4 次元変分法の基礎

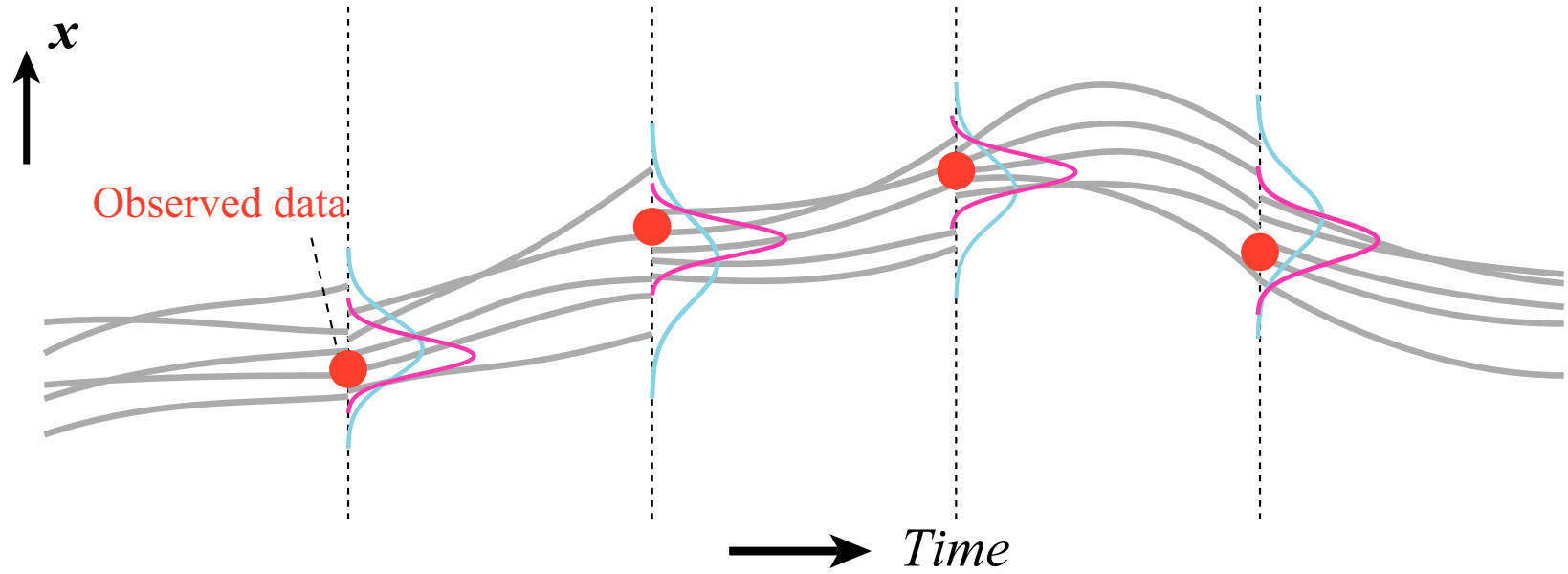
中野 慎也

「データ科学: 理論から実用へ」

14 September 2023

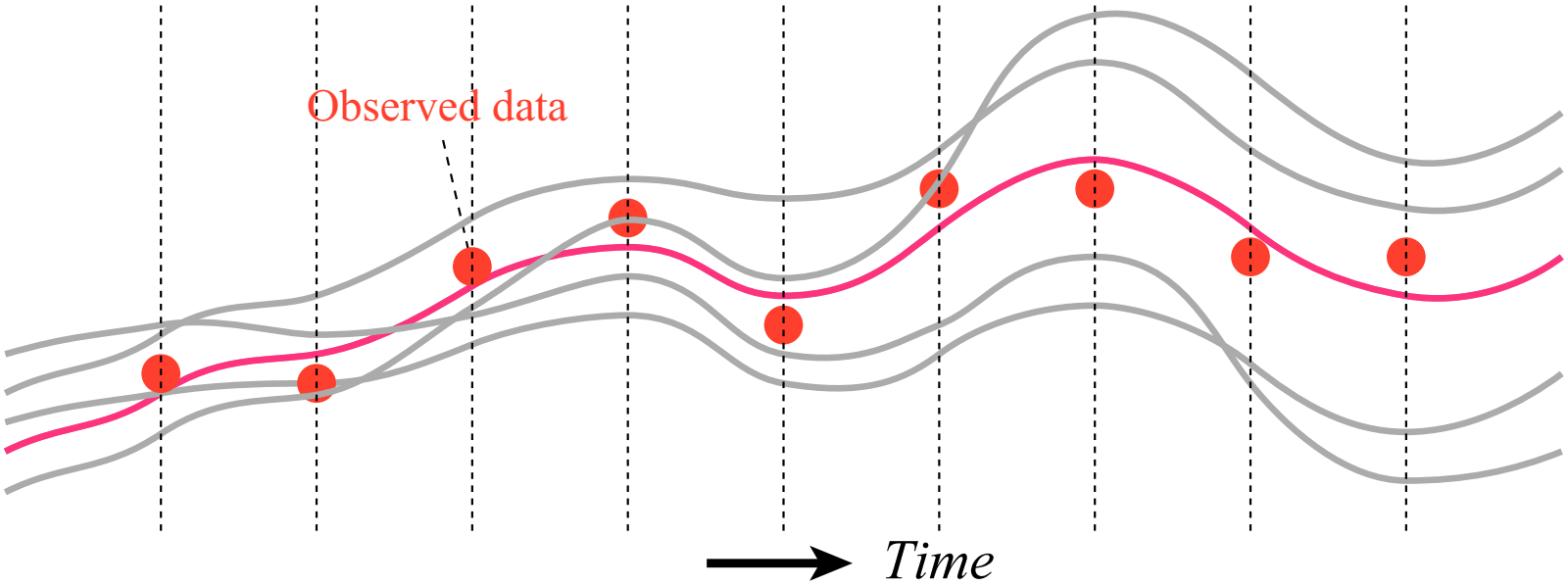
逐次データ同化手法

- 時間ステップごとに推定を行う．
- まず前の時間ステップからの予測を行い，データの情報を取り入れて，その予測を修正する．
- 得られた推定結果を，さらにその次の時間ステップの予測に用いる．



4 次元変分法

- 初期値やパラメータを調整して，ある一定の期間に得られたデータすべてに適合するようなシナリオを求める．
- 基本的にシミュレーションを信用し，$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1})$ という拘束条件の下で $\sum_{k=1}^{K} \|\boldsymbol{y}_k - \mathbf{H}_k \boldsymbol{x}_k\|^2$ を最小化する $\boldsymbol{x}_k$ を求めるという手続きになる．



4 次元変分法

- 4 次元変分法では，ある一定の期間に得られた観測データをすべて使い，様々な時刻，場所で得られた観測データと整合するようにシステムの時間発展を推定する．
- 確率分布の形で書くと，時刻 t_1 から t_K までに得られた観測 $\boldsymbol{y}_{1:K}$ が与えられた下で t_0 から t_K の状態 $\boldsymbol{x}_{0:K}$ に関する事後確率分布 $p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{1:K})$ を求めることになる．

これを分解すると，

$$
\begin{aligned}
p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{1:K}) &= \frac{p(\boldsymbol{y}_K|\boldsymbol{x}_{0:K},\boldsymbol{y}_{1:K-1})\,p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{1:K-1})}{p(\boldsymbol{y}_K|\boldsymbol{y}_{1:K-1})} \\
&= \frac{p(\boldsymbol{y}_K|\boldsymbol{x}_{0:K},\boldsymbol{y}_{1:K-1})\,p(\boldsymbol{x}_K|\boldsymbol{x}_{1:K-1},\boldsymbol{y}_{1:K-1})\,p(\boldsymbol{x}_{0:K-1}|\boldsymbol{y}_{1:K-1})}{p(\boldsymbol{y}_K|\boldsymbol{y}_{1:K-1})}.
\end{aligned}
\tag{1}
$$

4 次元変分法の定式化

ここで，状態空間モデル

$$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) + \boldsymbol{v}_k, \tag{2a}$$

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{w}_k \tag{2b}$$

を考える．(4 次元変分法では，観測が非線型であっても大きな問題はないので，$\mathbf{H}_k$ の代わりに $\boldsymbol{h}_k(\boldsymbol{x}_k)$ とおいた非線型観測モデルとしている．)

状態空間モデルでは，$\boldsymbol{y}_k$ は $\boldsymbol{x}_k$ が与えられた下で過去の観測 $\boldsymbol{y}_{1:k-1}$ や $\boldsymbol{x}_k$ と条件付き独立であると見なせ，$p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{y}_k|\boldsymbol{x}_k)$, $\boldsymbol{x}_k$ は $\boldsymbol{x}_{k-1}$ が与えられた下で過去の観測 $\boldsymbol{y}_{1:k-1}$ や過去の状態 $\boldsymbol{x}_{1:k-2}$ と条件付き独立と見なせ，$p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$.

4 次元変分法の定式化

$p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{y}_k|\boldsymbol{x}_k)$, $p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ を用いると，

$$
\begin{aligned}
&p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{1:K}) \\
&= \frac{p(\boldsymbol{y}_K|\boldsymbol{x}_{0:K}, \boldsymbol{y}_{1:K-1})\, p(\boldsymbol{x}_K|\boldsymbol{x}_{1:K-1}, \boldsymbol{y}_{1:K-1})\, p(\boldsymbol{x}_{0:K-1}|\boldsymbol{y}_{1:K-1})}{p(\boldsymbol{y}_K|\boldsymbol{y}_{1:K-1})} \\
&= \frac{p(\boldsymbol{y}_K|\boldsymbol{x}_K)\, p(\boldsymbol{x}_K|\boldsymbol{x}_{K-1})\, p(\boldsymbol{x}_{0:K-1}|\boldsymbol{y}_{1:K-1})}{p(\boldsymbol{y}_K|\boldsymbol{y}_{1:K-1})} \\
&= \frac{p(\boldsymbol{y}_K|\boldsymbol{x}_K)\, p(\boldsymbol{x}_K|\boldsymbol{x}_{K-1})}{p(\boldsymbol{y}_K|\boldsymbol{y}_{1:K-1})} \cdot \frac{p(\boldsymbol{y}_{K-1}|\boldsymbol{x}_{K-1})\, p(\boldsymbol{x}_{K-1}|\boldsymbol{x}_{K-2})\, p(\boldsymbol{x}_{0:K-2}|\boldsymbol{y}_{1:K-2})}{p(\boldsymbol{y}_{K-1}|\boldsymbol{y}_{1:K-2})} \\
&= \cdots \\
&= p(\boldsymbol{x}_0) \prod_{k=1}^{K} \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)\, p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})}.
\end{aligned}
\tag{3}
$$

4 次元変分法の定式化

システムノイズ $\boldsymbol{v}_k$, 観測ノイズ $\boldsymbol{w}_k$ がそれぞれガウス分布 $\mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$, $\mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ に従うとすると，

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \propto \exp\left[-\frac{1}{2}\big(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1})\big)^{\mathsf{T}}\mathbf{Q}_k^{-1}\big(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1})\big)\right], \tag{4}$$

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k) \propto \exp\left[-\frac{1}{2}\big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\big)^{\mathsf{T}}\mathbf{R}_k^{-1}\big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\big)\right]. \tag{5}$$

$p(\boldsymbol{x}_0)$ についてもガウス分布を仮定して

$$p(\boldsymbol{x}_0) \propto \exp\left[-\frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)^{\mathsf{T}}\mathbf{P}_b^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)\right] \tag{6}$$

とする．以上を

$$p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{1:K}) = p(\boldsymbol{x}_0)\prod_{k=1}^{K}\frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)\,p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})}$$

に代入して $\boldsymbol{x}_{0:K}$ の事後分布を求める．

4 次元変分法の定式化

先程の式を $p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{1:K})$ の式に代入すると，

$$
\begin{aligned}
&p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{1:K}) \\
&= p(\boldsymbol{x}_0) \prod_{k=1}^{K} \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)\, p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}{p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1})} \\
&\propto \exp\left[-\frac{1}{2} \left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)^{\mathsf{T}} \mathbf{P}_b^{-1} \left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) - \frac{1}{2} \sum_{k=1}^{K} \left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right)^{\mathsf{T}} \mathbf{R}_k^{-1} \left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right) \right. \\
&\qquad \left. -\frac{1}{2} \sum_{k=1}^{K} \left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1})\right)^{\mathsf{T}} \mathbf{Q}_k^{-1} \left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1})\right) \right].
\end{aligned}
$$

この指数部分を最大化すれば，事後確率 $p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{1:K})$ を最大化する $\boldsymbol{x}_{0:K}$ が得られる．

4 次元変分法の目的関数

事後分布 $p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{1:K})$ の指数部分を $-J$ とおくと，

$$J = \frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)^{\mathsf{T}} \mathbf{P}_b^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) + \frac{1}{2}\sum_{k=1}^{K}\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\Big)^{\mathsf{T}} \mathbf{R}_k^{-1}\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\Big) + \frac{1}{2}\sum_{k=1}^{K}\Big(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1})\Big)^{\mathsf{T}} \mathbf{Q}_k^{-1}\Big(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1})\Big) \tag{7}$$

となる．4 次元変分法では，基本的にこの J を最小化することで，事後確率密度 $p(\boldsymbol{x}_{0:K}|\boldsymbol{y}_{1:K})$ を最大化する $\boldsymbol{x}_{0:K}$ を求める．

4 次元変分法 (強拘束)

実際の応用ではシステムノイズ $\boldsymbol{v}_k$ を $\mathbf{0}$ とし，

$$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1}), \quad (k = 1, \ldots, K) \tag{8}$$

が成り立つという条件を課して推定を行うことが多い．この場合，目的関数は

$$J_s = \frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)^{\mathsf{T}} \mathbf{P}_b^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) + \frac{1}{2}\sum_{k=1}^{K}\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\Big)^{\mathsf{T}} \mathbf{R}_k^{-1}\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\Big) \tag{9}$$

となる．式 (8) のような条件を課すことを強拘束 (strong constraint) と呼ぶ．それに対して，式 (8) が成り立たない解も許容することを弱拘束 (weak constraint) と呼ぶ．

強拘束と弱拘束

- 強拘束のもとでは，初期値 x_0 が与えられると，

$$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) = \cdots = \boldsymbol{f}_k(\boldsymbol{f}_{k-1}(\cdots \boldsymbol{f}_1(\boldsymbol{x}_0)\cdots))$$

  のようにして，各時刻の x_k が決まる．

  各時刻の x_k は x_0 が与えられれば決まるので，x_0 のみ推定すればよいことになる．

- 強拘束では f_k が正しいという仮定の下で推定を行うのに対して，弱拘束ではデータ同化を行う場合はシミュレーションモデル f_k 自体の持つ誤差も考慮される．しかし，推定すべき変数が多くなるので，一般には推定が難しい．

強拘束の方が弱拘束よりも解きやすいので，以下でも基本的に強拘束の問題を考える．

評価関数に基づく推定

- 一般には，J_s を最小化する x_0 を解析に求めることはできない．
- そこで，J_s の x_0 についての勾配 $\nabla_{x_0} J$ を求め，降下法によって J_s の最適値を見つけ出す必要がある．
- J_s はかなり複雑な関数で，勾配の計算自体も大変だが，データ同化では $\nabla_{x_0} J$ をうまく求めるためにアジョイント法と呼ばれる手法が広く使われている．

降下法: $x_i \longleftarrow x_{i-1} - \alpha \nabla J$ を繰り返して J を最小化する x を求める．

拘束条件下での勾配

拘束条件が与えられたもとでの J_s の勾配を計算するには，ラグランジュ未定乗数法の考え方を応用するのが便利である．まず，以下の定理を確認する．

定理: $\boldsymbol{s}, \boldsymbol{t}$ をそれぞれ m 次元，p 次元の実ベクトルとする．また，$\boldsymbol{s}$ と $\boldsymbol{t}$ が満たすべき拘束条件が p 次元のベクトル値関数 $\boldsymbol{g}$ で

$$\boldsymbol{g}(\boldsymbol{s}, \boldsymbol{t}) = \boldsymbol{0} \tag{10}$$

のように与えられているとする．$\psi(\boldsymbol{s}, \boldsymbol{t})$ を実数値のスカラー関数，$\boldsymbol{\lambda}$ を p 次元の実ベクトルとして，

$$L = \psi(\boldsymbol{s}, \boldsymbol{t}) + \boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{s}, \boldsymbol{t}) \tag{11}$$

のような関数 L を導入したとき，$\nabla_{\boldsymbol{t}} L = \boldsymbol{0}$ が満たされるように $\boldsymbol{\lambda}$ を与えれば式 (10) の拘束条件の下での $\psi(\boldsymbol{s}, \boldsymbol{t})$ の $\boldsymbol{s}$ による勾配は $\nabla_{\boldsymbol{s}} L$ で得られる．

補足

なお，$\nabla_{\boldsymbol{t}} L$, $\nabla_{\boldsymbol{s}} L$ は，それぞれ L の t, s に関する勾配を表す．つまり，

$$\nabla_{\boldsymbol{t}} L = \begin{pmatrix} \frac{\partial L}{\partial t_1} \\ \vdots \\ \frac{\partial L}{\partial t_p} \end{pmatrix}, \qquad \nabla_{\boldsymbol{s}} L = \begin{pmatrix} \frac{\partial L}{\partial s_1} \\ \vdots \\ \frac{\partial L}{\partial s_m} \end{pmatrix} \tag{12}$$

である．

定理の証明

関数 $\psi(s,t)$ の s, t による偏微分を

$$\nabla_{\boldsymbol{s}}\,\psi = \frac{\partial \psi}{\partial \boldsymbol{s}}, \qquad \nabla_{\boldsymbol{t}}\,\psi = \frac{\partial \psi}{\partial \boldsymbol{t}}, \tag{13}$$

関数 $g(s,t)$ の s, t による偏微分で得られるヤコビ行列を

$$\tilde{\mathbf{G}}_{\boldsymbol{s}} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{s}^{\mathsf{T}}}, \qquad \tilde{\mathbf{G}}_{\boldsymbol{t}} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{t}^{\mathsf{T}}}, \tag{14}$$

と書くことにする．微分積分学の教科書に載っている陰関数定理により，$\boldsymbol{g}(s_0,t_0)=\mathbf{0}$ かつ $\tilde{\mathbf{G}}_{\boldsymbol{t}}$ が正則のとき，$s=s_0$, $t=t_0$ の近傍で以下を満たす p 次元ベクトル値関数 u が存在することが言える:

$$\boldsymbol{g}(\boldsymbol{s},\boldsymbol{u}(\boldsymbol{s})) = \mathbf{0}, \qquad \frac{d\boldsymbol{u}}{d\boldsymbol{s}^{\mathsf{T}}} = -\tilde{\mathbf{G}}_{\boldsymbol{t}}^{-1}\tilde{\mathbf{G}}_{\boldsymbol{s}}. \tag{15}$$

定理の証明

$\boldsymbol{s = s_0}$, $\boldsymbol{t = t_0}$ の近傍で，$\boldsymbol{t = u(s)}$ ならば $\boldsymbol{g(s,t) = 0}$ となることから，

$$\Psi(\boldsymbol{s}) = \psi\left(\boldsymbol{s}, \boldsymbol{u}(\boldsymbol{s})\right) \tag{16}$$

とおくと，Ψ の微分 $\nabla_{\boldsymbol{s}}\Psi$ が，$\boldsymbol{g(s,t) = 0}$ の制約の下での ψ の微分となる．計算すると，

$$\nabla_{\boldsymbol{s}}\Psi = \nabla_{\boldsymbol{s}}\,\psi + \left(\frac{d\boldsymbol{u}}{d\boldsymbol{s}^{\mathsf{T}}}\right)^{\mathsf{T}} \nabla_{\boldsymbol{t}}\,\psi = \nabla_{\boldsymbol{s}}\,\psi - \left(\tilde{\mathbf{G}}_{\boldsymbol{t}}^{-1}\tilde{\mathbf{G}}_{\boldsymbol{s}}\right)^{\mathsf{T}} \nabla_{\boldsymbol{t}}\psi. \tag{17}$$

定理の証明

一方，$L(\boldsymbol{s},\boldsymbol{t}) = \psi(\boldsymbol{s},\boldsymbol{t}) + \boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{g}(\boldsymbol{s},\boldsymbol{t})$ を $\boldsymbol{s}$, $\boldsymbol{t}$ についてそれぞれ偏微分を取ると，

$$\nabla_{\boldsymbol{s}} L = \nabla_{\boldsymbol{s}}\,\psi + \tilde{\mathbf{G}}_{\boldsymbol{s}}^{\mathsf{T}}\boldsymbol{\lambda}, \tag{18}$$

$$\nabla_{\boldsymbol{t}} L = \nabla_{\boldsymbol{t}}\,\psi + \tilde{\mathbf{G}}_{\boldsymbol{t}}^{\mathsf{T}}\boldsymbol{\lambda}, \tag{19}$$

$\nabla_{\boldsymbol{t}} L = \mathbf{0}$ とすると，

$$\boldsymbol{\lambda} = -\left(\tilde{\mathbf{G}}_{\boldsymbol{t}}^{\mathsf{T}}\right)^{-1}\nabla_{\boldsymbol{t}}\,\psi \tag{20}$$

が成り立つ．これを式 (18) に代入すると，

$$\nabla_{\boldsymbol{s}} L = \nabla_{\boldsymbol{s}}^{\mathsf{T}}\,\psi - \tilde{\mathbf{G}}_{\boldsymbol{s}}^{\mathsf{T}}\left(\tilde{\mathbf{G}}_{\boldsymbol{t}}^{\mathsf{T}}\right)^{-1}\nabla_{\boldsymbol{t}}\,\psi = \nabla_{\boldsymbol{s}}\,\psi - \left(\tilde{\mathbf{G}}_{\boldsymbol{t}}^{-1}\tilde{\mathbf{G}}_{\boldsymbol{s}}\right)^{\mathsf{T}}\nabla_{\boldsymbol{t}}\psi \tag{21}$$

となり，式 (17) の $\nabla_{\boldsymbol{s}}\Psi$ と一致する．したがって，$\nabla_{\boldsymbol{t}} L = \mathbf{0}$ のとき，$\nabla_{\boldsymbol{s}} L$ が $\boldsymbol{g}(\boldsymbol{s},\boldsymbol{t}) = \mathbf{0}$ の制約の下での ψ の微分となる．

補足

今，示したのは，

$$L = \psi(\boldsymbol{s}, \boldsymbol{t}) + \boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{s}, \boldsymbol{t}) \tag{11}$$

に対して，$\nabla_{\boldsymbol{t}} L = \boldsymbol{0}$ が満たされるように $\boldsymbol{\lambda}$ を決めれば拘束条件 $\boldsymbol{g}(\boldsymbol{s}, \boldsymbol{t}) = \boldsymbol{0}$ の下での $\psi(\boldsymbol{s}, \boldsymbol{t})$ の $\boldsymbol{s}$ による勾配が $\nabla_{\boldsymbol{s}} L$ で得られるということである．

さらに $\nabla_{\boldsymbol{s}} L = 0$ が成り立てば，拘束条件下で ψ の極値を与える $\boldsymbol{s}$ が得られる (通常のラグランジュ未定乗数法)．

定理を利用して，$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1}), \quad (k = 1, \ldots, K)$ という拘束条件の下で，$\boldsymbol{x}_0$ に関する J_s の勾配を得るには，

$$L_s = J_s + \sum_{k=1}^{K} \boldsymbol{\lambda}_k^{\mathsf{T}} \left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) \right) \tag{22}$$

という関数 L_s を導入する．

勾配の計算

関数

$$L_s = J_s + \sum_{k=1}^{K} \boldsymbol{\lambda}_k^{\mathsf{T}} \left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) \right)$$

は，拘束条件の項 $\boldsymbol{\lambda}_k^{\mathsf{T}} \left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) \right)$ が K 個足し合わされているが，
$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$ を $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1^{\mathsf{T}} & \ldots & \boldsymbol{x}_K^{\mathsf{T}} \end{pmatrix}^{\mathsf{T}}$ のように 1 つのベクトルにまとめ，さらに

$$\boldsymbol{\lambda} = \begin{pmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_K \end{pmatrix}, \qquad \boldsymbol{g}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{x}_1 - \boldsymbol{f}_1(\boldsymbol{x}_0) \\ \vdots \\ \boldsymbol{x}_K - \boldsymbol{f}_K(\boldsymbol{x}_{K-1}) \end{pmatrix} \tag{23}$$

とおくと，

$$\sum_{k=1}^{K} \boldsymbol{\lambda}_k^{\mathsf{T}} \left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) \right) = \boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{x}) \tag{24}$$

と書けるので，先程の定理の式と同じ形である．

勾配の計算

あとは，関数 $g(x)$ の x に関するヤコビ行列が正則であればよい．そこでヤコビ行列を計算してみると，

$$\tilde{\mathbf{G}}_{\boldsymbol{x}} = \begin{pmatrix} \mathbf{I} & \mathbf{O} & & & \\ \tilde{\mathbf{F}}_{2,\boldsymbol{x}_1} & \mathbf{I} & \mathbf{O} & & \\ \mathbf{O} & \tilde{\mathbf{F}}_{3,\boldsymbol{x}_2} & \mathbf{I} & \ddots & \\ & & \ddots & \ddots & \\ & & & \tilde{\mathbf{F}}_{K,\boldsymbol{x}_{K-1}} & \mathbf{I} \end{pmatrix} \tag{25}$$

のように前対角要素が 1 の下三角行列になるので，$\det \tilde{\mathbf{G}}_{\boldsymbol{x}} = 1$ であり，$\tilde{\mathbf{G}}_{\boldsymbol{x}}$ は必ず正則行列となる．

したがって，

$$L_s = J_s + \sum_{k=1}^{K} \boldsymbol{\lambda}_k^{\mathsf{T}} \left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) \right) \tag{22}$$

を導入すると，先程の定理が適用できる．

勾配の計算

$$L_s = J_s + \sum_{k=1}^{K} \boldsymbol{\lambda}_k^{\mathsf{T}} \left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) \right) \tag{22}$$

に基づいて，J_s の勾配 $\nabla_{\boldsymbol{x}_0} J_s$ を得るには，まず，

$$\nabla_{\boldsymbol{x}} L_s = \begin{pmatrix} \nabla_{\boldsymbol{x}_1} L_s \\ \vdots \\ \nabla_{\boldsymbol{x}_K} L_s \end{pmatrix} = \mathbf{0} \tag{26}$$

つまり，$\nabla_{\boldsymbol{x}_1} L_s = \nabla_{\boldsymbol{x}_2} L_s = \cdots = \nabla_{\boldsymbol{x}_K} L_s = \mathbf{0}$ を満たす $\boldsymbol{\lambda} = \begin{pmatrix} \boldsymbol{\lambda}_1^{\mathsf{T}} & \cdots \boldsymbol{\lambda}_K^{\mathsf{T}} \end{pmatrix}^{\mathsf{T}}$ を求めればよい．そうすれば，あとは $\nabla_{\boldsymbol{x}_0} L_s = \nabla_{\boldsymbol{x}_0} J_s$ より，$\nabla_{\boldsymbol{x}_0} J_s$ が得られる．

勾配の計算

$$L_s = J_s + \sum_{k=1}^{K} \boldsymbol{\lambda}_k^{\mathsf{T}} \left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) \right) \tag{22}$$

に，J_s の式を代入すると，

$$\begin{aligned} L_s = & \frac{1}{2} \left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b \right) \mathbf{P}_b^{-1} \left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b \right) + \frac{1}{2} \sum_{k=1}^{K} \left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \right) \mathbf{R}_k^{-1} \left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \right) \\ & + \sum_{k=1}^{K} \boldsymbol{\lambda}_k^{\mathsf{T}} \left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) \right) . \end{aligned} \tag{27}$$

この勾配を計算するには，ベクトル a とベクトル値関数 $g(x)$ の内積，およびベクトル値関数 $g(x)$ の 2 次形式 $g(x)^{\mathsf{T}} \mathbf{A} g(x)$ の微分が必要になる．先にそれを計算しておく．

ベクトル a とベクトル値関数 $g(x)$ の内積の勾配

$\boldsymbol{a}$ を n 次元ベクトル，$\boldsymbol{x}$ を m 次元ベクトル，$\boldsymbol{g}(\boldsymbol{x})$ を n 次元ベクトル値関数とする．

$$\psi_1 = \boldsymbol{a}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{x}) = \sum_{i=1}^{m} a_i g_i(\boldsymbol{x}) \tag{28}$$

の $\boldsymbol{x}$ に関する勾配は

$$\nabla_{\boldsymbol{x}} \psi_1 = \sum_{i=1}^{m} a_i \begin{pmatrix} \frac{\partial g_i}{\partial x_1}(\boldsymbol{x}) \\ \vdots \\ \frac{\partial g_i}{\partial x_m}(\boldsymbol{x}) \end{pmatrix} = \tilde{\mathbf{G}}_{\boldsymbol{x}}^{\mathsf{T}} \boldsymbol{a} \tag{29}$$

となる．但し，$\tilde{\mathrm{G}}_{\boldsymbol{x}}$ は関数 $\boldsymbol{g}$ の $\boldsymbol{x}$ に関するヤコビ行列である．

二次形式の勾配

次に，m 次の対称行列 $\mathbf{A}$ を考え，二次形式

$$\psi_2 = \boldsymbol{g}(\boldsymbol{x})^{\mathsf{T}} \mathbf{A} \boldsymbol{g}(\boldsymbol{x}) = \sum_{i,j} A_{ij}\, g_i(\boldsymbol{x})\, g_j(\boldsymbol{x}) \tag{30}$$

の勾配

$$\nabla_{\boldsymbol{x}} \psi_2 = \begin{pmatrix} \frac{\partial \psi_2}{\partial x_1} \\ \vdots \\ \frac{\partial \psi_2}{\partial x_m} \end{pmatrix} \tag{31}$$

を考える．l 番目の要素のみ考えると $\mathbf{A}$ は対称行列 $(A_{ij} = A_{ji})$ なので，

$$\frac{\partial \psi_2}{\partial x_l} = 2 \sum_{i,j} A_{ij} g_i(\boldsymbol{x}) \frac{\partial g_j}{\partial x_l}.$$

となる．

二次形式の勾配

$l = 1, \ldots, m$ に対し，

$$\frac{\partial \psi_2}{\partial x_l} = 2 \sum_{i,j} A_{ij} g_i(\boldsymbol{x}) \frac{\partial g_j}{\partial x_l}.$$

となるので，まとめると，

$$\nabla_{\boldsymbol{x}} \psi_2 = 2 \tilde{\mathbf{G}}_{\boldsymbol{x}}^{\mathsf{T}} \mathbf{A} \boldsymbol{g}(\boldsymbol{x}).$$

但し，$\tilde{\mathbf{G}}_{\boldsymbol{x}}$ は関数 $\boldsymbol{g}(x)$ のヤコビ行列

$$\tilde{\mathbf{G}}_{\boldsymbol{x}} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}^{\mathsf{T}}} = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_m} \end{pmatrix}.$$

勾配の計算

今述べた結果を用いて，

$$L_s = \frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)\mathbf{P}_b^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) + \frac{1}{2}\sum_{k=1}^{K}\left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right)\mathbf{R}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right)$$
$$+ \sum_{k=1}^{K}\boldsymbol{\lambda}_k^{\mathsf{T}}\left(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1})\right)$$

の $\boldsymbol{x}_k$ $(k = 1, \ldots, K-1)$ についての勾配を取ると，

$$\nabla_{\boldsymbol{x}_k} L_s = -\tilde{\mathbf{H}}_{k,\boldsymbol{x}_k}^{\mathsf{T}}\mathbf{R}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right) + \boldsymbol{\lambda}_k - \tilde{\mathbf{F}}_{k+1,\boldsymbol{x}_k}^{\mathsf{T}}\boldsymbol{\lambda}_{k+1}. \tag{32}$$

また，$\boldsymbol{x}_K$ についての勾配は

$$\nabla_{\boldsymbol{x}_K} L_s = -\tilde{\mathbf{H}}_{K,\boldsymbol{x}_K}^{\mathsf{T}}\mathbf{R}_K^{-1}\left(\boldsymbol{y}_K - \boldsymbol{h}_K(\boldsymbol{x}_K)\right) + \boldsymbol{\lambda}_K. \tag{33}$$

勾配の計算

$\nabla_{\boldsymbol{x}_k} L_s$ が $k = 1, \ldots, K$ に対してすべて $\mathbf{0}$ となるには，$k = 1, \ldots, K-1$ について

$$\boldsymbol{\lambda}_k = \tilde{\mathbf{H}}_{k,\boldsymbol{x}_k}^{\mathsf{T}} \mathbf{R}_k^{-1} \left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \right) + \tilde{\mathbf{F}}_{k+1,\boldsymbol{x}_k}^{\mathsf{T}} \boldsymbol{\lambda}_{k+1}, \tag{32}$$

が成り立ち，さらに

$$\boldsymbol{\lambda}_K = \tilde{\mathbf{H}}_{K,\boldsymbol{x}_K}^{\mathsf{T}} \mathbf{R}_K^{-1} \left(\boldsymbol{y}_K - \boldsymbol{h}_K(\boldsymbol{x}_K) \right) \tag{33}$$

が成り立てばよい．

これをすべて満たす $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_K$ は，以下の手順で得られる．

1. まず $\boldsymbol{x}_0$ から，$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1})$ に従って，$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$ を求める．
2. 次に，式 (33) から $\boldsymbol{\lambda}_K$ を求める．
3. 式 (32) を $k = K-1, \ldots, 1$ の順に再帰的に適用して $\boldsymbol{\lambda}_K, \ldots, \boldsymbol{\lambda}_1$ を得る．

$*$ なお，このように後ろ向きに $\tilde{\mathbf{F}}^{\mathsf{T}}$ を適用する手続きは，人工ニューラルネットワークの学習に使われる back-propagation 法とほぼ同じ．

勾配の計算

$$\boldsymbol{\lambda}_k = \tilde{\mathbf{H}}_{k,\boldsymbol{x}_k}^{\mathsf{T}} \mathbf{R}_k^{-1} \left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \right) + \tilde{\mathbf{F}}_{k+1,\boldsymbol{x}_k}^{\mathsf{T}} \boldsymbol{\lambda}_{k+1}, \tag{32}$$

$$\boldsymbol{\lambda}_K = \tilde{\mathbf{H}}_{K,\boldsymbol{x}_K}^{\mathsf{T}} \mathbf{R}_K^{-1} \left(\boldsymbol{y}_K - \boldsymbol{h}_K(\boldsymbol{x}_K) \right) \tag{33}$$

を満たす $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_K$ が求まったら，

$$\nabla_{\boldsymbol{x}_0} J_s = \nabla_{\boldsymbol{x}_0} L_s = \mathbf{P}_b^{-1} \left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b \right) - \tilde{\mathbf{F}}_{1,\boldsymbol{x}_0}^{\mathsf{T}} \boldsymbol{\lambda}_1 \tag{34}$$

より，$\nabla_{\boldsymbol{x}_0} J_s$ が求まる．

$\nabla_{\boldsymbol{x}_0} J_s$ が得られれば，最急降下法や準ニュートン法などを用いて J_s を最小化する $\boldsymbol{x}_0$ を求めることができる．このようにして，最適な初期値 x_0 を求めるのがアジョイント法である．

アジョイント法のアルゴリズム (まとめ)

- $\boldsymbol{x}_0$ に関する初期推定値 $\boldsymbol{x}_{0,0}$ を設定する.
- $m := 0$ とする.
- 収束するまで以下を行う.
  - 初期値 $\boldsymbol{x}_{0,m}$ の下で，シミュレーションを時間ステップ K まで実行し，$\boldsymbol{x}_{1,m}, \ldots, \boldsymbol{x}_{K,m}$ を得る
  - $\boldsymbol{\lambda}_{K,m} = \tilde{\mathbf{H}}^{\mathsf{T}}_{K,\boldsymbol{x}_{K,m}} \mathbf{R}_K^{-1} \left(\boldsymbol{y}_K - \boldsymbol{h}_K(\boldsymbol{x}_{K,m})\right)$ と設定する.
  - $k = K-1, \ldots, 1$ について，

$$\boldsymbol{\lambda}_{k,m} = \tilde{\mathbf{H}}^{\mathsf{T}}_{k,\boldsymbol{x}_{k,m}} \mathbf{R}_k^{-1} \left(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_{k,m})\right) + \tilde{\mathbf{F}}^{\mathsf{T}}_{k+1,\boldsymbol{x}_{k,m}} \boldsymbol{\lambda}_{k+1,m}$$

    を順々に計算する.
  - $\nabla_{\boldsymbol{x}_0} J_s$ を

$$\nabla_{\boldsymbol{x}_{0,m}} J_s = \mathbf{P}_b^{-1} \left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) - \tilde{\mathbf{F}}^{\mathsf{T}}_{1,\boldsymbol{x}_0} \boldsymbol{\lambda}_{1,m}$$

    のように求める.
  - $\boldsymbol{x}_{0,m+1} = \boldsymbol{x}_{0,m} - \alpha \nabla_{\boldsymbol{x}_{0,m}} J_s$ により，J_s が小さくなるように $\boldsymbol{x}_0$ の値を修正する.
  - $m := m + 1$ とおく.

アジョイント法のまとめ

アジョイント法は，元のシミュレーションモデルによる forward 計算と，アジョイントモデルによる backward 計算を行うことで，評価関数の勾配を計算し，初期値 x_0 の最適化を行う手法である．

- アジョイント法の利点
  - アジョイント法は，アンサンブルカルマンフィルタなどよりは，シミュレーションの実行回数を抑えられる．
  - $\boldsymbol{f}_k(\cdot)$ のヤコビ行列 $\tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1,m}}$ が出てくるが，線型近似を用いているわけではないので，非線型の問題でも基本的には妥当な解が得られる．
- アジョイント法の欠点
  - ヤコビ行列 $\tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1,m}}$ の転置に相当するプログラム (アジョイントコード) が必要になる．一般のシミュレーションモデルでこれを開発するのは，大きな手間になる．
  - 降下法を使うため，局所解に陥る可能性がある．また，適切な前処理をしないと収束が遅くなる場合がある．

弱拘束の場合のアジョイント法

$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1})$ が成り立たない弱拘束の場合

$$J = \frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)^{\mathsf{T}} \mathbf{P}_b^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) + \frac{1}{2}\sum_{k=1}^{K}\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\Big)^{\mathsf{T}} \mathbf{R}_k^{-1}\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\Big)$$
$$+ \frac{1}{2}\sum_{k=1}^{K}\Big(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1})\Big)^{\mathsf{T}} \mathbf{Q}_k^{-1}\Big(\boldsymbol{x}_k - \boldsymbol{f}_k(\boldsymbol{x}_{k-1})\Big) \tag{7}$$

であっても，アジョイント法は使える．

弱拘束の場合は，$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) + \boldsymbol{v}_k$ より，J を

$$J = \frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)^{\mathsf{T}} \mathbf{P}_b^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) + \frac{1}{2}\sum_{k=1}^{K}\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\Big)^{\mathsf{T}} \mathbf{R}_k^{-1}\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\Big)$$
$$+ \frac{1}{2}\sum_{k=1}^{K}\boldsymbol{v}_k^{\mathsf{T}}\mathbf{Q}_k^{-1}\boldsymbol{v}_k$$

のように書き換え，$\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_K$ も同時に推定する．

アジョイント法のもう一つの導出

今，拘束条件 $\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1})$ のもとで目的関数 J_s の勾配を求めたが，アジョイント法のアルゴリズムは，そのまま目的関数の微分を取って導出することもできる．目的関数を明示的に $\boldsymbol{x}_0$ の関数の形で書くために

$$\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) = \boldsymbol{f}_k(\boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-2})) = \cdots = \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{x}_0)$$

を用いると，

$$\begin{aligned} J_s &= \frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)^{\mathsf{T}} \mathbf{P}_b^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) + \frac{1}{2}\sum_{k=1}^{K}\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\Big)^{\mathsf{T}} \mathbf{R}_k^{-1}\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\Big) \\ &= \frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)^{\mathsf{T}} \mathbf{P}_b^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) \\ &\quad + \frac{1}{2}\sum_{k=1}^{K}\left(\boldsymbol{y}_k - \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{x}_0)\right)^{\mathsf{T}} \mathbf{R}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{x}_0)\right). \end{aligned} \tag{35}$$

アジョイント法のもう一つの導出

ここで関数 $\boldsymbol{g}_k$ を

$$\boldsymbol{g}_k(\boldsymbol{x}_0) = \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{x}_0) \tag{36}$$

と定義すると，

$$J_s = \frac{1}{2}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right)^{\mathsf{T}} \mathbf{P}_b^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) + \frac{1}{2}\sum_{k=1}^{K}\left(\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)\right)^{\mathsf{T}} \mathbf{R}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)\right)$$

と書ける．J_s の勾配を取ると，関数 $\boldsymbol{g}_k$ の $\boldsymbol{x}_0$ におけるヤコビ行列を $\tilde{\mathbf{G}}_{k,\boldsymbol{x}_0}$ として，

$$\nabla_{\boldsymbol{x}_0} J_s = \mathbf{P}_b^{-1}\left(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b\right) - \sum_{k=1}^{K} \tilde{\mathbf{G}}_{k,\boldsymbol{x}_0}^{\mathsf{T}} \mathbf{R}_k^{-1}\left(\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)\right). \tag{37}$$

アジョイント法のもう一つの導出

今，関数 g_k のヤコビ行列 $\tilde{\mathrm{G}}_{k,x_0}$ を考えたが，関数 g_k は元々

$$g_k(x_0) = h_k(f_k(\cdots f_1(x_0)\cdots) = h_k \circ f_k \circ \cdots \circ f_1(x_0)$$

のように関数 h_k と k 個の f の合成関数となっていた．

合成関数の微分は，基本的には連鎖率

$$\frac{d}{dx}(f(g(x)) = \frac{df}{dg}\frac{dg}{dx}$$

を用いればよいが，ベクトル値関数なので，まずは $h_k \circ f_k(x_{k-1})$ のヤコビ行列を確認しておく．

合成関数の微分

$\boldsymbol{h}_k(\boldsymbol{x}_k) = \boldsymbol{h}_k(\boldsymbol{f}_k(\boldsymbol{x}_{k-1}))$ を $\boldsymbol{x}_{k-1}$ の各要素で微分すると，

$$\frac{\partial \boldsymbol{h}_k}{\partial x_{k-1,i}} = \sum_j \frac{\partial \boldsymbol{h}_k}{\partial x_{k,j}} \frac{\partial f_{k,j}}{\partial x_{k-1,i}}$$

なので，$\boldsymbol{h}_k \circ \boldsymbol{f}_k$ のヤコビ行列は，

$$\frac{\partial (\boldsymbol{h}_k \circ \boldsymbol{f}_k)}{\partial \boldsymbol{x}_{k-1}^{\mathsf{T}}} = \begin{pmatrix} \frac{\partial h_{k,1}}{\partial x_{k-1,1}} & \cdots & \frac{\partial h_{k,1}}{\partial x_{k-1,m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{k,n}}{\partial x_{k-1,1}} & \cdots & \frac{\partial h_{k,n}}{\partial x_{k-1,m}} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{\partial h_{k,1}}{\partial x_{k,1}} & \cdots & \frac{\partial h_{k,1}}{\partial x_{k,m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{k,n}}{\partial x_{k,1}} & \cdots & \frac{\partial h_{k,n}}{\partial x_{k,m}} \end{pmatrix} \begin{pmatrix} \frac{\partial f_{k,1}}{\partial x_{k-1,1}} & \cdots & \frac{\partial f_{k,1}}{\partial x_{k-1,m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{k,m}}{\partial x_{k-1,1}} & \cdots & \frac{\partial f_{k,m}}{\partial x_{k-1,m}} \end{pmatrix} = \tilde{\mathbf{H}}_{k,\boldsymbol{x}_k} \tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1}}$$

となる．$\tilde{\mathbf{H}}_{k,\boldsymbol{x}_k}$, $\tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1}}$ それぞれ関数 $\boldsymbol{h}_k(\cdot)$, $\boldsymbol{f}_k(\cdot)$ のヤコビ行列であり，その積が $\boldsymbol{h}_k \circ \boldsymbol{f}_k$ のヤコビ行列になることが分かる．

評価関数の微分

同様の計算を繰り返すと，合成関数 $\boldsymbol{g}_k(\boldsymbol{x}_0) = \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \boldsymbol{f}_{k-1} \cdots \circ \boldsymbol{f}_1(\boldsymbol{x}_0)$ のヤコビ行列 $\tilde{\mathbf{G}}_{k,\boldsymbol{x}_0}$ は

$$\tilde{\mathbf{G}}_{k,\boldsymbol{x}_0} = \frac{\partial(\boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \boldsymbol{f}_{k-1} \cdots \circ \boldsymbol{f}_1)}{\partial \boldsymbol{x}_0^{\mathsf{T}}} = \tilde{\mathbf{H}}_{k,\boldsymbol{x}_k} \tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1}} \cdots \tilde{\mathbf{F}}_{1,\boldsymbol{x}_0} \tag{38}$$

となる．

したがって，

$$\nabla_{\boldsymbol{x}_0} J_s = \mathbf{P}_b^{-1} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b) - \sum_{k=1}^{K} \tilde{\mathbf{F}}_{1,\boldsymbol{x}_0}^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1}}^{\mathsf{T}} \tilde{\mathbf{H}}_{k,\boldsymbol{x}_k}^{\mathsf{T}} \mathbf{R}^{-1} (\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_0)) . \tag{39}$$

評価関数の微分

これで $\nabla_{\boldsymbol{x}_0} J_s$ は得られるが，$\tilde{\mathbf{F}}^{\mathsf{T}}$ の乗算をまとめると効率的である．

$$\nabla_{\boldsymbol{x}_0} J_s = \mathbf{P}_b^{-1}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b) - \sum_{k=1}^{K} \tilde{\mathbf{F}}_{1,\boldsymbol{x}_0}^{\mathsf{T}} \tilde{\mathbf{F}}_{2,\boldsymbol{x}_1}^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1}}^{\mathsf{T}} \tilde{\mathbf{H}}_{k,\boldsymbol{x}_0}^{\mathsf{T}} \mathbf{R}_k^{-1} \Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \Big)$$

の右辺第二項の $k = k'$ 以降の和を取り出すと

$$\begin{aligned}
&\sum_{k=k'}^{K} \tilde{\mathbf{F}}_{1,\boldsymbol{x}_0}^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1}}^{\mathsf{T}} \tilde{\mathbf{H}}_{k,\boldsymbol{x}_0}^{\mathsf{T}} \mathbf{R}_k^{-1} \Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \Big) \\
&= \tilde{\mathbf{F}}_{1,\boldsymbol{x}_0}^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_{k',\boldsymbol{x}_{k'-1}}^{\mathsf{T}} \Bigg[\tilde{\mathbf{H}}_{k',\boldsymbol{x}_{k'}}^{\mathsf{T}} \mathbf{R}_{k'}^{-1} \Big(\boldsymbol{y}_{k'} - \boldsymbol{h}_{k'}(\boldsymbol{x}_{k'}) \Big) \\
&\qquad + \sum_{k=k'+1}^{K} \tilde{\mathbf{F}}_{k'+1,\boldsymbol{x}_{k'}}^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1}}^{\mathsf{T}} \tilde{\mathbf{H}}_{k,\boldsymbol{x}_k}^{\mathsf{T}} \mathbf{R}_k^{-1} \Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \Big) \Bigg]
\end{aligned} \tag{40}$$

のようになる．

評価関数の微分

ここで，$k' = 1, \ldots, K-1$ に対して

$$\boldsymbol{\lambda}_{k'} = \tilde{\mathbf{H}}_{k',\boldsymbol{x}_{k'}}^{\mathsf{T}} \mathbf{R}_{k'}^{-1} \Big(\boldsymbol{y}_{k'} - \boldsymbol{h}_{k'}(\boldsymbol{x}_{k'}) \Big) + \sum_{k=k'+1}^{K} \tilde{\mathbf{F}}_{k'+1,\boldsymbol{x}_{k'}}^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1}}^{\mathsf{T}} \tilde{\mathbf{H}}_{k,\boldsymbol{x}_k}^{\mathsf{T}} \mathbf{R}_k^{-1} \Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \Big)$$

のように $\boldsymbol{\lambda}_{k'}$ を定義すると，

$$\begin{aligned}
\boldsymbol{\lambda}_{k'} &= \tilde{\mathbf{H}}_{k',\boldsymbol{x}_{k'}}^{\mathsf{T}} \mathbf{R}_{k'}^{-1} \Big(\boldsymbol{y}_{k'} - \boldsymbol{h}_{k'}(\boldsymbol{x}_{k'}) \Big) + \sum_{k=k'+1}^{K} \tilde{\mathbf{F}}_{k'+1,\boldsymbol{x}_{k'}}^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1}}^{\mathsf{T}} \tilde{\mathbf{H}}_{k,\boldsymbol{x}_k}^{\mathsf{T}} \mathbf{R}_k^{-1} \Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \Big) \\
&= \tilde{\mathbf{H}}_{k',\boldsymbol{x}_{k'}}^{\mathsf{T}} \mathbf{R}_{k'}^{-1} \Big(\boldsymbol{y}_{k'} - \boldsymbol{h}_{k'}(\boldsymbol{x}_{k'}) \Big) + \tilde{\mathbf{F}}_{k'+1,\boldsymbol{x}_{k'}}^{\mathsf{T}} \Bigg[\tilde{\mathbf{H}}_{k'+1,\boldsymbol{x}_{k'+1}}^{\mathsf{T}} \mathbf{R}_{k'+1}^{-1} \Big(\boldsymbol{y}_{k'+1} - \boldsymbol{h}_{k'+1}(\boldsymbol{x}_{k'+1}) \Big) \\
&\qquad + \sum_{k=k'+2}^{K} \tilde{\mathbf{F}}_{k'+2,\boldsymbol{x}_{k'+1}}^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_{k,\boldsymbol{x}_{k-1}}^{\mathsf{T}} \tilde{\mathbf{H}}_{k,\boldsymbol{x}_k}^{\mathsf{T}} \mathbf{R}_k^{-1} \Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \Big) \Bigg]
\end{aligned}$$

したがって，以下のアジョイント法の式が得られる:

$$\boldsymbol{\lambda}_{k'} = \tilde{\mathbf{H}}_{k',\boldsymbol{x}_{k'}}^{\mathsf{T}} \mathbf{R}_{k'}^{-1} \Big(\boldsymbol{y}_{k'} - \boldsymbol{h}_{k'}(\boldsymbol{x}_{k'}) \Big) + \tilde{\mathbf{F}}_{k'+1,\boldsymbol{x}_{k'}}^{\mathsf{T}} \boldsymbol{\lambda}_{k'+1}. \tag{41}$$

アジョイント法

$\boldsymbol{\lambda}_K$ を与えると，漸化式

$$\boldsymbol{\lambda}_{k'} = \tilde{\mathbf{H}}^{\mathsf{T}}_{k',\boldsymbol{x}_{k'}} \mathbf{R}^{-1}_{k'} \Big(\boldsymbol{y}_{k'} - \boldsymbol{h}_{k'}(\boldsymbol{x}_{k'}) \Big) + \tilde{\mathbf{F}}^{\mathsf{T}}_{k'+1,\boldsymbol{x}_{k'}} \boldsymbol{\lambda}_{k'+1} \tag{41}$$

を $k' = K-1, K-2, \ldots$ と順次適用することで $\boldsymbol{\lambda}_K$ から $\boldsymbol{\lambda}_1$ まで求まる．なお，

$$\begin{aligned}
&\sum_{k=K-1}^{K} \tilde{\mathbf{F}}^{\mathsf{T}}_{1,\boldsymbol{x}_0} \cdots \tilde{\mathbf{F}}^{\mathsf{T}}_{k,\boldsymbol{x}_{k-1}} \tilde{\mathbf{H}}^{\mathsf{T}}_{k,\boldsymbol{x}_k} \mathbf{R}^{-1}_k \Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \Big) \\
&= \tilde{\mathbf{F}}^{\mathsf{T}}_{1,\boldsymbol{x}_0} \cdots \tilde{\mathbf{F}}^{\mathsf{T}}_{K-1,\boldsymbol{x}_{K-2}} \Bigg[\tilde{\mathbf{H}}^{\mathsf{T}}_{K-1,\boldsymbol{x}_{K-1}} \mathbf{R}^{-1}_{K-1} \Big(\boldsymbol{y}_{K-1} - \boldsymbol{h}_{K-1}(\boldsymbol{x}_{K-1}) \Big) \\
&\qquad + \tilde{\mathbf{F}}^{\mathsf{T}}_{K,\boldsymbol{x}_{K-1}} \tilde{\mathbf{H}}^{\mathsf{T}}_{K,\boldsymbol{x}_K} \mathbf{R}^{-1}_K \Big(\boldsymbol{y}_K - \boldsymbol{h}_K(\boldsymbol{x}_K) \Big) \Bigg]
\end{aligned}$$

なので，$\boldsymbol{\lambda}_K = \tilde{\mathbf{H}}^{\mathsf{T}}_{K,\boldsymbol{x}_K} \mathbf{R}^{-1}_K \Big(\boldsymbol{y}_K - \boldsymbol{h}_K(\boldsymbol{x}_K) \Big)$ とおけば，$k' = K-1$ でも式 (41) が成り立つ．

アジョイント法

$$\sum_{k=1}^{K} \tilde{\mathbf{F}}_1^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_k^{\mathsf{T}} \tilde{\mathbf{H}}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \Big)$$

$$= \tilde{\mathbf{F}}_1^{\mathsf{T}} \left[\tilde{\mathbf{H}}_1^{\mathsf{T}} \mathbf{R}_1^{-1} \Big(\boldsymbol{y}_1 - \boldsymbol{h}_1(\boldsymbol{x}_1) \Big) + \sum_{k=2}^{K} \tilde{\mathbf{F}}_2^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_k^{\mathsf{T}} \tilde{\mathbf{H}}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \Big) \right] = \tilde{\mathbf{F}}_1^{\mathsf{T}} \boldsymbol{\lambda}_1$$

となるので，$\boldsymbol{\lambda}_1$ が求まれば

$$\nabla_{\boldsymbol{x}_0} J_s = \mathbf{P}_b^{-1} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b) - \tilde{\mathbf{F}}_1^{\mathsf{T}} \boldsymbol{\lambda}_1 \tag{42}$$

のように $\nabla_{\boldsymbol{x}_0} J_s$ も得られる．

以上のようにして，J_s の勾配からアジョイント法の手続きを導出することもできる．

アジョイントモデルの作成例

アジョイントモデルの作り方を示すため，修正オイラー法で，以下の単振り子方程式を解くモデルを例として取り上げる:

$$\frac{d^2\phi}{dt^2} = -\alpha^2 \sin\phi. \tag{43}$$

を考える．

$\dot{\phi} = d\phi/dt$ とおくと

$$\frac{d\phi}{dt} = \dot{\phi}, \qquad \frac{d\dot{\phi}}{dt} = -\alpha^2 \sin\phi \tag{44}$$

となるので，これを修正オイラー法で解く．

修正オイラー法

修正オイラー法とは，

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{g}(\boldsymbol{x}, t) \tag{45}$$

の形で与えられた常微分方程式を数値的に解くために，時刻 t_k から $t_{k+1}(= t_k + \Delta t)$ の間の x の値の時間発展を以下のように計算する方法である:

$$\boldsymbol{x}_{k+1/2} = \boldsymbol{x}_k + \frac{1}{2}\boldsymbol{g}(\boldsymbol{x}_k, t_k)\Delta t, \tag{46}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{g}\left(\boldsymbol{x}_{k+1/2}, t_k + \frac{\Delta t}{2}\right)\Delta t. \tag{47}$$

2 次の Runge-Kutta 法に相当.

単振動モデル

修正オイラー法で単振動の方程式を解くには，各時間ステップ k において

$$\phi_{k+1/2} = \phi_k + \frac{1}{2}\dot{\phi}_k \Delta t, \tag{48a}$$

$$\dot{\phi}_{k+1/2} = \dot{\phi}_k - \frac{1}{2}\alpha^2 (\sin \phi_k) \Delta t \tag{48b}$$

を求めた上で，

$$\phi_{k+1} = \phi_k + \dot{\phi}_{k+1/2} \Delta t, \tag{49a}$$

$$\dot{\phi}_{k+1} = \dot{\phi}_k - \alpha^2 (\sin \phi_{k+1/2}) \Delta t \tag{49b}$$

を計算する．
ここで，式 (49a), (49b) に式 (48a), (48b) を代入すると，

$$\phi_{k+1} = \phi_k - \frac{\alpha^2 \Delta t^2}{2} \sin \phi_k + \dot{\phi}_k \Delta t, \tag{50a}$$

$$\dot{\phi}_{k+1} = \dot{\phi}_k - \left(\alpha^2 \Delta t\right) \sin \left(\phi_k + \frac{\Delta t}{2} \dot{\phi}_k \right). \tag{50b}$$

システムモデル

今の式の時間ステップを 1 つずらして，

$$\phi_k = \phi_{k-1} - \frac{\alpha^2 \Delta t^2}{2} \sin \phi_{k-1} + \dot{\phi}_{k-1} \Delta t,$$
$$\dot{\phi}_k = \dot{\phi}_{k-1} - \left(\alpha^2 \Delta t\right) \sin \left(\phi_{k-1} + \frac{\Delta t}{2} \dot{\phi}_{k-1} \right).$$

これをシステムモデル $\boldsymbol{x}_k = \boldsymbol{f}_k(\boldsymbol{x}_{k-1})$ の形に書き換えるには，

$$\boldsymbol{x}_k = \begin{pmatrix} \phi_k \\ \dot{\phi}_k \end{pmatrix}, \quad \boldsymbol{f}(\boldsymbol{x}_{k-1}) = \begin{pmatrix} \phi_{k-1} - \frac{\alpha^2 \Delta t^2}{2} \sin \phi_{k-1} + \dot{\phi}_{k-1} \Delta t \\ \dot{\phi}_{k-1} - \left(\alpha^2 \Delta t\right) \sin \left(\phi_{k-1} + \frac{\Delta t}{2} \dot{\phi}_{k-1} \right) \end{pmatrix} \tag{51}$$

とおけばよい．

システムモデルのヤコビ行列

$$\boldsymbol{x}_k = \begin{pmatrix} \phi_k \\ \dot{\phi}_k \end{pmatrix}, \quad \boldsymbol{f}_k(\boldsymbol{x}_{k-1}) = \begin{pmatrix} \phi_{k-1} - \frac{\alpha^2 \Delta t^2}{2} \sin \phi_{k-1} + \dot{\phi}_{k-1} \Delta t \\ \dot{\phi}_{k-1} - \left(\alpha^2 \Delta t\right) \sin \left(\phi_{k-1} + \frac{\Delta t}{2} \dot{\phi}_{k-1}\right) \end{pmatrix}$$

とおいたとき，関数 $\boldsymbol{f}_k$ の $\boldsymbol{x}_{k-1}$ におけるヤコビ行列を求めると

$$\frac{\partial \phi_k}{\partial \phi_{k-1}} = 1 - \frac{\alpha^2 \Delta t^2}{2} \cos \phi_{k-1}, \qquad \frac{\partial \phi_k}{\partial \dot{\phi}_{k-1}} = \Delta t,$$

$$\frac{\partial \dot{\phi}_k}{\partial \phi_{k-1}} = -(\alpha^2 \Delta t) \cos \left(\phi_{k-1} + \frac{\Delta t}{2} \dot{\phi}_{k-1}\right),$$

$$\frac{\partial \dot{\phi}_k}{\partial \dot{\phi}_{k-1}} = 1 - \frac{\alpha^2 \Delta t^2}{2} \cos \left(\phi_{k-1} + \frac{\Delta t}{2} \dot{\phi}_{k-1}\right)$$

なので，

$$\tilde{\mathbf{F}}_{\boldsymbol{x}_k} = \begin{pmatrix} 1 - \frac{\alpha^2 \Delta t^2}{2} \cos \phi_{k-1} & \Delta t \\ -(\alpha^2 \Delta t) \cos \left(\phi_{k-1} + \frac{\Delta t}{2} \dot{\phi}_{k-1}\right) & 1 - \frac{\alpha^2 \Delta t^2}{2} \cos \left(\phi_{k-1} + \frac{\Delta t}{2} \dot{\phi}_{k-1}\right) \end{pmatrix}. \tag{52}$$

アジョイントモデル

したがって，アジョイントモデルは，

$$\tilde{\mathbf{F}}_{\boldsymbol{x}_k}^{\mathsf{T}} = \begin{pmatrix} 1 - \frac{\alpha^2 \Delta t^2}{2} \cos\phi_{k-1} & -(\alpha^2 \Delta t) \cos\left(\phi_{k-1} + \frac{\Delta t}{2}\dot{\phi}_{k-1}\right) \\ \Delta t & 1 - \frac{\alpha^2 \Delta t^2}{2} \cos\left(\phi_{k-1} + \frac{\Delta t}{2}\dot{\phi}_{k-1}\right) \end{pmatrix}. \tag{53}$$

一般に非線型モデルでは，アジョイントモデルに前の時間ステップの状態変数 $\boldsymbol{x}_{k-1}$ の値が含まれる．$\boldsymbol{x}_{k-1}$ の値には，その時点で得られている x_0 の推定値からスタートさせたシミュレーションの実行結果を入れる．つまり，アジョイント法で $\nabla_{\boldsymbol{x}_0} J_s$ を求める際には，シミュレーションを実行した結果得られる時間ステップごとの x_k の値を保存しておく必要がある．

変数変換

ここで，分散共分散行列を $\mathbf{P}_b = \mathbf{X}_b \mathbf{X}_b^{\mathsf{T}}$ のように分解して $\boldsymbol{x}_0$ を

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}}_b + \mathbf{X}_b \boldsymbol{w} \tag{54}$$

のように書き直してみる．このときの $\mathbf{X}_b$ は，固有値分解で作ってもよいし，コレスキー分解を用いてもよい．

行列 $\mathbf{P}_b$ が正則なら，$\mathbf{X}_b$ も正則で，

$$\mathbf{P}_b^{-1} = (\mathbf{X}_b \mathbf{X}_b^{\mathsf{T}})^{-1} = \mathbf{X}_b^{\mathsf{T}\,-1} \mathbf{X}_b^{-1}. \tag{55}$$

変数変換

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}}_b + \mathbf{X}_b \boldsymbol{w}$$

としたとき，評価関数

$$J_s(\boldsymbol{x}_0) = \frac{1}{2}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b)^{\mathsf{T}} \mathbf{P}_b^{-1} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b) + \frac{1}{2} \sum_{k=1}^{K} \left[\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right]^{\mathsf{T}} \mathbf{R}_k^{-1} \left[\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right]$$

の右辺第 1 項は，

$$\begin{aligned} \frac{1}{2}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b)^{\mathsf{T}} \mathbf{P}_b^{-1} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b) &= \frac{1}{2} \boldsymbol{w}^{\mathsf{T}} \mathbf{X}_b^{\mathsf{T}} \mathbf{P}_b^{-1} \mathbf{X}_b \boldsymbol{w} = \frac{1}{2} \boldsymbol{w}^{\mathsf{T}} \mathbf{X}_b^{\mathsf{T}} \mathbf{X}_b^{\mathsf{T}\,-1} \mathbf{X}_b^{-1} \mathbf{X}_b \boldsymbol{w} \\ &= \frac{1}{2} \boldsymbol{w}^{\mathsf{T}} \boldsymbol{w} \end{aligned} \tag{56}$$

となる．

変数変換

$$\boldsymbol{h}_k(\boldsymbol{x}_k) = \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{x}_0) = \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\bar{\boldsymbol{x}}_b + \mathbf{X}_b \boldsymbol{w})$$

より，評価関数は $\boldsymbol{w}$ の関数

$$\begin{aligned} J_s(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^{\mathsf{T}}\boldsymbol{w} + \frac{1}{2}\sum_{k=1}^{K} &\left[\boldsymbol{y}_k - \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\bar{\boldsymbol{x}}_b + \mathbf{X}_b \boldsymbol{w})\right]^{\mathsf{T}} \mathbf{R}_k^{-1} \\ &\times \left[\boldsymbol{y}_k - \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\bar{\boldsymbol{x}}_b + \mathbf{X}_b \boldsymbol{w})\right] \end{aligned} \tag{57}$$

の形に書ける．

$J_s(\boldsymbol{w})$ の勾配 $\nabla_{\boldsymbol{w}} J_s$ は，

$$\frac{\partial(\bar{\boldsymbol{x}}_b + \mathbf{X}_b \boldsymbol{w})}{\partial \boldsymbol{w}^{\mathsf{T}}} = \mathbf{X}_b$$

となることを考慮すれば，

$$\nabla_{\boldsymbol{w}} J_s = \boldsymbol{w} - \sum_{k=1}^{K} \mathbf{X}_b^{\mathsf{T}} \tilde{\mathbf{F}}_1^{\mathsf{T}} \tilde{\mathbf{F}}_2^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_k^{\mathsf{T}} \tilde{\mathbf{H}}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \left[\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right]. \tag{58}$$

変数変換

$$\nabla_{\boldsymbol{w}} J_s = \boldsymbol{w} - \sum_{k=1}^{K} \mathbf{X}_b^{\mathsf{T}} \tilde{\mathbf{F}}_1^{\mathsf{T}} \tilde{\mathbf{F}}_2^{\mathsf{T}} \cdots \tilde{\mathbf{F}}_k^{\mathsf{T}} \tilde{\mathbf{H}}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \left[\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k) \right].$$

は，先程のアジョイント法の手続きの最後で，

$$\nabla_{\boldsymbol{x}_0} J_s = \mathbf{P}_b^{-1} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b) - \tilde{\mathbf{F}}_1^{\mathsf{T}} \boldsymbol{\lambda}_1$$

を計算する代わりに

$$\nabla_{\boldsymbol{w}} J_s = \boldsymbol{w} - \mathbf{X}_b^{\mathsf{T}} \tilde{\mathbf{F}}_1^{\mathsf{T}} \boldsymbol{\lambda}_1 \tag{59}$$

のようにすれば得られる．降下法で J_s を最小化していく際には，この形を用いた方が収束が速くなることが多い．

4 次元変分法

- アジョイント法
  - Forward と backward の計算を何回か繰り返す.
  - Backward の計算にアジョイントコード (foward モデルの転置行列に相当) が必要.
- アンサンブル 4 次元変分法 (4-dimensional Ensemble Variational method; 4DEnVar)
  - Forward のアンサンブル計算 (数十〜数百程度の異なる設定による計算) を何回か繰り返す (非線型性が強くなければ 1 回でよいかもしれない).
  - アジョイントコードが不要.

アンサンブルによる方法

- $\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$ を $\boldsymbol{x}_0$ の事前分布 $\mathcal{N}(\bar{\boldsymbol{x}}_b, \mathbf{P}_b)$ に従う N 個の独立な乱数とする．
- N 個の異なる初期設定 $\{\boldsymbol{x}_0^{(1)}, \ldots, \boldsymbol{x}_0^{(N)}\}$ のそれぞれに対してシミュレーションを

$$\boldsymbol{x}_k^{(i)} = \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{x}_0^{(i)})$$

のように実行し，出力 $\{\boldsymbol{x}_{1:K}^{(1)}, \ldots, \boldsymbol{x}_{1:K}^{(N)}\}$ を得たとする．

但し，$\boldsymbol{x}_{1:K}^{(i)} = \{\boldsymbol{x}_1^{(i)}, \ldots, \boldsymbol{x}_K^{(i)}\}$　(各時刻の出力を合わせたもの)．

- 初期値 $\boldsymbol{x}_0$ を $\{\boldsymbol{x}_0^{(i)}\}$ の線型結合

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}}_b + \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w^{(i)} (\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_b) \tag{60}$$

で表せると仮定する．

アンサンブルによる方法

行列 $\mathbf{X}_b$ を

$$\mathbf{X}_b = \frac{1}{\sqrt{N}} \left(\begin{array}{ccc} \boldsymbol{x}_0^{(1)} - \bar{\boldsymbol{x}}_b & \cdots & \boldsymbol{x}_0^{(N)} - \bar{\boldsymbol{x}}_b \end{array} \right) \tag{61}$$

と定義すると，

$$\boldsymbol{x}_0 = \bar{\boldsymbol{x}}_b + \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w^{(i)} (\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_b) = \bar{\boldsymbol{x}}_b + \mathbf{X}_b \boldsymbol{w}. \tag{62}$$

なお，$\{\boldsymbol{x}_0^{(i)}\}_{i=1}^N$ は，$\mathcal{N}(\bar{\boldsymbol{x}}_b, \mathbf{P}_b)$ に従うので，

$$\mathbf{X}_b \mathbf{X}_b^{\mathsf{T}} = \frac{1}{N} \sum_{i=1}^{N} \left(\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_b \right) \left(\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_b \right)^{\mathsf{T}} \approx \mathbf{P}_b. \tag{63}$$

逆に $\boldsymbol{w}$ が標準正規分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ に従うとすると，$\boldsymbol{x}_0 = \bar{\boldsymbol{x}}_b + \mathbf{X}_b \boldsymbol{w}$ で得られる $\boldsymbol{x}_0$ は，近似的に $\boldsymbol{x}_0$ の事前分布 $\mathcal{N}(\bar{\boldsymbol{x}}_b, \mathbf{P}_b)$ に従う．

評価関数

$\boldsymbol{x}_0 = \bar{\boldsymbol{x}}_b + \mathbf{X}_b \boldsymbol{w}$ とした上で，改めて評価関数

$$J(\boldsymbol{x}_0) = \frac{1}{2}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b)^{\mathsf{T}} \mathbf{P}_b^{-1} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b) + \frac{1}{2} \sum_{k=1}^{K} \left[\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right]^{\mathsf{T}} \mathbf{R}_k^{-1} \left[\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{x}_k)\right]$$

の最小化を考える．

まず，$\mathbf{X}_b \mathbf{X}_b^{\mathsf{T}} \approx \mathbf{P}_b$ を利用すると，

$$\frac{1}{2}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b)^{\mathsf{T}} \mathbf{P}_b^{-1} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b) \approx \frac{1}{2} \boldsymbol{w}^{\mathsf{T}} \boldsymbol{w} \tag{64}$$

が言える．

線型近似

$\boldsymbol{h}_k(\boldsymbol{x}_k) = \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{x}_0)$ だが，$\boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1$ をまとめて

$$\boldsymbol{g}_k(\boldsymbol{x}_0) = \boldsymbol{h}_k \circ \boldsymbol{f}_k \circ \cdots \circ \boldsymbol{f}_1(\boldsymbol{x}_0) \tag{65}$$

とおくことにする．さらにこの $\boldsymbol{g}_k(\boldsymbol{x}_0)$ を線型近似する:

$$\boldsymbol{g}_k(\boldsymbol{x}_0) \approx \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) + \tilde{\mathbf{G}}_k(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_b), \quad \left(\tilde{\mathbf{G}}_k = \left. \frac{\partial \boldsymbol{g}_k}{\partial \boldsymbol{x}_0^{\mathsf{T}}} \right|_{\boldsymbol{x}_0 = \bar{\boldsymbol{x}}_b} \right). \tag{66}$$

以上から，評価関数 J は以下のような $\boldsymbol{w}$ の関数 $\hat{J}(\boldsymbol{w})$ で近似できる:

$$\hat{J}(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^{\mathsf{T}}\boldsymbol{w} + \frac{1}{2}\sum_{k=1}^{K} \left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) - \tilde{\mathbf{G}}_k \mathbf{X}_b \boldsymbol{w} \right]^{\mathsf{T}} \mathbf{R}_k^{-1} \left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) - \tilde{\mathbf{G}}_k \mathbf{X}_b \boldsymbol{w} \right]. \tag{67}$$

4 次元アンサンブル変分法

$\hat{J}(\boldsymbol{w})$ の微分をとって

$$\nabla_{\boldsymbol{w}} \hat{J} = \boldsymbol{w} - \sum_{k=1}^{K} \left[\tilde{\mathbf{G}}_k \mathbf{X}_b\right]^{\mathsf{T}} \mathbf{R}_k^{-1} \left[\boldsymbol{y}_k - \boldsymbol{g}_k(\boldsymbol{x}_b) - \tilde{\mathbf{G}}_k \mathbf{X}_b \boldsymbol{w}\right]. \tag{68}$$

評価関数 $\hat{J}$ の $\boldsymbol{w}$ についての勾配 $\nabla_{\boldsymbol{w}} \hat{J}$ が得られれば，降下法によって最適値 $\hat{\boldsymbol{w}}$ が求まる．

$\longrightarrow$ さらに $\hat{\boldsymbol{x}}_0 = \bar{\boldsymbol{x}}_b + \mathbf{X}_b \hat{\boldsymbol{w}}$ から $\boldsymbol{x}_0$ の最適値 (の近似) $\hat{\boldsymbol{x}}_0$ も求まる．

実際の計算方法

- $\tilde{\mathbf{G}}_k\mathbf{X}_b$ の部分は，$\tilde{\mathbf{G}}_k(\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_b) \approx \boldsymbol{g}_k(\boldsymbol{x}_0^{(i)}) - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b)$ となることから，

$$\tilde{\mathbf{G}}_k\mathbf{X}_b \approx \frac{1}{\sqrt{N}}\left(\boldsymbol{g}_k(\boldsymbol{x}_0^{(1)}) - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) \quad \cdots \quad \boldsymbol{g}_k(\boldsymbol{x}_0^{(N)}) - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) \right) := \boldsymbol{\Gamma}_k. \tag{69}$$

で近似できる．結局，関数 g の微分は計算しなくてもよい．

- また，線型近似を用いると，

$$\begin{aligned} \langle \boldsymbol{g}_k(\boldsymbol{x}_0) \rangle &= \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{g}_k(\boldsymbol{x}_0^{(i)}) \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \left(\boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) + \tilde{\mathbf{G}}_k(\boldsymbol{x}_0^{(i)} - \bar{\boldsymbol{x}}_b) \right) \approx \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) \end{aligned} \tag{70}$$

なので，$\boldsymbol{g}_k(\bar{\boldsymbol{x}}_b)$ を $\langle \boldsymbol{g}_k(\boldsymbol{x}_0) \rangle$ で代用してもよい．この場合，シミュレーションを 1 回省略できる．

4 次元アンサンブル変分法のアルゴリズム

1. N 個の異なる初期値 $\{\boldsymbol{x}_0^{(1)}, \ldots, \boldsymbol{x}_0^{(N)}\}$ に対してシミュレーションを実行し，各 i について各時間ステップの観測の予測値 $\boldsymbol{g}_k(\boldsymbol{x}_0^{(i)})$ $(k = 1, \ldots, K)$ を得る．
2. 次に

$$\boldsymbol{\Gamma}_k = \frac{1}{\sqrt{N}} \left(\begin{array}{ccc} \boldsymbol{g}_k(\boldsymbol{x}_0^{(1)}) - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) & \cdots & \boldsymbol{g}_k(\boldsymbol{x}_0^{(N)}) - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) \end{array} \right)$$

を求める．
3. 適当な初期値 $\boldsymbol{w}$ を与え，勾配

$$\nabla_{\boldsymbol{w}} \hat{J} = \boldsymbol{w} - \sum_{k=1}^{K} \boldsymbol{\Gamma}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) - \boldsymbol{\Gamma}_k \boldsymbol{w} \right]$$

を使って，$\hat{J}$ を最小にする $\hat{w}$ を求める．

解き方について

ここで使用している評価関数の近似

$$\hat{J}(\boldsymbol{w}) \approx \frac{1}{2}\boldsymbol{w}^{\mathsf{T}}\boldsymbol{w} + \frac{1}{2}\sum_{k=1}^{K}\left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) - \boldsymbol{\Gamma}_k\boldsymbol{w}\right]^{\mathsf{T}}\mathbf{R}_k^{-1}\left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) - \boldsymbol{\Gamma}_k\boldsymbol{w}\right].$$

が，w に対して 2 次関数になっていることに注意すると，この近似評価関数の下での最適な w は，$\mathbf{I}$ を単位行列として

$$\hat{\boldsymbol{w}} = \left(\sum_k[\boldsymbol{\Gamma}_k^{\mathsf{T}}\mathbf{R}_k^{-1}\boldsymbol{\Gamma}_k] + \mathbf{I}\right)^{-1}\sum_k\left(\boldsymbol{\Gamma}_k^{\mathsf{T}}\mathbf{R}_k^{-1}\left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b)\right]\right) \tag{71}$$

と直ちに求まる．

∗ ベイズの定理に基づく事後分布の計算で，

$$\frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{x}}_b)^{\mathsf{T}}\mathbf{P}_b^{-1}(\boldsymbol{x}-\bar{\boldsymbol{x}}_b) + \frac{1}{2}\left(\boldsymbol{y}-\mathbf{H}\boldsymbol{x}\right)^{\mathsf{T}}\mathbf{R}^{-1}\left(\boldsymbol{y}-\mathbf{H}\boldsymbol{x}\right)$$

を最小化する $\hat{x}$ が以下だったことに注意．

$$\hat{\boldsymbol{x}} = \bar{\boldsymbol{x}}_b + (\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}_b^{-1})^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}(\boldsymbol{y} - \mathbf{H}\bar{\boldsymbol{x}}_b).$$

解き方について

$$\hat{J}(\boldsymbol{w}) \approx \frac{1}{2}\boldsymbol{w}^{\mathsf{T}}\boldsymbol{w} + \frac{1}{2}\sum_{k=1}^{K}\left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) - \boldsymbol{\Gamma}_k\boldsymbol{w}\right]^{\mathsf{T}}\mathbf{R}_k^{-1}\left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) - \boldsymbol{\Gamma}_k\boldsymbol{w}\right]$$

という近似評価関数は，$\boldsymbol{x}^* = \boldsymbol{w}$ とし，さらに

$$\boldsymbol{y}^* = \begin{pmatrix} \boldsymbol{y}_1 - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) \\ \vdots \\ \boldsymbol{y}_K - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) \end{pmatrix}, \boldsymbol{\Gamma}^* = \begin{pmatrix} \boldsymbol{\Gamma}_1 \\ \vdots \\ \boldsymbol{\Gamma}_K \end{pmatrix}, \mathbf{R}^* = \begin{pmatrix} \mathbf{R}_1 & & O \\ & \ddots & \\ O & & \mathbf{R}_K \end{pmatrix}$$

とおくと，

$$-\frac{1}{2}\boldsymbol{x}^{*\mathsf{T}}\boldsymbol{x}^* - \frac{1}{2}\left(\boldsymbol{y}^* - \boldsymbol{\Gamma}^*\boldsymbol{x}^*\right)^{\mathsf{T}}\mathbf{R}^{*-1}\left(\boldsymbol{y}^* - \boldsymbol{\Gamma}^*\boldsymbol{x}^*\right)$$

の形になる．

解き方について

$$\begin{pmatrix} \boldsymbol{\Gamma}_1^{\mathsf{T}} & \cdots & \boldsymbol{\Gamma}_K^{\mathsf{T}} \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{R}_K \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_K \end{pmatrix} = \sum_{k=1}^{K} \left(\boldsymbol{\Gamma}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \boldsymbol{y}_k \right),$$

$$\begin{pmatrix} \boldsymbol{\Gamma}_1^{\mathsf{T}} & \cdots & \boldsymbol{\Gamma}_K^{\mathsf{T}} \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{R}_K \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{\Gamma}_1 \\ \vdots \\ \boldsymbol{\Gamma}_K \end{pmatrix} = \sum_{k=1}^{K} \left(\boldsymbol{\Gamma}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \boldsymbol{\Gamma}_k \right)$$

などに注意すると，先程の式

$$\hat{\boldsymbol{w}} = \left(\sum_k [\boldsymbol{\Gamma}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \boldsymbol{\Gamma}_k] + \mathbf{I} \right)^{-1} \sum_k \left(\boldsymbol{\Gamma}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \left[\boldsymbol{y}_k - \boldsymbol{g}_k(\bar{\boldsymbol{x}}_b) \right] \right)$$

が求まる．$\left(\sum_k [\boldsymbol{\Gamma}_k^{\mathsf{T}} \mathbf{R}_k^{-1} \boldsymbol{\Gamma}_k] + \mathbf{I} \right)^{-1}$ の部分は，N 次の行列 (N はアンサンブル数) なので，逆行列の計算もさほど大変ではない．

解き方について

アンサンブル変分法は，1 回適用するだけでは $g(x)$ の線型近似に基づく解しか得られないが，以下のような繰り返しアルゴリズムも考えられる．

- j ステップ目において，以下を実行する．
  1. $j-1$ ステップ目の推定値 $\hat{\boldsymbol{x}}_{0,j-1}$ の周りに N 個の異なる初期値 $\{\boldsymbol{x}_{0,j-1}^{(1)}, \ldots, \boldsymbol{x}_{0,j-1}^{(N)}\}$ を以下が満たされるように生成する．

$$\hat{\boldsymbol{x}}_{0,j-1} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_{0,j-1}^{(i)}$$

  2. $\{\boldsymbol{x}_{0,j-1}^{(1)}, \ldots, \boldsymbol{x}_{0,j-1}^{(N)}\}$ のそれぞれ対してシミュレーションを実行し，各 i について各時間ステップの観測の予測値 $\boldsymbol{g}_k(\boldsymbol{x}_{0,j-1}^{(i)})$ $(k = 1, \ldots, K)$ を得る．
  3. 以下の行列を求める:

$$\boldsymbol{\Gamma}_{k,j-1} = \frac{1}{\sqrt{N}} \left(\begin{array}{ccc} \boldsymbol{g}_k(\boldsymbol{x}_{0,j-1}^{(1)}) - \boldsymbol{g}_k(\hat{\boldsymbol{x}}_{0,j-1}) & \cdots & \boldsymbol{g}_k(\boldsymbol{x}_{0,j-1}^{(N)}) - \boldsymbol{g}_k(\hat{\boldsymbol{x}}_{0,j-1}) \end{array} \right).$$

  4. 以下の式により，$\boldsymbol{w}$ の推定値を求める．

$$\hat{\boldsymbol{w}}_j = \left(\sum_k [\boldsymbol{\Gamma}_{k,j-1}^{\mathsf{T}} \mathbf{R}_k^{-1} \boldsymbol{\Gamma}_{k,j-1}] + \mathbf{I} \right)^{-1} \sum_k \left(\boldsymbol{\Gamma}_{k,j-1}^{\mathsf{T}} \mathbf{R}_k^{-1} \left[\boldsymbol{y}_k - \boldsymbol{g}_k(\hat{\boldsymbol{x}}_{0,j-1}) \right] \right).$$

  5. $\hat{\boldsymbol{x}}_{0,j} = \hat{\boldsymbol{x}}_{0,j-1} + \mathbf{X}_j \hat{\boldsymbol{w}}_j$ より，$\boldsymbol{x}_0$ の推定値を求める．

補足

先程の繰り返しアルゴリズムを適用した場合，レーベンバーグ・マーカート (Levenberg-Marquardt) 法と呼ばれる最適化手法の近似という解釈もできる．
レーベンバーグ・マーカート法は以下のニュートン法と関係している．

ニュートン法
$f(x)$ の最小にする x を求めるために以下を繰り返す:

$$\boldsymbol{x}_j = \boldsymbol{x}_{j-1} - \left(\nabla^2 f(\boldsymbol{x}_{j-1})\right)^{-1} \nabla f(\boldsymbol{x}_{j-1}) \tag{72}$$

但し，∇f は f の勾配, $\nabla^2 f$ は f のヘッセ行列で

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} \partial f/\partial x_1 \\ \vdots \\ \partial f/\partial x_m \end{pmatrix}, \quad \nabla^2 f(\boldsymbol{x}) = \begin{pmatrix} \partial^2 f/\partial x_1^2 & \cdots & \partial^2 f/(\partial x_1 \partial x_m) \\ \vdots & \ddots & \vdots \\ \partial^2 f/(\partial x_m \partial x_1) & \cdots & \partial^2 f/\partial x_m^2 \end{pmatrix} \tag{73}$$

補足

関連手法として，$f(\boldsymbol{x}) = (\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x}))^{\mathsf{T}} \mathbf{R}^{-1} (\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x}))$ の最小にする $\boldsymbol{x}$ を求めるために，ニュートン法の近似を適用するガウス・ニュートン法がある．

$f(\boldsymbol{x})$ の勾配は $\boldsymbol{g}$ のヤコビ行列を $\tilde{\mathbf{G}}$ とすると

$$\nabla f(\boldsymbol{x}) = -2\tilde{\mathbf{G}}^{\mathsf{T}} \mathbf{R}^{-1} (\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x})), \tag{74}$$

$f(\boldsymbol{x})$ のヘッセ行列は $\boldsymbol{g}$ の 1 次近似 $\boldsymbol{g}(\boldsymbol{x}) \approx \boldsymbol{g}(\bar{\boldsymbol{x}}_b) + \tilde{\mathbf{G}}(\boldsymbol{x} - \bar{\boldsymbol{x}}_b)$ を用いると，

$$\nabla^2 f(\boldsymbol{x}) \approx 2\tilde{\mathbf{G}}^{\mathsf{T}} \mathbf{R}^{-1} \tilde{\mathbf{G}}. \tag{75}$$

ガウス・ニュートン法では，ニュートン法の式に以上を代入して $f(x)$ を最小化する $\boldsymbol{x}$ を得る．すなわち，以下を繰り返す:

$$\boldsymbol{x}_j = \boldsymbol{x}_{j-1} + \left(\tilde{\mathbf{G}}^{\mathsf{T}} \mathbf{R}^{-1} \tilde{\mathbf{G}}\right)^{-1} \tilde{\mathbf{G}}^{\mathsf{T}} \mathbf{R}^{-1} (\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x})). \tag{76}$$

補足

ガウス・ニュートン法が適用できるのは，行列 $\tilde{\mathbf{G}}^{\mathsf{T}}\mathbf{R}^{-1}\tilde{\mathbf{G}}$ が正則な場合であるが，正則であっても 0 に近い固有値を持つ場合は安定しない．

そこで，レーベンバーグ・マーカート法は，計算を安定させるためにガウス・ニュートン法の式を以下のように修正して繰り返す:

$$\boldsymbol{x}_j = \boldsymbol{x}_{j-1} + \left(\tilde{\mathbf{G}}^{\mathsf{T}}\mathbf{R}^{-1}\tilde{\mathbf{G}} + \lambda^2\mathbf{I}\right)^{-1}\tilde{\mathbf{G}}^{\mathsf{T}}\mathbf{R}^{-1}\left(\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{x})\right). \tag{77}$$

先程の繰り返しアルゴリズムは，レーベンバーグ・マーカート法において，大規模時系列データを y とし，ヤコビ行列 $\tilde{\mathbf{G}}$ をアンサンブルを用いて近似したことに相当する．

まとめ

- 4 次元変分法では，初期値やパラメータを調整して，ある一定の期間に得られたデータすべてに適合するようなシナリオを求める．
- 代表的な手法はアジョイント法である．アジョイント法は適当な初期値シミュレーションを実行した後，後ろ向きにアジョイントモデルを実行することで，評価関数の微分を求める方法である．
- アジョイントモデルの作成は大変なので，アンサンブルを用いてアジョイントモデルを使わずに評価関数の最小化を行う方法も検討されている．