Dominant Mode Extraction Based on the Four-Dimensional Variational Method

Hiromichi Nagao
Earthquake Research Institute
The University of Tokyo
Tokyo, Japan
Email: nagaoh@eri.u-tokyo.ac.jp

Shin-ichi Ito
Earthquake Research Institute
The University of Tokyo
Tokyo, Japan
Email: ito@eri.u-tokyo.ac.jp

Mitsuru Matsumura
CJS Inc.
Kyoto, Japan
Email: matsumura.mitsuru.public@chino-js.com

***Abstract*—Understanding the formation mechanism of magnetic domain patterns is important to improve the performance of magnetic materials. The magnetic domain patterns depend on the parameters of the time-dependent Ginzburg-Landau (TDGL) equation and the sweep rate of the external magnetic field. Although conventional analytical approaches can predict the patterns formed in the case of a high sweep rate, more versatile methods are required to understand the formation mechanism of complicated patterns for any temporal variation in the external field. This study proposes a method that extracts dominant modes from a posterior distribution based on the four-dimensional variational method (4DVar). The method decomposes the magnetic domain patterns into eigenvectors of the Hessian matrix of the cost function, defined as the difference between the observed and simulated magnetic domain patterns. The eigenvectors are extracted using a second-order adjoint method (SOA) and power iteration. The patterns are reconstructed by superimposing the extracted eigenvectors, and their time evolution can be obtained from the weights of the eigenvectors. Experiments demonstrate that the patterns are sufficiently reconstructed using a small number of the eigenvectors. This enables us to understand the pattern evolution as the change of the dominant eigenvectors.**

I. Introduction

Magnetic domain patterns are essential since they reflect the properties of magnetic materials [1]. Understanding the formation of domain patterns often provides a direction for improving the performance of magnetic materials. Domain patterns are observed as images [2]. A magnetic domain comprises a set of magnetizations, and a domain pattern image is represented by a vector whose element is the magnetization at each pixel. The dimension of the magnetization vector is so high that it must be reduced to understand the spatio-temporal variation of the vector. Mode decomposition approaches are popular for dimension reduction. If a vector can be approximated as a linear combination of a small number of dominant modes, the pattern formation can be easily understood by monitoring the time evolution of the weights of the modes. The Discrete Fourier decomposition [2][3] and the proper orthogonal decomposition (POD) [4] are popular mode decomposition methods. The POD decomposes the variance-covariance matrix of a given spatio-temporal vector field into eigenvectors corresponding to spatial and temporal modes, providing information on coherent structures in the data. The eigenvalue of each POD mode represents its magnitude, which indicates its relative importance among the modes. The dynamic mode decomposition (DMD), similar to the POD, can extract modes with periodic time variations and obtain their growth and decay rates.

This study proposes a new mode decomposition method that focuses on the Hessian matrix of a cost function, which is defined as the difference between physical simulation and observation data. The proposed method calculates the eigenvectors of the Hessian matrix and uses them to extract several dominant modes in the posterior probability density distribution (PDF) (Fig. 1) with evaluating the uncertainty of each mode. Moreover, the temporal evolution of magnetic patterns can be sufficiently reconstructed using the dominant modes.

II. Method and Model

A. Four-Dimensional Variational Method

Let $\boldsymbol{x}_t$ be a n-dimensional time-dependent state vector that follows an autonomous ordinary differential equation:

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{f}\left(\boldsymbol{x}_t\right) \ , \tag{1}$$

where the function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is assumed to be second-order differentiable with respect to $\boldsymbol{x}_t$. Note that the initial condition $\boldsymbol{x}_{t_0}$ is the unique control variable of the time evolution of $\boldsymbol{x}_t$. Suppose we have a set of observational data from $t = t_0$ to $t = t_k$, *i.e.*, $\boldsymbol{Y}_{0:k} = \left(\boldsymbol{y}_{t_0}^\top, \boldsymbol{y}_{t_1}^\top, \ldots, \boldsymbol{y}_{t_k}^\top\right)^\top$, where $\top$ denotes the transposition. Each observation $\boldsymbol{y}_t \in \mathbb{R}^m$ is assumed to have a relation with the state vector $\boldsymbol{x}_t$ as

$$\boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_t) + \boldsymbol{\epsilon}_t \ , \tag{2}$$

where $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^m$ is an observational operator that maps $\boldsymbol{x}_t$ to a quantity comparable with the data $\boldsymbol{y}_t$, and $\boldsymbol{\epsilon}_t \in \mathbb{R}^m$ is an observational noise that follows a PDF $q(\boldsymbol{\epsilon})$. Based on the given model and observational data, a likelihood or cost function $p\left(\boldsymbol{Y}_{0:k} \mid \boldsymbol{x}_{t_0}\right)$ can be calculated. Then the posterior PDF at $t = t_0$, *i.e.*, $p\left(\boldsymbol{x}_{t_0} \mid \boldsymbol{Y}_{0:k}\right)$, is given by the Bayes' theorem as

$$p\left(\boldsymbol{x}_{t_0} \mid \boldsymbol{Y}_{0:k}\right) = \frac{p\left(\boldsymbol{x}_{t_0}\right) p\left(\boldsymbol{Y}_{0:k} \mid \boldsymbol{x}_{t_0}\right)}{p\left(\boldsymbol{Y}_{0:k}\right)} \ , \tag{3}$$

where $p(\boldsymbol{x}_{t_0})$ is a prior PDF that describes *a priori* information on the initial condition. Using the PDF for the observational data, the posterior PDF is given by

$$p(\boldsymbol{x}_{t_0} \mid \boldsymbol{Y}_{0:k}) = \frac{p(\boldsymbol{x}_{t_0})}{p(\boldsymbol{Y}_{0:k})} \prod_{j=0}^{k} q\left(\boldsymbol{y}_{t_j} - \boldsymbol{h}(\boldsymbol{x}_{t_j})\right) . \quad (4)$$

The four-dimensional variational (4DVar) method (or adjoint method) is widely used in modern weather forecasting to optimize the initial condition $\boldsymbol{x}_{t_0}$, maximizing the posterior PDF $p(\boldsymbol{x}_{t_0} \mid \boldsymbol{Y}_{0:k})$. Considering that $p(\boldsymbol{Y}_{0:k})$ is constant, maximization of Eq ((4)) is equivalent to minimization of the negative logarithm of $p(\boldsymbol{x}_{t_0} \mid \boldsymbol{Y}_{0:k})\, p(\boldsymbol{Y}_{0:k})$ given by

$$J = \underbrace{-\log p(\boldsymbol{x}_{t_0})}_{=I} + \sum_{j=0}^{k} \underbrace{\left[-\log q\left(\boldsymbol{y}_{t_j} - h(\boldsymbol{x}_{t_j})\right)\right]}_{=\mathcal{J}_j} . \quad (5)$$

Note that $\mathcal{J}_j$ is not directly differentiable with respect to $\boldsymbol{x}_{t_0}$ since it implicitly depends on $\boldsymbol{x}_{t_0}$ through Eq. (1). To obtain the gradient $\sum_{j=0}^{k} \mathrm{d}\mathcal{J}_j/\mathrm{d}\boldsymbol{x}_{t_0}$, the 4DVar applies a variational approach to an augmented Lagrangian function defined by

$$\begin{aligned}\mathcal{L} &= \sum_{j=0}^{k} \mathcal{J}_j + \int_{t_0}^{t_k} dt\, \boldsymbol{\lambda}_t^{\top} \left(\boldsymbol{f}(\boldsymbol{x}_t) - \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}\right) \\ &= \int_{t_0}^{t_k} dt \left[\sum_{j=0}^{k} \delta(t - t_j)\, \mathcal{J}_j + \boldsymbol{\lambda}_t^{\top} \left(\boldsymbol{f}(\boldsymbol{x}_t) - \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}\right)\right] ,\end{aligned} \quad (6)$$

where $\delta(t)$ is a Dirac delta function and $\boldsymbol{\lambda}_t$ is a Lagrange multiplier. The calculus of variation of the Lagrangian function provides us an adjoint model, which describes the time evolution of $\boldsymbol{\lambda}_t$:

$$-\frac{\mathrm{d}\boldsymbol{\lambda}_t}{\mathrm{d}t} = \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}_t}\right)^{\top} \boldsymbol{\lambda}_t + \sum_{j=1}^{k} \delta(t - t_j) \frac{\partial \mathcal{J}_j}{\partial \boldsymbol{x}_{t_j}} , \quad (7)$$

together with the conditions at the initial and end times

$$\boldsymbol{\lambda}_{t_0} = \sum_{j=0}^{k} \frac{\mathrm{d}\mathcal{J}_j}{\mathrm{d}\boldsymbol{x}_{t_0}} , \quad \boldsymbol{\lambda}_{t_k} = 0 . \quad (8)$$

See [5][6] for the derivation of Eqs. (7) and (8) from Eq. (6). The gradient of the cost function at the initial time, *i.e.*, $\mathrm{d}J/\mathrm{d}\boldsymbol{x}_{t_0}$, is obtained by (a) $\boldsymbol{x}_t$ is simulated by Eq. (1) starting from the given $\boldsymbol{x}_{t_0}$, (b) the adjoint model Eq. (7) is solved backwardly in time starting from $\boldsymbol{\lambda}_{t_k}$ to $t = t_0$, and (c) the objective gradient is obtained by adding $\mathrm{d}I/\mathrm{d}\boldsymbol{x}_{t_0}$ to $\boldsymbol{\lambda}_{t_0}$. The initial condition $\boldsymbol{x}_{t_0}$ can be updated using an appropriate gradient method and the obtained gradient $\mathrm{d}J/\mathrm{d}\boldsymbol{x}_{t_0}$. Repeating this, we can eventually obtain an optimal solution $\hat{\boldsymbol{x}}_{t_0}$ for $\boldsymbol{x}_{t_0}$.

B. Orthogonal Decomposition of Variance-Covariance Matrix

A Laplace approximation of the cost function J in the neighborhood of $\boldsymbol{x}_{t_0} = \hat{\boldsymbol{x}}_{t_0}$ is

$$p(\boldsymbol{x}_{t_0} \mid \boldsymbol{Y}_{0:k}) \approx \sqrt{\frac{\det \mathsf{H}}{(2\pi)^n}} \exp\left[-\frac{1}{2}(\boldsymbol{x}_{t_0} - \hat{\boldsymbol{x}}_{t_0})^{\top} \mathsf{H} (\boldsymbol{x}_{t_0} - \hat{\boldsymbol{x}}_{t_0})\right] , \quad (9)$$

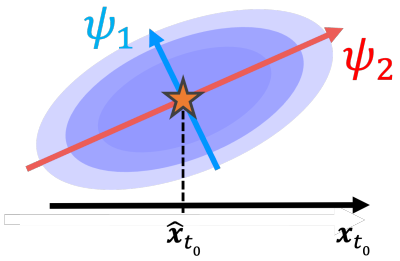


Fig. 1: Schematic of a posterior decomposed into dominant modes by the proposed method.

where the matrix $\mathsf{H} \in \mathbb{R}^{n \times n}$ is a Hessian matrix evaluated at the optimal solution given by

$$\mathsf{H} = \frac{\mathrm{d}^2 J}{\mathrm{d}\hat{\boldsymbol{x}}_{t_0}^2} . \quad (10)$$

We consider an eigenvalue decomposition of the Hessian matrix H

$$\mathsf{H} = \sum_{j=1}^{n} \nu_j \boldsymbol{\psi}_j^{\top} \boldsymbol{\psi}_j , \quad (11)$$

where ν_j $(j = 1, .., n)$ are the eigenvalues sorted in descending order, and the vectors $\boldsymbol{\psi}_j \in \mathbb{R}^n$ $(j = 1, .., n)$ are the corresponding eigenvectors.

The Laplace approximation states that the variance of each element in $\hat{\boldsymbol{x}}_{t_0}$ is given by the diagonal element of the inverse matrix of Hessian H^{-1}. However, in general, a direct evaluation of the diagonal elements of H^{-1} is computationally massive. Here we estimate the computational complexity to obtain H^{-1} when approximating each element in H via a finite difference of J. The finite difference approximation of each element in H needs a few times of computations of J, which is proportional to the complexity C to solve Eq. (1). This means that we need $O(n^2 C)$ computational complexity to obtain all of the elements in H. Additionally, taking an inverse of H needs the computations proportional to $O(n^3)$ since H is a dense matrix in general. In summary, the finite difference approximation needs $O(n^2 C)$+$O(n^3)$ computations. This is unrealistic when n is large; therefore, we employ the alternate method for the computation of H^{-1} proposed by [7]. The method can avoid such a large computational complexity since it is dedicated to extracting only the uncertainties of interest. The key algorithm is the second-order adjoint (SOA) method, which enables the computation of a product of the Hessian H and an arbitrary vector $\boldsymbol{z}$ with the same computational complexity as the one needed to solve Eq. (1). The 4DVar-based uncertainty quantification method [7] can compute specified elements in H^{-1} by combining the SOA method and a Krylov subspace method. The SOA method is composed of solving a set of the tangent linear (TL) model and the SOA model, which are derived from the perturbations of Eqs. (1) and (7) [8][9]. The TL model is given by

$$\frac{\mathrm{d}\boldsymbol{\xi}_t}{\mathrm{d}t} = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_t} \boldsymbol{\xi}_t , \quad (12)$$

with its inital condition $\boldsymbol{\xi}_{t_0} = \boldsymbol{z}$, and the SOA model is given by

$$-\frac{\mathrm{d}\boldsymbol{\eta}_t}{\mathrm{d}t} = \left(\frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_t}\right)^\top \boldsymbol{\eta}_t + \left(\frac{\partial^2 \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_t^2}\boldsymbol{\xi}_t\right)^\top \hat{\boldsymbol{\lambda}}_t + \sum_{j=0}^{k} \delta\left(t - t_j\right) \frac{\partial^2 \mathcal{J}_j}{\partial \hat{\boldsymbol{x}}_{t_j}^2}\boldsymbol{\xi}_{t_j} \ , \tag{13}$$

together with its initial and end-point conditions

$$\boldsymbol{\eta}_{t_0} = \sum_{j=0}^{k} \frac{\mathrm{d}^2 \mathcal{J}_j}{\mathrm{d}\hat{\boldsymbol{x}}_0^2}\boldsymbol{\xi}_{t_0} \ , \quad \boldsymbol{\eta}_{t_k} = 0 \ , \tag{14}$$

where $\hat{\boldsymbol{x}}_t$ and $\hat{\boldsymbol{\lambda}}_t$ are the solutions of Eqs. (1) and (7), in which $\boldsymbol{x}_{t_0} = \hat{\boldsymbol{x}}_{t_0}$ is used, and the derivative with respect to $\hat{\boldsymbol{x}}_t$ means that with respect to $\boldsymbol{x}_t$, to which $\boldsymbol{x}_t = \hat{\boldsymbol{x}}_t$ is substituted. Using these models allows us to compute a product of the Hessian and an arbitrary vector, in that, after solving the TL model starting from an initial condition $\boldsymbol{\xi}_{t_0} = \boldsymbol{z}$, a product of the Hessian and the vector $\boldsymbol{z}$ is given by using η_{t_0} as

$$\mathsf{H}\boldsymbol{z} = \frac{\mathrm{d}^2 I}{\mathrm{d}\hat{\boldsymbol{x}}_{t_0}^2}\boldsymbol{z} + \boldsymbol{\eta}_{t_0} \ . \tag{15}$$

Using the SOA method enables us to access a column vector of the inverse H^{-1} by solving a linear equation

$$\mathsf{H}\boldsymbol{z} = \boldsymbol{b} \ , \tag{16}$$

where the vector $\boldsymbol{b}$ as a one-hot vector, based on a relevant Krylov subspace method such as the conjugate gradient or conjugate residual methods. The uncertainty is given by the square root of the diagonal elements in H^{-1}. The computational complexity needed to solve the set of the TL and SOA models is almost same as that needed to solve Eq. (1); therefore, the total computational complexity to obtain an uncertainty through solving Eq. (16) is $O(C)$ if we can start the Krylov subspace method using a good initial estimate.

C. Time-Dependent Ginzburg-Landau Model

To generate magnetic domain patterns, we utilize the Ising-like spin model with time-dependent Ginzburg-Landau (TDGL) dynamics [2][3], described as:

$$\begin{aligned} \frac{\mathrm{d}\phi(\boldsymbol{r})}{\mathrm{d}t} &= f_\mathrm{a} + f_\mathrm{e} + f_\mathrm{d} + B(t) \ , \\ f_\mathrm{a} &= \alpha\Lambda(\boldsymbol{r})\left\{\phi(\boldsymbol{r}) - \phi(\boldsymbol{r})^3\right\} \ , \\ f_\mathrm{e} &= \beta\nabla^2\phi(\boldsymbol{r}) \ , \\ f_\mathrm{d} &= -\gamma\int \mathrm{d}^2\boldsymbol{r}' G\left(\boldsymbol{r} - \boldsymbol{r}'; A\right)\phi(\boldsymbol{r}) \ , \end{aligned} \tag{17}$$

where $\boldsymbol{r} = (x, y)$ is an element of the two-dimensional discrete space, and $\phi(\boldsymbol{r})$ is a scalar spin field, i.e., a magnetic dipole restricted to the z direction located at $\boldsymbol{r}$. The terms f_a, f_e, f_d correspond to the anisotropy, exchange, and dipole-dipole interactions, respectively. We define $\Lambda(\boldsymbol{r}) = 1 + \mu(\boldsymbol{r})/4, \mu(\boldsymbol{r}) \sim \mathcal{N}(0, 0.3^2)$, which expresses disorder. $G(\boldsymbol{r}; A)$ is a Green's function that includes the effect of the material thickness A, defined as

$$G(\boldsymbol{r}; A) = \frac{2}{A^2}\left(\frac{1}{|\boldsymbol{r}|} - \frac{1}{\sqrt{|\boldsymbol{r}|^2 + A^2}}\right) \ . \tag{18}$$

We let $B(t) = R(B_0 - v_B t), R(x) = \max(x, 0)$ represent the external magnetic field intensity, where B_0 is the initial external field, and v_B is the sweep rate.

If $1 \le x, y \le L$, we denote subscript $i(= 1, 2, ..., n(= L^2))$ for a variable at r_i, *e.g.*, $\phi_i = \phi(r_i)$, and descretize Eq. (17) as

$$\begin{aligned} \frac{\mathrm{d}\phi_i}{\mathrm{d}t} &= \alpha\Lambda_i\left(\phi_i - \phi_i^3\right) + \beta\sum_j \Delta_{ij}\phi_j \\ &\quad - \gamma\sum_j F_{ij}^{(R)}\phi_j + B(t) \ , \end{aligned} \tag{19}$$

where Δ_{ij} expresses Laplacian descretized with the first-order finite difference method with periodic boundary conditions. $F_{ij}^{(R)}$ is defined as

$$\begin{aligned} F_{ij}^{(R)} &= \frac{2}{A^2}\sum_{r=-R}^{R}\sum_{r'=-R}^{R}\left[\frac{1}{d(r, r'|0)_{ij}} - \frac{1}{d(r, r'|A)_{ij}}\right] \\ d(r, r'|h)_{ij} &= \\ &\sqrt{(Lr + x(i) - x(j))^2 + (Lr + y(i) - y(j))^2 + h^2} \ , \end{aligned} \tag{20}$$

where $R(> 0)$ is the number of the replicated computation regions generated due to the periodic boundary conditions.

The adjoint model for Eq.(19) is as follows:

$$\begin{aligned} -\frac{\mathrm{d}\lambda_i}{\mathrm{d}t} &= \alpha\Lambda_i\left(1 - 3\phi_i^2\right)\lambda_i + \sum_j\left[\beta\Delta_{ji} - \gamma F_{ji}^{(R)}\right]\lambda_j \\ &\quad + \sum_{j=1}^{k}\delta\left(t - t_j\right)\frac{\partial \mathcal{J}_j}{\partial \phi_i} \ . \end{aligned} \tag{21}$$

The TL model for Eq.(19) is as follows:

$$\frac{\mathrm{d}\xi_i}{\mathrm{d}t} = \alpha\Lambda_i\left(1 - 3\phi_i^2\right)\xi_i + \sum_j\left[\beta\Delta_{ij} - \gamma F_{ij}^{(R)}\right]\xi_j \ . \tag{22}$$

The SOA model for Eq.(19) is as follows:

$$\begin{aligned} -\frac{\mathrm{d}\eta_i}{\mathrm{d}t} &= \alpha\Lambda_i\left(1 - 3\phi_i^2\right)\eta_i - 6\alpha\Lambda_i\phi_i\xi_i\lambda_i \\ &\quad + \sum_j\left[\beta\Delta_{ji} - \gamma F_{ji}^{(R)}\right]\eta_j + \sum_{j=1}^{k}\delta\left(t - t_j\right)\frac{\partial^2 \mathcal{J}_j}{\partial \phi_i^2}\xi_i \ . \end{aligned} \tag{23}$$

We integrate these equations in time using the explicit forth-order Runge-Kutta methods.

D. Mode extraction via power iteration

Eigenvectors of the Hessian matrix H with $n \times n$ elements are computed with the power iteration, where $n = L^2$ and $1 \leq x, y \leq L$. These eigenvectors are orthogonal bases for any n-dimensional vectors because the Hessian matrix is symmetric. Using the eigenvectors $\boldsymbol{\psi}_i$ $(i = 1, 2, ...n)$, the spin vector $\boldsymbol{\phi} = (\phi(r_1), \phi(r_2), ..., \phi(r_n))^\top$ is decomposed as

$$\boldsymbol{\phi} = \sum_{i=1}^{n} w_i \boldsymbol{\psi}_i \ , \tag{24}$$

where $w_i = \boldsymbol{\psi}_i^\top \boldsymbol{\phi}$ is the weight of the eigenvector $\boldsymbol{\psi}_i$ and measures the effectiveness of the mode $\boldsymbol{\psi}_i$. In other words, we can detect dominant modes for the spin vector $\boldsymbol{\phi}$ corresponding to the temporal magnetic domain pattern.

We propose a new method to compute the eigenvectors $\boldsymbol{\psi}_i$. Using the eigenvectors, a unit vector $\boldsymbol{u}_0$ is decomposed as,

$$\boldsymbol{u}_0 = \sum_{i=1}^{n} c_i \boldsymbol{\psi}_i \ , \tag{25}$$

where $\boldsymbol{\psi}_1$, $\boldsymbol{\psi}_2$,..., $\boldsymbol{\psi}_n$ correspond to the eigenvalues of the Hessian matrix in descending order, ν_1, ν_2,..., ν_n. Defining $\boldsymbol{v}_1 \equiv \mathsf{H}\boldsymbol{u}_0$, the unit vector parallel to this can be computed as

$$\begin{aligned} \boldsymbol{u}_1 \equiv \frac{\boldsymbol{v}_1}{|\boldsymbol{v}_1|} &= \frac{\mathsf{H}\boldsymbol{u}_0}{|\boldsymbol{v}_1|} = \frac{1}{|\boldsymbol{v}_1|} \sum_{i=1}^{n} c_i \boldsymbol{\psi}_i \\ &= \frac{\nu_1}{|\boldsymbol{v}_1|} \left\{ c_1 \boldsymbol{\psi}_1 + \sum_{i=2}^{n} c_i \left(\frac{\nu_i}{\nu_1} \right) \boldsymbol{\psi}_i \right\} \ . \end{aligned} \tag{26}$$

Repeting this, the unit vector parallel to $\boldsymbol{v}_k \equiv \mathsf{H}\boldsymbol{u}_{k-1}$ is obtained as

$$\boldsymbol{u}_k \equiv \frac{\nu_1^k}{\prod_{i=1}^{k} |\boldsymbol{v}_i|} \left\{ c_1 \boldsymbol{\psi}_1 + \sum_{i=2}^{n} c_i \left(\frac{\nu_i}{\nu_1} \right)^k \boldsymbol{\psi}_i \right\} \ . \tag{27}$$

If k is sufficiently large,

$$\boldsymbol{u}_k \sim \frac{\nu_1^k}{\prod_{i=1}^{k} |\boldsymbol{v}_i|} c_1 \boldsymbol{\psi}_1 \ . \tag{28}$$

Defining $\widetilde{\boldsymbol{\psi}}_1 \equiv \boldsymbol{u}_k$,

$$\boldsymbol{v}_{k+1} = \mathsf{H}\boldsymbol{u}_k = \nu_1 \widetilde{\boldsymbol{\psi}}_1 \ , \boldsymbol{u}_k \cdot \boldsymbol{v}_{k+1} \sim \nu_1 \ . \tag{29}$$

Defining the provisional value of ν_1 as $p = \boldsymbol{u}_k \cdot \boldsymbol{v}_{k+1}$,

$$\begin{aligned} |\boldsymbol{v}_{k+1} - p\boldsymbol{u}_k|^2 &= |\boldsymbol{v}_{k+1}|^2 - 2p\boldsymbol{v}_{k+1} \cdot \boldsymbol{u}_k + p^2 |\boldsymbol{u}_k|^2 \\ &= |\boldsymbol{v}_{k+1}|^2 - p^2 \ . \end{aligned} \tag{30}$$

Using a convergence criteria ϵ, we set the convergence condition of the power iteration as $(|\boldsymbol{v}_{k+1}|^2 - p^2)/p^2 < \epsilon$. To extract the second largest eigenvalue ν_2 and the corresponding eigenvector $\widetilde{\boldsymbol{\psi}}_2$ parallel to $\boldsymbol{\psi}_2$, we begin by subtracting the component parallel to $\widetilde{\boldsymbol{\psi}}_1$ from $\boldsymbol{u}_0$ and normalize it as follows:

$$\begin{aligned} \boldsymbol{u'}_0 &\equiv \boldsymbol{u}_0 - \left(\boldsymbol{u}_0 \cdot \widetilde{\boldsymbol{\psi}}_1 \right) \widetilde{\boldsymbol{\psi}}_1, \\ \boldsymbol{u}_0^{(2)} &\equiv \frac{\boldsymbol{u'}_0}{|\boldsymbol{u'}_0|}. \end{aligned} \tag{31}$$

If the extracted eigenvector $\widetilde{\boldsymbol{\psi}}_1$ is parallel to the true eigenvector $\boldsymbol{\psi}_1$, $\boldsymbol{u}_0^{(2)}$ is expressed as

$$\boldsymbol{u}_0^{(2)} = \sum_{i=2}^{n} c_i^{(2)} \boldsymbol{\psi}_i. \tag{32}$$

In practice, however, $\widetilde{\boldsymbol{\psi}}_1$ contains very small components parallel to $\boldsymbol{\psi}_2$, $\boldsymbol{\psi}_3$,..., $\boldsymbol{\psi}_n$. The true expression of $\boldsymbol{u}_0^{(2)}$ is

$$\boldsymbol{u}_0^{(2)} = c_1^{(2)} \boldsymbol{\psi}_1 + \sum_{i=2}^{n} c_i^{(2)} \boldsymbol{\psi}_i; . \tag{33}$$

When we repeatedly perform the SOA method and the power iteration, $\boldsymbol{\psi}_1$ results in dominant, and $\boldsymbol{u}_k^{(2)} \sim \widetilde{\boldsymbol{\psi}}_1$ and $\boldsymbol{u}_k^{(2)} \cdot \boldsymbol{v}_{k+1}^{(2)} \sim \nu_1$, although the magnitude of $c_1^{(2)}$ is small, where $\boldsymbol{v}_{k+1}^{(2)} = \mathsf{H}\boldsymbol{u}_k^{(2)}$.

Thus, we need to suppress the influence of the $\boldsymbol{\psi}_1$ term to extract the second largest eigenvalue ν_2 and the corresponding eigenvector $\widetilde{\boldsymbol{\psi}}_2$. We subtract the component parallel to $\widetilde{\boldsymbol{\psi}}_1$ every time we run the SOA model so that the $\boldsymbol{\psi}_1$ term does not become dominant. The correction of $\boldsymbol{v}_k^{(2)}$ and $\boldsymbol{u}_k^{(2)}$ is

$$\begin{aligned} \boldsymbol{v'}_k^{(2)} &= \boldsymbol{v}_k^{(2)} - \left(\boldsymbol{v}_k^{(2)} \cdot \widetilde{\boldsymbol{\psi}}_1 \right) \widetilde{\boldsymbol{\psi}}_1 \\ \boldsymbol{u}_k^{(2)} &\sim \frac{\nu_2^k}{\prod_{i=1}^{k} \left| \boldsymbol{v'}_i^{(2)} \right|} \left\{ c_2 \boldsymbol{\psi}_2 + \sum_{i=3}^{n} c_i \left(\frac{\nu_i}{\nu_2} \right)^k \boldsymbol{\psi}_i \right\} \ . \end{aligned} \tag{34}$$

If k is sufficiently large,

$$\begin{aligned} \widetilde{\boldsymbol{\psi}}_2 \equiv \boldsymbol{u}_k^{(2)} &\sim \frac{\nu_2^k}{\prod_{i=1}^{k} \left| \boldsymbol{v'}_i^{(2)} \right|} c_2 \boldsymbol{\psi}_2 \\ \nu_2 &\sim \boldsymbol{u}_k^{(2)} \cdot \boldsymbol{v'}_{k+1}^{(2)} \ . \end{aligned} \tag{35}$$

We can extract the third to n-th eigenvalues and eigenvectors in a similar manner to the second ones. When we extract the m-th $(m \leq n)$ ones,

$$\begin{aligned} \boldsymbol{u'}_0^{(m-1)} &\equiv \boldsymbol{u}_0^{(m-1)} - \left(\boldsymbol{u}_0^{(m-1)} \cdot \widetilde{\boldsymbol{\psi}}_{m-1} \right) \widetilde{\boldsymbol{\psi}}_{m-1}, \\ \boldsymbol{u}_0^{(m)} &\equiv \frac{\boldsymbol{u'}_0^{(m-1)}}{\left| \boldsymbol{u'}_0^{(m-1)} \right|}, \\ \boldsymbol{v}_k^{(m)} &\equiv \mathsf{H}\boldsymbol{u}_{k-1}^{(m)} \quad (k \geq 1), \\ \boldsymbol{v'}_k^{(m)} &\equiv \boldsymbol{v}_k^{(m)} - \sum_{j=1}^{m-1} \left(\boldsymbol{v}_k^{(m)} \cdot \widetilde{\boldsymbol{\psi}}_j \right) \widetilde{\boldsymbol{\psi}}_j, \\ \boldsymbol{u}_k^{(m)} &\equiv \frac{\boldsymbol{v'}_k^{(m)}}{\left| \boldsymbol{v'}_k^{(m)} \right|} \ . \end{aligned} \tag{36}$$

If k is sufficiently large,

$$\begin{aligned} \widetilde{\boldsymbol{\psi}}_m \equiv \boldsymbol{u}_k^{(m)} &\sim \frac{\nu_m^k}{\prod_{i=1}^{k} \left| \boldsymbol{v'}_i^{(m)} \right|} c_m \boldsymbol{\psi}_m, \\ \nu_m &\sim \boldsymbol{u}_k^{(m)} \cdot \boldsymbol{v'}_{k+1}^{(m)}. \end{aligned} \tag{37}$$

Using Eq. (9), posterior distribution function $p\left(\phi \mid \boldsymbol{Y}_{0:k}\right) \propto e^{-J}$ is written as

$$p\left(\phi \mid \boldsymbol{Y}_{0:k}\right) \propto \exp\left[-\frac{1}{2}(\boldsymbol{\phi}-\hat{\boldsymbol{\phi}})^{\top}\mathsf{H}(\boldsymbol{\phi}-\hat{\boldsymbol{\phi}})\right] . \tag{38}$$

When we decompose $\boldsymbol{\phi}$ and $\hat{\boldsymbol{\phi}}$ into the eigenvectors as Eq.(24), the posterior distribution function is also decomposed as

$$\begin{aligned} p(\phi|\boldsymbol{D}) &\propto \exp\left[-\frac{1}{2}\sum_{i=1}^{n}\nu_i(w_i-\hat{w}_i)^2\right] \\ &= \prod_{i=1}^{n}\exp\left[-\frac{1}{2}\nu_i(w_i-\hat{w}_i)^2\right] , \end{aligned} \tag{39}$$

where $\hat{w}_i$ is the optimum value of the weight coefficient. That is, the posterior distribution function is given as the product of normal distribution functions with mean $\hat{w}_i$ and variance $1/\nu_i$, and $\sqrt{1/\nu_i}$ indicates the uncertainty of weight w_i.

III. Experiment

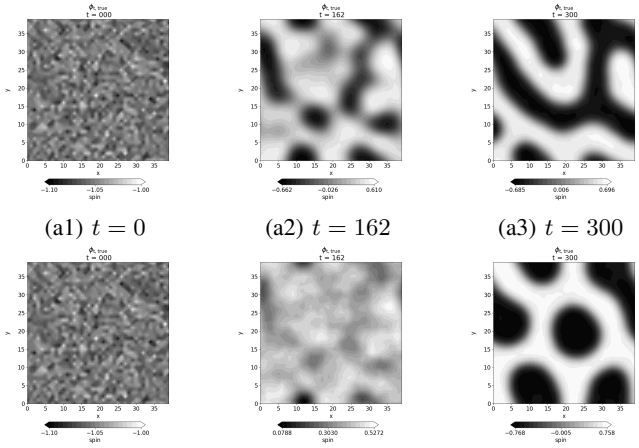
We conduct an experiment of decomposing the magnetic domain patterns in the following procedures; (1) we run the TDGL model to generate "true" data, the simulation data, and the artificial observation data of the magnetization $\phi(\boldsymbol{r})$, (2) we calculate cost function values by using the simulation and observation data, (3) we estimate the optimized magnetization $\hat{\phi}(\boldsymbol{r})$ by using the Adjoint method, (4) we extract the eigenvalues and eigenvectors of the Hessian matrix for the cost function by using the SOA method and the power iteration, and (5) we decompose the $\hat{\phi}(\boldsymbol{r})$ by using the extracted eigenvectors. The settings of the TDGL model to obtain the true data is shown in Table I. The initial values of ϕ_i is set randomly in the range $-1.1 \leq \phi_i \leq -1.0$ In the following section, we perform two patterns of experiments by setting the thickness of the material $A = 3.00, 3.75$.

TABLE I: Parameter setting in the TDGL model.

parameters	values
node count in x, y directions (L, L)	$(40, 40)$
grid spacing in x, y directions $(\mathrm{d}x, \mathrm{d}y)$	$(1, 1)$
time step $\mathrm{d}t$	0.25
time step count $N_t = 300/\mathrm{d}t$	1200
anisotropy strength α	1.0
exchange interaction strength β	1.0
dipole interaction strength γ	0.3
replica region count R	3
initial value of external magnetic field B_0	1.5
sweep rate of external magnetic field v_{B}	0.01

Figure 2 shows the time evolution of the true magnetic domain patterns simulated by the TDGL model. In the initial state, the magnetization at grid points are distributed at random. The random pattern gradually diffuses as the external magnetic field is weaken. After the external field is vanished at $t = 150$, characteristic domain patterns start to appear from the the diffused structure. The patterns develop depending on the material thickness A. The final patterns are like a labyrinth for $A = 3.00$ (a3), like sea-islands for $A = 3.75$ (b3), as are already reported in the previous studies [2][3].

In the labyrinth pattern (a3), a tongue-like domain with negative magnetization (black colored) extends from $(x, y) = (10, 40)$ to $(20, 25)$. The tongue-like domain is surrounded by a positive-magnetization domain (white colored), which is further surrounded by another negative-magnetization domain. Each domain has the width about 5. In the sea-island pattern (b3), an island domain with negative magnetization and with the diameter of 10 is located near the center of the pattern. The central island is surrounded by a positive domain, which is further surrounded by other negative island domains. The magnitude of the magnetization in a domain is almost uniform for both (a3) and (b3).



(a1) $t = 0$ (a2) $t = 162$ (a3) $t = 300$

(b1) $t = 0$ (b2) $t = 162$ (b3) $t = 300$

Fig. 2: Time evolution of magnetic domain patterns for (a) $A = 3.00$ and (b) $A = 3.75$.

Snapshots of the observed magnetization $\phi_{t,\mathrm{obs}}$ are synthesized by adding noise to the snapshots of the true magnetization $\phi_{t,\mathrm{true}}$. The observation noise ϵ_{obs} follows the normal distribution with mean zero and variance σ^2_{obs}, which gives the observed magnetization as follows:

$$\phi_{t,\mathrm{obs}} = \phi_{t,\mathrm{true}} + \epsilon_{\mathrm{obs}} . \tag{40}$$

Snapshots of the simulated magnetization are calculated with TDGL model in the same way as true value, except the slightly different initial value. The initial magnetization of the simulation ϕ_0 is generated by adding the noise to that of the true one $\phi_{0,\mathrm{true}}$. The noise follows the normal distribution with mean zero and variance σ^2_{sim}, which gives the simulated initial magnetization as follows:

$$\phi_0 = \phi_{0,\mathrm{true}} + \epsilon_{\mathrm{sim}} . \tag{41}$$

The simulated magnetization at each time step ϕ_t is temporally evolve from ϕ_0 following the TDGL model.

The cost function is defined by using the difference between the simulated and observed magnetization as follows:

$$J = \sum_{t=0}^{N_t} \frac{\left(\phi_t - \phi_{t,\mathrm{obs}}\right)^2}{2\sigma^2_{\mathrm{obs}}} . \tag{42}$$

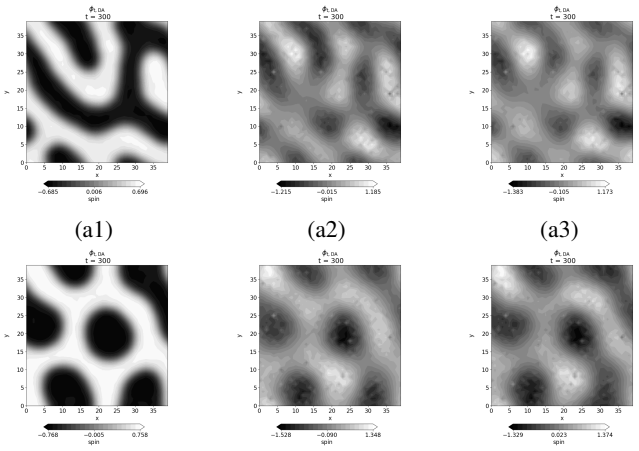
IV. Results and Discussion

A. Reconstruction of magnetic domain patterns

Figure 3 shows the original and reconstructed magnetic domain patterns by using decomposed eigenvectors extracted via the SOA and the power iteration. The eigenvectors are linearly combined as Eq. (24) to reconstruct the patterns, except that the number of the superimposed modes are reduced $\leq$ 20. Note that the magnetization in the original patterns, (a1) and (b1), is optimized via the 4DVar. Although the initial values of the optimized magnetization are different from the true magnetization (not shown here), the final values are the same as the one shown in Figures 2 (a3) and (b3). This indicates that the magnetic domain patterns generated by the TDGL model is not dependent on the initial values.

Figures 3 (a2) and (b2) are reconstructed patterns by using modes corresponding to the 20 largest eigenvalues. The reconstructed domain patterns have the same local structures as the original patterns. For Figure 3 (a2), the labyrinth pattern with local structures as the tongue-like domain and its surrounding double domains are reproduced, although the magnitude of the magnetization in each domain is not uniform. For Figure 3 (b2), all of the domains including the central and its surrounding sea islands are well reproduced.

Figures 3 (a3) and (b3) are reconstructed patterns by using 10 modes selected from the 20 modes. The 10 modes have the largest weights in absolute values in the 20 modes. The patterns are quite similar to the ones reconstructed by using 20 modes ((a2) and (b2)), and reproduce the original pattern ((a1) and (b1)). Although the original patterns have the dimension of 1,600 (40 $\times$ 40), out results demonstrate that we can explain the formation of the patterns by using only 10 modes.

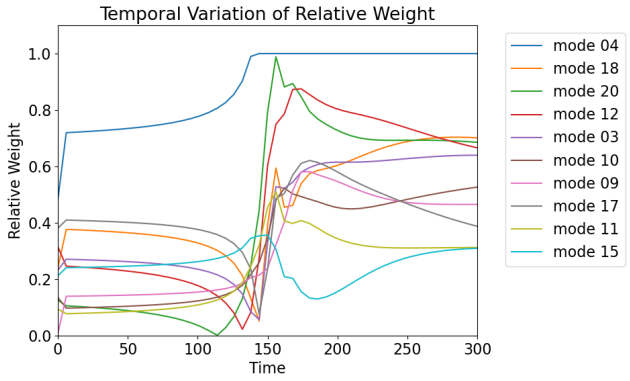


(a1) (a2) (a3)

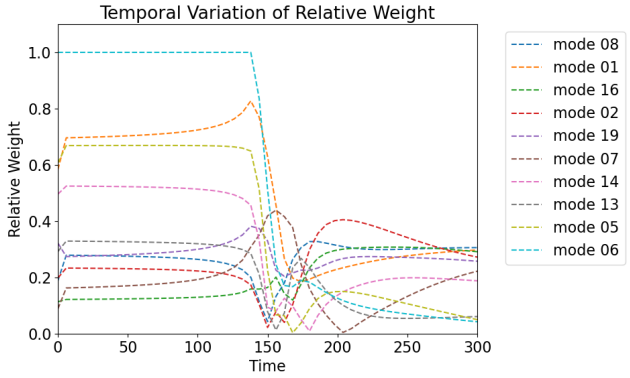
(b1) (b2) (b3)

Fig. 3: Original and reconstructed magnetic domain patterns at $t = 300$ for (a)$A = 3.00$ and (b)$A = 3.75$. (a1) and (b1) are the original patterns with magnetization optimized via 4DVar. (a2) and (b2) are reconstructed by using 20 modes with largest eigenvalues. (a3) and (b3) are reconstructed by using 10 modes with largest weights in the 20 modes.

B. Transition of Dominant modes



(a) Larger 10 weights at $t = 300$.



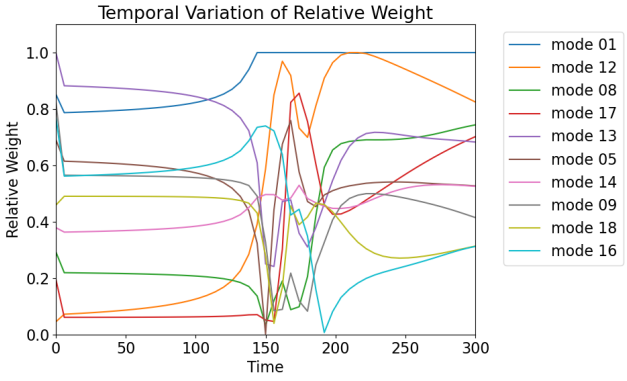
(b) Smaller 10 weights at $t = 300$.

Fig. 4: Time evolution of relative weight for $A = 3.00$. In the 20 weights corresponding to the eigenvectors used in Figure 3 (a2), larger 10 weights at $t = 300$ are plotted in (a), and smaller 10 weights are plotted in (b). The mode numbers correspond to the descending order of the eigenvalues. The weights in (a) are used to reconstruct the pattern in Figure 3 (a3).

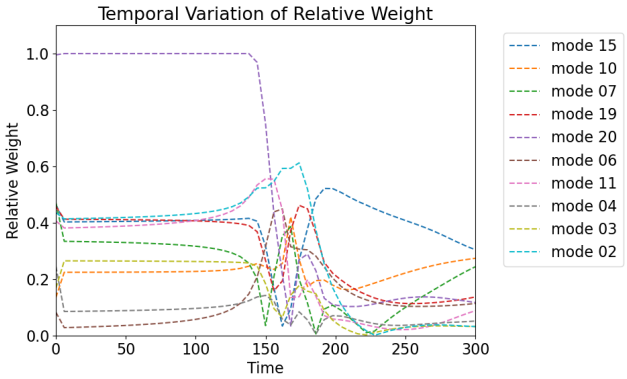
To investigate the modes that dominate the process of pattern formation, we trace the time evolution of the weights of the modes. Figures 4 and 5 show the weights in absolute values, which are divided by the maximum weight at each time point. The larger 10 weights at $t = 300$ (Figures 4a and 5a) correspond to the eigenvectors shown in Figures 6 and 7.

The magnitude of many of the weights drastically varies around $t = 150$, when the external vanishes, so that the dominant modes also changes. This corresponds to the formation of the magnetic domain after $t = 150$, as shown in Figure 2.

In Figure 4a, the weights of the modes 18, 20, 12, 03, 10 and 09 greatly increases around $t = 150$ from less than 0.4 to 0.4-1.0. This indicates that these mode are dominant in forming the labyrinth domain pattern. The mode 04 has the largest weight at $t > 150$, and it already has the second largest weight at $t < 150$. This suggest that the the mode 04 contributes to the pattern formation in a different manner from the above



(a) Larger 10 weights at $t = 300$.



(b) Smaller 10 weights at $t = 300$.

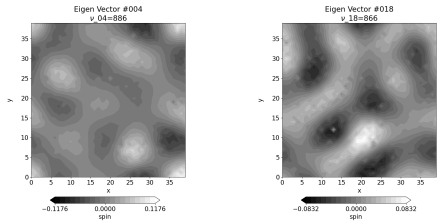
Fig. 5: Same as Figure 4 except $A = 3.75$.

six modes. The other modes less contribute to the labyrinth pattern because their weights are smaller.

In Figure 5a, the weights of the modes 12, 08 and 17 greatly increases around $t = 150$ from less than 0.3 to more than 0.6. This indicates that these mode are dominant in forming the sea-island domain pattern. The mode 01 has the largest weight at $t > 150$, and it already has the second largest weight at $t < 150$. This suggest that the the mode 01 contributes to the pattern formation in a different manner from the above three modes. The modes 13 and 05 have the large weights at $t = 300$, while their magnitude is larger at $t < 150$. The mode 14 also have the large weight at $t = 300$, although its enhancement from $t < 150$ is relatively small. This suggest that these modes are involved with the pattern formation through other mechanisms.

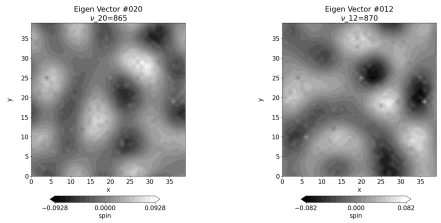
C. Eigenvectors of the Hessian matrix

As described in section II-D, The eigenmodes and the corresponding eigenvalues decomposes the posterior PDF. The inverse of an eigenvalue represents the uncertainty of the optimized weight. Since the mode numbers correspond to the descending order of the eigenvalues, the mode numbers also correspond to the ascending order of the uncertainties. It is difficult to find physical meanings from the eigenmodes in Figures 6 and 7. However, we can evaluate how uncertain each mode is. For example, considering the Figures 4a and 6, the mode 18 has the larger weight than mode 03, while the mode 18 is more uncertain than mode 03. It is important to neglect eigenmodes with smaller eigenvalues not only because it enhances the interpretability of the patterns but also because it reduces the uncertainty of the reconstruction.



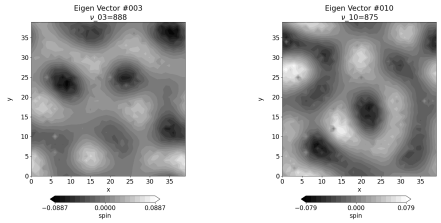
(a) Mode 04.

(b) Mode 18.



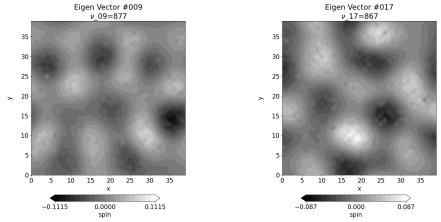
(c) Mode 20.

(d) Mode 12.



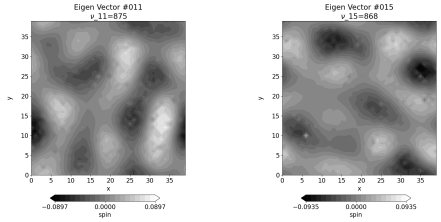
(e) Mode 03.

(f) Mode 10.



(g) Mode 09.

(h) Mode 17.



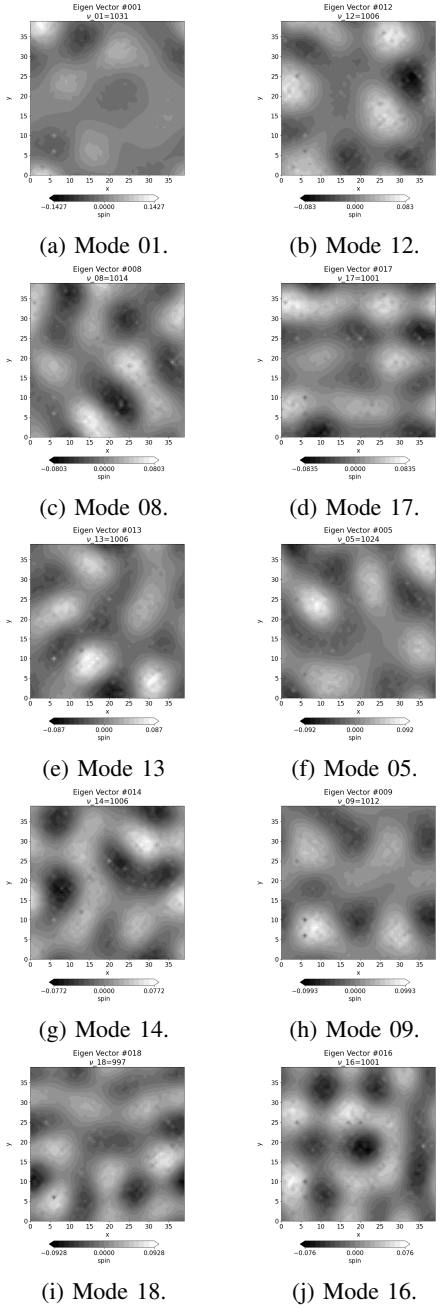
(i) Mode 11

(j) Mode 15.

Fig. 6: Eigenvectors for $A = 3.00$. These 10 modes are used to reconstruct the pattern in Figure 3 (a3). They also correspond to the weights in Figure 4.

V. Conclusions

We invented a method using the SOA model and the power iteration to extract eigenmodes of the Hessian matrix



(a) Mode 01. (b) Mode 12.

(c) Mode 08. (d) Mode 17.

(e) Mode 13 (f) Mode 05.

(g) Mode 14. (h) Mode 09.

(i) Mode 18. (j) Mode 16.

Fig. 7: Eigenvectors for $A = 3.75$. These 10 modes are used to reconstruct the pattern in Figure 3 (b3). They also correspond to the weights in Figure 5.

of the cost function for data fusion. We used this method to decompose the magnetic domain patterns generated by the TDGL model. We investigated the time evolution of the weights of the eigenmodes, and extracted only 10 modes to reconstruct the domain patterns with 1,600 dimension. In this study, we first extracted 20 modes with the largest eigenvalues, in other words, with the smallest uncertainties. These modes, however, does not necessarily have the largest weights in absolute values. Although the modes not extracted could also have large weight, the extracted modes successfully reconstructed the original domain patterns. This indicates that the extracted 10 modes are sufficient to explain the patterns domain patterns. If dozens or hundreds of the modes have similar weights, it may be difficult to reconstruct the domain patterns with 10-20 modes. In such cases, we might need to extract hundreds of modes and would find it difficult to explain the patterns by using the too many modes. We assume that parameters of the TDGL equation, α, β and γ, are known. Even if these parameters are unknown, we can optimize them with the 4DVar and can apply our method to decompose the magnetic domain patterns. Our method is applicable not only to the TDGL model, but also to any system with spatio-temporal data obtained by its system model and observations.

ACKNOWLEDGMENTS

We are grateful to Profs. Masato Okada, Ichiro Akai, and Dr. Ryoji Anzaki for their valuable discussions. This study was supported by JST CREST grants JPMJCR1761 and JPMJCR1763, MEXT Project for Seismology Toward Research Innovation with Data of Earthquake (STAR-E) grant JPJ010217, JSPS KAKENHI Grant-in-Aid for Scientific Research (A) No. 23H00466, Grant-in-Aid for Fund for the Promotion of Joint International Research (International Collaborative Research) No. 23KK0181, Grant-in-Aid for Challenging Research (Exploratory) No. 20K21785, and ERI UTokyo Joint Research ERI JURP 2024-A-04, 2022-B-06, and 2024-B-01.

REFERENCES

[1] C. Kittel, "Physical theory of ferromagnetic domains," *Rev. Mod. Phys.*, vol. 21, pp. 541–583, Oct 1949. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.21.541

[2] K. Kudo, M. Mino, and K. Nakamura, "Magnetic domain patterns depending on the sweeping rate of magnetic fields," *Journal of the Physical Society of Japan*, vol. 76, no. 1, p. 013002, 2007. [Online]. Available: https://doi.org/10.1143/JPSJ.76.013002

[3] R. Anzaki, S. Ito, H. Nagao, M. Mizumaki, M. Okada, and I. Akai, "Phase prediction method for pattern formation in time-dependent ginzburg-landau dynamics for kinetic ising model without a priori assumptions of domain patterns," *Phys. Rev. B*, vol. 103, p. 094408, Mar 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.103.094408

[4] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, "Modal analysis of fluid flows: An overview," *AIAA Journal*, vol. 55, no. 12, pp. 4013–4041, 2017. [Online]. Available: https://doi.org/10.2514/1.J056060

[5] F. X. L. Dimet and O. Talagrand, "Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects," *Tellus A*, vol. 38A, no. 2, pp. 97–110, 1986. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1600-0870.1986.tb00459.x

[6] Z. Wang, I. M. Navon, F.-X. Le Dimet, and X. Zou, "The second order adjoint analysis: theory and applications," *Meteorology and atmospheric physics*, vol. 50, no. 1, pp. 3–20, 1992.

[7] S. Ito, H. Nagao, A. Yamanaka, Y. Tsukada, T. Koyama, M. Kano, and J. Inoue, "Data assimilation for massive autonomous systems based on a second-order adjoint method," *Physical Review E*, vol. 94, no. 043307, 2016.

[8] Z. Wang, K. Droegemeier, and L. White, "The adjoint newton algorithm for large-scale unconstrained optimization in meteorology applications," *Computational Optimization and Applications*, vol. 10, no. 3, pp. 283–320, 1998.

[9] F.-X. Le Dimet, I. M. Navon, and D. N. Daescu, "Second-order information in data assimilation," *Monthly Weather Review*, vol. 130, no. 3, pp. 629–648, 2002.