

第3章 データ同化システム

3.1 asuca 変分法データ同化システム¹

3.1.1 はじめに

本章では、*asuca* を基にした変分法データ同化システム (*asuca-Var*) の概要を示す。数値予報におけるデータ同化の主な役割は、観測データと数値予報モデルを利用して数値予報モデルの初期値を作成することにある。変分法データ同化手法は、変分原理と最尤推定に基づくデータ同化手法であり、背景場の時間発展を考慮しない3次元変分法、時間発展を考慮する4次元変分法、そして、アンサンブル摂動を利用して背景誤差に流れ依存性を考慮するハイブリッド変分法等がある。*asuca-Var* は、それら変分法データ同化手法を用いて *asuca* の初期値を作成するデータ同化システムである。

まず、*asuca-Var* の構成と、その構成要素の役割を示す。次に、変分法に基づいた解析値の導出を行い、*asuca-Var* の各種設定を示す。そして、背景誤差に流れ依存性を考慮したデータ同化手法への拡張を述べる。各項目において、計算資源の節約や開発効率の向上のための取り組みも紹介する。

なお、ここでの記述は、気象庁予報部 (2010)、Banister (2008)、Lahoz et al. (2010)、淡路ほか (2009) を参考にした。適宜、ご参照いただきたい。

3.1.2 *asuca-Var* の構成

asuca-Var は、*asuca* の初期値を作成することを目的とした変分法データ同化システムである。その構成要素は、第1.4.3項で述べられているとおり、予報モデル「*asuca*」、観測値を用いて最適解の計算を行う解析本体「*asuca-Var* コア」、予報値に作用し観測相当量を求める「観測演算子」である。そして、*asuca* と観測演算子は、気象庁の物理過程ライブラリ (原 2012) を利用している。特に、観測演算子は、*asuca* の開発プロジェクト以外の「外部パッケージ」の利用を想定している。ここで、外部パッケージとは汎用性のある観測シミュレーションのためのプログラム群を指し、欧州衛星開発機構が開発を主導する高速放射伝達モデル (the fast Radiative Transfer model for TOVS: RTTOV; Saunders et al. 1999) が、その一例である。

図 3.1.1 は、*asuca-Var* の構成要素である *asuca*、*asuca-Var* コア、物理過程ライブラリ、観測演算子と外部パッケージの関係を示したものである。*asuca*、物理過程ライブラリ、観測演算子は、それぞれ非線形 (NL)、接線形 (TL)、随伴 (AD) コードを持つ。*asuca-Var* コアは、それらを統合し、プレコンディショニング、限定メモリ準ニュートン法による極小値探索、システム全体の制御等を行うデータ同化システムの心臓部として機能する。また、3次元変分法 (3D-Var)、4次元変分

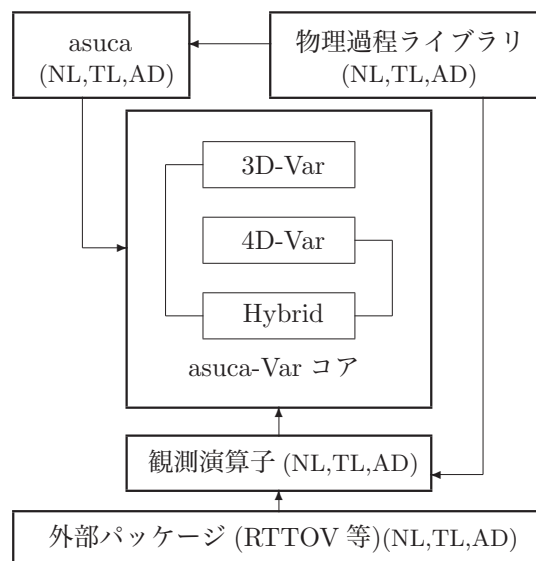


図 3.1.1 *asuca*、物理過程ライブラリ、観測演算子、外部パッケージ、*asuca-Var* コアの関係。矢印は、参照元と参照先の依存方向を表す。

法 (4D-Var)、流れに依存した背景誤差の利用 (Hybrid) など、その目的に応じて機能をパラメータの設定で切り替えることができるように実装している。

なお、上記で述べたように、NL、TL、AD コードをデータ同化システムの心臓部から分離することにより、予報モデルや観測演算子の更新に伴う *asuca-Var* コアの修正を最小限に抑えることができ、バグの混入防止や複数開発者による作業効率の向上が期待できる。特に、予報モデルが、NL の他に TL、AD を持つ意義は、データ同化システムと予報モデルとの乖離を防止できる点にある。その詳細は次節で議論される。

3.1.3 解析値の導出

時刻 t におけるモデル変数を要素に持つ状態ベクトルを \mathbf{x}_t とし、時刻 $t=0$ における第一推定値を \mathbf{x}_0^b とする。また、それぞれの真値からの差を $\delta\mathbf{x}_t$ と $\delta\mathbf{x}_0^b$ と表す。このとき、背景誤差共分散行列は、

$$\mathbf{B}_0 = \langle \delta\mathbf{x}_0^b, \delta\mathbf{x}_0^{bT} \rangle \quad (3.1.1)$$

で与えられる。ここで、 $\langle \cdot, \cdot \rangle$ は、平均を表す。

次に、時刻 t における観測値を要素にもつ観測ベクトルを \mathbf{y}_t と定義する。状態ベクトルから、 \mathbf{y}_t に対応する観測相当量ベクトルへの変換を行う非線形観測演算子を \mathcal{H}_t とする。また、 \mathbf{y}_t と真値との差を $\delta\mathbf{y}_t$ とすると、観測誤差共分散行列は、

$$\mathbf{R}_t = \langle \delta\mathbf{y}_t, \delta\mathbf{y}_t^T \rangle \quad (3.1.2)$$

で与えられる。同化期間を $t=0$ から $t=t_a$ までとす

¹ 幾田 泰醇

ると、評価関数 J は、

$$J = \frac{1}{2} (\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}_0^{-1} (\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2} \sum_{t=0}^{t_a} (\mathcal{H}_t(\mathbf{x}_t) - \mathbf{y}_t)^T \mathbf{R}_t^{-1} (\mathcal{H}_t(\mathbf{x}_t) - \mathbf{y}_t) + J_p \quad (3.1.3)$$

となる。ここで、 J_p は拘束項を表す。また、システム誤差は考慮しないものとする。

なお、状態ベクトルの時間発展は、時刻 t から $t + \Delta t$ までの非線形時間発展演算子 $\mathcal{M}_{t,t+\Delta t}$ を用いて

$$\mathbf{x}_{t+\Delta t} = \mathcal{M}_{t,t+\Delta t}(\mathbf{x}_t) \quad (3.1.4)$$

と表す。ここで、 Δt は、モデルの積分時間間隔である。

次に、数値的極小値探索の繰り返し回数を i とおき、探索過程で更新された状態ベクトルを $\mathbf{x}_0^{(i)}$ とする。そのときの修正量を

$$\delta \mathbf{x}_0^{(i)} = \mathbf{x}_0^{(i)} - \mathbf{x}_0^b \quad (3.1.5)$$

とし、 \mathbf{y}_t と時刻 t の第一推定値 $\mathbf{x}_t^{(0)} (= \mathcal{M}_{0,t}(\mathbf{x}_0^b))$ からシミュレートした観測相当量との差を

$$\mathbf{d}_t = \mathbf{y}_t - \mathcal{H}_t(\mathbf{x}_t^{(0)}) \quad (3.1.6)$$

とする。また $\mathcal{M}_{t,t+\Delta t}(\cdot)$ の接線形演算子 $\mathbf{M}_{t,t+\Delta t}$ は、

$$\mathcal{M}_{t,t+\Delta t}(\mathbf{x}_t^b + \delta \mathbf{x}_t^{(i)}) = \mathcal{M}_{t,t+\Delta t}(\mathbf{x}_t^b) + \mathbf{M}_{t,t+\Delta t} \delta \mathbf{x}_t^{(i)} + \mathcal{O}\left(|\delta \mathbf{x}_t^{(i)}|^2\right) \quad (3.1.7)$$

を満たすヤコビ行列として定義し、同様に、 $\mathcal{H}_t(\cdot)$ の接線形演算子 \mathbf{H}_t は、

$$\mathcal{H}_t(\mathbf{x}_t^b + \delta \mathbf{x}_t^{(i)}) = \mathcal{H}_t(\mathbf{x}_t^b) + \mathbf{H}_t \delta \mathbf{x}_t^{(i)} + \mathcal{O}\left(|\delta \mathbf{x}_t^{(i)}|^2\right) \quad (3.1.8)$$

を満たすヤコビ行列として定義する。このとき、接線形モデルの時間発展は、 $\mathbf{M}_{t,t+\Delta t}$ を用いて

$$\delta \mathbf{x}_{t+\Delta t}^{(i)} = \mathbf{M}_{t,t+\Delta t} \delta \mathbf{x}_t^{(i)} \quad (3.1.9)$$

と書ける。

時刻 t_a までの接線形時間発展演算子を含めた観測演算子を

$$\hat{\mathbf{H}} = \begin{pmatrix} \mathbf{H}_0 \\ \mathbf{H}_{\Delta t} \mathbf{M}_{0,\Delta t} \\ \vdots \\ \mathbf{H}_{t_a} \mathbf{M}_{0,t_a} \end{pmatrix} \quad (3.1.10)$$

と定義し、(3.1.10) 式と同様に、 \mathbf{d}_t で構成される行列を $\hat{\mathbf{d}}$ とする。このときの観測誤差共分散行列は、

$$\hat{\mathbf{R}} = \text{diag}[\mathbf{R}_0, \dots, \mathbf{R}_{t_a}] \quad (3.1.11)$$

となる。さらに、 \mathbf{B}_0 の演算を簡略化するためモデル変数 $\delta \mathbf{x}_0$ から制御変数 $\delta \chi_0$ への変数変換

$$\delta \chi_0 = \mathbf{B}_0^{-1/2} \delta \mathbf{x}_0 \quad (3.1.12)$$

を行う。ここで、 $\delta \chi_0$ は、

$$\langle \delta \chi_0, \delta \chi_0^T \rangle = \mathbf{I} \quad (3.1.13)$$

を満たすベクトルであり、 $\mathbf{B}_0^{1/2}$ は、

$$\mathbf{B}_0 = \mathbf{B}_0^{1/2} \langle \delta \chi_0, \delta \chi_0^T \rangle (\mathbf{B}_0^{1/2})^T \quad (3.1.14)$$

$$\equiv \mathbf{B}_0^{1/2} \mathbf{B}_0^{T/2} \quad (3.1.15)$$

を満たす行列である。(3.1.10) 式、(3.1.11) 式及び (3.1.12) 式の変数変換の結果、評価関数 (3.1.3) 式の最小化問題は、 $\delta \chi_0$ 表現の次式に帰着する。

$$\min_{\delta \chi_0^{(i)}} J(\delta \chi_0^{(i)}) = \min_{\delta \chi_0^{(i)}} \left[\frac{1}{2} \left\| \delta \chi_0^{(i)} \right\|_2^2 + \frac{1}{2} \left\| \hat{\mathbf{R}}^{-1/2} \hat{\mathbf{H}} \mathbf{B}_0^{1/2} \delta \chi_0^{(i)} - \hat{\mathbf{R}}^{-1/2} \hat{\mathbf{d}} \right\|_2^2 + J_p(\delta \chi_0^{(i)}) \right] \quad (3.1.16)$$

ここで、 $\|\cdot\|_2$ は、 L^2 ノルムを表す。また、(3.1.16) 式の勾配は、

$$\frac{\delta J}{\delta \chi_0^{(i)}} = \delta \chi_0^{(i)} + \mathbf{B}_0^{T/2} \hat{\mathbf{H}}^T \hat{\mathbf{R}}^{-1} (\hat{\mathbf{H}} \mathbf{B}_0^{1/2} \delta \chi_0^{(i)} - \hat{\mathbf{d}}) + \frac{\delta J_p}{\delta \chi_0^{(i)}} \quad (3.1.17)$$

となる。 n 回の数値的極小値探索の結果、得られた解を $\delta \chi_0^{(n)}$ とすると時刻 $t = 0$ の解析値 \mathbf{x}_0^a は、

$$\mathbf{x}_0^a = \mathbf{x}_0^b + \mathbf{B}_0^{1/2} \delta \chi_0^{(n)} \quad (3.1.18)$$

で与えられる。

モデルの初期値は、(3.1.18) 式から同化窓最後の $t = t_a$ まで \mathcal{M}_{0,t_a} により時間積分した

$$\mathbf{x}_{t_a}^a = \mathcal{M}_{0,t_a}(\mathbf{x}_0^a) \quad (3.1.19)$$

で与えられる。

3.1.4 制御変数

解析対象の変数は、東西風 u 、南北風 v 、地中温度 T_g 、地上気圧 p_s と温位 θ 、地表面体積含水率 w_g 、偽相対湿度 $\mu_p = q_v/q_{\text{sat}0}^b$ (Dee and da Silva 2003) とする。 q_v は、水蒸気の密度と全密度の比である。 $q_{\text{sat}0}^b$ は、第一推定値の飽和水蒸気密度と全密度の比であり、第一推定値の気温 T_0^b 、水飽和水蒸気の密度比 $q_{\text{sat},w0}^b$ 、水飽和水蒸気の密度比 $q_{\text{sat}i0}^b$ を用いて次式で与える。

$$q_{\text{sat}0}^b = \begin{cases} q_{\text{sat},w0}^b & (T_0^b > T_0) \\ q_{\text{sat}i0}^b & (T_0^b < T_i) \\ q_{\text{sat}i0}^b + (q_{\text{sat},w0}^b - q_{\text{sat}i0}^b) \frac{T_0^b - T_i}{T_0 - T_i} & (\text{otherwise}) \end{cases} \quad (3.1.20)$$

ここで、 $T_0 = 273.15[\text{K}]$ 、 $T_0 - T_i = 36[\text{K}]$ である。また、すべての制御変数は、スカラーポイントに配置する。制御変数間の誤差相関については、 T_g, p_s, θ 間の誤差相関と w_g, μ_p 間の誤差相関があると仮定する。それ以外の変数は、本田 (2010) の調査と議論に基づき独立であるとする。

3.1.5 背景誤差共分散行列

背景誤差共分散行列 \mathbf{B}_0 は、(3.1.15) 式から

$$\mathbf{B}_0 = \mathbf{B}_0^{1/2} \mathbf{B}_0^{T/2} \quad (3.1.21)$$

と書け、その構造は、以下で示す変数変換 (3.1.12) 式の構成により与えられる。 $\mathbf{B}_0^{1/2}$ をパラメータ変換 \mathbf{K}_p と鉛直座標変換 \mathbf{C}_v 、空間鉛直方向の誤差共分散部分 $\mathbf{B}_v^{1/2}$ 、空間水平方向の誤差相関部分 $\mathbf{B}_h^{1/2}$ に分解すると、(3.1.12) 式は、

$$\delta \mathbf{x}_0 = \mathbf{K}_p \mathbf{C}_v \mathbf{B}_v^{1/2} \mathbf{B}_h^{1/2} \delta \chi_0 \quad (3.1.22)$$

と書ける。

$\mathbf{C}_v \mathbf{B}_v^{1/2} \mathbf{B}_h^{1/2}$ は、誤差相関の無い制御変数を関係づける成分が 0 であるブロック対角行列となる。以下では、水平誤差相関 $\mathbf{B}_h^{1/2}$ 、鉛直誤差共分散 $\mathbf{B}_v^{1/2}$ 、鉛直座標変換 \mathbf{C}_v 、パラメータ変換 \mathbf{K}_p の設定を述べる。

(1) 水平誤差相関

水平誤差相関は、水平 2 次元のガウシアンを仮定し、その分散は、東西、南北の 2 方向で与え水平一様とする。ここでは、鉛直格子番号 k の任意の制御変数を表す χ_k に対応する \mathbf{B}_h を \mathbf{B}_{h,χ_k} とおく。この、 \mathbf{B}_{h,χ_k} の演算には、自己随伴型の準ガウシアンフィルタとして機能する等方リカーシブフィルタ (Purser et al. 2003) を採用した。以下では、簡単のため 1 次元ベクトル ψ を用いてリカーシブフィルタの概要を説明する。まず、微分演算子 $-d^2/dx^2$ の有限差分演算子 K を次のように定義する。

$$\frac{K(\psi)_i}{\Delta x^2} = -\frac{\psi_{i-1} - 2\psi_i + \psi_{i+1}}{\Delta x^2} \quad (3.1.23)$$

このとき、拡散方程式の解法と同様にフーリエ変換を行うと、波数 k によって記述されるスペクトル表現の演算子

$$\hat{K}(k) = \left[2 \sin \left(\frac{k\Delta x}{2} \right) \right]^2 \quad (3.1.24)$$

が得られる。(3.1.24) 式を $k^2 \Delta x^2$ について解くと、

$$\sin^{-1} z = \sum_{i=0}^{\infty} \gamma_i z^{2i+1}, \quad |z| < 1 \quad (3.1.25)$$

$$\gamma_i = \frac{1}{2i+1} \cdot \frac{(2i-1)!!}{(2i)!!} \quad (3.1.26)$$

であることから、

$$(k^2 \Delta x^2)^i = \sum_{j \geq i} b_{i,j} \hat{K}^j \quad (3.1.27)$$

と書ける。ここで、

$$b_{1,j} \equiv \sum_{k=0}^{j-1} \frac{\gamma_k \gamma_{j-1-k}}{4^{j-1}} = \frac{2(j!)^2}{j^2 (2j)!} \quad (3.1.28)$$

$$b_{i,j} = \sum_{k=1}^{j+1-i} b_{1,k} b_{i-1,j-k} \quad (3.1.29)$$

である。スケール因子 $a = \sigma \Delta x$ を用いて微分演算子 $\mathcal{D}_{(n)}$ を

$$\mathcal{D}_{(n)} = 1 + \dots + \frac{1}{n!} \left(-\frac{a^2}{2} \frac{d^2}{dx^2} \right)^n \quad (3.1.30)$$

で定義すると、(3.1.30) 式の波数 k のスペクトルは、

$$\hat{\mathcal{D}}_{(n)} = 1 + \dots + \frac{1}{n!} \left(\frac{a^2 k^2}{2} \right)^n \quad (3.1.31)$$

となる。(3.1.31) 式に (3.1.27) 式を代入し、 \hat{K} の関数 $\hat{\mathcal{D}}_{(n)}$ から K の関数 $\mathcal{D}_{(n)}^*$ に逆変換する。さらに、次数 n で切断すると次式が得られる。

$$\mathcal{D}_{(n)}^* = 1 + \dots + \left[\sum_{j=1}^n \frac{b_{j,n}}{j!} \left(\frac{\sigma^2}{2} \right)^j \right] K^n \quad (3.1.32)$$

ここで、(3.1.31) 式は、

$$\lim_{n \rightarrow \infty} \hat{\mathcal{D}}_{(n)} = \exp \left(\frac{a^2 k^2}{2} \right) \quad (3.1.33)$$

を満たすことから、(3.1.33) 式の逆数は、ガウシアンカーネルをもつ畳み込み演算のフーリエ変換に相当し、 $(\mathcal{D}_{(n)}^*)^{-1}$ は、 n 次の準ガウシアンフィルタと見なすことが出来る。リカーシブフィルタは、 x 方向に作用させた後に y 方向に作用させるため、 x 方向の演算を $\mathbf{B}_{h_x, \chi_k}^{1/2}$ 、 y 方向の演算を $\mathbf{B}_{h_y, \chi_k}^{1/2}$ とおくと、 $\mathbf{B}_{h, \chi_k}^{1/2}$ は、

$$\mathbf{B}_{h, \chi_k}^{1/2} \equiv \mathbf{B}_{h_y, \chi_k}^{1/2} \mathbf{B}_{h_x, \chi_k}^{1/2} \quad (3.1.34)$$

と書ける。無限領域で相関距離が水平一様な場合、 $\mathbf{B}_{h_x, \chi_k}^{-1/2}$ は、 $\mathbf{B}_{h_x, \chi_k}^{-1/2} = \mathbf{U}_{h_x, \chi_k}^T \mathbf{U}_{h_x, \chi_k}$ のようにコレスキー分解が可能であり、この逆行列は

$$\mathbf{B}_{h_x, \chi_k}^{1/2} = \mathbf{U}_{h_x, \chi_k}^{-1} \mathbf{U}_{h_x, \chi_k}^{-T} \quad (3.1.35)$$

となる。ここで \mathbf{U}_{h_x, χ_k} は、上三角行列である。任意の x 方向の 1 次元ベクトル \mathbf{p} が入力として与えられたとき、中間ベクトル \mathbf{q} 、出力ベクトル \mathbf{s} とすると、 $\mathbf{U}_{h_x, \chi_k}^{-1} \mathbf{U}_{h_x, \chi_k}^{-T}$ の演算は、次の 2 つの計算に分割できる。

$$\mathbf{q} = \mathbf{U}_{h_x, \chi_k}^{-T} \mathbf{p} \quad (3.1.36)$$

$$\mathbf{s} = \mathbf{U}_{h_x, \chi_k}^{-1} \mathbf{q} \quad (3.1.37)$$

(3.1.36) 式、(3.1.37) 式は、次の漸化式に書き下せる。

$$q_i = \beta p_i + \sum_{j=1}^n \alpha_j q_{i-j} \quad (3.1.38)$$

$$s_i = \beta q_i + \sum_{j=1}^n \alpha_j s_{i+j} \quad (3.1.39)$$

ここで、 i は、 x 方向の格子番号である。まず、(3.1.38) 式を、 x 方向前方に向かって解き、次に (3.1.39) 式を、 x 方向後方に向かって解く。ここで、漸化式の係数は $\beta = 1/(\mathbf{U}_{h_x, \chi_k})_{ii}$, $\alpha_j = -(\mathbf{U}_{h_x, \chi_k})_{i, i+j}/\beta$ で与えられる。 $\mathbf{B}_{h_y, \chi_k}^{-1/2}$ の演算も $\mathbf{B}_{h_x, \chi_k}^{-1/2}$ の演算と同様である。有限領域 $i \in [1, N]$ における境界条件は、フィルタ演算の前方・後方演算の可換性から要請される

$$\mathbf{s} = \mathbf{U}_{h_x, \chi_k}^{-1} \mathbf{U}_{h_x, \chi_k}^{-T} \mathbf{p} \equiv \mathbf{U}_{h_x, \chi_k}^{-1} \mathbf{q} \quad (3.1.40)$$

$$= \mathbf{U}_{h_x, \chi_k}^{-T} \mathbf{U}_{h_x, \chi_k}^{-1} \mathbf{p} \equiv \mathbf{U}_{h_x, \chi_k}^{-T} \mathbf{r} \quad (3.1.41)$$

を利用して与える。ここで、 \mathbf{r} は、 \mathbf{q} と同様な中間ベクトルである。 \mathbf{s} の一部を $\hat{\mathbf{s}}_j = (s_{j+1-n}, \dots, s_j)^T$ と表記し、下三角行列 $L_{i,i} = 1, L_{i+j,i} = -\alpha_j$ 、上三角行列 $U_{i,i+j} = -\alpha_{n-j}$ とする。 $\hat{\mathbf{r}}_{N+n} = 0$ であることから、(3.1.40) 式と (3.1.41) 式から

$$\mathbf{L} \hat{\mathbf{s}}_{N+n} = \mathbf{U} \hat{\mathbf{s}}_N \quad (3.1.42)$$

$$\mathbf{L}^T \hat{\mathbf{s}}_N = \mathbf{U}^T \hat{\mathbf{s}}_{N+n} + \beta \hat{\mathbf{q}}_N \quad (3.1.43)$$

となり、境界条件である

$$(\mathbf{L}^T - \mathbf{U}^T \mathbf{L}^{-1} \mathbf{U}) \hat{\mathbf{s}}_N = \beta \hat{\mathbf{q}}_N \quad (3.1.44)$$

が与えられる。境界では、(3.1.44) 式を用いて、 x 方向後方に向かって \mathbf{s} を求める。リカーシブフィルタは、次数 n の取り方と適用回数に任意性があり、次数や適用回数を増やすと、近似誤差が縮小する。しかし、次数や適用回数の増加は、計算量と MPI 通信量の増加を招き計算時間の増大に繋がる。そこで、 $\mathbf{B}_{h_x, \chi_k}^{1/2}$ と $\mathbf{B}_{h_y, \chi_k}^{1/2}$ に利用するリカーシブフィルタでは、実用上十分な精度が得られる次数 $n = 4$ を採用し適用回数を 1 回に設定している。

(2) 鉛直誤差共分散

水平格子番号 (i, j) の任意の制御変数を表す χ_{ij} に対応する \mathbf{B}_v を $\mathbf{B}_{v, \chi_{ij}}$ とおくと、 $\mathbf{B}_{v, \chi_{ij}}^{1/2}$ は、 $\mathbf{B}_{v, \chi_{ij}}$ の固有値を対角要素を持つ対角行列 $\mathbf{\Lambda}_{v, \chi_{ij}}$ の平方根を用いて次の様に書ける。

$$\mathbf{B}_{v, \chi_{ij}}^{1/2} = \mathbf{U}_{v, \chi_{ij}} \mathbf{\Lambda}_{v, \chi_{ij}}^{1/2} \mathbf{U}_{v, \chi_{ij}}^T \quad (3.1.45)$$

ここで、 $\mathbf{U}_{v, \chi_{ij}}$ は、 $\mathbf{B}_{v, \chi_{ij}}$ の固有ベクトルを列に持つ直交行列である。

なお、 $\mathbf{B}_{v, \chi_{ij}}^{1/2}$ は、水平格子点 (i, j) 毎に異なる値を持つことができ、鉛直背景誤差の場所依存性を表現できる。例えば、海格子と陸格子で鉛直誤差共分散の統計をとると、境界層に相当する高度以下で海と陸で異なる共分散が得られる。そのような場所依存性を持つ鉛直誤差共分散の利用は、地形や地表面種別の影響を解析値に反映することに繋がり、局地的な気象現象を対象としたデータ同化を行う際に有益である。

(3) 鉛直座標変換

水平格子番号 (i, j) の任意の制御変数を表す χ_{ij} に対応する鉛直座標変換 \mathbf{C}_v を $\mathbf{C}_{v, \chi_{ij}}$ とおく。制御変数に作用する水平方向の空間フィルタ $\mathbf{B}_{h_x, \chi_k}^{1/2}$ は、モデル座標系の鉛直軸に沿った方向に離散化された層毎に計算を行うため、観測による修正を層に沿って広げる性質を持つ。例えば、モデルに z^* 座標系 (Gal-Chen and Somerville 1975) が採用されている場合、大気上層の観測による修正が、モデル地形に依存することになる (藤田・倉橋 2010)。これを軽減するため、 χ が配置される座標系に、地形の影響が大気上層に及ばないように z_h を低く設定した鉛直ハイブリッド座標系 (石田 2008) を採用することとした。この鉛直ハイブリッド座標系からモデルの層配置への変換は、 $\mathbf{B}_{h, \chi}$ と $\mathbf{B}_{v, \chi_{ij}}$ の演算の後、 $\mathbf{C}_{v, \chi_{ij}}$ を作用させることで実現される。この鉛直座標変換は、プログラム開始時に、一度だけ計算し、 $\mathbf{C}_{v, \chi_{ij}} \mathbf{B}_{v, \chi_{ij}}^{1/2}$ を保存しておく。これにより、極小値探索の繰り返し回数分の鉛直座標変換を省略できる。

(4) パラメータ変換

\mathbf{K}_p を書き下すため、(3.1.22) 式を $\delta \mathbf{x}_0 = \mathbf{K}_p \delta \tilde{\chi}_0$ と書き直す。モデル変数の変分ベクトル $\delta \mathbf{x}_0$ と、解析変数に対応する要素の変分ベクトル $\delta \tilde{\chi}_0$ は、

$$\delta \mathbf{x}_0 = \begin{pmatrix} \delta(\rho u) \\ \delta(\rho v) \\ \delta(\rho \theta_m) \\ \delta(\rho q_v) \\ \delta \rho \\ \delta T_g \\ \delta w_g \end{pmatrix}, \quad \delta \tilde{\chi}_0 = \begin{pmatrix} \delta \tilde{u} \\ \delta \tilde{v} \\ \delta \tilde{\theta} \\ \delta \tilde{p}_s \\ \delta \tilde{\mu}_p \\ \delta \tilde{T}_g \\ \delta \tilde{w}_g \end{pmatrix} \quad (3.1.46)$$

で与えられる。ここで、 $\delta \mathbf{x}_0$ の要素は、 $\delta \tilde{\chi}_0$ によって摂動を考慮する必要がある変数のみを記述する。

\mathbf{K}_p の構成要素を $K_{x_0, \tilde{\chi}_0}$ と置き変換内容を書き下す。添字は、変換対象の (3.1.46) 式に表れる構成要素の対を表す。まず、 $K_{\rho u, \tilde{u}} = \rho, K_{\rho v, \tilde{v}} = \rho, K_{T_g, \tilde{T}_g} = \mathbf{I}, K_{w_g, \tilde{w}_g} = \mathbf{I}$ となる。これら以外で 0 ではない \mathbf{K}_p の部分行列は、

$$\begin{pmatrix} K_{\rho u, \tilde{\theta}} & K_{\rho u, \tilde{p}_s} & K_{\rho u, \tilde{\mu}_p} \\ K_{\rho v, \tilde{\theta}} & K_{\rho v, \tilde{p}_s} & K_{\rho v, \tilde{\mu}_p} \\ K_{\rho \theta_m, \tilde{\theta}} & K_{\rho \theta_m, \tilde{p}_s} & K_{\rho \theta_m, \tilde{\mu}_p} \\ K_{\rho q_v, \tilde{\theta}} & K_{\rho q_v, \tilde{p}_s} & K_{\rho q_v, \tilde{\mu}_p} \\ K_{\rho, \tilde{\theta}} & K_{\rho, \tilde{p}_s} & K_{\rho, \tilde{\mu}_p} \end{pmatrix} = \begin{pmatrix} 0 & 0 & u \\ 0 & 0 & v \\ \rho & 0 & \theta_m \\ 0 & \rho & q_v \\ 0 & 0 & \mathbf{I} \end{pmatrix} \times \begin{pmatrix} \mathcal{T}_\theta & 0 & \mathcal{T}_{\mu_p} \\ 0 & 0 & \mathcal{Q}_{\mu_p} \\ \mathcal{D}_{\theta_m} \mathcal{T}_\theta & \mathcal{D}_{p_s} & \mathcal{D}_{\theta_m} \mathcal{T}_{\mu_p} \end{pmatrix} \quad (3.1.47)$$

である。(3.1.47) 式の行列内の記号は、以下の関係式を満たすものである。

$$\delta q_v = \frac{\partial q_v}{\partial \tilde{\mu}_p} \delta \tilde{\mu}_p = \mathcal{Q}_{\mu_p} \delta \tilde{\mu}_p \quad (3.1.48)$$

$$\begin{aligned}\delta\theta_m &= \frac{\partial\theta_m}{\partial\tilde{\theta}}\delta\tilde{\theta} + \frac{\partial\theta_m}{\partial\tilde{\mu}_p}\delta\tilde{\mu}_p \\ &= \mathcal{T}_{\tilde{\theta}}\delta\tilde{\theta} + \mathcal{T}_{\tilde{\mu}_p}\delta\tilde{\mu}_p\end{aligned}\quad (3.1.49)$$

$$\delta\rho = \mathcal{D}_{\theta_m}\delta\theta_m + \mathcal{D}_{p_s}\delta p_s \quad (3.1.50)$$

なお、(3.1.50)式は、静力学平衡の関係 $\nabla\pi = -\frac{g}{C_p\theta_m}$ の接線形方程式から $\delta\pi$ を求め、 ρ の変分を求める演算を表す。

3.1.6 質量保存の拘束条件

asuca-Var では、制御変数を定義する際に、質量に関する制御変数と、風に関する制御変数との間に誤差相関が無いと仮定した。そのため、モデルの強拘束の無い3次元変分法から得られる解析値は、質量と風が無関係に修正されたものになる。その様な解析値から時間積分を開始すると不自然な波が発生する。そこで、波の発生を抑えるため、質量と風の修正が相互に関係するように、連続の式を満たす拘束条件を導入する。まず変換演算子 \mathbf{C}_c を

$$\begin{aligned}\mathbf{C}_c\delta\mathbf{x} &= \left[\frac{\delta(\rho U)}{J}\right]_{i-\frac{1}{2}}^{i+\frac{1}{2}} + \left[\frac{\delta(\rho V)}{J}\right]_{j-\frac{1}{2}}^{j+\frac{1}{2}} \\ &\quad + \left[\frac{\delta(\rho W)}{J}\right]_{k-\frac{1}{2}}^{k+\frac{1}{2}}\end{aligned}\quad (3.1.51)$$

と定義する。これは、有限体積法表記の連続の式である(2.2.5)式から、摂動を考慮していない降水の落下項と時間変化の項を除いたものに相当する。拘束項 J_{p_c} は、この \mathbf{C}_c を用いて

$$J_{p_c} = \frac{1}{2}\delta\chi_0^T \mathbf{B}_0^T \mathbf{C}_c^T \mathbf{\Lambda}_c \mathbf{C}_c \mathbf{B}_0^{1/2} \delta\chi_0 \quad (3.1.52)$$

となる。 J_{p_c} の勾配は

$$\frac{\delta J_{p_c}}{\delta\chi_0} = \mathbf{B}_0^T \mathbf{C}_c^T \mathbf{\Lambda}_c \mathbf{C}_c \mathbf{B}_0^{1/2} \delta\chi_0 \quad (3.1.53)$$

である。ここで $\mathbf{\Lambda}_c$ は、格子点毎の拘束の強さを規定する対角行列である。その採り方は任意であり、予報開始時に発生する不自然な波の抑制効果と、評価関数の振る舞いを基に決めるチューニングパラメータである。

3.1.7 変分法品質管理

変分法品質管理 (Andersson and Järvinen 1999) は、変分法データ同化において、評価関数の極小値探索を行いながら観測値の品質管理を行う手法である。ある観測 y_t に対して、観測測器の障害や人為的ミス等により、通常の測定誤差より明らかに大きな誤差の発生する確率を p_g とし、発生する区間を観測誤差の標準偏差 σ_o の l 倍で定義する。評価関数(3.1.3)式の第2項から任意の観測の項を一つ抜きだし

$$J_o^N = (\mathcal{H}_t(\mathbf{x}_t) - y_t)^2 / 2\sigma_o^2 \quad (3.1.54)$$

とすると、 p_g を考慮した確率密度分布は、

$$p_{qc} = \frac{1-p_g}{\sqrt{2\pi}\sigma_o} e^{-J_o^N} + \frac{p_g}{2l\sigma_o} \quad (3.1.55)$$

となる。ここで、 p_g は、一様分布を仮定した。観測値の確率密度分布を p_{qc} とした場合の評価関数の観測に関する項を J_o^{QC} とすると、 J_o^N を用いて

$$J_o^{QC} = -\ln\left(\frac{\gamma + e^{-J_o^N}}{\gamma + 1}\right) \quad (3.1.56)$$

と書ける。この勾配は、

$$\frac{\delta J_o^{QC}}{\delta\chi_0} = \frac{\delta J_o^N}{\delta\chi_0} \cdot \left(1 - \frac{\gamma}{e^{-J_o^N} + \gamma}\right) \quad (3.1.57)$$

で与えられる。ここで、 $\gamma = \sqrt{2\pi}p_g / [2l(1-p_g)]$ である。

3.1.8 観測演算子

観測演算子 \mathcal{H} は、次式のように空間変換演算子 \mathcal{H}_s と、物理要素変換演算子 \mathcal{H}_p に分離し実装する。

$$\mathcal{H}(\mathbf{x}) = \mathcal{H}_s[\mathcal{H}_p(\mathbf{x})] \quad (3.1.58)$$

この接線形演算子 \mathbf{H} は、 \mathcal{H}_p , \mathcal{H}_s の接線形演算子 \mathbf{H}_p , \mathbf{H}_s を用いて、

$$\mathbf{H}\delta\mathbf{x} = \mathbf{H}_s\mathbf{H}_p\delta\mathbf{x} \quad (3.1.59)$$

と書ける。 \mathcal{H}_s ($= \mathbf{H}_s$) は、近接8格子からの線形内挿である。ただし、地上観測や輝度温度などの平面上のデータの場合は、近接4格子からの線形内挿である。観測演算子の新規追加は、 \mathbf{H}_s を意識せずに、決められたインターフェースに合うように、新たな観測の \mathcal{H}_p と \mathbf{H}_p 及びその随伴コード \mathbf{H}_p^T を作成し実装すればよい。

実装している観測演算子は、大気東西風、南北風、気温、相対湿度に加え、地上観測の東西風、南北風、気温、比湿、相対湿度、気圧である。更に、ドップラー速度、地上可降水量、輝度温度、土壌含水率についても実装している。

なお、相対湿度の観測演算子 \mathcal{H}_{rh} は、

$$\mathcal{H}_{rh}(\mathbf{x}) = \frac{p q_v \frac{R_v}{R_d}}{\left[1 - \sum_{\alpha \neq v} q_\alpha + \left(\frac{R_v}{R_d} - 1\right) q_v\right]} e_{\text{sat}}(T) \quad (3.1.60)$$

で与えられる。ここで、 $\sum_{\alpha \neq v} q_\alpha$ は、 q_v 以外の水物質の和を表し、 $e_{\text{sat}}(T)$ は、WMO で定められたラジオゾンデや地上観測の通報基準に従い、水の飽和水蒸気圧である (WMO 2008)。また、可降水量の観測演算子 \mathcal{H}_{tpw} は、次式で与えられる。

$$\mathcal{H}_{tpw}(\mathbf{x}) = \sum_{k=1}^{n_z} \left(z_{k+\frac{1}{2}} - z_{k-\frac{1}{2}}\right) (\rho q_v)_k \quad (3.1.61)$$

ドップラー速度の観測演算子 \mathcal{H}_{dpr} は、サイトからの距離を d 、ビームの方位角を θ 、仰角を ϕ 、観測高度を z として

$$\mathcal{H}_{\text{dpr}}(\mathbf{x}) = \frac{\sum_{k=1}^{n_z} (u_k \sin \theta + v_k \cos \theta) \exp\left[-\frac{(z_k - z)^2}{l^2}\right]}{\sum_{k=1}^{n_z} \exp\left[-\frac{(z_k - z)^2}{l^2}\right]} \quad (3.1.62)$$

で与えられる。ここで、 $l = d\delta\phi$ であり、ビームの広がりを表す $\delta\phi$ は、0.3 度とした (石川 2007; 小泉 2004; Seko et al. 2004)。

地上観測の観測演算子は、気象庁の物理過程ライブラリの地上物理量診断 (原 2008) を利用している。初期設定では、地上物理量診断で用いるバルク係数は、陸上と海上共に Beljaars and Holtslag (1991) のフラックススキームによって計算される。また、陸地の蒸発効率 β は、Deardorff (1978) に基づき、 w_g から診断される。

衛星輝度温度の観測演算子は、欧州気象衛星開発機構 (EUMETSAT) の高速放射伝達モデル (RTTOV v10.2; Saunders et al. 2012) を利用している。asuca-Var の観測演算子は、パラメータで接線形や随伴の演算機能を切り替える仕様としていることから、K-matrix は用いず、接線形演算子のサブルーチン `rttov_t1` と随伴演算子のサブルーチン `rttov_ad` を個別に呼び出している。また、非線形演算が選択された場合には直接演算用のサブルーチン `rttov_direct` を利用することになる。オゾン、予報モデルの放射過程と同じ気候値を与えている。処理対象は、晴天域の輝度温度のみとしており、雲水量の入力は行っておらず、高速マイクロ波射出モデル (FASTEM) も無効としている。

3.1.9 並列化

asuca-Var は、計算効率を向上させるため asuca と同様に領域 2 次元分割を行っている。第 3.1.5 項で述べた水平誤差相関の計算に用いられるリカーシブフィルタは、全計算領域を x 方向前方・後方、 y 方向前方・後方の順にノード間をまたいで計算していく。この計算を最小の通信量で行うため、各ノードでリカーシブフィルタの近似次数分の格子数を「のりしろ」として用意し、この「のりしろ」を順次交換していく方法を採用した。また、観測演算子の呼び出しは、各ノードに割り当てられた領域内に含まれる観測の数だけ繰り返される。この繰り返し部分を OpenMP で並列化し、計算時間の短縮を図っている。

3.1.10 背景場に流れ依存性を考慮する手法

(1) 4次元変分法データ同化

本稿の執筆時 (2014 年 1 月) では、asuca の力学過程の TL、AD コードが概ね完成し、物理を含まない 4 次

元変分法データ同化の動作確認を行っている。4 次元化に際し、数分または数秒間隔で観測値が得られる気象レーダー・ウィンドプロファイラや GPS ゾンデ等の高頻度観測データ同化を効率的に行うため \mathbf{d}_t の計算をモデルの時間積分中に行う仕組みを実装した。これによって、 \mathbf{d}_t の計算のために NL の結果を出力する必要がなくなり、データ出力の負荷を抑えながら、実際の観測時刻に近い時刻で観測データを同化することが可能となる。

(2) ハイブリッド変分法データ同化

asuca-Var では、Lorenz (2003)、Buehner (2005) に基づき、変分法データ同化における背景誤差にアンサンブル摂動を利用して背景場の流れ依存性を考慮する試みを行っている。ここでは、アンサンブル摂動を用いたハイブリッド変分法手法の概略を示す。統計的な背景誤差 $\mathbf{B}_{n,0}$ を用いる制御変数 $\chi_{n,0}$ に加えて、アンサンブル摂動から得られる背景誤差 $\mathbf{B}_{e,0}$ を利用する拡張制御変数 $\chi_{e,0}$ を追加する。それぞれの重みを β_n, β_e とし、(3.1.12) 式を

$$\delta\mathbf{x}_0^b = \beta_n \mathbf{B}_{n,0}^{1/2} \delta\chi_{n,0} + \beta_e \mathbf{B}_{e,0}^{1/2} \delta\chi_{e,0} \quad (3.1.63)$$

のように拡張する。このとき、背景誤差 \mathbf{B}_0 は

$$\mathbf{B}_0 = \beta_n^2 \mathbf{B}_{n,0} + \beta_e^2 \mathbf{B}_{e,0} \quad (3.1.64)$$

となる。ここで、 $\mathbf{B}_{e,0}^{1/2}$ は、 i 番目のメンバーのアンサンブル摂動 $\delta\mathbf{x}_{e,i}$ 、アンサンブルメンバー総数の N_e 、局所化演算子 \mathbf{L} を用いて

$$\mathbf{B}_{e,0}^{1/2} = (N_e - 1)^{-1/2} \times \left[\text{diag}(\delta\mathbf{x}_{e,1}) \mathbf{L}^{1/2}, \dots, \text{diag}(\delta\mathbf{x}_{e,N_e}) \mathbf{L}^{1/2} \right] \quad (3.1.65)$$

で与える。(3.1.63) 式、(3.1.64) 式と (3.1.65) 式から、最小化問題 (3.1.16) 式は、

$$\begin{aligned} & \min_{\delta\chi_0^{(i)}} J \left(\delta\chi_{n,0}^{(i)}, \delta\chi_{e,0}^{(i)} \right) \\ &= \min_{\delta\chi_0^{(i)}} \left[\frac{1}{2} \left\| \delta\chi_{n,0}^{(i)} \right\|_2^2 + \frac{1}{2} \left\| \delta\chi_{e,0}^{(i)} \right\|_2^2 \right. \\ & \quad \left. + \frac{1}{2} \left\| \hat{\mathbf{R}}^{-1/2} \hat{\mathbf{H}} \left(\beta_n \mathbf{B}_{n,0}^{1/2} \delta\chi_{n,0}^{(i)} + \beta_e \mathbf{B}_{e,0}^{1/2} \delta\chi_{e,0}^{(i)} \right) \right. \right. \\ & \quad \left. \left. - \hat{\mathbf{R}}^{-1/2} \hat{\mathbf{d}} \right\|_2^2 + J_p \left(\delta\chi_{n,0}^{(i)}, \delta\chi_{e,0}^{(i)} \right) \right] \quad (3.1.66) \end{aligned}$$

に拡張される。局地解析におけるハイブリッド変分法手法の有用性は、藤田 (2010) で実証されているが、局所化のスケール、背景誤差の混合の割合等の詳細は、アンサンブル摂動を与えるアンサンブルシステムの設定に依存するため、性能評価を行いつつ最適な設定を検討していく必要がある。

参考文献

- Andersson, E. and H. Järvinen, 1999: Variational quality control. *Quart. J. Roy. Meteor. Soc.*, **125**, 697–722.
- 淡路敏之, 池田元美, 石川洋一, 蒲地政文, 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.
- Bannister, R. N., 2008: A review of forecast error covariance statistics in atmospheric variational data assimilation. II: Modelling the forecast error covariance statistics. *Quart. J. Roy. Meteor. Soc.*, **134**, 1971–1996.
- Beljaars, A. C. M. and A. A. M. Holtslag, 1991: Flux Parameterization over Land Surfaces for Atmospheric Models. *J. Appl. Meteor.*, **30**, 327–341.
- Buehner, M., 2005: Ensemble-derived stationary and flow-dependent background error covariances: Evaluation in a quasi-operational NWP setting. *Quart. J. Roy. Meteor. Soc.*, **131**, 1013–1043.
- Deardorff, J. W., 1978: Efficient Prediction of Ground Surface Temperature and Moisture, With Inclusion of a Layer of Vegetation. *J. Geophys. Res.*, **83**, 1889–1903.
- Dee, D. P. and A. M. da Silva, 2003: The Choice of Variable for Atmospheric Moisture Analysis. *Mon. Wea. Rev.*, **131**, 155–171.
- 藤田匡, 倉橋永, 2010: 局地解析. 数値予報課報告・別冊第 56 号, 気象庁予報部, 68–72.
- 藤田匡, 2010: 流れに依存する背景誤差. 数値予報課報告・別冊第 56 号, 気象庁予報部, 73–83.
- Gal-Chen, T. and R.C.J. Somerville, 1975: On the use of a coordinate transformation for the solution of the Navier-Stokes equation. *J. Comput. Phys.*, **17**, 209–228.
- 原旅人, 2008: 地上物理量診断. 数値予報課報告・別冊第 54 号, 気象庁予報部, 181–184.
- 原旅人, 2012: 物理過程ライブラリの開発. 数値予報課報告・別冊第 58 号, 気象庁予報部, 205–208.
- 本田有機, 2010: 制御変数. 数値予報課報告・別冊第 56 号, 気象庁予報部, 14–16.
- 石田純一, 2008: 気象庁非静力学モデルの支配方程式系と地形に沿う鉛直ハイブリッド座標の導入. 数値予報課報告・別冊第 54 号, 気象庁予報部, 27–43.
- 石川宜広, 2007: ドップラーレーダーデータの利用. 平成 19 年度数値予報研修テキスト, 気象庁予報部, 102–103.
- 気象庁予報部, 2010: 非静力学メソ 4 次元変分法. 数値予報課報告・別冊第 56 号, 気象庁予報部, 106pp.
- 小泉耕, 2004: 非静力学メソ数値予報モデルの現業化. 平成 16 年度数値予報研修テキスト, 気象庁予報部, 71–74.
- Lahoz, W., B. Khattatov, and R. Menard, 2010: *Data Assimilation: Making Sense of Observations*. Springer, 475pp.
- Lorenc, A. C., 2003: The potential of the ensemble Kalman filter for NWP—a comparison with 4D-Var. *Quart. J. Roy. Meteor. Soc.*, **129**, 3183–3203.
- Purser, R. J., D. F. Parrish, and N. M. Roberts, 2003: Numerical Aspects of the Application of Recursive Filters to Variational Statistical Analysis. Part I: Spatially Homogeneous and Isotropic Gaussian Covariances. *Mon. Wea. Rev.*, **131**, 1536–1548.
- Saunders, R., M. Matricardi, and P. Brunel, 1999: An improved fast radiative transfer model for assimilation of satellite radiance observations. *Quart. J. Roy. Meteor. Soc.*, **125**, 1407–1425.
- Saunders, R., J. Hocking, P. Rayner, M. Matricardi, A. Geer, N. Bormann, P. Brunel, F. Karbou, and F. Aires, 2012: RTTOV-10 science and validation report. available from http://research.metoffice.gov.uk/research/interproj/nwpsaf/rtm/rtm_rttov10.html.
- Seko, H., T. Kawabata, T. Tsuyuki, H. Nakamura, K. Koizumi, and T. Iwabuchi, 2004: Impacts of GPS-derived Water Vapor and Radial Wind Measured by Doppler Radar on Numerical Prediction of Precipitation. *J. Meteor. Soc. Japan*, **82**, 473–489.
- WMO, 2008: *Guide to Meteorological Instruments and Methods of Observation*.

3.2 接線形・随伴モデルの実装¹

3.2.1 はじめに

第 1.4.3 項及び第 3.1 節で述べられているように、asuca を基にした変分法データ同化システム (asuca-Var) の開発が既に始まっており、現在 3 次元変分法ベースのシステムが完成しつつある。今後は asuca-Var を 3 次元変分法から 4 次元変分法へと拡張する計画であるが、そのためには asuca の接線形・随伴モデル (tangent linear/adjoint model: TL/AD) が必要となる。

数値予報課では asuca-Var の 3 次元変分法の開発時点で既に 4 次元変分法への拡張を見据え、TL/AD の開発方針の検討を行った。この際、現在運用されている気象庁非静力学モデル (JMA-NHM) に基づく非静力学メソ 4 次元変分法データ同化システム (JNoVA) の TL/AD 開発時の問題点を議論し、後述するように最新の予報モデルに追従可能な TL/AD の開発体制を構築することを目標とした方針を定めた上で、まず物理過程を考慮しない asuca 力学過程の TL/AD の開発から着手した。

この節では、今後 TL/AD の新規開発を行う際の検討及び新規に取り組む開発者の参考となることを目的として、asuca-Var における TL/AD の開発方針を定めるにあたり検討した内容及び実際の開発手順やその具体例を述べる。最後に今後の開発について述べる。

3.2.2 接線形・随伴モデル

TL/AD については露木 (1997) や石川・藤田 (2013) などで詳しい解説がなされているが、ここでも簡単に解説する。TL/AD は 4 次元変分法における評価関数の勾配計算やアンサンブル予報用の初期値摂動のための特異ベクトル計算に用いられる。TL は予報モデルの各処理を線形化したモデルであり、摂動の時間発展を計算する。TL 内の各処理は線形化されているため行列を用いて表すことができる。そのため、TL による 1 タイムステップの演算は行列の積を用いて表すことができる。ここで 1 タイムステップ当たり N 個の処理から成ると仮定し、摂動変数から成るベクトルを \mathbf{x} 、各処理を行列 $\mathbf{M}_i (i = 1, \dots, N)$ で書けるとすると、TL の 1 タイムステップにおける処理は、

$$\mathbf{x}(t + \Delta t) = \mathbf{M}_N \mathbf{M}_{N-1} \cdots \mathbf{M}_1 \mathbf{x}(t)$$

と表すことができる。対応する AD の予報変数 $\hat{\mathbf{x}}(t)$ についての式は、

$$\hat{\mathbf{x}}(t) = \mathbf{M}_1^T \cdots \mathbf{M}_{N-1}^T \mathbf{M}_N^T \hat{\mathbf{x}}(t + \Delta t)$$

と表される (添字 T は転置を表す)。この転置行列の積の部分が 1 タイムステップにおける AD の処理に相当する²。AD は TL で時間発展した摂動の初期時刻にお

¹ 小野 耕介

² 実行列において転置行列と随伴行列は同義であり、このため随伴モデルと呼ばれる。また、 $\mathbf{M}_i^T = \mathbf{M}_i^{-1}$ の場合に限り

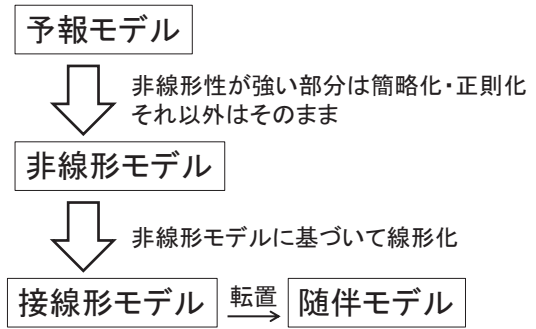


図 3.2.1 予報モデルと非線形・接線形・随伴モデルの関係。

ける感度を計算し、TL とは処理の流れが逆になり、摂動の時間積分は未来から初期時刻に遡って行われる。

TL 及び AD は予報モデルに基づいて作成される。しかし、予報モデルの定式化、特に物理過程において、非線形性が強い部分や条件分岐によって演算が変わる部分では、TL による予報モデルの線形近似の精度が悪化し得る。このような場合、予報モデルを簡略化あるいは正則化 (石川・藤田 2013) した上で TL/AD を作成する。簡略化・正則化が行われたモデルを予報モデルとは区別して非線形モデル (nonlinear model: NL) と呼ぶ³。以上の予報モデルと NL 及び TL/AD の関係を図 3.2.1 に示す。

3.2.3 開発の方針

(1) 開発の着手について

新規数値予報モデルの開発初期においては、ソースコードの大規模な改変 (多数のサブルーチン間にまたがるようなコードの修正) を伴うことが多い。そのため、開発の手戻りが発生しないよう、TL/AD の開発は予報モデルの開発がある程度進んで完成度が高くなってから開始することが肝要である。実際に asuca-Var の開発の検討を始めたのは、asuca による実データを用いた予報実験が行われ、その結果を観測データや現業モデルと比較できる段階 (河野ほか (2011) など) に達してからである。実データ実験が行える段階では、理想実験を通した力学過程のフレームに関わる基礎開発は概ね完成していると考えられ、これが TL/AD 開発検討のタイミングとして適切と考えられる。実際の作業にあたっては、今後の大規模なモデルの改変の有無についてモデル開発者と認識を共通にした上で、開発検討を開始することが望ましい。asuca 力学過程の TL/AD の開発では、このような手順を踏んだ結果、実際に作業を開始してから今日までソースコードの大規模な改変は行われていない。また asuca-Var では、3 次元変分法の開発に着手すると同時に 4 次元変分法への拡張を見据えて、力学過程の TL/AD の開発に着手した。

$\mathbf{x} = \hat{\mathbf{x}}$ が成り立つが、通常は成り立たないため $\hat{\mathbf{x}}$ を用いた。

³ 予報モデル自体非線形なモデルであるが、ここでは接線形・随伴モデルとの対比としてこのように呼ぶ。

(2) 開発体制について

asuca-Var の TL/AD 開発にあたっては、以下で述べるような JNoVA における開発経験を踏まえ効率的な開発体制を構築した。

JNoVA における TL/AD の問題点は、最新の予報モデルに基づいておらず、TL/AD の鉛直座標系が JMA-NHM と異なっている点が挙げられる。また、乱流や地表面過程に古いスキームが用いられているなど、最新の開発成果が反映されていない点もある。これらの影響は 3次元変分法も含めシステム全体に影響が及んでいる (本田 2010)。さらに、JNoVA における予報モデルと TL/AD との乖離は高速化にも影響を与える。JMA-NHM は当初南北方向のみの 1次元分割による MPI を用いた並列計算が可能であったが (石田・荒波 2003)、2006 年の水平格子間隔 5km への高解像度化の前に東西方向にも拡張されて 2次元分割による並列計算が可能ないように変更された (荒波・石田 2008)。一方で、JNoVA の開発開始は 2002 年であり (本田 2003)、それ以後に行われた 2次元分割による並列化は TL/AD を含めコードの変更が広範囲に及ぶため反映できておらず、十分な高速化が図られていない原因となっている。

このように JNoVA において TL/AD が最新の予報モデルに基づいていない状況の背景には、従来の同化と予報モデルの開発が分離した体制の下で進められたことにより、TL/AD の開発を限られた同化開発者が手がけ、モデル開発者が直接関与する機会がほとんどなかったことが挙げられる。このような体制の下、同化開発者はまず予報モデルの特定の時点のバージョンを基に TL/AD の開発を開始する。TL/AD の開発では、サブルーチンごとに TL/AD を作成しそのコードを検証していくが、加えてモデル全体として時間積分も含めたコードの検証を行う必要がある。モデル全体の検証では、些細なミスにより検証がうまくいかないことも多く、その原因を特定するためにモデル全体から誤りを探し出す必要があるため、作業に時間がかかる傾向がある。この一方で、モデル開発者は最新の知見に基づきモデルの改良あるいは新規スキームの導入を継続していく。そのため、ある版に基づく TL/AD が一通り完成する間に、予報モデルはさらに開発が進むことから、TL/AD は最新の予報モデルより古いままという状況が続く。JNoVA の TL/AD 完成以後、幾度か予報モデルの開発成果を TL/AD に反映させる作業を続けてはいるが、その成果は一部の内容に限定されている。また、予報モデルは複数の過程から成り、各過程を専門的に扱う複数の開発者が改良を進める一方、TL/AD 開発は必ずしも各過程に関する専門的な知識及び経験を持たない同化開発者が多くの過程に対応せざるを得ない。このように、予報モデルのあらゆる過程の開発成果を少数の同化開発者だけで適切に TL/AD に反映させ維持管理していくことは作業負荷の面から

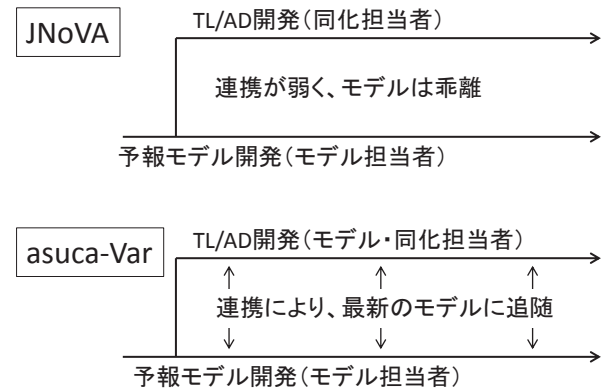


図 3.2.2 JNoVA と asuca-Var における開発体制の違い。

困難で、モデル開発者の協力が必須である⁴。また、物理過程の TL/AD においては簡略化・正則化の作業が必要とされ、物理過程開発者の知見を活かすことにより、効率的かつ性能の高い TL/AD を構築することができる。

以上の JNoVA の開発における問題点を踏まえ、asuca 力学過程の TL/AD 開発では、同化開発者が力学過程の TL/AD 開発を開始し、そのコードレビューをモデル開発者が担当することで、モデル開発者が TL/AD 開発に習熟できる体制をとった。また、TL/AD が完成したサブルーチンにモデル開発者が変更を加える場合は TL/AD も修正することで、最新版の内容を TL/AD に反映できるようにした (図 3.2.2)。以上のような開発者間の連携意識を高めるために、TL/AD 開発をする上で以下のルールを取り決めた。

1. NL 及び TL/AD は予報モデルと同一ファイル内に記述する。
2. コードレビュー体制を利用する。

JNoVA では予報モデルと TL/AD のソースコードが別管理されており、モデルと同化のそれぞれの開発者が別々に開発を行ってきたが、asuca の開発ではソースコードを一元管理することで、モデル開発者が TL/AD の開発についても意識できるようにした。このようなコードの一元管理によって、出力や物理量計算などのライブラリを、予報モデルと TL/AD で共有できるメリットもある。また開発内容のレビュー体制を通じて、コードチェックに加え、前述の通りモデル開発者が TL/AD に習熟することも目指している。

(3) コーディングルールについて

asuca-Var における TL/AD 開発においてはいくつかのコーディングルールを取り決めた。NL 及び

⁴ 一方で、JNoVA では TL/AD の開発中に予報モデルの誤りを見つけるなど、モデル開発へのメリットもあった。

TL/AD は前述の通り予報モデルと同一ファイルで管理するが、サブルーチン名は基となる予報モデルのサブルーチン名に「_tlad」をつけ区別した上で、同一ファイル内に記述した。また、TL/AD のサブルーチンの呼び出しには `ksw_runmode` という変数を用いる。`ksw_runmode` はさらに上位の `isw_runmode` によって制御される。`isw_runmode` は予報モデルの実行あるいは 3 次元変分法の実行など、システム全体の動作を制御する。そして、`isw_runmode` で選択された全体の実行モードに応じた処理過程の実行時に、その都度必要となる適切な `ksw_runmode` を動的に指定した上で、各 TL/AD サブルーチンが呼び出される。実際には、`isw_runmode` をモデルの実行前に陽に設定し、`ksw_runmode` は `isw_runmode` に従って設定されるという動作になる。

また摂動変数を明確にするために、摂動変数には変数名に「7」をつけるように取りきめた⁵。一方で、基本場変数⁶ はそのままの名前を用いている⁷。基本場変数の名前をそのままにすることで、予報モデルが更新された際に、NL の変数名を修正する手間がかからないというメリットがある (具体例は後述の図 3.2.3 を参照)。

3.2.4 実際の開発

TL/AD の作成及び検証方法については、露木 (1997)、露木 (2002)、門脇 (2002)、石川・藤田 (2013) などの文献があつて、ローレンツモデルなどを用いたわかりやすい解説があり参考になる。ここでは、`asuca` 力学過程の TL/AD の実際の開発手順や具体例、注意すべき点を述べる。

(1) 開発手順

`asuca-Var` における力学過程の TL/AD 作成は以下の手順で行っている。

1. 開発ルール及びコーディングルール策定
2. 摂動変数の前処理
3. 単体で閉じたサブルーチンから TL/AD 作成
4. MPI 通信のための随伴コードを作成
5. 単体サブルーチンの集合からなる上位サブルーチンの TL/AD を作成
6. 時間積分に関する部分の作成及びコード全体の検証

⁵ 一般に摂動変数を表す際にはダッシュ「`'`」がよく使われ、「7」に似ているため。

⁶ 予報モデルが非線形な場合、予報変数の値が TL の摂動計算において必要となり、予報モデルにおける予報変数とは区別してこのように呼ぶ。具体例は図 3.2.3 及び 3.2.4 参照。

⁷ JNoVA では基本場変数名の最後に「9」をつけ、摂動変数はそのまま用いていた。

開発を始めるにあたっては、前述のように開発体制とコーディングルールを取り決めてから開発に着手した (1)。まず最初に診断変数や予報変数の摂動の前処理が行えるよう前処理に摂動変数を追加し (2)、その後単体で閉じた (他サブルーチン呼び出さない) サブルーチンから作成を進めた (3)。閉じたサブルーチンの開発は前述の文献を参考に作成し (具体例は後述)、TL 及び AD それぞれについてコード検証を行う。これらの中には通信を必要とするサブルーチンがあるので、MPI 通信部分の AD を作成する必要もある (後述、4)。単体のサブルーチンの作成が一通り終了したら、次にその集合からなる上位のサブルーチンの TL/AD を作成する (5)。これは単体のサブルーチンのコード検証が済んでいれば、問題無く作成できる。以上が完成すると、全体の時間積分に関する部分を作成し、コード全体を検証する (6)。時間積分の際、後述のように基本場の値を保持する方法が問題となる。

(2) 開発の具体例

1. 接線形/随伴コードの具体例

TL の具体例として、図 3.2.3 及び図 3.2.4 に流束制限関数を用いた物理量の x 方向のフラックス計算 (第 2.4 節) のサブルーチン `calcf1xx.tlad` から抜粋したコードを示す。サブルーチンでは物理量 (コード上で `p`) から流束制限関数 (`factor`) を計算し、計算空間での x 方向の運動量 (`mom`) を用いてフラックス (`flux_x`) を計算するという手順になっている。この例では、予報モデルを簡略化あるいは正則化する必要がないことが確認されているため、NL は予報モデルの `calcf1xx` と同じ内容である。

TL では、NL に IF 文による条件分岐があることと、フラックス計算が非線形であることにより基本場の値が必要となる。また NL では `max` 及び `min` 関数が使われているが、石川・藤田 (2013) で述べられている通り、条件分岐は基本場の値を用いて行うため⁸、全て IF 文を用いた条件分岐に書き直す必要がある。

一方 AD については、TL の転置を取り流れを反転させればよいが、ループの最初に基本場を計算しておく必要がある (図 3.2.4)。また次に述べるように並列処理を行うための工夫が必要となる。

2. 随伴コードと並列処理

`asuca` では OpenMP によるスレッド並列及び MPI によるプロセス並列の処理を導入し高速化を図っている (付録 A 参照)。ここでは、並列化の際に AD において注意すべき点を述べる。

`asuca` では OpenMP を用いたスレッド並列が用いら

⁸ TL のコード検証では、NL による摂動 $M(x+\Delta x)-M(x)$ を計算する必要があるが、 $M(x+\Delta x)$ の計算における条件分岐では摂動を含む変数 $x+\Delta x$ を用いる。

```

subroutine calcflxx_tlad( flux_x7, p7, mom7, &
    & flux_x, is, ie, js, je, ks, ke, p, mom )

use parameter_vars, only : ksw_runmode, &
    & ksw_runmode_nl, ksw_runmode_bs, &
    & ksw_runmode_tl, ksw_runmode_ad
integer(4), parameter:: dp = kind(1.0d0)
integer(4), parameter:: rp = dp
real(rp), parameter :: r3 = 1._rp / 3._rp

! nonlinear
if ( ksw_runmode == ksw_runmode_nl ) then
do j = js, je
do i = is, ie
do k = ks, ke
    if( mom(k,i,j) > 0.d0 )then
        x1 = p(k,i ,j) - p(k,i-1,j)
        y1 = p(k,i+1,j) - p(k,i,j)
        z1 = (x1 * 0.5_rp + y1) * r3
        if ( y1 > 0._rp ) then
            factor = max( 0._rp, min(x1, y1, z1) )
        else
            factor = min( 0._rp, max(x1, y1, z1) )
        endif
        flux_x(k,i,j) = p(k,i ,j) + factor
        flux_x7(k,i,j) = mom(k,i,j) * flux_x(k,i,j)
    else
        .
        .
    endif
end do
end do
end do

! tangent linear
elseif ( ksw_runmode == ksw_runmode_tl ) then
do j = js, je
do i = is, ie
do k = ks, ke
    if( mom(k,i,j) > 0.d0 ) then
        ! Basic field
        x1 = p(k,i ,j) - p(k,i-1,j)
        y1 = p(k,i+1,j) - p(k,i,j)
        z1 = (x1 * 0.5_rp + y1) * r3
        ! TL
        x17 = p7(k,i,j) - p7(k,i-1,j)
        y17 = p7(k,i+1,j) - p7(k,i,j)
        z17 = (x17 * 0.5_rp + y17) * r3
        if ( y1 > 0._rp ) then
            ! Basic field
            factor = max( 0._rp, min(x1, y1, z1) )
            ! TL
            if (x1<y1.and.x1<z1.and.x1>0._rp) then
                factor7 = x17
            elseif(y1<x1.and.y1<z1.and.y1>0._rp) then
                factor7 = y17
            elseif(z1<x1.and.z1<y1.and.z1>0._rp) then
                factor7 = z17
            else
                factor7 = 0._rp
            endif
        else
            .
            .
        endif
        ! Basic field
        flux_x(k,i,j) = p(k,i,j) + factor
        ! TL
        flux_x7(k,i,j) = p7(k,i,j) + factor7
        flux_x7(k,i,j) = mom7(k,i,j) * flux_x(k,i,j)
        & + mom(k,i,j) * flux_x7(k,i,j)
    else
        .
        .
    endif
end do
end do
end do

```

図 3.2.3 サブルーチン calcflxx_tlad から非線形コード及び接線形コードを抜粋したもの。p が任意の物理量、mom が x 方向の運動量を表し、そのフラックス flux_x を計算する。

```

! adjoint
elseif ( ksw_runmode == ksw_runmode_ad ) then
do j = js, je
do i = is, ie
do k = ks, ke
    if( mom(k,i,j) > 0.d0 ) then
        ! Basic Field
        x1 = p(k,i ,j) - p(k,i-1,j)
        y1 = p(k,i+1,j) - p(k,i,j)
        z1 = (x1 * 0.5_rp + y1) * r3
        if ( y1 > 0._rp ) then
            factor = max( 0._rp, min(x1, y1, z1) )
        else
            factor = min( 0._rp, max(x1, y1, z1) )
        endif

        ! AD
        mom7(k,i,j) = &
            & (p(k,i,j) + factor) * flux_x7(k,i,j)
        p7(k,i,j) = &
            & p7(k,i,j) + mom(k,i,j) * flux_x7(k,i,j)
        factor7 = mom(k,i,j) * flux_x7(k,i,j)
        flux_x7(k,i,j) = 0._rp
        if ( y1 > 0._rp ) then
            if ( x1 < y1 .and. x1 < z1 &
                & .and. x1 > 0._rp ) then
                x17 = factor7
                factor7 = 0._rp

                p7(k,i,j) = p7(k,i,j) + x17
                p7(k,i-1,j) = p7(k,i-1,j) - x17
                x17 = 0._rp
            elseif ( y1 < x1 .and. &
                & y1 < z1 .and. y1 > 0._rp ) then
                y17 = factor7*x
                factor7 = 0._rp

                p7(k,i+1,j) = p7(k,i+1,j) + y17
                p7(k,i,j) = p7(k,i,j) - y17
                y17 = 0._rp
            elseif ( z1 < x1 .and. &
                & z1 < y1 .and. z1 > 0._rp ) then
                z17 = factor7
                factor7 = 0._rp

                x17 = 0.5_rp * r3 * z17
                y17 = r3 * z17
                z17 = 0._rp

                p7(k,i+1,j) = p7(k,i+1,j) + y17
                p7(k,i,j) = p7(k,i,j) - y17
                y17 = 0._rp

                p7(k,i,j) = p7(k,i,j) + x17
                p7(k,i-1,j) = p7(k,i-1,j) - x17
                x17 = 0._rp
            else
                factor7 = 0._rp
            endif
        else
            .
            .
        endif
    endif
end do
end do
end do
return
end subroutine calcflxx_tlad

```

図 3.2.4 図 3.2.3 の続き。サブルーチン calcflxx_tlad から随伴コードを抜粋したもの。

(a) 格子点(k, i, j)の周囲の格子も計算を行うため、分割領域の端では隣接する分割領域の格子点まで計算を行う

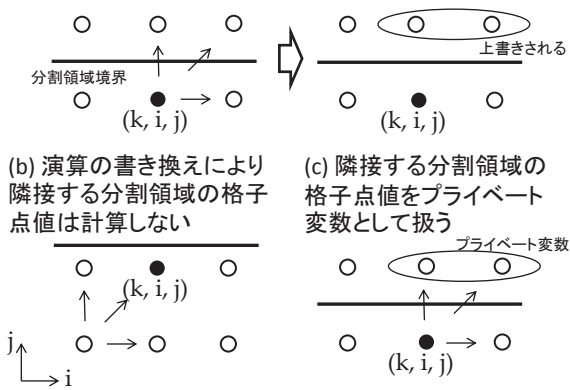


図 3.2.5 平均値計算に対する AD 及びその並列化ための模式図。実線がスレッド並列分割領域の境界を表し、丸が格子点を表す。

れており、kij 方式 (付録 A 参照) では最も外側の j 方向 (南北方向) においてスレッド並列を行う。一般にループの中で、NL と TL で計算対象となる格子が変わることは無いが、AD では変わる場合がある。簡単な例として、ある摂動変数 (x7) の周囲の格子点の平均 (y7) を求める計算を考える⁹。このときの TL は、

$$y7(k, i, j) = (x7(k, i, j) + x7(k, i+1, j) + x7(k, i, j+1) + x7(k, i+1, j+1)) * 0.25$$

で表される。これに対応する AD は TL を機械的に変換すると (変数 x7 が TL においてこの処理の後でも使われると仮定する)、

$$\begin{aligned} x7(k, i, j) &= x7(k, i, j) + 0.25 * y7(k, i, j) \\ x7(k, i+1, j) &= x7(k, i+1, j) + 0.25 * y7(k, i, j) \\ x7(k, i, j+1) &= x7(k, i, j+1) + 0.25 * y7(k, i, j) \\ x7(k, i+1, j+1) &= x7(k, i+1, j+1) + 0.25 * y7(k, i, j) \\ y7(k, i, j) &= 0.0 \end{aligned}$$

と書くことができ、周囲の格子点を含む加算に置き換わることがわかる。上記のコードにおいて並列化による領域分割を考えると、分割領域の端では分割領域境界をまたいで隣接する他スレッドの格子点値まで計算してしまう (図 3.2.5(a) 参照)。このため隣接スレッドでは、既に計算済みであった格子点値が他スレッドによって上書きされ得る状況が発生する。このようにループの中で、自スレッドで計算済みの値が他スレッドによって上書きされることをループ運搬依存と呼ぶ。ループ運搬依存が発生すると正しい計算結果が得られないため、NL や TL と同じように並列化を適用することができない。したがって、コードに何らかの工夫が必要となる。

⁹ 同じような計算として、asuca ではベクトルポイントの予報変数である風をスカラーポイントで用いる際の診断などがある。

現在の開発では、kij 方式の k の手前に OpenMP 指示文を入れ、鉛直方向にスレッド並列を行うようにしている。しかしこれだけでは i, j ループのたびにスレッド並列の開始と終了処理が行われ、演算速度が低下してしまう可能性がある。そこでループの順序を kij から ijk へと変更し k 方向にスレッド並列を行うように変更した。これにより、メモリアクセスは不連続になるがスレッド並列の開始と終了処理を減らすことができる。

この方法の他に上記の x7 の演算を、

$$x7(k, i, j) = (y7(k, i, j) + y7(k, i-1, j) + y7(k, i, j-1) + y7(k, i-1, j-1)) * 0.25$$

と書き換え、ループ内における隣接格子への依存をなくすことでスレッド並列が可能となる¹⁰(図 3.2.5(b))。ただしループの端の x7 の演算は別途書き直し、y7=0.0 の演算をループの外で行うよう修正が必要である。

また、OpenMP の REDUCTION 変数を用いる方法も考えられる。これは隣接スレッドの格子点値をプライベート変数として扱い、ループ終了後に共有変数として合算を行うことで、コードの書き換えなしにスレッド並列が可能になる¹¹(図 3.2.5(c))。これは機械的に作成した AD に変更を加えず、指示文を挿入するだけでスレッド並列を行うことができ、簡便な手法であると考えられる。

以上のように AD に OpenMP を用いたスレッド並列を適用する際は、いずれの手法を採用するにしても、その手法は開発者間で統一しておく必要がある。また AD の作成には時間を要するため、一度 OpenMP による並列化を意識せずに AD を構築し、完成後に採用した手法によって OpenMP 化を機械的に進めることも開発上効率的であると考えられる。

次に MPI によるプロセス並列を用いる場合を考える。ノード間通信のためには AD が必要であり、これは以下の簡単な例からわかる。例えば、計算領域上の変数 x, y が共に MPI 分割領域の端に位置し、互いに分割領域境界をまたいで隣り合っているとす。また、それぞれの分割領域ののりしろ変数を a, b とする。ここでは x を b へ、y を a へ通信することを考える (図 3.2.6)。この処理を行列で表すと、

$$\begin{pmatrix} x \\ a \\ b \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ a \\ b \\ y \end{pmatrix}$$

¹⁰ JNoVA では、スーパーコンピュータ上のコンパイラによる自動スレッド並列を用いており、AD ではこの手法を採用している。

¹¹ プライベート変数は各スレッドに独自の値を持たせる変数であり、共有変数は各スレッドが値を共有する変数である。

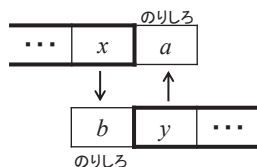


図 3.2.6 MPI 通信の模式図。太枠内が計算領域を表す。

と書ける。この行列の転置をとると、

$$\begin{pmatrix} x \\ a \\ b \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ a \\ b \\ y \end{pmatrix}$$

が得られ、 b の値を x に、 a の値を y に加算した上で、 a 、 b をゼロクリアする処理となる。NL/TL と AD は MPI 通信の処理が異なり、通信の方向が反転している (AD では図 3.2.6 の矢印の向きが逆になる) ことに注意を払う必要がある。

(3) 開発の進捗

本稿の執筆時点では、各サブルーチンの TL/AD 開発は完了しており、TL の時間積分を行いその線形性が成り立つことを確認したところであり、AD の時間積分を行う作業に取り組んでいる。

TL の基本場の保存方法については、JNoVA ではショートタイムステップを含め、サブルーチン内で基本場を直接計算する場合を除き、基本場は全てメモリ上に保持しており (本田 2010)、今回の開発でもこの方法を踏襲した。しかし asuca では、時間積分に 3 段階ルンゲクッタ法を採用しており (第 2.3 節参照)、1 タイムステップの時間積分を行う際に、ショートタイムステップの中にもルンゲクッタステップが存在する¹²。将来インナーモデルがさらに高解像度化することを想定すると、全ての基本場をメモリ上に保持できない可能性もある。この場合は、各タイムステップにおける基本場のみを保持し、TL/AD 計算の際にルンゲクッタステップにおける基本場を計算し直す必要があると考えている。

3.2.5 今後の開発

ここまで述べてきたように、asuca-Var における力学過程の TL/AD 開発では、JNoVA での開発経験を踏まえ、将来に渡り継続可能な拡張性の高い開発を目指す方針を定めた上で作業を進めており、力学過程の

¹² asuca ではショートタイムステップのステップ数は音波による CFL 条件を満たすよう格子間隔及び積分時間間隔から自動的に計算している。現 JNoVA を想定して水平解像度 15km、時間積分間隔 40 秒でステップ数を見積ると 12 ステップとなり、現 JNoVA のインナーモデルにおけるショートタイムステップ数 7 に比べ多くなることがわかる。

TL/AD の完成後に物理過程の TL/AD の開発に着手する予定である。物理過程については前述のとおり、力学過程と比較してコードが複雑であり、改良によるコードの変更も頻繁に行われるため、モデル開発者による TL/AD の修正が開発継続にとって必須である。実際の物理過程の TL/AD 開発を通し、ここで述べた開発ルールに対する更なる改善等を行った場合については別の機会に報告したい。

参考文献

荒波恒平, 石田純一, 2008: 並列化と高速化. 数値予報課報告・別冊第 54 号, 気象庁予報部, 58–65.

本田有機, 2003: 4 次元変分法データ同化システム. 開発報告. 数値予報課報告・別冊第 49 号, 気象庁予報部, 144–147.

本田有機, 2010: 接線形・随伴モデル. 数値予報課報告・別冊第 56 号, 気象庁予報部, 9–13.

石田純一, 荒波恒平, 2003: 並列化. 数値予報課報告・別冊第 49 号, 気象庁予報部, 107–111.

石川宜広, 藤田匡, 2013: 4 次元変分法の開発と物理過程の課題. 数値予報課報告・別冊第 59 号, 気象庁予報部, 152–187.

門脇隆志, 2002: 変分法で使用するコードの検証方法. 数値予報課報告・別冊第 48 号, 気象庁予報部, 85.

河野耕平, 荒波恒平, 原旅人, 北村祐二, 室井ちあし, 2011: 気象庁非静力学モデル asuca の開発状況. 第 13 回非静力学モデルに関するワークショップ講演予稿集.

露木義, 1997: アジョイントコードの書き方. 数値予報課報告・別冊第 43 号, 気象庁予報部, 155–165.

露木義, 2002: アジョイントコードの書き方の補足. 数値予報課報告・別冊第 48 号, 気象庁予報部, 65–70.