

データ同化実習

伊藤耕介 (琉球大学)

本マニュアルについて

データ同化実習では、簡単なモデルを用いて、代表的なデータ同化手法であるカルマンフィルタ系列の手法とアジョイント法（一般に4次元変分法として知られる手法）がどのように働いているのかを確認する。

まず、1節で実習の進め方について記し、2節では必要な部分に絞って同化手法の基礎を記している。3節と4節では実習で同化に用いる2つの方程式を具体的に記している。5節でFortranコードの流れを説明し、6節で参加者が取り組むべき課題について挙げている。付録AではFortranを使ったことがない人向けに簡単な解説を書いているので、必要に応じてご覧いただきたい。

1 実習の進め方について

今日の実習では双子実験を行う。教科書¹にもある通り、双子実験とは、同化手法の動作を確認するために行う試行実験の一つである。ある初期値を仮に「真の初期値」とみなして数値実験を行い、その数値実験結果に誤差を加えたものを「観測値」と呼ぶ。そして、「誤った初期値」から始めた数値実験結果に「観測値」を同化することによって、「真の初期値」や「真の数値実験結果」が復元できれば、同化手法が（双子実験の枠組みの中ではあるが）有効であることがわかる。

一般に、気象学や海洋物理学で用いるモデルは、非常に複雑なシステムであり、数万から数百億といった自由度（格子点の数×1格子点あたりの変数の数）をもっているが、ここでは、自由度が2の「減衰項付きの振動」（線形モデル）と自由度が3の「Lorenz(1963)モデル」（非線形モデル）を題材に、カルマンフィルタ系列の手法とアジョイント法を実際に理解しながら動かしてみる。

実習の時間には限りがあるので、「減衰項付きの振動」に関するカルマンフィルタ、アンサンブルカルマンフィルタ、アジョイント法のプログラムコードを配布する。このプログラムコードをコンパイル・実行しながら、まずは、これらの同化手法が何をしているのか、という点を理論とコードを見比べながらじっくりと確認しよう。その後、このプログラムをベースとして、Lorenz(1963)モデルを基礎とするデータ同化プログラムを自作してもらおう。基本的には、理論と「減衰項付き振動」のプログラムコードをなぞりながら進めばよい。これらのプログラムに慣れ親しんだら、発展問題や自発的に課題を見つけて取り組んでいただきたい。今日の実習を通じて、カルマンフィルタやアジョイント法といったデータ同化手法が、どこか遠い国の話などではなく、「意外に簡単なんだな」と思ってもらえることを期待している。

今日の実習では、本質的な内容の理解を容易にするため、簡単なモデルを使うことに加えて、以下のようなかなりの簡単化を施していることをあらかじめお断りしておく。

- 内積は自然内積（ベクトル要素どうしの積の単純和）として定義されている。

¹淡路敏之・蒲池政文・池田元美・石川洋一, 2009, データ同化：観測・実験とモデルを融合するイノベーション, 京都大学学術出版会, 284pp.

- 背景誤差共分散行列²，及び，観測誤差共分散行列について，対角成分のみを考えている．
- 観測誤差共分散行列を既知としており，観測値と真値の差の統計量は観測誤差共分散と整合的になっている．
- 時間方向には1次精度のオイラー法で離散化している．
- 観測演算子は線形の観測行列で表されるとき，観測される量はそのままモデル変数に対応しており，内挿やリトリバルを介さないとする．
- 観測値として変数 x だけを考えている．
- アジョイント法における最小値探索の方法として，評価関数の初期値に対する勾配に，ある負の係数をかけて初期値を修正する単純な方法を採用している．

つまり，お渡しする（した）コードはあくまで「学習用」なので，同様のものが現業のデータ同化システムで使われているとは考えない方がよい．と同時に，現業運用されている精巧なデータ同化システムは，美しい理論だけではなく，観測値の取得や品質管理といった多大なる苦勞のもとに成り立っているという点にも思いを馳せていただけると嬉しい．

2 同化手法の概要

2.1 カルマンフィルタ・拡張カルマンフィルタ

観測される変数の数を N_o ，制御変数の数を N_x とする．カルマンフィルタにおいては，観測値をベクトル y ，モデルの状態ベクトルを x であらわすとき，

$$x_t^a = x_t^f + K_t(y_t - Hx_t^f) \quad (1)$$

で値を更新する．ここで，上付きの a はデータ同化の結果得られた解析値， f はデータ同化前の予報値であり，下付きの t はタイムステップを表している． $N_o \times N_x$ の行列 H は観測行列と呼ばれ，モデル変数を観測値空間に内挿する役割を担っている³． K_t は，カルマンゲインと呼ばれる重み行列である．行列のサイズは， $N_x \times N_o$ であり，以下の式で計算される．

$$K_t = P_t^f H^T (R_t + H P_t^f H^T)^{-1} \quad (2)$$

$$P_t^f = M P_{t-1}^a M^T \quad (3)$$

ここでは， $N_x \times N_x$ の行列 P_t^f は，モデル変数の予報誤差共分散を表しており，システムノイズを考えていない． $N_o \times N_o$ の行列 R_t は，観測誤差共分散を表している．予報誤差共分散の時間発展を表す二つ目の式は，リアプノフ方程式と呼ばれている．ここで $N_x \times N_x$ の行列 M は状態遷移行列と呼ばれる行列で，線形モデルを用いた摂動の時間発展演算によって決まる．今日の実習で用いるシステムについては3節に具体的に記している．最後に，解析誤差共分散行列 $P_t^a (N_x \times N_x \text{ の行列})$ は

²カルマンフィルタ系列の場合は，予報誤差共分散行列の同化開始時における設定値

³例えば，モデル変数が $x = (x_1, x_2, x_3)^T$ のとき，変数 x_1 に対応した観測値 $x_{1,obs}$ が得られたなら， $H = (1, 0, 0)$ であり， $Hx = x_1$ となる．もし，すべてのモデル変数について観測が得られているような場合には $H = I$ となる．

$$\mathbf{P}_t^a = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^f \quad (4)$$

で計算される．この \mathbf{P}_t^a を新たに \mathbf{P}_t^f とすれば，その時刻を起点として，誤差共分散の時間発展が計算できる．ひとことで言うと，計算の初めに設定された \mathbf{P}^f が，モデルに沿った時間発展と観測値の取り込みによってどんどん変わっていくというところがミソである（図 1）．

非線形モデルに適用可能な拡張カルマンフィルタを考える場合も定式化はほぼ同じであるが，状態遷移行列 \mathbf{M} の要素が時間によって変化するモデル変数に依存するという点が異なる．状態ベクトル \mathbf{x} の非線形モデルによる 1 タイムステップ分の時間発展を $\mathbf{x}_{t+1} = \mathcal{M}(\mathbf{x}_t)$ と書くとき，テイラー展開を用いて，

$$\begin{aligned} \delta \mathbf{x}_{t+1} &= \mathcal{M}(\mathbf{x}_t + \delta \mathbf{x}_t) - \mathcal{M}(\mathbf{x}_t) \\ &= \frac{\partial \mathcal{M}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}_t} \delta \mathbf{x}_t + \mathcal{O}(\delta \mathbf{x}^2) \end{aligned} \quad (5)$$

$$\sim \frac{\partial \mathcal{M}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}_t} \delta \mathbf{x}_t \quad (6)$$

と変形できるので， $\mathbf{M}_t = \frac{\partial \mathcal{M}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}_t}$ とおく．具体例については，4 節に示してある．

2.2 アンサンブルカルマンフィルタ

アンサンブルカルマンフィルタで用いる式は，基本的には（拡張）カルマンフィルタの定式化と同じであるが， \mathbf{P}_t^f をモデルの方程式から陽に計算した状態遷移行列ではなく，アンサンブルランの結果に基づいて構成するという点が異なる．いま，モデルの状態ベクトルの予報値 \mathbf{x}_t^f に関して，アンサンブルメンバー数を N とし，そのうち m 番目のメンバーにおける値を $\mathbf{x}_t^{f,m}$ ，アンサンブル平均値を上付きのバーで表すことにする．行列 \mathbf{X}_t^f を

$$\mathbf{X}_t^f \equiv \frac{1}{\sqrt{N-1}} \left(\mathbf{x}_t^{f,1} - \bar{\mathbf{x}}_t^f, \mathbf{x}_t^{f,2} - \bar{\mathbf{x}}_t^f, \dots, \mathbf{x}_t^{f,N} - \bar{\mathbf{x}}_t^f \right) \quad (7)$$

と定義すれば，データ同化によって更新された値と更新のために必要となるカルマンゲインは以下のように書くことができる．

$$\mathbf{x}_t^{a,m} = \mathbf{x}_t^{f,m} + \mathbf{K}_t (\mathbf{y}_t - \mathbf{H} \mathbf{x}_t^{f,m} + \epsilon^m) \quad (8)$$

$$\mathbf{K}_t = \mathbf{X}_t^f (\mathbf{X}_t^f)^T \mathbf{H}^T (\mathbf{R}_t + \mathbf{H} \mathbf{X}_t^f (\mathbf{X}_t^f)^T \mathbf{H}^T)^{-1} \quad (9)$$

つまり，アンサンブルカルマンフィルタにおいては， \mathbf{P}_t^f を $\mathbf{X}_t^f (\mathbf{X}_t^f)^T$ と表していることになる．ここでは，観測値ベクトル \mathbf{y} に観測誤差分散に対応した摂動 ϵ を加えて，メンバーごとに異なる「観測値」を使っている．

2.3 アジョイント法

第一推定値と同化結果の初期値との差異，及び，観測値とモデルの結果の差異を定量化した評価関数 J を定義し，初期値を制御変数として最適化するという問題を解く（図 2 上）．観測ベクトルが \mathbf{y} の場合，

$$J = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_b) + \frac{1}{2} \sum_{t'} (\mathbf{H}\mathbf{x}_{t'} - \mathbf{y}_{t'})^T \mathbf{R}^{-1}(\mathbf{H}\mathbf{x}_{t'} - \mathbf{y}_{t'}) \quad (10)$$

となる．ここで， \mathbf{B} は背景誤差分散行列 ($N_x \times N_x$ の行列)， \mathbf{R} ($N_o \times N_o$ の行列) は観測誤差分散行列で，それぞれ，初期値の第一推定値（すなわち，同化を行う前の初期値）と観測値にどれだけ不確定性があるかを表す（今日の実習では，値を適当な対角行列と仮定する）．この関数は，初期値の第一推定値から値が離れたり，観測値と数値モデル結果の差が大きくなると値が大きくなる．右辺第二項は，観測値と数値モデルの結果との差を足し合わせたものであり，こちらが主要項となるのが一般的である．

時刻 $t = 0$ から $t = t'$ を同化期間とするとき，アジョイント変数（評価関数に対する勾配の情報を伝える変数） $\lambda_{t'} = 0$ として，アジョイントモデル \mathbf{M}_t^* による後方時間積分，

$$\lambda_{t-1} = \mathbf{M}_{t-1}^* \lambda_t + \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{H}\mathbf{x}_{t-1} - \mathbf{y}_{t-1}) \quad (t' \geq t > 1) \quad (11)$$

$$\lambda_0 = \mathbf{M}_0^* \lambda_1 + \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_b) + \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{H}\mathbf{x}_0 - \mathbf{y}_0) \quad (12)$$

を初期時刻まで行くと， J の初期値に対する偏微分．すなわち，

$$\frac{\partial J}{\partial \mathbf{x}_0} = \lambda_0^x \quad (13)$$

が計算できる（図2下）．この情報から， J を小さくするためには， x を増やすべきか減らすべきか， v を増やすべきか減らすべきかを知ることができる．配布したコードでは，勾配が得られたとき，単純に

$$\mathbf{x}_0^{\text{new}} = \mathbf{x}_0^{\text{old}} - \alpha_x \frac{\partial J}{\partial \mathbf{x}_0} \quad (14)$$

$$(15)$$

として，係数 α_x をかけることで，初期値を更新している．この作業を繰り返すことによって，評価関数を小さくする \mathbf{x}_0 を探索する．

3 減衰項つきの振動

3.1 方程式系

教科書 p.131 の式 (5.1) から強制力 w を除き，さらに簡単にした以下の式について考える．

$$\frac{dx}{dt} = v \quad (16)$$

$$m \frac{dv}{dt} = -kx - rv \quad (17)$$

2 節の \mathbf{x} は， $(x, v)^T$ に当たる．時間方向にオイラー法で離散化すると，以下ようになる．

$$x_t = x_{t-1} + \Delta t v_{t-1} \quad (18)$$

$$v_t = -\left(\frac{k\Delta t}{m}\right)x_{t-1} + \left(1 - \frac{r\Delta t}{m}\right)v_{t-1} \quad (19)$$

これに対応する接線形モデル（摂動の時間発展を記述するモデル）は，

$$x'_t = x'_{t-1} + \Delta t v'_{t-1} \quad (20)$$

$$v'_t = -\left(\frac{k\Delta t}{m}\right)x'_{t-1} + \left(1 - \frac{r\Delta t}{m}\right)v'_{t-1} \quad (21)$$

である（'は摂動であることを表す）．すなわち，状態遷移行列は

$$\mathbf{M} = \begin{bmatrix} 1 & \Delta t \\ -k\Delta t/m & 1 - r\Delta t/m \end{bmatrix} \quad (22)$$

と表される．ここでは，内積を自然内積として定義する．すると，状態遷移行列の随伴演算子（アジョイント演算子）は転置行列に等しく，

$$\mathbf{M}^* = \begin{bmatrix} 1 & -k\Delta t/m \\ \Delta t & 1 - r\Delta t/m \end{bmatrix} \quad (23)$$

となる．

いま， x の観測値が得られるとしたとき，評価関数は，

$$2J = (x_0 - x_b)^2/B_{xx} + (v_0 - v_b)^2/B_{vv} + \sum (x_t - x_{t,\text{obs}})^2/R_{xx} \quad (24)$$

と書くことができる．ここで， B_{xx}, B_{vv} は \mathbf{B} のうち，それぞれ， x と v の分散に対応する要素であり， R_{xx} は \mathbf{R} のうち， x の分散に対応する要素である．

アジョイントモデルは， $t > 1$ において，

$$\lambda_{t-1}^x = \lambda_t^x - \left(\frac{k\Delta t}{m}\right)\lambda_t^v + (x_{t-1} - x_{t-1,\text{obs}})/R_{xx} \quad (25)$$

$$\lambda_{t-1}^v = \Delta t \lambda_t^x + \left(1 - \frac{r\Delta t}{m}\right)\lambda_t^v \quad (26)$$

時刻 $t = 0$ においては，初期値が第一推定値から離れすぎないようにするための項が追加され，

$$\lambda_0^x = \lambda_1^x - \left(\frac{k\Delta t}{m}\right)\lambda_1^v + (x_0 - x_b)/B_{xx} + (x_0 - x_{0,\text{obs}})/R_{xx} \quad (27)$$

$$\lambda_0^v = \Delta t \lambda_1^x + (v_0 - v_b)/B_{vv} + \left(1 - \frac{r\Delta t}{m}\right)\Delta t \lambda_1^v \quad (28)$$

となる．

ただし，式 (25) の右辺第 3 項と式 (27) の右辺第 4 項は，観測値が得られたときにのみ追加される項である．

4 Lorenz(1963) モデル

4.1 方程式系

Lorenz(1963) のモデルは以下の式で表される .

$$\frac{dx}{dt} = -\sigma(x - y) \quad (29)$$

$$\frac{dy}{dt} = -y - xz + rx \quad (30)$$

$$\frac{dz}{dt} = xy - bz \quad (31)$$

ローレンツが有名な論文の中で与えた設定は $\sigma = 10, r = 28, b = 8/3$ である . この式を時間方向にオイラー法で離散化した式は ,

$$x_t = x_{t-1} - \sigma(x_{t-1} - y_{t-1})\Delta t \quad (32)$$

$$y_t = y_{t-1} + (-y_{t-1} - x_{t-1}z_{t-1} + rx_{t-1})\Delta t \quad (33)$$

$$z_t = z_{t-1} + (x_{t-1}y_{t-1} - bz_{t-1})\Delta t \quad (34)$$

である . 接線形モデルは ,

$$x'_t = x'_{t-1} + (-\sigma(x'_{t-1} - y'_{t-1}))\Delta t \quad (35)$$

$$y'_t = y'_{t-1} + (-y'_{t-1} - x_{t-1}z'_{t-1} - z_{t-1}x'_{t-1} + rx'_{t-1})\Delta t \quad (36)$$

$$z'_t = z'_{t-1} + (x_{t-1}y'_{t-1} + y_{t-1}x'_{t-1} - bz'_{t-1})\Delta t \quad (37)$$

すなわち , 状態遷移行列は ,

$$\mathbf{M} = \begin{bmatrix} 1 - \Delta t\sigma & \Delta t\sigma & 0 \\ \Delta t(r - z_{t-1}) & 1 - \Delta t & -\Delta tx_{t-1} \\ \Delta ty_{t-1} & \Delta tx_{t-1} & 1 - \Delta tb \end{bmatrix} \quad (38)$$

であり , 先ほどと同様に , 随伴演算子 (アジョイント演算子) は ,

$$\mathbf{M}^* = \begin{bmatrix} 1 - \Delta t\sigma & \Delta t(r - z_{t-1}) & \Delta ty_{t-1} \\ \Delta t\sigma & 1 - \Delta t & \Delta tx_{t-1} \\ 0 & -\Delta tx_{t-1} & 1 - \Delta tb \end{bmatrix} \quad (39)$$

となる .

評価関数は , x の観測値が得られるとしたとき ,

$$2J = (x_0 - x_b)^2/B_{xx} + (y_0 - y_b)^2/B_{yy} + (z_0 - z_b)^2/B_{zz} + \sum (x_t - x_{t,\text{obs}})^2/R_{xx} \quad (40)$$

と書くことができ , アジョイントモデルは ,

$$\lambda_{t-1}^x = (1 - \Delta t\sigma)\lambda_t^x + \Delta t(r - z_{t-1})\lambda_t^y + \Delta ty_{t-1}\lambda_t^z + (x_{t-1} - x_{t-1,\text{obs}})/R_{xx} \quad (41)$$

$$\lambda_{t-1}^y = \Delta t\sigma\lambda_t^x + (1 - \Delta t)\lambda_t^y + \Delta tx_{t-1}\lambda_t^z \quad (42)$$

$$\lambda_{t-1}^z = -\Delta tx_{t-1}\lambda_t^y + (1 - \Delta tb)\lambda_t^z \quad (43)$$

となる．時刻 $t = 0$ においては，初期値が第一推定値から離れすぎないようにするための項が追加され，

$$\lambda_0^x = (1 - \Delta t \sigma) \lambda_1^x + \Delta t (r - z_0) \lambda_1^y + \Delta t y_0 \lambda_1^z + (x_0 - x_b)/B_{xx} + (x_0 - x_{\text{obs}})/R_{xx} \quad (44)$$

$$\lambda_0^y = \Delta t \sigma \lambda_1^x + (1 - \Delta t) \lambda_1^y + \Delta t x_0 \lambda_1^z + (y_0 - y_b)/B_{yy} \quad (45)$$

$$\lambda_0^z = -\Delta t x_0 \lambda_1^y + (1 - \Delta t b) \lambda_1^z + (z_0 - z_b)/B_{zz} \quad (46)$$

と表される．ここで， x_t, y_t, z_t が必要となるが，これらは，非線形モデルの計算結果を記憶しておいたものを用いる．先ほどと同様に，式 (41) の右辺第 4 項と式 (44) の右辺第 5 項は，観測値が得られたときにのみ追加される．

5 Fortran コードの構成

Fortran のコードは，わざと冗長にしているが，それでも，宣言文と画面への出力の部分を除けば，わずか 70-100 行なので，怖がらずに取り組んでいただきたい．

カルマンフィルタによる「減衰項付きの振動」の同化のプログラム (Oscillation_KF.f90) は，宣言部と画面への出力部を除いて，以下のように構成されている．

- Sec. 1: 変数の初期化．
- Sec. 2: 真の初期値から始める数値計算，及び，観測値の生成．
- Sec. 3: 誤った初期値から始める数値計算．
- Sec. 4: 誤った初期値から始め，観測値を取り込む同化計算．
- Sec. 4-1: 数値積分を行う．
 - Sec. 4-2-1: 状態遷移行列を作成する．
 - Sec. 4-2-2: リアプノフ方程式をもとに予報誤差分散の時間発展を計算する．
 - Sec. 4-2-3: 観測値がある時刻に，カルマンゲイン（修正量に関する重みづけ）を計算する．
 - Sec. 4-2-4: カルマンゲインに応じた値の修正を行う．
 - Sec. 4-2-5: カルマンゲインに応じた値の修正を行う．
- Sec. 5: 同化終了時の値から，予報実験を行う．

アンサンブルカルマンフィルタを実装する際には，誤った初期値に擾乱を加えてアンサンブルメンバーを作成する点，4-2-1 と 4-2-2 の部分の代わりに擾乱の統計処理に基づいて予報誤差分散を計算する点，計算されたカルマンゲインに基づいて各メンバーの結果を修正する点が異なる（2.2 節参照）．

アジョイント法による「減衰項付き振動」の同化のプログラム (Oscillation_4DVAR.f90) は，宣言部と画面への出力部を除いて，以下のように構成されている．

- Sec. 1: 変数の初期化．
- Sec. 2: 真の初期値から始める数値計算，及び，観測値の生成．
- Sec. 3: 誤った初期値から始める数値計算．
- Sec. 4: 誤った初期値から始め，観測値を取り込む同化計算．
 - Sec. 4-1: 各サイクルの初めに必要となる変数の初期化
 - Sec. 4-2: 前方数値積分と評価関数のうちミスフィットで定量化される部分の計算
 - Sec. 4-3: ミスフィットの情報の投入とアジョイントモデルを用いた後方時間積分
 - Sec. 4-4: アジョイント変数と評価関数のうち背景値に関連する部分

Sec. 4-5 イテレーションの終了を判定する部分

Sec. 4-6 イテレーションの終了判定に必要な変数の保存

Sec. 4-7 降下法を用いた初期値の修正

Sec. 5: 同化終了時の値から，予報実験を行う．

以上のセクションの番号は，Fortran プログラムのコメント文に書いてあるので，確認いただきたい．ここでは，カルマンフィルタ系列，アジョイント法の両手法で，ともに，同化実験 (Sec. 4) と予報実験 (Sec. 5) を一つのプログラム内で行っている．観測値が得られる同化期間 (Assimilation Period) においては，同化結果が真の実験結果や観測値に近づいているかどうかを確認し，同化が終了した時点から行う予報期間 (Prediction Period) においては，誤った初期値から始める場合に比べて，同化結果を用いて予報実験を行った方が真の値に近づいているかどうかを確認する．

6 課題

6.1 基礎的な課題

- まずは動作確認

コンソールやコマンドプロンプト上で

```
g95 Oscillation_KF.f90 (または g95 Oscillation_4DVAR.f90)
```

としてコンパイルしたのち生成されたファイル (a.out や a.exe) を

```
./a.out (または a.exe)
```

として実行し，出力を確認する (次ページ参照)．もし Fortran の使い方がわからなければ，付録 A を参考にしていきたい．


```
./a.exe
```

```
#####  
###    Identical Twin Experiment    ###  
###    Kalman Filter (page 131)     ###  
#####
```

```
Assmilation Period: t=  0.00- 10.00
```

```
Prediction Period:  t= 10.00- 20.00
```

```
*****Assimilation Period: x *****
```

[Time]	[True]	[No Assim]	[Assim]	[Observation]
0.00	5.000	2.000	2.000	No obs
2.00	1.487	1.702	1.503	1.488
4.00	-2.623	-0.758	-2.576	-3.136
6.00	-1.585	-1.225	-1.606	-1.510
8.00	1.164	0.135	1.130	1.122
10.00	1.261	0.774	1.316	1.699

```
*****Prediction Period: x *****
```

12.00	-0.370	0.122	-0.334
14.00	-0.857	-0.434	-0.881
16.00	-0.002	-0.188	-0.030
18.00	0.516	0.213	0.523
20.00	0.137	0.169	0.156

```
*****Assimilation Period: v *****
```

[Time]	[True]	[No Assim]	[Assim]
0.00	0.000	1.000	1.000
2.00	-2.768	-1.142	-1.142
4.00	-0.727	-0.903	-0.793
6.00	1.477	0.451	1.453
8.00	0.826	0.663	0.839
10.00	-0.673	-0.098	-0.655

```
*****Prediction Period: v *****
```

12.00	-0.675	-0.425	-0.706
14.00	0.228	-0.053	0.209
16.00	0.467	0.242	0.480
18.00	-0.015	0.095	0.000
20.00	-0.285	-0.121	-0.289

この実験では、同化期間が $t=0$ から $t=10$ であり、 $t=10$ の結果を基にして、 $t=20$ まで予報実験を行っている。数字 4 列 (x の同化期間については 5 列) は x と v のそれぞれについて、時刻と計算結果を示しており、[True] は双子実験の真値、[No Assim] は $t=0$ において誤った初期値から開始した数値計算結果、[Assim] は観測値を同化した結果を示している。[Observation] は x の同化期間における観測値を示している。観測値を同化することによって、徐々に x も v も真値に近づいていると同時に、同化後の予報の結果が同化を行わない実験 (No Assim) に比べて真値に近づいていることがわかるであろう。

また、カルマンフィルタの場合には、このほかに解析誤差共分散行列 P^a も表示されるほか、4 次元変分法の場合には、イテレーションの途中に、

```
iteration= 23 ; x(0)= 4.293 ; v(0)= 0.212 ; J= 9.400
adjoint x= -4.810 adjoint v= 1.288
```

のような出力も表示される。左から順に、何回目のイテレーションか、 x や v の同化期間初期時刻における値、評価関数の値、評価関数の $x(0)$ と $v(0)$ に対する勾配 ($\partial J/\partial x(0)$ と $\partial J/\partial v(0)$) を表している。

- 同化期間・予報期間・観測間隔・観測誤差共分散行列・背景誤差共分散行列の値を変化させると結果はどう変わるだろうか？また、結果が変わった理由について教科書を参考にして考えてみよう。
- 手法について理解したら、このプログラムを踏み台にして、Lorenz(1963) モデルに関する拡張カルマンフィルタ・アジョイント法・アンサンブルカルマンフィルタのコードを作成してみよう。ちなみに、ローレンツが有名な論文の中で与えた設定は $\sigma = 10, r = 28, b = 8/3$ である（時間差分にはオイラー法を用いているので、 $\Delta t \leq 0.001$ を推奨する。また、アジョイントモデルが動かない場合には、同化期間や修正係数が大きすぎないか？を確認しよう）。

データ同化を学び始めた山崎哲さん（海洋研究開発機構）にご協力いただいたところ、振動のシステムで練習をし、Lorenz(1963) モデルに関する拡張カルマンフィルタ及びアジョイント法のコードを作成するのに、あわせて 3 時間程度を要した。個人の経験に応じ、異なる複数の手法を作成してもよいし、いずれかの同化システムを作った後で次小節の発展問題に挑んでもよいだろう。

6.2 基礎的な課題が終わったら

手法の概要がつかめたら、以下に示す課題のうち興味のある問題から順に取り組んでみるとよいだろう。

6.2.1 全般的な話題

- デフォルトでは、 x だけを観測値としているが、そのほかの変数を観測値として加えられるようにしたらプログラムはどのように変わるか？
- モデルにバイアスがある場合には、結果はどのように変わるだろうか？（真値にバイアスとガウシアンノイズを加えたものを観測値としてみればよい）

- 真の計算を行う際に用いるパラメータと誤った計算を行う際に用いるパラメータの値を変えた場合に、推定される初期値はどうなるだろうか？(また、それに関連し、初期値問題としてではなく、パラメータと初期値を同時に修正する問題として定式化したらどうなるだろうか？)
- 線形性のチェック：もし、線形性が成り立っているならば、擾乱を加えた非線形モデル計算の結果から非線形モデルを引いた差は、擾乱を接線形モデルを用いて時間発展させたものに等しくなる。線形性がどれぐらいの期間保たれているか、そして、その時間スケールを超えた場合にデータ同化の性能がどのように変わるかを調べてみよう。また、線形性が成り立つかどうか、擾乱の大きさや初期値の値に依存していることも確認しよう。

6.2.2 カルマンフィルタ系列の手法に関する話題

- Lorenz モデルにおけるアンサンブルメンバー数を 2,10,100 と変えて、誤差分散の大きさを比べ、一般的なアンサンブルカルマンフィルタにおいて共分散膨張 (Inflation) が導入されていることとの関連を考えてみよう。
- 種々のスムーザを導入し、カルマンフィルタの結果がどう変わるかを確認してみよう。また、RTS スムーザを導入したときに、アジョイント法の結果に近くなるかどうかを確認しよう。
- アンサンブルカルマンフィルタによって確率密度関数がどのように変化するかを確認しよう。また、真の確率密度関数を記述するためにはどれだけのメンバーが集まれば必要かを考えてみよう。

6.2.3 アジョイント法に関する話題

- 誤差共分散行列に非対角成分があるとしたら、コードはどのように変わるだろうか？
- 線形性が満たされている場合には、アジョイント変数が感度に対応していることを確認しよう。すなわち、評価関数のある時刻の物理量として得られるアジョイント変数 (たとえば、 $\partial x(t=20)/\partial v(t=0)$) が、入力値に加えた摂動と出力として現れる摂動の比に等しいことを確認しよう。
- それぞれの問題で、同化期間を長くしたときにアジョイント法がうまくいくかどうかを確認しよう。また、同化期間を長くにとってデータ同化がうまくいかなかった際、誤差のサイズを小さくしたらどうなるかを調べてみよう。
- 複数のサイクルを繰り返し適用する実験を試みよう。
- 変分法のための降下法として、共役勾配法や準ニュートン法を導入してみよう。

6.2.4 先端的同化手法に関する話題

- カルマンフィルタで時間発展された背景誤差共分散行列を、アジョイント法の背景誤差分散として用いる研究 (ハイブリッド法) を導入してみよう。
- 粒子フィルターを導入してみよう。また、メンバー数がどれだけあれば十分かを確認しよう。

A Fortran の使い方

エディタ (メモ帳や秀丸など, Word はダメ) で書かれたプログラムを, コンソールやコマンドプロンプト上 (黒い画面もしくは白い画面) でコンパイル (人間が作成したプログラムをコンピュータ上で実行可能な形式に変換) することによって, 実行ファイル (a.out や a.exe) が生成される. Fortran はそのために必要となる「コンパイラ」としての役割を提供している.

Windows マシンであれば, Fortran コンパイラをインストールするにあたって一番簡単なのは, <http://www.g95.org/downloads.shtml> から, g95 という Fortran コンパイラをダウンロードする方法である. ダウンロードした exe ファイルを実行し, 指示に従うだけでインストールが完了する.

プログラムとコンパイルの実例を示そう. メモ帳などで, 以下のプログラムを書き, Hello.f90 として保存する.

```
program Hello
implicit none
write(*,*) "Hello World!"
end program Hello
```

次に, コンソールや Windows のコマンドプロンプト (アクセサリ コマンドプロンプト) において, コンパイラを使ってプログラムを実行ファイルに変換する. g95 を用いている場合には, Hello.f90 があるディレクトリに移動 (Windows のコマンドプロンプトでは dir xxx など) したあとで,

```
g95 Hello.f90
```

と入力する. うまくいくと, a.out (Linux, Mac の場合) もしくは a.exe (Windows コマンドプロンプトの場合) という実行ファイルが生成される.

実行ファイルができたなら, コンソールやコマンドプロンプト上で,

```
./a.out
```

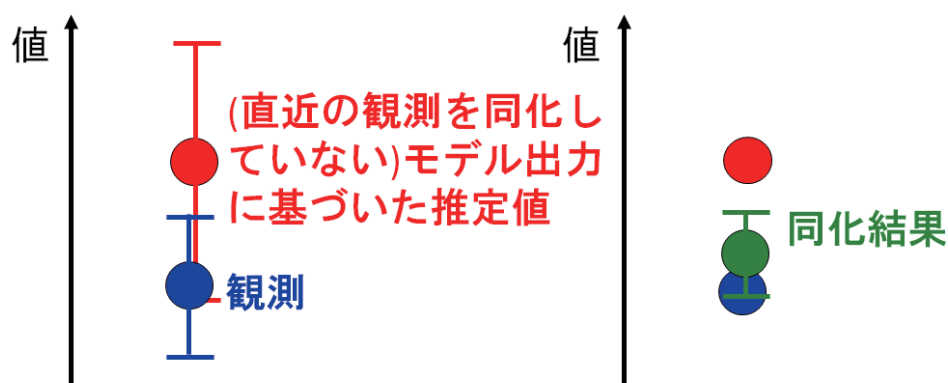
または

```
a.exe
```

などと入力し, 画面に”Hello World!”と表示されたら自分で作ったプログラムを実行できたことになる.

カルマンフィルター系列の手法の概念(1/2)

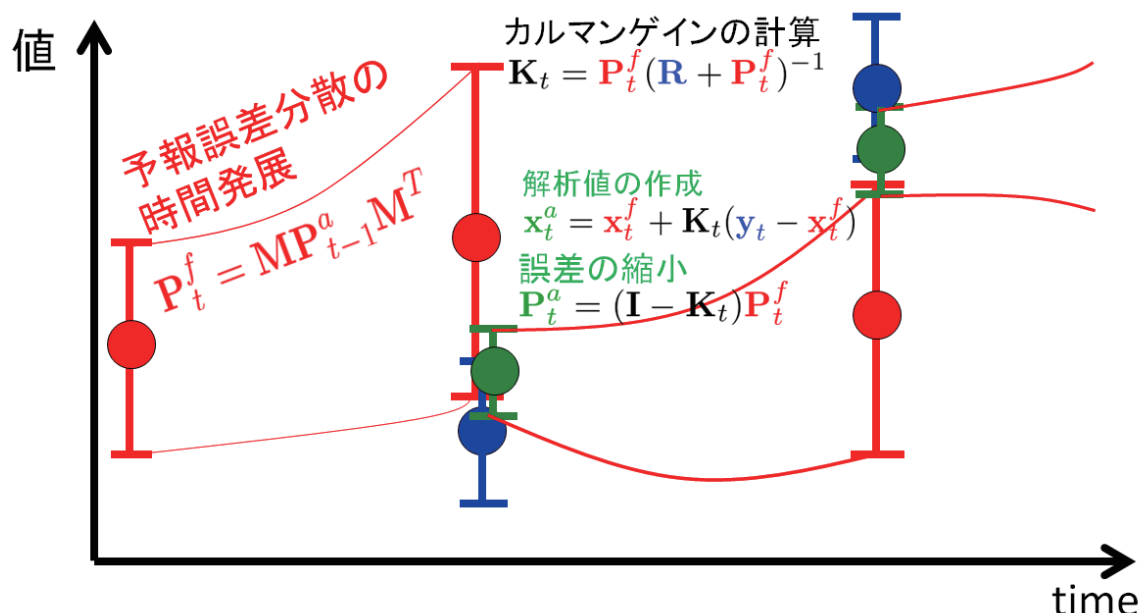
観測 + 数値モデル結果 ⇒ 同化結果



$$x_{assim} = x_{model} + \frac{\sigma_{model}^2}{\sigma_{obs}^2 + \sigma_{model}^2} (x_{obs} - x_{model})$$

$$\sigma_{assim}^2 = \left(1 - \frac{\sigma_{model}^2}{\sigma_{obs}^2 + \sigma_{model}^2} \right) \sigma_{model}^2$$

カルマンフィルター系列の手法の概念(2/2)



- 予報誤差分散の時間発展を計算し、観測値が得られたら、重みづけをして修正する。
- 新たな誤差分散を予報誤差分散に置き換え計算を行う

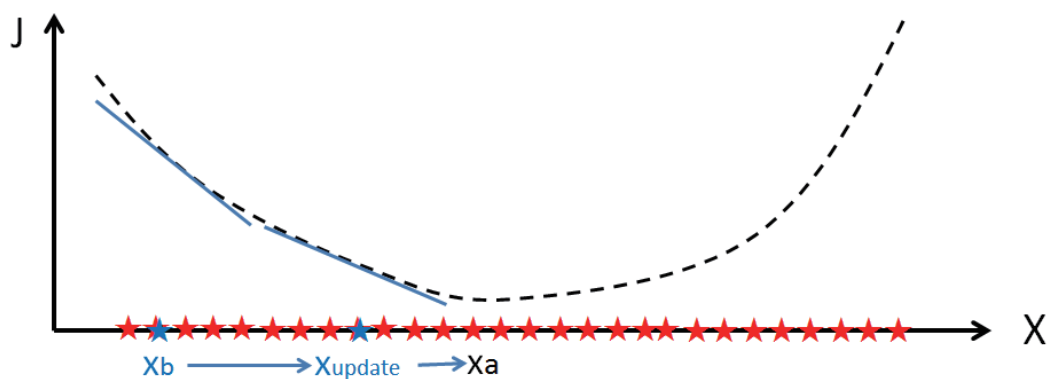
図 1: カルマンフィルタの模式図．簡単のため，観測値演算子 (H) は省略してある．

アジョイント法(4次元変分法)の概念(1/2)

評価関数 J を最小化するような初期値 x を効率的に見つけたい

$$J = \underbrace{\frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_b)}_{\text{初期値の第一推定値からの離れ具合}} + \underbrace{\frac{1}{2} \sum_{t'} (\mathbf{x}_{t'} - \mathbf{y}_{t'})^T \mathbf{R}^{-1}(\mathbf{x}_{t'} - \mathbf{y}_{t'})}_{\text{観測値と数値モデル結果の差(こちらが主要項)}}$$

- J の分布は事前に分からない
- 方法1: x の値を変えて実験を繰り返す.
⇒ 単純明快だが、変数が多いと実行不可能
- 方法2: $\partial J / \partial x$ をアジョイント法で計算し、 x を改善する
⇒ 効率的に (x_a) にたどり着く



アジョイント法(4次元変分法)の概念(2/2)

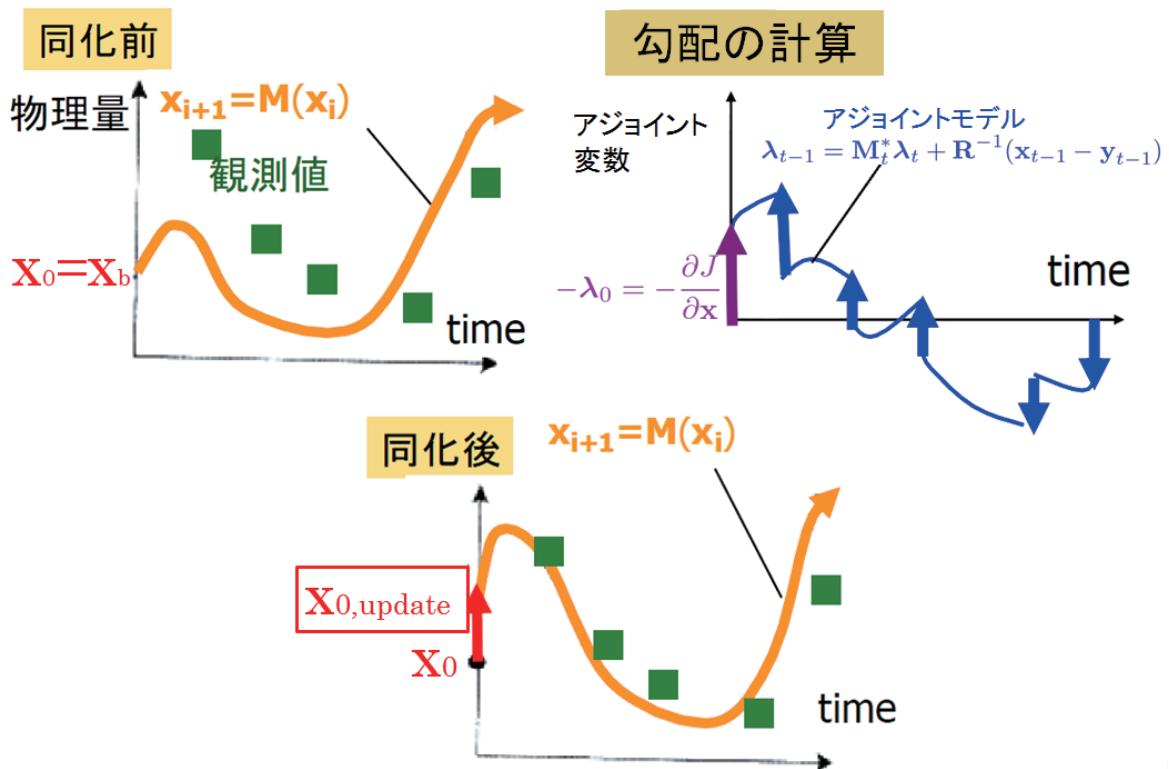


図 2: アジョイント法の模式図．簡単のため，観測値演算子 (H) は省略してある．