u(x) \text{ and } y^u(x) \text{ the unique solution of the direct problem } (\mathcal{D}), \\ \text{find } p(x) \text{ such that :} \\ \left[\partial_y A(u; y^u)\right]^T \cdot p(x) = (J_{obs})'(y^u(x)) \end{cases} \tag{17}$$

▷**The gradient components using the adjoint model** read :

$$\boxed{\nabla j(u)(x) =< -[\left(\partial_u A(u; y^u)\right] + [F'(u)])^T (x), p^u(x) > \ + \alpha_{reg} \nabla J_{reg}(u)(x)} \tag{18}$$

▷For complex real-world models, the adjoint codes may be an issue to derive and update...
→ **Automatic Differentiation (AD)** is a good solution.
However, for Fortran and C++ codes, AD requires quite good programming know-hows .

▷For time-dependent problems,
- the control / uncertain parameter is $c(x,t) = (y_{\mathbf{0}}(x), u(x,t))$ and $\nabla j(c) = (\partial_{y_{\mathbf{0}}} j(c), \partial_u j(c))^T$.
- the adjoint model is retroacting in time : $-\partial_t p(x,t) + [\partial_y A(u; y^u)]^T \cdot p(x,t) = (J_{obs})'(y^u(x,t))$ with $p(x,T)$ given.

DA & Neural Networks

Work done with Hugo Boulenc, INSA-IMT, PhD 2022-25.

▷Given numerous examples $(u(x); y(u)(x))$ (e.g. model outputs), one may be able to learn the map : $\boxed{\mathcal{N}_\theta : (u; x) \mapsto y(\theta)(u; x) \text{ with } \mathcal{N}_\theta(x) = (f_{L+1} \circ \ldots \circ f_1)(x)}$

θ : the huge dimensional vector of parameters of the NN. θ =(weight matrices W, bias vectors b).

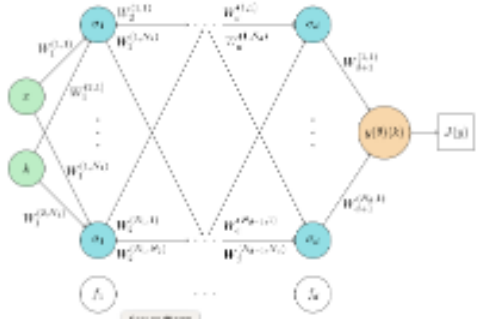


Figure – A classical NN to learn the map $\mathcal{F} : (x; k) \mapsto y(x; k)$.

▷**Training phase of a u-parametrized NN**

- Let $\mathcal{D}$ be a dataset containing pairs $\left((U; X)_s^{obs}), Y_s^{obs}\right)$, $s = 1, \cdots, N_s$.

Let $J_{obs}(\mathcal{D})$ be the misfit functional : $\boxed{J_{obs}(\mathcal{D}) = \| Y^{obs} - \mathcal{N}_\theta(U^{obs}; X^{obs})\|^2_{2,N_s}}$.

Training : $\boxed{\theta^* = \min_\theta j_{\mathcal{D},obs}(\theta)}$ with $\boxed{j_{\mathcal{D},obs}(\theta) = J_{obs}(\mathcal{D})}$.

- Optimization problem tractable and reliable **iff** $dim(\mathcal{X}) = O(1)$ **and** $dim(\mathcal{U}) = O(1)$.

Otherwise, dimensionality reduction techniques on the input variables $(U; X)$ is required.

▷**Parameter estimation.** If the NN has been well-trained, next one can easily solve :

$\boxed{u^* = arg \min_u \mathcal{J}_{\theta*}(u; x)}$.

The PINNs as proposed in [Raissi et al.'19]

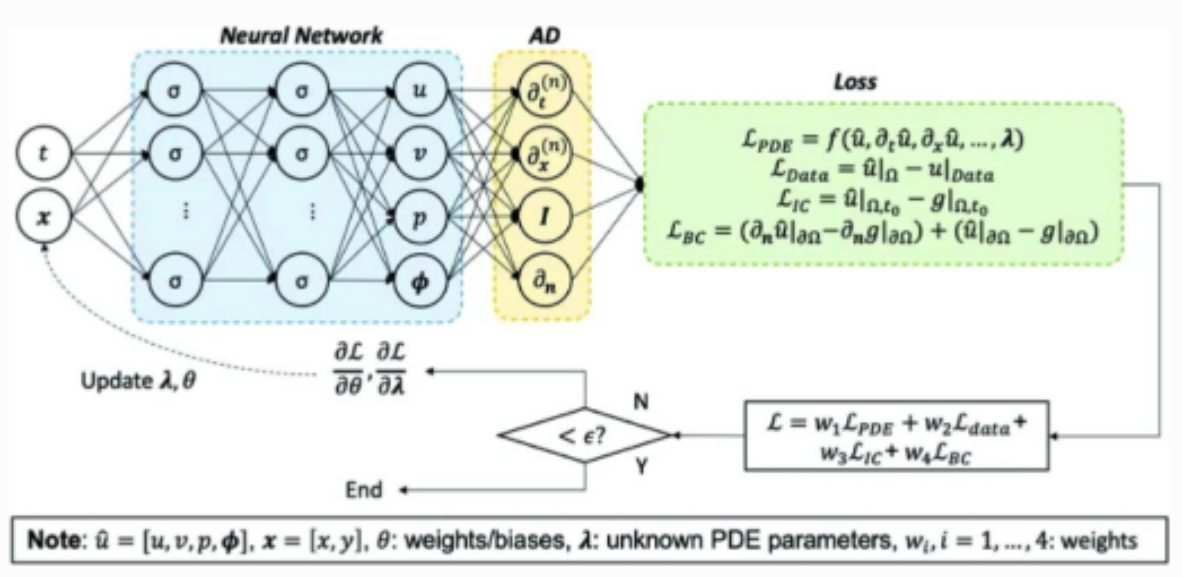


Figure – Flow chart of the PINNs as in [Cai et al. review '22]. $J \equiv \mathcal{L}$, $u \equiv \lambda$.

The loss is defined following [Lagaris et al.'97] :

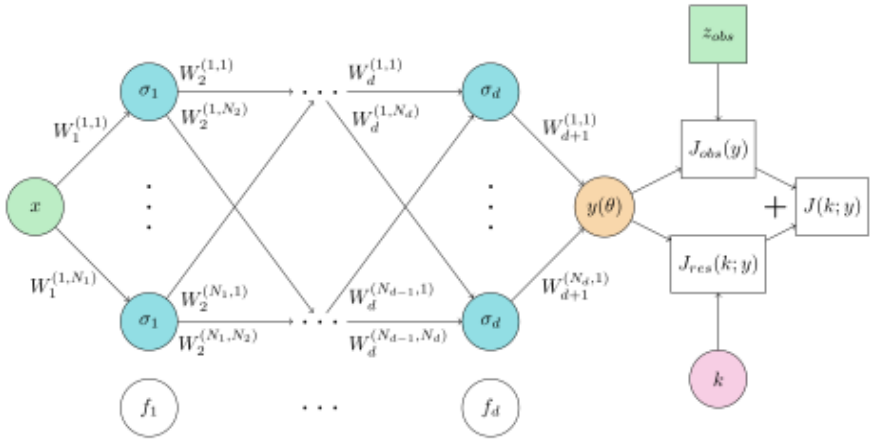
$$J(u;y) \sim \underbrace{\|A(u;y) - B(u)\|^2_{\omega_{col}}}_{J_{residual}(u;y)} + \underbrace{\|Z(y) - z^{obs}\|^2_{\omega_{obs}}}_{J_{obs}(y)} \tag{19}$$

1) Calculations of the derivatives wrt u are straightforward.
2) The derivatives wrt y in $A(u;y)$ required to evaluate $J_{residual}(\cdot;y)$ (e.g. $\Delta y(x)$) are performed by **Automatic Differentiation of the NN**. (Very nice trick !).

- *Recall :* $\mathcal{N}_\theta(x) = (f_{L+\mathbf{1}} \circ \ldots \circ f_{\mathbf{1}})(x)$. *Therefore :* $\mathcal{N}'_\theta(x) = \cdots$.

- *NB. This differentiation step was done by hand in [Lagaris et al. '97] with 1 hidden layer, direct modeling only.*

"PINNs as [Raissi et al.'19]" ≡ "Semi-parametrized" NNs, "Physically-Informed"



Following the ideas as [Lagaris et al.'97][Raissi et al.'19], the **loss (cost function)** is defined as :

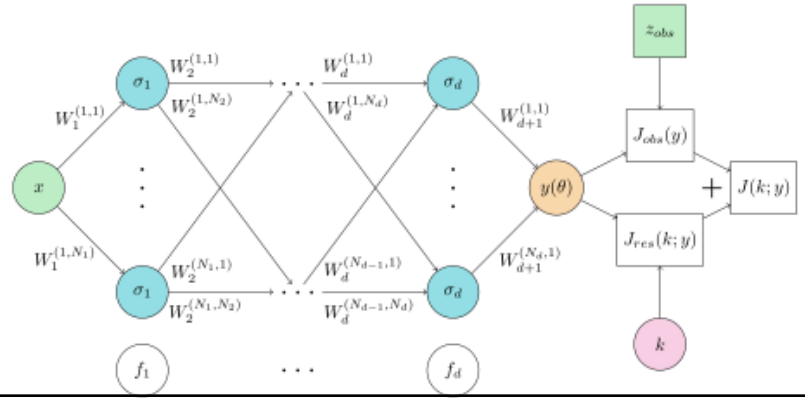
$$J(u;y) \sim \underbrace{\|A(u;y) - B(u)\|^2_{Q,\omega_{col}}}_{J_{residual}(u;y)} + \underbrace{\|Z(y) - z^{obs}\|^2_{R,\omega_{obs}}}_{J_{obs}(y)} \quad (20)$$

▷**Pros :**
- a well-trained NN constitutes a surrogate model,
- DA with NNs is easily formulated compared to VDA.

▷**Cons** :
- accuracy of $y(\theta^*)(u;x) \approx y^{true}(u;x)$ compared to (high-order) PDE numerical schemes...
- accuracy of u^* ?...

On-going investigations of connections between DA & NNs by Hugo Boulenc (PhD studies) with R. Bouclier (INSA-IMT) and P.-A. Garambois (INRAe Aix).

"PINNs as [Raissi et al.'19]" ≡ "Semi-parametrized" NNs, "Physically-Informed"



$$J(u;y) \sim \underbrace{\|A(u;y)-L(u)\|^2_{Q,\omega_{col}}}_{J_{residual}(u;y)} + \underbrace{\|Z(y)-z^{obs}\|^2_{R,\omega_{obs}}}_{J_{obs}(y)} \tag{21}$$

▷PINNs : the PDE solver = vanishing residual.

↪ Similar idea to the "weakly constrained VDA method" in [Tremolet'06] or residual-based solvers for non-linear PDEs.

▷"Semi-parametrized" NN : only the partial derivative wrt u is taken into account...
Indeed,

$$j'(u)\cdot\delta u = \partial_u J(u;y^u)\cdot\delta u + \underbrace{\partial_y J(u;y^u)\cdot(d_u y^u\cdot\delta u)}_{\text{not considered}} \tag{22}$$

↪ Well-known trick in optimal control e.g. in optimal design : the direct dependency of $J(u;\cdot)$ is taken into account while the dependency of $y(u)$ is not.
This incomplete gradient may, however, enable to already optimize something...

⇒ New class of computational methods, easy to implement...

On bi-objective optimization (both for VDA and NN formulations...)

How to tune the weight parameter α_{reg} ?

▷Recall : $\boxed{j(u) = J(u; y^u)}$ with $y^u(x)$ the unique solution of the direct model given u,

$$\boxed{J(u; y) \;=\; J_{obs}(y) \;+ \alpha_{reg}\; J_{reg}(u)} \tag{23}$$

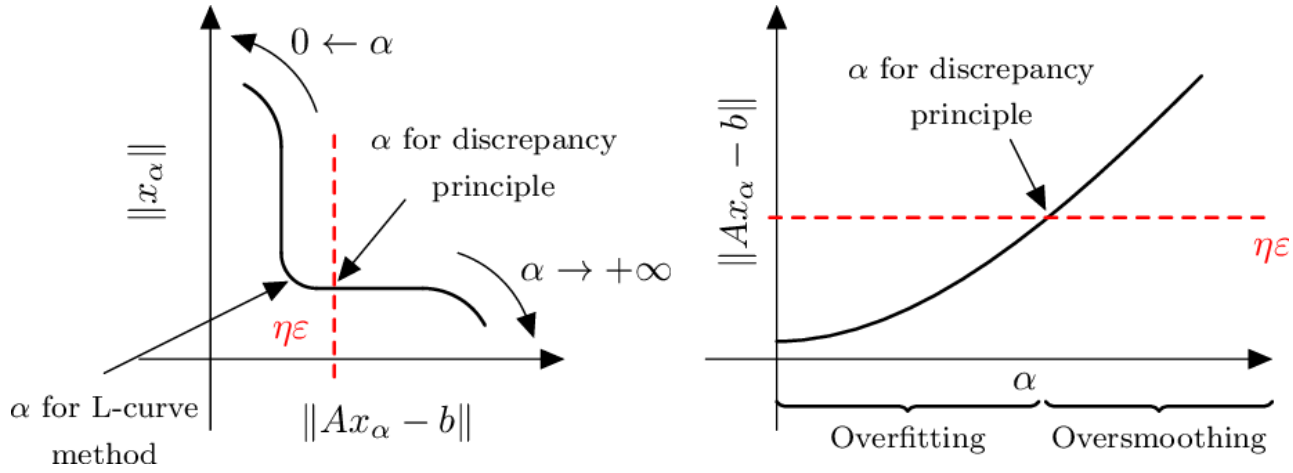


Figure – Bi-objective optimization and functional values obtained with different weight parameter values α_{reg} : (L) the L-curve criteria ; (R) : The Morozov's principle.

Summary & a few perspectives

▷Once the technical aspects Automatic Differentiation - HPC well controlled for the adjoint method, **the VDA approach remains a reference approach for high dimensional (non linear) problems**.

▷The complete analysis in the LQG case show what **norms R^{-1} and B^{-1}** should be considered even for non linear, high dimensional problems : central but difficult question(s)...

It exists intensive investigations for the atmosphere and oceans (and a very few in hydrology). The answers are often strongly connected to the precise studied phenomena...

▷**A few "research" questions to address to improve the accuracy and robustness of NN-based models** :

- what adaptive strategy for ω_{col} in PINNs approach ?
- what "optimal" norm Q to consider in $J_{residual}$ in PINNs approach ?
- how to certify the reliability of the NN-based surrogate models ?