# Chapter 4
# Nudging methods

Nudging, or Newtonian relaxation, is a simple yet dynamic method that aims to dynamically adjust the model toward the observations. The idea is simply to insert a feedback term into the model equation that is proportional to the observation–model misfit and *nudges* the model state toward the observations, as shown in Figure 4.1.

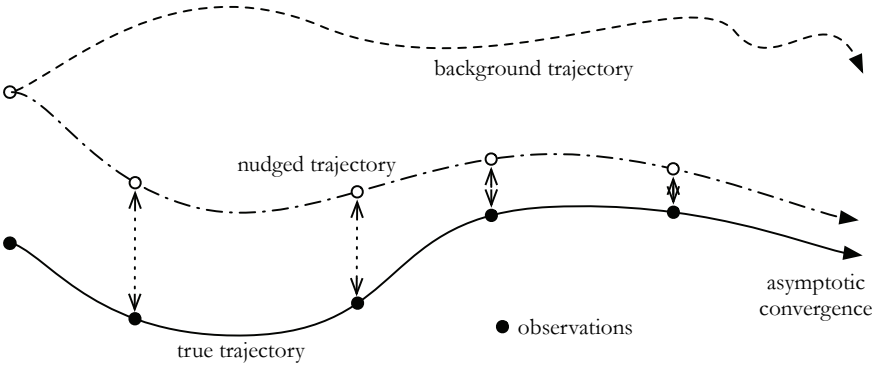


**Figure 4.1.** *Schematic representation of the nudging method.*

In geosciences nudging was first called the "dynamic initialization technique" and was introduced in the 1970s by Anthes [1974] for hurricane forecasting and by Hoke and Anthes [1976] for weather forecasting. Since then it has been used in a wide range of applications, as a first step toward more sophisticated DA methods. Its main advantage is its simplicity of implementation since there is no need for adjoint development, nor for a complex analysis/best linear unbiased estimator (BLUE) computation step. It is sufficient to implement an observation operator and to slightly modify the model to add a feedback term.

It is still used nowadays when more complex methods need to be avoided. However, more recent implementations are preferable, e.g., spectral nudging (see Section 4.1.5) or back and forth nudging (BFN; see Section 4.2).

# 4.1 ▪ Nudging

## 4.1.1 ▪ Principle

The nudging method can be explained with a very simple ODE. Let us assume that the model state variable $\mathbf{x}$ is a vector-valued real function of time $t \in \mathbb{R}, \mathbf{x}(t) \in \mathbb{R}^n$, and a solution of the differential model equation

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathscr{M}(\mathbf{x}(t)), \quad \mathbf{x}(t=0) = \mathbf{x}_0,$$

where $\mathscr{M} : \mathbb{R}^n \to \mathbb{R}^n$ is a given function, possibly nonlinear. We then assume that we have $p$ observations of the system $\mathbf{y}^o(t) \in \mathbb{R}^p$ distributed over time, and that we have an observation operator $\mathscr{H}$, possibly nonlinear, that maps, at each observation time, the state $\mathbf{x} \in \mathbb{R}^n$ into the observation space $\mathbb{R}^p$, so that $\mathscr{H}(\mathbf{x}(t)) \in \mathbb{R}^p$ and $\mathbf{y}^o(t)$ are comparable objects. We call the vector

$$\boldsymbol{\epsilon}^o(t) = \mathscr{H}(\mathbf{x}(t)) - \mathbf{y}^o(t) \in \mathbb{R}^p$$

the observation error. The nudging method, also called Newtonian relaxation, consists of looking for a solution, $\mathbf{x}(t)$, of the equation

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathscr{M}(\mathbf{x}(t)) - \mathbf{K}\{\mathscr{H}(\mathbf{x}(t)) - \mathbf{y}^o(t)\}, \quad \mathbf{x}(t=0) = \mathbf{x}_0,$$

where $\mathbf{K}$ is a matrix with $n$ rows and $p$ columns. The idea is simply to relax, or nudge, the model state vector $\mathbf{x}$ toward the observations. In the limit where $\mathbf{K}$ is very small, $\mathbf{x}$ only solves the model equation; if $\mathbf{K}$ is very large, then $\mathscr{H}(\mathbf{x}(t))$ is close in some sense to $\mathbf{y}^o$. An ideal choice of $\mathbf{K}$ is then a compromise between being close to the observations and satisfying the model equation, which is a classical motto in DA. A simplified linear case will allow us to see this more clearly. So we now assume that $\mathscr{M} = \mathbf{M}$ is a matrix, $p = n$, and $\mathscr{H} = \mathbf{I}$ is the identity. In this case, the nudging equation becomes

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{M}\mathbf{x} - \mathbf{K}(\mathbf{x} - \mathbf{y}^o), \quad \mathbf{x}(t=0) = \mathbf{x}_0,$$

and its solution is

$$\mathbf{x}(t) = e^{-(\mathbf{K}-\mathbf{M})t} \int_0^t e^{(\mathbf{K}-\mathbf{M})s} \mathbf{K}\mathbf{y}^o \, \mathrm{d}s + e^{-(\mathbf{K}-\mathbf{M})t} \mathbf{x}_0.$$

If $\mathbf{K}$ is symmetric positive definite, then we have (see, e.g., Auroux and Blum [2005])

$$\mathbf{x}(t) \underset{\mathbf{K}\to\infty}{\longrightarrow} \mathbf{y}^o(t) \quad \forall t > 0.$$

Let us mention that the nudging method has been well known since the 1960s in automatics and control theory, in the linear case where the model, $\mathbf{M}$, and observation operator, $\mathbf{H}$, are both matrices. In these domains it is called the Luenberger observer [Luenberger, 1966] or asymptotic observer. Indeed, under some hypotheses, $\mathbf{K}$ can be chosen so that the error between the true model state and the nudging equation solution tends to zero when $t$ goes to infinity, thus the name *asymptotic* observer.

### 4.1.2 ▪ Link with the KF

In a discrete-time framework, one time step of the nudging equation can be written as

$$\mathbf{x}_{k+1} = \mathbf{M}_{k+1:k}\mathbf{x}_k - \mathbf{K}\left\{\mathbf{H}_{k+1}(\mathbf{M}_{k+1:k}(\mathbf{x}_k)) - \mathbf{y}^{\mathrm{o}}_{k+1}\right\}.$$

If we call the vector $\mathbf{M}_{k+1:k}(\mathbf{x}_k)$ the forecast, then this equation can be seen as a predictor-corrector scheme that is very similar to the Kalman filter (KF) equations in the "no model error" case,

$$\mathbf{x}^{\mathrm{a}}_{k+1} = \mathbf{x}^{\mathrm{f}}_{k+1} + \mathbf{K}\left\{\mathbf{y}^{\mathrm{o}}_{k+1} - \mathbf{H}_{k+1}(\mathbf{x}^{\mathrm{f}}_{k+1})\right\}, \qquad \mathbf{x}^{\mathrm{f}}_{k+1} = \mathbf{M}_{k+1:k}(\mathbf{x}^{\mathrm{a}}_{k}).$$

The similarity between these two equations is obvious; they only differ by the choice of the $\mathbf{K}$ matrix. As we know that the KF is optimal in the linear Gaussian case, then this is the same for the nudging, provided the $\mathbf{K}$ matrix is exactly the associated Kalman gain matrix. Of course, in real nudging applications, the Kalman gain matrix is out of reach, so that the nudging method may be seen as a degraded yet simple KF.

### 4.1.3 ▪ Advantages and drawbacks

The nudging method is very simple, which sums up both its advantages and drawbacks: no complex implementation, but suboptimal results. Compared to variational assimilation, nudging does not require any adjoint model, whose development is a time-consuming task. Compared to Kalman filtering, it does not require complex implementation or the initialization of covariance matrices. Its implementation consists of, essentially

1. coding the observation operator as in 4D-Var and Kalman filtering;
2. tuning the $\mathbf{K}$ matrix, which can be chosen diagonal or even scalar; and
3. slightly modifying the model equations to incorporate the additional term $\mathbf{K}\{\mathscr{H}(\mathbf{x}(t)) - \mathbf{y}^{\mathrm{o}}(t)\}$.

It can be a first step toward a more elaborate DA method, which will require the same observation operator anyway, and it can give some hints about the potential of the observations to correct toward the model state. In some cases, such as small observation error, decorrelated in space and constant over time, it can be sufficient to achieve the desired results.

However, the nudging method with a diagonal $\mathbf{K}$ can produce poor results if, for example, the observation errors are correlated in space, or if the errors vary much in time—see Wunsch [2006] for more details. In this kind of framework, the specification of $\mathbf{K}$ is crucial, but its tuning is not automated and relies on numerical experimentation as well as the skill of the practitioner. Of course the tuning of $\mathbf{K}$ could be automated as well as optimized—see above the relationship with the KF gain matrix and below the optimal nudging scheme. However, in this case the ease of implementation and very likely the low computational cost would be lost.

### 4.1.4 ▪ Optimal nudging

As we have seen before, the nudging scheme can be seen, in the linear case, as a degraded, suboptimal form of the KF. It is therefore very tempting to improve the quality of the nudging matrix $\mathbf{K}$ to be closer to the KF optimality. However, the Kalman

gain matrix can be very costly to implement and compute, so it may seem preferable to find a way to improve $\mathbf{K}$ at a lower cost. As we have seen before, if the nudging coefficient is too small, then the model equation dominates and the observation has little effect on the variable. On the contrary, if it is too large, then the observation dominates, as well as the observation error if it is substantial. We want to find the optimal compromise between these two effects.

To do so, an optimal nudging procedure has been proposed by Zou et al. [1992]. They apply variational parameter estimation to the determination of the $\mathbf{K}$ matrix coefficients, in the case where $\mathbf{K}$ is chosen diagonal—but not space dependent, i.e., one coefficient is required for each physical variable. To do so, they form the cost function

$$J(\mathbf{K}) = J^{\mathrm{o}}(\mathbf{x}(\mathbf{K})) + J^{\mathrm{b}}(\mathbf{K}),$$

where

- $\mathbf{x}$ follows the nudging equation

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathcal{M}(x) - \mathbf{K}(\mathbf{x} - \mathbf{x}^{\mathrm{o}}), \quad \mathbf{x}(t=0) = \mathbf{x}_0;$$

- $J^{\mathrm{o}}$ measures the observation misfit, $\mathbf{x} - \mathbf{x}^{\mathrm{o}}$; and
- $J^{\mathrm{b}}$ is a regularization term for $\mathbf{K}$, i.e., a norm of $\mathbf{K} - \mathbf{K}^{\mathrm{b}}$, where $\mathbf{K}^{\mathrm{b}}$ is a first guess for $\mathbf{K}$.

The estimation of $\mathbf{K}$ is then performed as in regular variational assimilation (see Chapter 2): implementation of the adjoint model, computation of the gradient of the cost function with respect to $\mathbf{K}$, and iterative optimization of $\mathbf{K}$ using a gradient descent type minimization algorithm.

This method retains the advantage of simplicity, because the optimization concerns a small number of coefficients, as $\mathbf{K}$ is kept diagonal and the same nudging coefficient is used for a given variable at all grid points. Therefore, the optimization takes place in a low-dimensional space and can be very efficiently implemented at a small additional cost. The gain in using optimized coefficients is, however, great, as demonstrated by the numerical experiments of Zou et al. [1992]. A parallel study by Stauffer and Bao [1993] also highlighted the interest of this method while pointing out some difficulties, namely the sensitivity of the optimized nudging coefficients to their first guess and to the choice of the regularization term $J^{\mathrm{b}}(\mathbf{K})$, which could lead to poor results if badly specified.

A more recent study [Vidard et al., 2003] on the subject of optimal nudging proposed to use it in the framework of model error control in variational assimilation. While being similar to Zou et al. [1992], the approach is formulated slightly differently and allows us to take into account model error in some sense, which is quite a difficult problem in variational assimilation. The model equation is the same as in Zou et al. [1992],

$$\mathbf{x}_{k+1} = \mathcal{M}_{k+1:k}(\mathbf{x}_k) - \mathbf{K}_{k+1}\left\{\mathcal{H}_{k+1}(\mathcal{M}_{k+1:k}(\mathbf{x}_k)) - \mathbf{y}^{\mathrm{o}}_{k+1}\right\},$$

except that the nudging term is more general because it includes an observation operator, and also because it is seen as a model error term. Then the control vector is not only the initial condition, as in traditional 4D-Var, but augmented by the coefficients of the matrix $\mathbf{K}$. The cost function now has three terms,

$$J(\mathbf{x}_0, \mathbf{K}) = J^{\mathrm{o}}(\mathbf{x}) + J^{\mathrm{b}}(\mathbf{x}_0) + J^{\mathrm{nudg}}(\mathbf{x}, \mathbf{K}),$$

where

- $\mathbf{x}$ follows the nudging equation above;
- $J^{\mathrm{o}}$ measures the observation misfit $\mathscr{H}(\mathbf{x})-\mathbf{x}^{\mathrm{o}}$;
- $J^{\mathrm{b}}$ is the background cost, i.e., a norm of $\mathbf{x}_0-\mathbf{x}^{\mathrm{b}}$, where $\mathbf{x}^{\mathrm{b}}$ is the background;
- $J^{\mathrm{nudg}}$ is the nudging cost, formulated in terms of model error control (see Section 2.4.7.5 for the control of model error in a variational framework),

$$J^{\mathrm{nudg}}=\sum_{k,\mathrm{time}}\left\{\mathbf{K}_k(\mathscr{H}_k(\mathscr{M}_k(\mathbf{x}_{k-1}))-\mathbf{y}_k^{\mathrm{o}})\right\}^{\mathrm{T}}\mathbf{Q}_k^{-1}\left\{\mathbf{K}_k(\mathscr{H}_k(\mathscr{M}_k(\mathbf{x}_{k-1}))-\mathbf{y}_k^{\mathrm{o}})\right\},$$

  with $\mathbf{Q}_k$ the model error covariance matrix; and
- $\mathbf{K}_k$ may vary over the time steps $k$, and $\mathbf{K}_k$ may not be diagonal.

As we can see, the nudging here has two roles: first, as expected, it nudges the model toward the observation; second, it allows some kind of model error control. If $\mathbf{K}$ is not assumed to be diagonal, then this 4D optimal nudging scheme (4D-ON) provides better results than 4D-Var with an imperfect model. The authors also showed that in some cases it is better to choose $\mathbf{K}$ diagonal, and 4D-ON still provides improved results compared to 4D-Var and allows for numerical simplifications that greatly reduce the computational cost.

### 4.1.5 ▪ Spectral nudging

In this subsection, the equations are presented in the *continuous* case, because it simplifies the spectral formulation, using Fourier decomposition.

Global atmosphere or ocean forecasting is often limited to coarse-scale predictions because of computing power limitations. Regional forecasting allows us to use a higher-resolution model and to predict finer-scale phenomena. However, when using a regional model, it is crucial to correctly specify the boundary conditions. Nudging was used to provide such specifications in Davies [1976]. Let us denote by $x_R$ the regional model state and assume that the model evolution equation is

$$\frac{\mathrm{d}x_R}{\mathrm{d}t}=\mathscr{M}(x_R).$$

Then the equation is modified by nudging using the global model state $x_G$,

$$\frac{\mathrm{d}x_R}{\mathrm{d}t}=\mathscr{M}(x_R)-K(x_R-x_G), \tag{4.1}$$

where the nudging parameter $K\in\mathbb{R}$ is space dependent: $K$ is positive around the boundary of the regional domain and decreases to zero in the interior of the domain, its decrease being a function of the distance to the boundary.

This method allows us to take into account the global model information in a smooth way, but only near the boundary. It was later modified to take into account the global model everywhere inside the regional model (see Waldron et al. [1996] and more recent applications in von Storch and Langenberg [2000] and Miguez-Macho [2004]). The idea is that the large-scale trend of the regional model, being a refinement of the global model, should coincide with the global one. Therefore, the spectral nudging

method performs nudging of the regional model toward the global one, as in equation (4.1), but only for large scales. Thus,

$$\frac{\mathrm{d}x_R}{\mathrm{d}t} = \mathcal{M}(x_R) - \sum_{|p|<P} \sum_{|q|<Q} K_{p,q}(x_{p,q,R} - x_{p,q,G}) e^{ix\frac{2\pi p}{D_x}} e^{iy\frac{2\pi p}{D_y}},$$

where the regional domain is a rectangle of size $D_x$ in longitude $x$ and $D_y$ in longitude $y$, and $x_{p,q,R}$ (resp. $x_{p,q,G}$) are the regional (resp. global) model Fourier coefficients. We can see in this equation that the nudging coefficient $K$ depends on the scale, and that it only happens for large scales, in other words, for Fourier frequencies smaller than $P$ and $Q$. For small scales there is no nudging and the regional model alone drives the dynamics.

## 4.2 ▪ Back-and-forth nudging

### 4.2.1 ▪ Backward nudging

We have seen that the standard nudging method, under the hypotheses of Section 4.1.1, has asymptotic properties: the model state converges toward the observations when the time tends to infinity. In DA, time windows are always finite so that the nudging is suboptimal. The best fit to the observations is generally obtained at the end of the time window, as the state is nudged over time to the observations. However, in DA one is often interested in the initial state of the system, not the final state. This remark is the starting point of the backward nudging idea [Auroux, 2003]: "reverse" the model and perform nudging backward in time. To do so we first write formally, in the ODE framework, the backward model equation with a final condition,

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathcal{M}(\mathbf{x}(t)), \quad \mathbf{x}(t=T) = \mathbf{x}_T. \tag{4.2}$$

A change of variable, $t' = T - t$, leads to

$$-\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t'} = \mathcal{M}(\mathbf{x}(t')), \quad \mathbf{x}(t'=0) = \mathbf{x}_T,$$

which is a forward equation in $t'$ with an initial condition and can be rewritten as

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t'} = -\mathcal{M}(\mathbf{x}(t')), \quad \mathbf{x}(t'=0) = \mathbf{x}_T. \tag{4.3}$$

The backward nudging then consists of performing nudging on this last equation,

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t'} = -\mathcal{M}(\mathbf{x}(t')) - \mathbf{K}\left\{\mathcal{H}(\mathbf{x}(t')) - \mathbf{y}^{\mathrm{o}}(t')\right\}, \quad \mathbf{x}(t'=0) = \mathbf{x}_T. \tag{4.4}$$

The formal backward equation (4.2) and its forward equivalent (4.3) are generally ill-posed (e.g., when the forward model contains some kind of diffusive process) and may explode numerically. The presence of the nudging term in (4.4) then plays a double role: first it acts as a stabilizer in the numerical equation (putting the eigenvalues of the system back in the negative real plane), and second it nudges the system to the observations, providing a best fit at the initial time.

Auroux [2003] provides applications of backward nudging to the Lorenz model, as well as a barotropic quasi-geostrophic ocean model, and shows improved results

compared to direct nudging, with an error one third to one fourth the size on future forecasts. Figure 4.2 presents the rms errors obtained after assimilation for this latter model, where the improvement associated with backward nudging is highlighted.

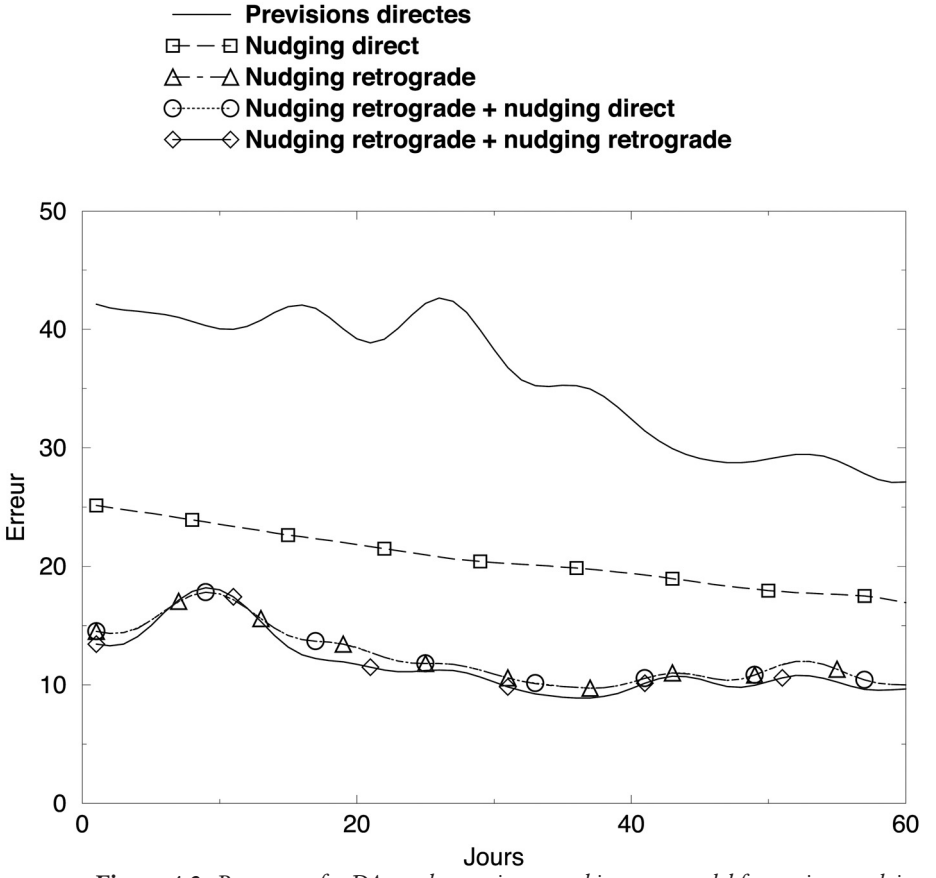


**Figure 4.2.** *Rms error for DA on the quasi-geostrophic ocean model for various nudging methods. Reprinted with permission from D. Auroux [2003].*

### 4.2.2 ▪ Back-and-forth nudging algorithm

Backward nudging improves forward nudging, but it still has the same major drawback, namely acting on a finite time window so that time stops before convergence happens. The BFN algorithm [Auroux and Blum, 2005] addresses this problem by alternating iteratively forward and backward nudging. The idea is that one iteration of forward and backward nudging over a duration $T$ could replace forward nudging during a double duration $2T$. Of course it is not strictly equivalent, but in essence this is the basis of the BFN algorithm: by iterating back and forth, we attain an asymptotic behavior, as if time were going to infinity.

Algorithm 4.1 can be implemented as follows (see also Figure 4.3):

1. Initialization of $\mathbf{x}_0$.
2. Forward nudging starting from $\mathbf{x}_0$. At the end of the time window it has reached the state $\mathbf{x}(T)$.

**Algorithm 4.1** BFN algorithm.

INITIALIZATION:

$$\tilde{\mathbf{x}}_{-1}(0) = \mathbf{x}_0$$

BFN LOOP OVER $j \geq 0$:
Forward nudging:

$$(F)\begin{cases} \dfrac{d\mathbf{x}_j}{dt} = \mathcal{M}(\mathbf{x}_j) - \mathbf{K}(\mathcal{H}(\mathbf{x}_j) - \mathbf{y}^o), \\ \mathbf{x}_j(0) = \tilde{\mathbf{x}}_{j-1}(0). \end{cases}$$

Backward nudging:

$$(B)\begin{cases} \dfrac{d\tilde{\mathbf{x}}_j}{dt} = \mathcal{M}(\tilde{\mathbf{x}}_j) + \mathbf{K}'(\mathcal{H}(\tilde{\mathbf{x}}_j) - \mathbf{y}^o), \\ \tilde{\mathbf{x}}_j(T) = \mathbf{x}_j(T). \end{cases}$$

3. Backward nudging starting at time $T$ from $\mathbf{x}(T)$. At the beginning of the time window it will have reached the state $\tilde{\mathbf{x}}(0)$.

4. Back to step 2: forward nudging with $\mathbf{x}(0)$ initialized with $\tilde{\mathbf{x}}(0)$.

5. Back to step 3, and repeat backward and forward nudging integrations until convergence.

We note that the forward nudging matrix $\mathbf{K}$ may differ from the backward nudging matrix $\mathbf{K}'$.

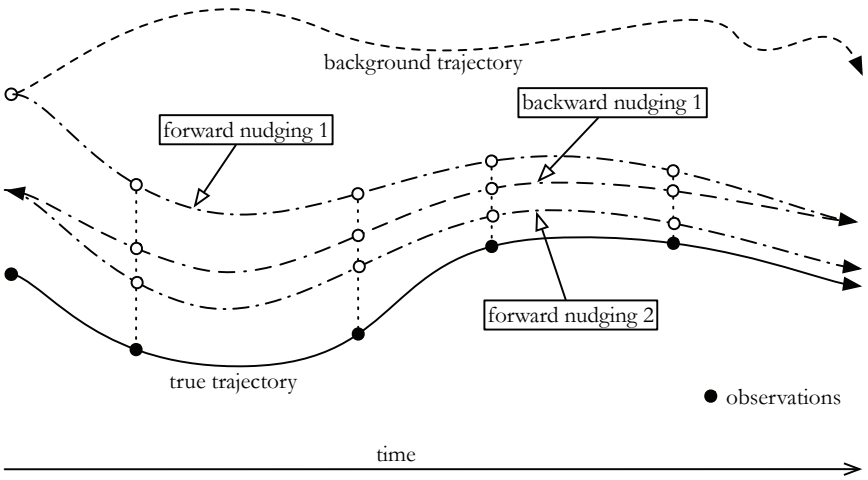


**Figure 4.3.** *Schematic representation of the BFN method.*

Auroux and Blum [2005] and Auroux and Blum [2008] detail some properties of the BFN algorithm, which we will recall in Section 4.2.3.

## 4.2.3 ▪ Algorithm properties

### 4.2.3.1 ▪ Convergence

Auroux and Blum [2005] prove the convergence in a very elementary case, where the model $\mathbf{M}$ is linear and $\mathbf{H}$ is the identity operator, so that the state vector is fully and perfectly observed. In this case, we can explicitly compute the state vectors $\mathbf{x}_j(t)$ and $\tilde{\mathbf{x}}_j(t)$ and prove that if $\mathbf{K}$ and $\mathbf{K}'$ are large enough, both converge as the iteration index $j$ goes to infinity toward limit trajectories $\mathbf{x}_\infty(t)$ and $\tilde{\mathbf{x}}_\infty(t)$. We can see in this special case that the convergence can be achieved in the finite time interval $[0,T]$, contrary to forward nudging, which requires $t \to \infty$, using the BFN trick of iterating nudging steps.

Moreover, if $\mathbf{K} = \mathbf{K}'$ and matrices $\mathbf{K}$ and $\mathbf{M}$ commute, then

$$\mathbf{x}_\infty(t) \underset{\mathbf{K}\to\infty}{\longrightarrow} \mathbf{y}^o(t), \quad \tilde{\mathbf{x}}_\infty(t) \underset{\mathbf{K}\to\infty}{\longrightarrow} \mathbf{y}^o(t) \quad \forall t \in [0,T].$$

Still, in the linear case, if $\mathbf{K} = \mathbf{K}'$ and if the limit trajectories are equal, $\mathbf{x}_\infty(t) = \tilde{\mathbf{x}}_\infty(t)$, then the limit satisfies both of the following equations:

$$\text{(F)} \quad \frac{d\mathbf{x}_\infty}{dt} = \mathbf{M}\mathbf{x}_\infty - \mathbf{K}(\mathbf{H}\mathbf{x}_\infty - \mathbf{y}^o),$$

$$\text{(B)} \quad \frac{d\mathbf{x}_\infty}{dt} = \mathbf{M}\mathbf{x}_\infty + \mathbf{K}(\mathbf{H}\mathbf{x}_\infty - \mathbf{y}^o).$$

Subtracting the equations, we obtain the following property for $\mathbf{x}_\infty$:

$$\mathbf{K}(\mathbf{H}\mathbf{x}_\infty(t) - \mathbf{y}^o(t)) = 0.$$

### 4.2.3.2 ▪ Variational interpretation

In Auroux and Blum [2008], the following variational interpretation of nudging is proposed in the simplified case where

- $\mathscr{H} = \mathbf{H}$ is linear;
- $\mathscr{M} = \mathbf{M}$ is linear and symmetric; and
- the nudging matrix is $\mathbf{K} = \mathbf{H}^T\mathbf{R}^{-1}$, where $\mathbf{R}$ is the observation error covariance matrix.

We denote by $k$ the time-stepping index. The nudging equation is discretized in time as

$$\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta t} = \mathbf{M}\mathbf{x}_{k+1} - \mathbf{K}(\mathbf{H}\mathbf{x}_{k+1} - \mathbf{y}^o),$$

so that $\mathbf{x}_{k+1}$ is the solution of the optimization problem

$$\min_{\mathbf{x}} \left[ \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k) - \frac{\Delta t}{2}\mathbf{x}^T\mathbf{M}\mathbf{x} + \frac{\Delta t}{2}(\mathbf{H}\mathbf{x} - \mathbf{y}^o)^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{x} - \mathbf{y}^o) \right].$$

The first two terms are the energy of the discrete model and the last one is the observation misfit. As in variational assimilation, the solution is therefore at each time step a compromise between minimizing the energy of the system and being close to the observations. This interpretation also provides a nice way to specify the nudging matrix, $\mathbf{K}$, while still retaining some error covariance information.

### 4.2.4 ▪ Well-posedness of the BFN: Theoretical studies

Theoretical results about one-dimensional transport equations have been given by Auroux and Nodet [2012]. The well-posedness of the BFN equations and/or the convergence of the algorithm were studied for various cases. In this framework, $x$ is the space variable, the model state is called $u$, and the model is continuous in space. A generic BFN step was then written as

$$\begin{aligned} &\text{(F)} \quad \partial_t u = M_1(u) + M_2(u) - K(t,x)(u - u^o), \quad u(0) = u_0, \\ &\text{(B)} \quad \partial_t \tilde{u} = M_1(\tilde{u}) + M_2(\tilde{u}) + K'(t,x)(\tilde{u} - u^o), \quad \tilde{u}(T) = u(T), \end{aligned}$$

where $u^o$ are observations of the system satisfying the forward equation without nudging. Let us remark that there is no observation operator $\mathcal{H}$, but $K$ depends on $t$ and $x$, so that it contains the information about where/when the observations are available or not. More precisely, if $K = 0$ at some point in space and time, then it means there is no observation.

Four types of equations were considered over the space interval $x \in [0,1]$:

- one-dimensional linear transport: $M_1(u) = -a(x)\partial_x u$, where $a(x)$ is a smooth given function;
- Burgers' equation: $M_1(u) = -u\partial_x u$;
- with a viscous term: $M_2(u) = \nu \partial_{xx} u$, where $\nu > 0$ is a given constant;
- without viscosity: $M_2(u) = 0$.

And various cases were considered for $K(t,x)$, with $K'(t,x) = \text{constant} \times K(t,x)$:

1. $K(t,x) = K \in \mathbb{R}$, so that $u$ is observed everywhere and for all time.
2. $K(t,x) = K 1_{[t_1,t_2]}(t)$ with $K \in \mathbb{R}$ and $0 < t_1 < t_2 < T$, so that $u$ is observed everywhere during a given time interval.
3. $K(t,x) = K(x)$ with $\text{Support}(K) \subset [a,b]$ and $0 < a < b < 1$, so that $u$ is unobserved on an open space with nonempty interior.

The results are summed up in Table 4.1. In the realistic case where the model state is not observed everywhere and there is viscosity, we can see that the BFN equations are not well-posed, even for a linear transport equation. Intuitively this is quite natural, because the backward equation contains a viscous term with the wrong sign, which is of course ill-posed. This is confirmed numerically, as the backward nudging constant $K'$ must be chosen very large for the backward equation to be stable.

### 4.2.5 ▪ BFN algorithm improvement

As has been studied by Auroux and Nodet [2012] and recalled above, the BFN algorithm is ill-posed in the presence of viscous terms and where the observations are sparse, because of the backward equation. Of course, numerically this is not a deadlock, because it is sufficient to set the backward nudging constant $K'$ to a very large number to counteract the largest eigenvalue of the viscous term. However, this can be an issue because the backward equation may force the model state toward the observations too brutally and one may wish to perform the backward nudging more subtly.

**Table 4.1.** *Summary of the theoretical results about BFN equations.*

| | Linear transport | Burgers' |
|---|---|---|
| Inviscid | • (B) and (F) equations well-posed.<br>• BFN algorithm converges.<br>• Rate of convergence is a function of $K(t,x)$. | • Results similar to the linear case. |
| Viscous | • Case (1) and (2) (observations everywhere): (B) and (F) equations well-posed; BFN converges.<br>• Case (3) (no observation on an open subset with nonempty interior):<br>**Ill-posed**; no solution backward. | • **Ill-posed** for every case (even (1)); no regularity of the solution. |

This is why the algorithm was modified by Auroux et al. [2013] to account for viscous equations. In this framework, it is recommended to change the sign of the viscosity in the backward equation so that it becomes well-posed and the nudging can be done with a reasonably small nudging parameter. More precisely, the model is separated into two parts, $\mathscr{M}$ for the nonviscous terms and $D$ for the viscous terms:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathscr{M}(\mathbf{x}) + D(\mathbf{x}), \quad \mathbf{x}(t=0) = \mathbf{x}_0.$$

The diffusive-BFN algorithm can be written as

$$\text{(F)} \quad \frac{\mathrm{d}\mathbf{x}_j}{\mathrm{d}t} = \mathscr{M}(\mathbf{x}_j) + D(\mathbf{x}_j) - \mathbf{K}(\mathscr{H}(\mathbf{x}_j) - \mathbf{y}^{\mathrm{o}}), \quad \mathbf{x}_j(0) = \tilde{\mathbf{x}}_{j-1}(0),$$

$$\text{(B)} \quad \frac{\mathrm{d}\tilde{\mathbf{x}}_j}{\mathrm{d}t} = \mathscr{M}(\tilde{\mathbf{x}}_j) - D(\tilde{\mathbf{x}}_j) + \mathbf{K}'(\mathscr{H}(\tilde{\mathbf{x}}_j) - \mathbf{y}^{\mathrm{o}}), \quad \tilde{\mathbf{x}}_j(T) = \mathbf{x}_j(T).$$

The motivation for doing so is twofold. First, it solves the problem of having to use a huge backward $\mathbf{K}'$. Second, in geoscience applications, the viscous term, $D$, is often an artificial parameterization of subscale phenomena and used to dissipate energy, so that it is "natural" to reverse it in the backward equation to maintain a physically realistic energy dissipation.

This improved version of the algorithm allowed the authors to obtain convergence where the classic BFN was unable to do so, with the presence of viscous terms, as well as sparse and noisy observations.

### 4.2.6 ▪ Why or when to choose the BFN algorithm?

The BFN algorithm is a cheap DA method, which has two major advantages. First, its implementation is extremely simple. Second, it converges fast. Therefore, it can be an ideal choice for a first dive into DA, with a minimal methodological and numerical investment.

Let us emphasize that the BFN is always preferable to the simple forward nudging method for three reasons. First, its implementation is almost equivalent and very simple either way. Second, it overcomes the finite-time window problem of forward

nudging, for which only asymptotic convergence is assured. Last, it provides fast convergence in a few iterations.

However, it can be too coarse for realistic applications. Indeed, although interesting results have been obtained so far with realistic oceanic or atmospheric models (see, e.g., Boilley and Mahfouf [2012] and Ruggiero et al. [2015]), it has not been tested operationally with real data. In the latter case, it could be recommended to use a few BFN iterations as a preprocessing tool to accelerate a more classic 4D-Var or KF algorithm (see, e.g., an academic illustration in Auroux [2009]).