International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 35, No. 4 (2021) 2150013 (20 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218001421500130

World Scientific
www.worldscientific.com

Automatic Binary Data Classification Using a Modified Allen–Cahn Equation

Sangkwon Kim* and Junseok Kim†

Department of Mathematics, Korea University
Seoul 02841, Republic of Korea
**ksk8863@korea.ac.kr*
†*cfdkim@korea.ac.kr*

Received 9 May 2019
Accepted 12 July 2020
Published 22 October 2020

In this paper, we propose an automatic binary data classification method using a modified Allen–Cahn (AC) equation. The modified AC equation was originally developed for image segmentation. The equation consists of the AC equation with a fidelity term which enforces the solution to be the given data. In the proposed method, we start from a coarse grid and refine the grid until the accuracy of the data classification reaches a given tolerance. Therefore, we can avoid a laborious trial and error procedure. For a numerical method for the modified AC equation, we use a recently developed explicit hybrid scheme. We perform several 2D and 3D computational tests to demonstrate the performance of the proposed method. The computational results confirm that the proposed algorithm is automatic.

Keywords: Binary data classification; modified Allen–Cahn equation; operator splitting method.

1. Introduction

Data classification is an important technique in analyzing data by separating unstructured data into meaningful structures. Classification is assigning one of the classes, which is agreed in advance to new data. Data classification involves learning labeled datasets and then generating decision boundary that separates each class. Therefore, techniques of generating decision boundary are such an important part of the classification. There are many kinds of techniques for generating decision boundary, such as support vector machine (SVM),[2,7] artificial neural networks (ANNs),[8,10] random forest (RF),[6] and so on. Advanced techniques of data classification producing good results are announced.[1,13,18,19] In addition to classification techniques mentioned above, classification methods based on solving partial differential equations are also being studied. One of them is the technique using the

† Corresponding author.

Allen–Cahn (AC) type equation. The AC equation was originally introduced as a phenomenological model for anti-phase domain coarsening in a binary alloy.[3] The AC equation has been successfully used to model a class of problems such as image in painting,[15] image segmentation,[4,14] and the mixture of two incompressible fluids.[12,16]

Using this feature, dual algorithm for segmentation has been proposed.[22] It is based on a penalty of dual variable along the edges, which only requires to solve a vectorial AC equation with linear ∇(div)-diffusion where operator splitting method (OSM) and fast Fourier transform (FFT) are implemented in time and space, respectively. Garcia-Cardona *et al.*[9] developed two graph-based algorithms for the multi-class segmentation of high-dimensional data by using a diffusive interface model. One is based on the gradient descent with convex splitting method and the other uses the Merriman–Bence–Osher (MBO) method. In this paper, we propose an automatic binary data classification method using AC type equation. Jeong and Kim[11] proposed an explicit hybrid numerical scheme to solve the AC equation. This algorithm proceeds in two steps. First, we solve the linear diffusion term by using the explicit scheme and solve the nonlinear term by using the closed-form solution thereafter. Bertozzi and Flenner presented the modified AC equation which adding a fitting term to the original AC equation for classification. In the previous research, there is a study on the accuracy and efficiency but not on mesh size of computational domain to solve the AC equation. Therefore, we propose an algorithm to find the mesh size automatically by solving the modified AC equation as an explicit hybrid scheme. The main purpose of this study is twofold: (a) the data conformation to the computational grid and (b) a new automatic binary classification algorithm.

The outline of this paper is as follows. We describe the governing equation in Sec. 2. Section 3 explains the numerical solution algorithm. We present the numerical results for several datasets in Sec. 4. Concluding remarks are given in Sec. 5.

2. Governing Equation

Because we are interested in binary data classification using the AC equation with a data fitting term, we briefly review the AC equation.[3]

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} = -\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t), \quad \mathbf{x}\in\Omega, \quad t>0,$$
$$\mathbf{n}\cdot\nabla\phi(\mathbf{x},t) = 0, \quad \mathbf{x}\in\partial\Omega, \quad t>0,$$

where $\Omega\subset\mathbb{R}^d$ $(d=2,3)$ is a domain, $\mathbf{x}=(x,y)$ or $\mathbf{x}=(x,y,z)$, $\mathbf{n}$ is outward unit normal vector on $\partial\Omega$, $\phi(\mathbf{x},t)$ is the phase-field function, $F(\phi)=0.25(\phi^2-1)^2$, ϵ is constant which is related to the interface transition thickness. It was originally introduced as a phenomenological model for anti-phase domain coarsening in a binary alloy.[3] The AC equation is the L^2-gradient flow of the Ginzburg–Landau free energy functional:

$$\mathcal{E}(\phi) = \int_\Omega \left(\frac{F(\phi)}{\epsilon^2} + \frac{1}{2}|\nabla\phi|^2 \right) d\mathbf{x}.$$

By minimizing this energy functional $\mathcal{E}(\phi)$, we have motion by mean curvature dynamics and smooth interfacial transitions. We apply the properties of the AC equation to data classification. In order to classify binary data, we want to minimize the following free energy functional:

$$\mathcal{E}(\phi) = \int_\Omega \left(\frac{F(\phi)}{\epsilon^2} + \frac{1}{2}|\nabla\phi|^2 + \frac{\lambda}{2}(\phi - f)^2 \right) d\mathbf{x}, \tag{1}$$

where λ is a fidelity parameter and f is a fitting term to a given data. By minimizing the free energy functional equation (1), the modified AC equation can be obtained in L^2 sense using the gradient descent method:

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} = -\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t) + \lambda[f(\mathbf{x}) - \phi(\mathbf{x},t)], \quad \mathbf{x} \in \Omega, \quad t > 0, \tag{2}$$

$$\mathbf{n} \cdot \nabla\phi(\mathbf{x},t) = 0, \quad \mathbf{x} \in \partial\Omega, \quad t > 0.$$

3. Numerical Scheme

In this section, a numerical scheme for the modified AC equation is presented in three-dimensional space $\Omega = (L_x, R_x) \times (L_y, R_y) \times (L_z, R_z)$. Two-dimensional numerical scheme can be similarly defined. Let N_x, N_y, and N_z be positive integers, $h = (R_x - L_x)/N_x = (R_y - L_y)/N_y = (R_z - L_z)/N_z$ be the uniform mesh size, and $\Omega_h = \{(x_i, y_j, z_k) | x_i = L_x + ih,\ y_j = L_y + jh, z_k = L_z + kh, 0 \le i \le N_x, 0 \le j \le N_y, 0 \le k \le N_z\}$ be the discrete computational domain. Let ϕ_{ijk}^n be approximations of $\phi(x_i, y_j, z_k, n\Delta t)$, where Δt is the time step. We split the governing equation into the following three equations using the operator splitting method:

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} = \Delta\phi(\mathbf{x},t), \tag{3}$$

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} = -\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2}, \tag{4}$$

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} = \lambda[f(\mathbf{x}) - \phi(\mathbf{x},t)]. \tag{5}$$

For $0 < i < N_x$, $0 < j < N_y$, $0 < k < N_z$, first we solve Eq. (3) using the explicit Euler method with the homogeneous Neumann boundary condition:

$$\frac{\phi_{ijk}^{n+\frac{1}{3}} - \phi_{ijk}^n}{\Delta t} = \Delta_h \phi_{ijk}^n, \tag{6}$$

where $\Delta_h \phi_{ijk}^n = (\phi_{i-1,jk}^n + \phi_{i+1,jk}^n + \phi_{i,j+1,k}^n + \phi_{i,j-1,k}^n + \phi_{ij,k+1}^n + \phi_{ij,k-1}^n - 6\phi_{ijk}^n)/h^2$. Here, we use the one-sided difference for the homogeneous Neumann boundary condition: $\phi_{0jk} = \phi_{1jk},\ \phi_{i0k} = \phi_{i1k},\ \phi_{ij0} = \phi_{ij1},\ \phi_{N_xjk} = \phi_{N_x-1,jk},\ \phi_{iN_yk} = \phi_{i,N_y-1,k},$ $\phi_{ijN_z} = \phi_{ij,N_z-1}$. Next, we solve the nonlinear equation

$$\frac{\partial\psi_{ijk}(t)}{\partial t} = \frac{\psi_{ijk}(t) - \psi_{ijk}^3(t)}{\epsilon^2}$$

analytically with the initial condition $\psi_{ijk}(0) = \phi_{ijk}^{n+\frac{1}{3}}$ and then set $\phi_{ijk}^{n+\frac{2}{3}} = \psi_{ijk}(\Delta t)$. That is,

$$\phi_{ijk}^{n+\frac{2}{3}} = \frac{\phi_{ijk}^{n+\frac{1}{3}}}{\sqrt{[1 - (\phi_{ijk}^{n+\frac{1}{3}})^2]e^{-\frac{2\Delta t}{\epsilon^2}} + (\phi_{ijk}^{n+\frac{1}{3}})^2}}.$$

Finally, we solve the fidelity equation for ϕ_{ijk}^{n+1}

$$\frac{\phi_{ijk}^{n+1} - \phi_{ijk}^{n+\frac{2}{3}}}{\Delta t} = \lambda(f_{ijk} - \phi_{ijk}^{n+1}). \tag{7}$$

Rewriting Eq. (7), we have

$$\phi_{ijk}^{n+1} = \frac{\phi_{ijk}^{n+\frac{2}{3}} + \lambda\Delta t f_{ijk}}{1 + \lambda\Delta t}.$$

This scheme is explicit, therefore, we do not need to solve a system of discrete equations implicitly and it is very fast. For the stability of the scheme, we have the constraint, $\Delta t < 0.5h^2/d$, where d is the dimension of space.[21]

4. Numerical Experiments

In this section, we perform numerical experiments to automatically find an efficient grid for several datasets such as two moons, Archimedean spiral, two linked tori, gyroid surface in $2D$ or $3D$. We start from a coarse grid and recursively refine the grid until the accuracy of the data classification reaches a given tolerance. We have different computational domains $\Omega = (L_x, R_x) \times (L_y, R_y) \times (L_z, R_z)$ for each case. We consider a time step $\Delta t = \alpha h^2/d$ with $\alpha = 0.24$ and $d = 2$ or $d = 3$, $N_x = (R_x - L_x)h + 1,\ N_y = (R_y - L_y)h + 1,\ N_z = (R_z - L_z)h + 1$, where h is grid size. Here, we use $\epsilon = hm/(2\sqrt{2}\tanh^{-1}(0.9))$ with positive constant m.

In the modified AC equation, the fidelity term is a fitting term to a given data. Because the given data is scattered in computational domain Ω, i.e. points from the given data may or may not be on grid points and the information should be distributed to adjacent grid points. Figure 1 illustrates the process of distributing information to adjacent grid points for one point data. Figure 1(b) illustrates the bilinear weighted distribution process to adjacent grid points when there is data of value 1 as shown in Fig. 1(a). Similarly, Fig. 1(d) illustrates the distribution process when there is data of value -1 as shown in Fig. 1(c).

If there are multiple data of 1 in a cell, then we assign the largest value to each grid point. Figures 2(a)–2(c) show the process of assigning values to each grid point

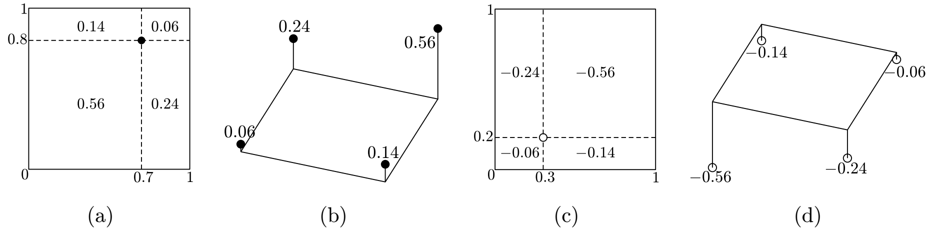


Fig. 1. Schematic illustration of the process of configuring a fidelity term one data in a cell. (a) is data with value 1, (b) is the linearly weighted distribution from (a), (c) is data with value −1, (d) is the linearly weighted distribution from (c).

when there are multiple data of 1 in a cell. Similarly, we assign the smallest value to each grid point if there are multiple data of −1 in a cell. Figures 2(e)–2(f) show the process of assigning values to each grid point when there are multiple data of −1 in a cell.

If there are multiple data mixed of 1 and −1 in a cell as shown in Fig. 3(a), we perform these positive and negative values separately and then sum these two matrix values as shown in Figs. 3(b)–3(d). Figure 4 shows the allocation of information for the two moons data to each grid point.

The accuracy of all experiments is calculated using the given data. The experimental class of data is determined using the solutions of the modified AC at the four vertices of the cell to which data belongs. The data is determined to be a class of −1 if

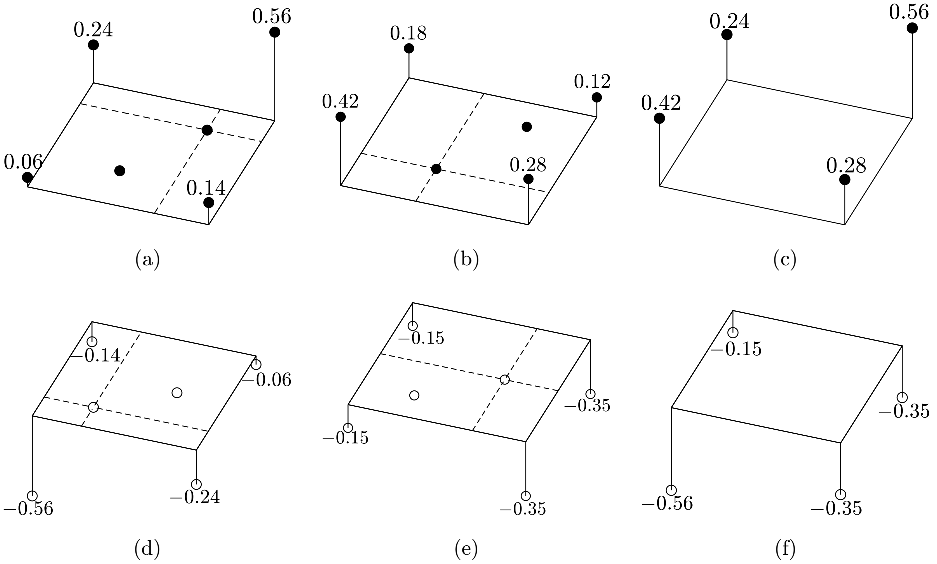


Fig. 2. Schematic illustration of the process of configuring a fidelity term multiple data in a cell. (a)–(b) are the linearly weighted distribution from data with value 1, (c) is a result to assign the largest value to each grid point, (d)–(e) are the linearly weighted distribution from data with value −1, (f) is a result to assign the smallest value to each grid point.

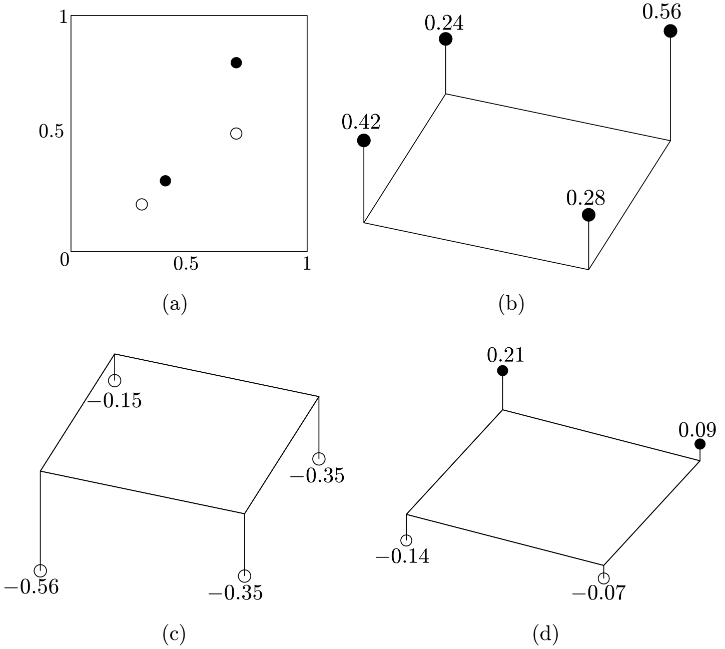


Fig. 3. Schematic illustration of the process of configuring a fidelity term mixed data. (a) is data mixed with value 1 and −1, (b)–(c) are the linearly weighted distribution separately, (d) is sum of (b) and (c).

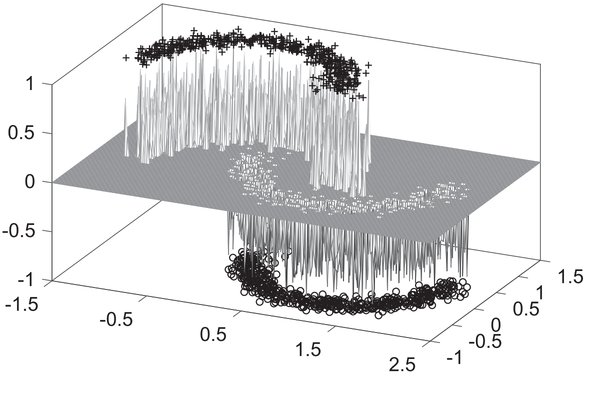


Fig. 4. Fidelity of two moons data.

the bilinear weighted sum of the four solutions is negative, otherwise a class of 1 is determined. The accuracy is defined as

$$E = \frac{1}{N} \sum_{i=1}^{N} \frac{\mathrm{TC}(i) + \mathrm{EC}(i)}{2},$$

where N is the number of data and TC and EC are true class and experimental class of data, respectively.

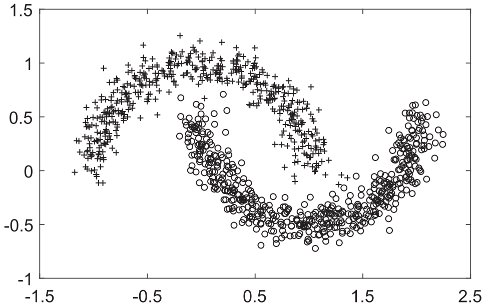


Fig. 5. Two moons data.

4.1. *Two-dimensional data classification*

The modified AC equation can be applied to classifying binary data. We consider two moons data, Archimedean spiral sample and Ho & Kleinberg's checkerboard dataset for $2D$ with a time step $\Delta t = \alpha h^2/2$, $\alpha = 0.24$, $\epsilon = \epsilon_4$, $\lambda = 50$ and tolerance 0.995.

4.1.1. *Two moons data*

Two moons data is constructed such that two half circles exist with radius of one. Each half circle consists of five hundred pieces of data, one half circle takes the value of -1 and the other takes the value of 1. That is, the data set has a thousand points. To generate data sets, first, the center of the upper half circle is the origin and the center of the lower half circle is $(1,\ 0.5)$. In other words, the data follows:

$$(x_1, y_1) = (\cos(\theta), \sin(\theta)) \quad \text{and} \quad (x_2, y_2) = (\cos(\theta) + 1, -\sin(\theta) + 0.5).$$

Then, adding random noise with a Gaussian distribution having standard deviation 0.1 to each data coordinate yields

$$(x_1, y_1) + (z_1, z_2) \quad \text{and} \quad (x_2, y_2) + (z_1, z_2),$$

where $z_1, z_2 \sim N(0, 0.1)$. The goal is to classify the two half circles by using the proposed algorithm for the modified AC equation in $\Omega = (1.5, 2.5) \times (-1, 1.5)$, see Fig. 5. We use parameters such as $h = 0.02$, $N_x = 201$, $N_y = 126$. Figure 6 shows the process of solving the modified AC equation using the parameters given above at $t = 0$, $50\Delta t$, $100\Delta t$ and $250\Delta t$. The left figures show the classification area of each class, the solid line is the area that classifies one class, and the dotted line is the area that classifies the other class. The right figures show the solution of the modified AC equation for each condition.

Next, we perform an adaptive mesh refinement by reducing mesh size h by half, beginning with a coarse mesh and until accuracy reaches a given tolerance. Figure 7 shows the process of classifying two moons data. The accuracies for the number of grid $(N_x, N_y) = (9, 6)$ and $(17, 11)$ are 0.990 and 0.995, respectively. Therefore, for this data, grid size $(N_x, N_y) = (17, 11)$ is guaranteed to be at least 0.995.

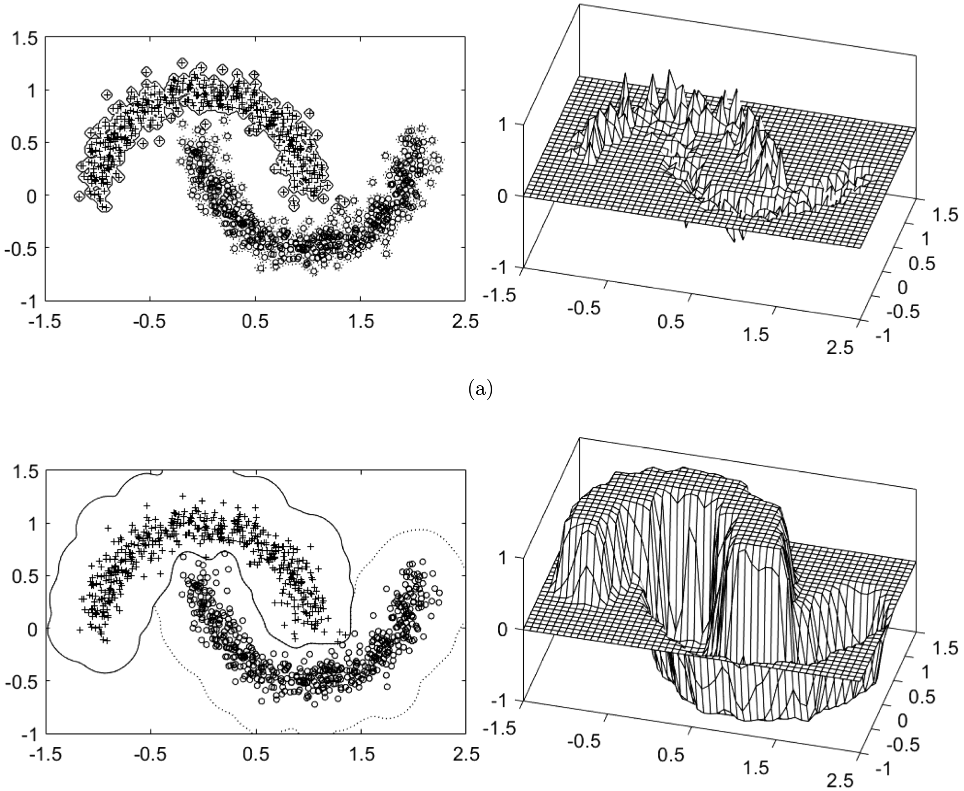
4.1.2. *Archimedean spiral*

In Fig. 8, the given data has two Archimedean spiral structures. The initial points are generated by Archimedean spiral function as

$$(x_1, y_1) = ((a + b\theta)\cos(\theta), (a + b\theta)\sin(\theta)),$$
$$(x_2, y_2) = (-(a + b\theta)\cos(\theta), -(a + b\theta)\sin(\theta)),$$

where $a = 0.9$ and $b = 0.3$ control the spiral rotation and the distance between successive turnings, respectively. We add random noise with a Gaussian distribution having standard deviation 0.15 to each data coordinate. Each data structure has one of the values -1 and 1. We apply the modified AC equation to this data samples to get a curve that classifies the data.

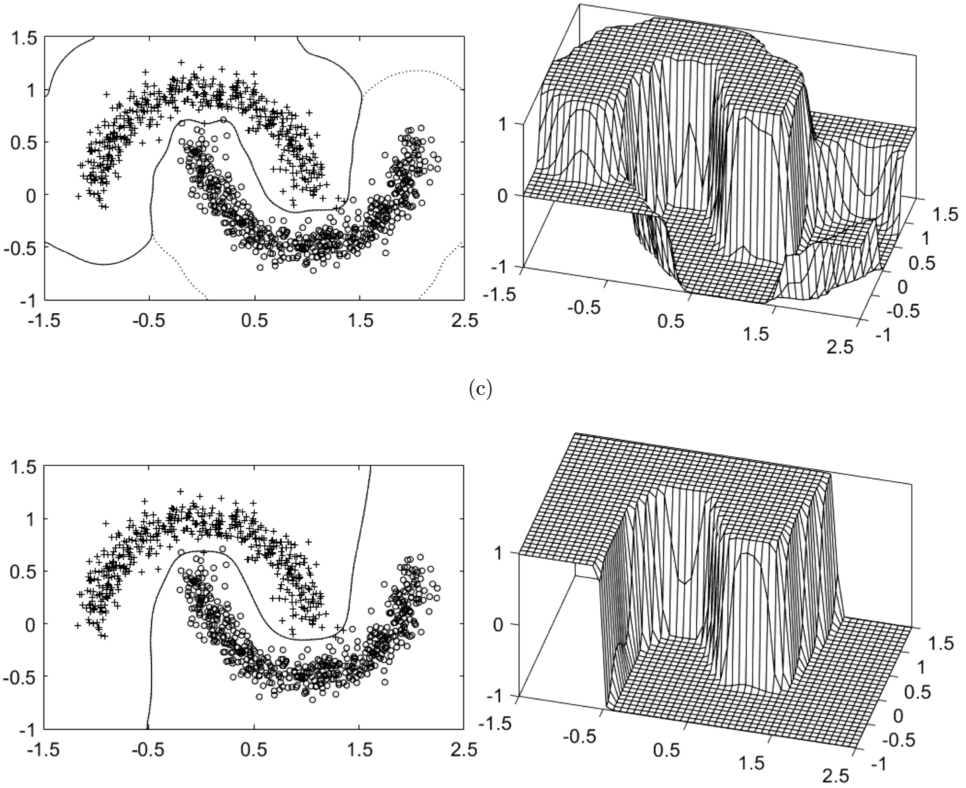
The goal is to find the grid size with the accuracy of the given tolerance as a result of classifying the two Archimedes spiral structure data using the proposed algorithm in $\Omega = (-4.5, 4.5) \times (-4, 4)$. The process of classifying the two Archimedes spiral



(a)

(b)

Fig. 6. Data classification process at (a) Initial condition, (b) 50 iterations, (c) 100 iterations and (d) 250 iterations.



(c)

(d)

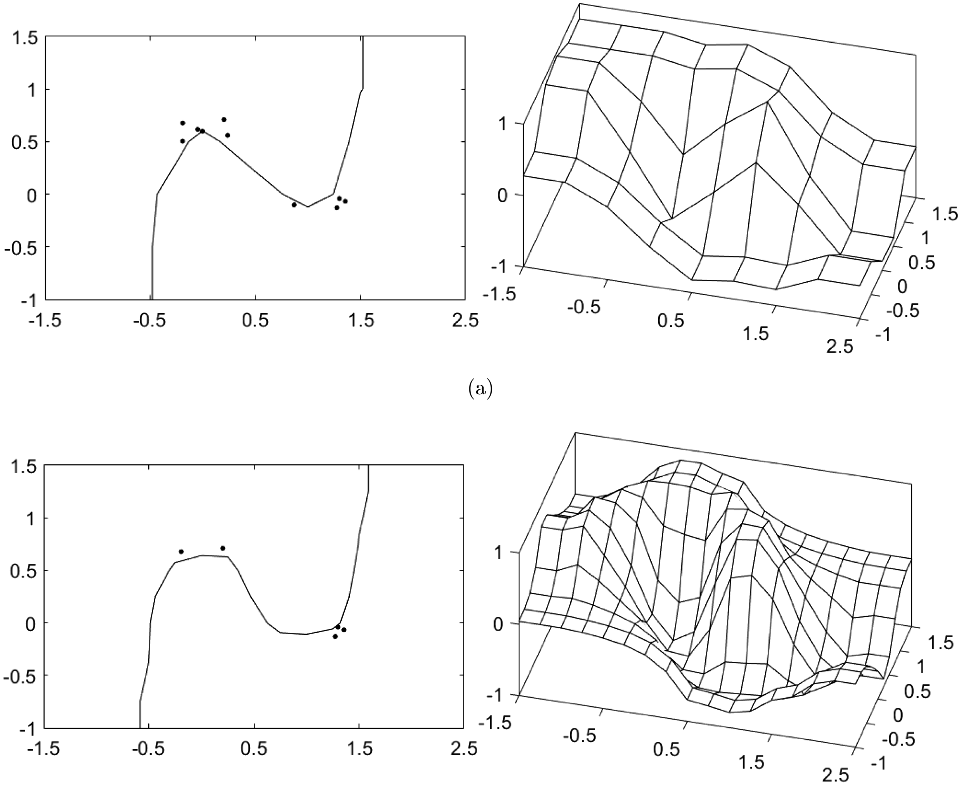
Fig. 6. (*Continued*)

structure data is shown in Fig. 9. The accuracies for the number of grid $(N_x, N_y) = (10, 9)$ and $(19, 17)$ are 0.950 and 0.998, respectively. Therefore, for this data, the grid size $(N_x, N_y) = (37, 33)$ is guaranteed to be at least 0.995.

4.1.3. *Ho & Kleinbergs checkerboard*

We consider Ho & Kleinbergs checkerboard dataset as shown in Fig. 10. It was used to show the nonlinear classification.[20] It contains two classes of samples produced under uniform distribution and its form is a checkerboard tile of 4×4. Each data class has one of the values -1 and 1 alternatively. We apply the modified AC equation to this data sample to get a curve that classifies the data.

The goal is to find the grid size with the accuracy of the given tolerance as a result of classifying the checkerboard dataset using the proposed algorithm in $\Omega = (-0.1, 1.1) \times (-0.1, 1.1)$. The process of classifying the checkerboard data is shown in Fig. 11. The accuracies for the number of grid $(N_x, N_y) = (13, 13)$, $(49, 49)$ and $(385, 385)$ are 0.559, 0.885 and 0.998, respectively. Therefore, for this data, the



(a)

(b)

Fig. 7. Accuracies with different mesh sizes: (a) $h = 0.5$, accuracy $= 0.990$ and (b) $h = 0.25$, accuracy $= 0.995$. In the left figure, solid line is the decision boundary for classifying data and dots are the data that failed classification.

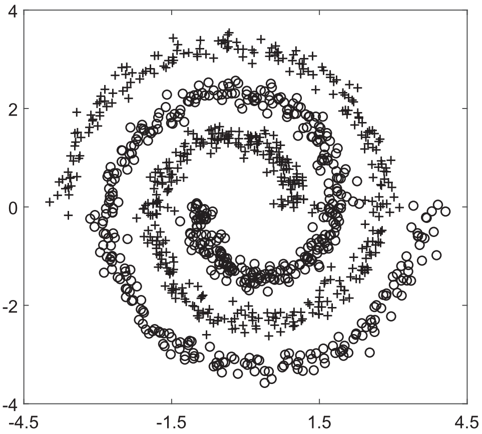


Fig. 8. Archimedean spiral.

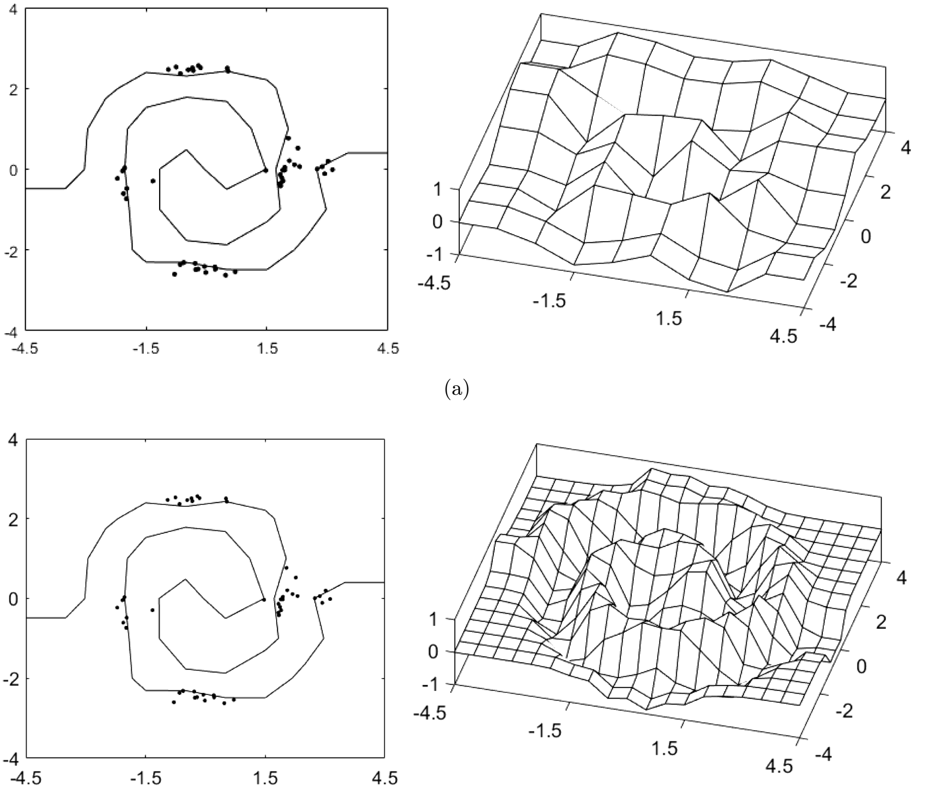


Fig. 9. Accuracies with mesh sizes: (a) $h = 1.0$, accuracy $= 0.950$ and (b) $h = 0.5$, accuracy $= 0.998$. In the left figure, solid line is the decision boundary for classifying data and dots are the data that failed classification.

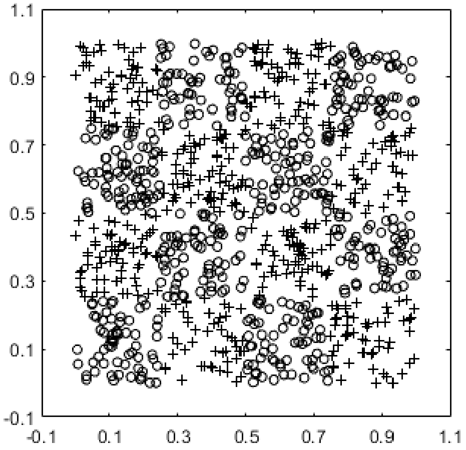
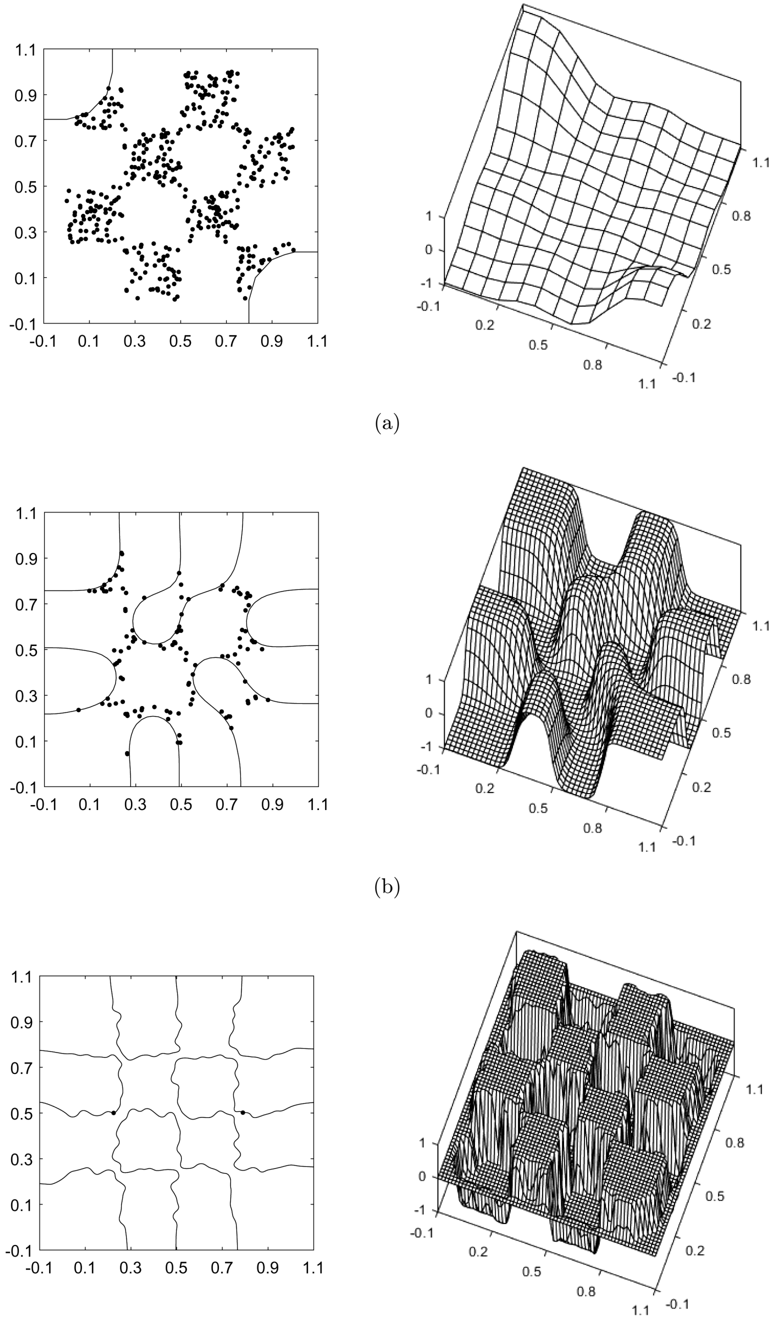


Fig. 10. Checker board dataset.



(a)

(b)

(c)

Fig. 11. Accuracy for different mesh sizes: (a) $h = 0.1$, accuracy $= 0.559$; (b) $h = 0.025$, accuracy $= 0.885$; and (c) $h = 0.003125$, accuracy $= 0.998$. In the left column, solid line is the decision boundary for classifying data and dots are the data that failed classification. In the right column, mesh plots are shown.

grid size $(N_x, N_y) = (385, 385)$ guarantees the accuracy of the classification to be at least 0.995.

4.2. *Three-dimensional data classification*

In this section, experiments are conducted in a similar way to experiments for $2D$. We consider two linked tori, two conchospiral and gyroid data as three-dimensional data. Unless otherwise noted, the following parameters are used: time step $\Delta t = \alpha h^2/3$ with $\alpha = 0.24$, $\epsilon = \epsilon_2$, $\lambda = 50$, computation domain $\Omega = (0,1) \times (0,1) \times (0,1)$, and tolerance 0.995.

4.2.1. *Two linked tori*

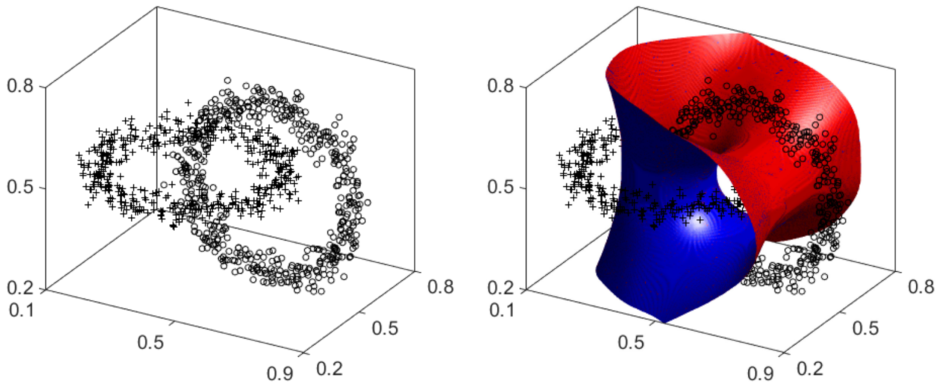
Two linked tori data are illustrated in Fig. 12(a). The initial data consist of two circles as follows:

$$(x_1(\theta,\rho), y_1(\theta,\rho), z_1(\theta,\rho)) = (a_1 + r\cos(\theta), b_1 + r\sin(\theta), c_1), \tag{8}$$

$$(x_2(\theta,\rho), y_2(\theta,\rho), z_2(\theta,\rho)) = (a_2 + r\cos(\theta), b_2, c_2 + r\sin(\rho)), \tag{9}$$

where $\theta, \rho \in [0, 2\pi)$, radius $r = 0.25$, and centers $(a_1, b_1, c_1) = (0.375, 0.5, 0.5)$, $(a_2, b_2, c_2) = (0.625, 0.5, 0.5)$. We add random noise with a Gaussian distribution having standard deviation 0.025 to each data coordinate. The values of the data generated by Eqs. (8) and (9) are 1 and -1, respectively. The number of data is five hundreds per circle, hence a thousand data are used.

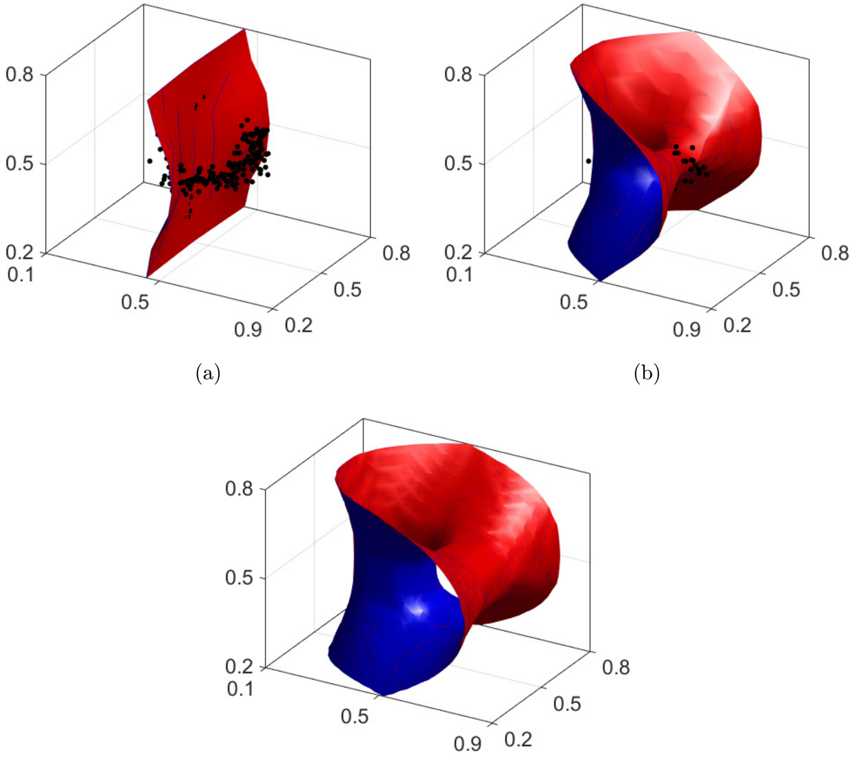
Figure 12(b) shows the decision boundary surface computed from the modified AC equation to classify two linked tori data with following parameters $h = 0.005$ and $N_x = N_y = N_z = 201$. Figures 13(a)–13(c) show the accuracy results with 0.726, 0.980 and 1.0 for the number of grid $(N_x, N_y, N_z) = (11, 11, 11), (21, 21, 21)$ and $(41, 41, 41)$, respectively. Therefore, for this data, the grid size $(N_x, N_y, N_z) = (41, 41, 41)$ guarantees the accuracy of the classification to be at least 0.995.



(a)

(b)

Fig. 12. (a) Two linked tori in three-dimensional space. (b) Surface classifying two linked tori data.



(a)

(b)

(c)

Fig. 13. Classification process in three-dimensional space with (a) $h = 0.1$, accuracy $= 0.726$, (b) $h = 0.05$, accuracy $= 0.980$ and (c) $h = 0.025$, accuracy $= 1.0$. Surface is the decision boundary classifying data and dots are the data that failed classification.

4.2.2. *Two conchospiral*

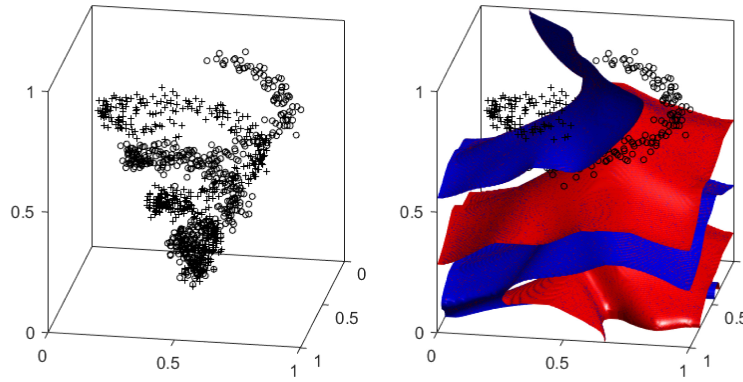
The two conchospiral dataset is illustrated in Fig. 12(a). The initial data consist of two conchospirals as follows:

$$(x_1, y_1, z_1) = (a + r\theta_1 \cos(\theta_1)/4\pi, b + r\theta_1 \sin(\theta_1)/4\pi, c + \alpha\theta_1), \tag{10}$$

$$(x_2, y_2, z_2) = (a + r\theta_2 \cos(\theta_2)/4\pi, b + r\theta_2 \sin(\theta_2)/4\pi, c + \alpha\theta_2), \tag{11}$$

where $0 \leq \theta_1 \leq 4\pi$, $\pi \leq \theta_2 \leq 5\pi$, radius $r = 0.4$, and center $(a, b, c) = (0.5, 0.5, 0.1)$. We add random noise with a Gaussian distribution having standard deviation 0.025 to each data coordinate. The values of the data generated by Eqs. (10) and (11) are 1 and -1, respectively. The number of data is five hundreds per conchospiral, hence a thousand data are used.

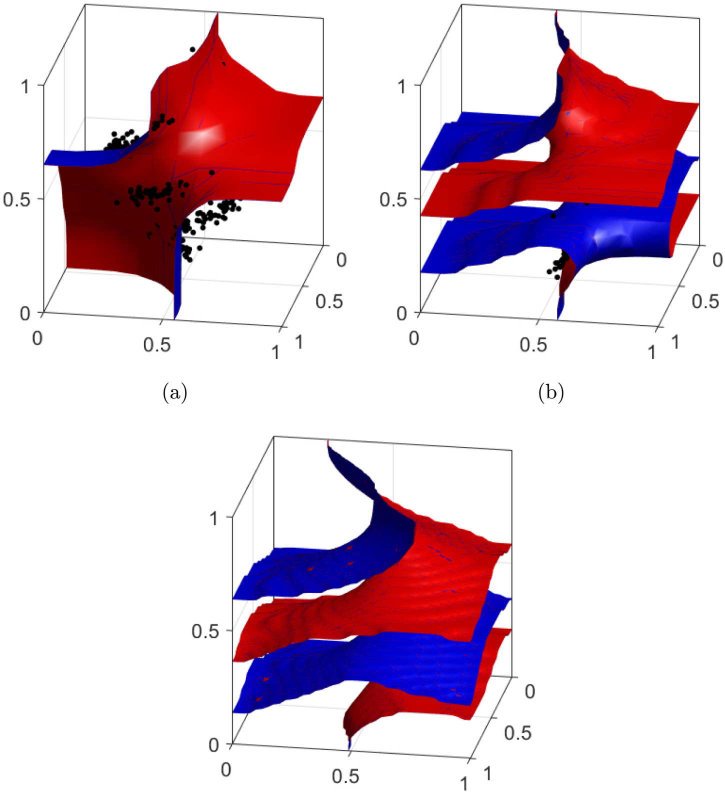
Figure 14(b) shows the decision boundary surface computed from the modified AC equation to classify two conchospiral data with following parameters $h = 0.005$ and $N_x = N_y = N_z = 201$. Figures 15(a)–15(c) show the accuracy results with 0.622, 0.970 and 0.996 for the number of grid $(N_x, N_y, N_z) = (11, 11, 11), (21, 21, 21)$ and



(a)

(b)

Fig. 14. (a) Two conchospiral in three-dimensional space. (b) Surface classifying two conchospiral data.



(a)

(b)

(c)

Fig. 15. Classification process in three-dimensional space with (a) $h = 0.1$, accuracy $= 0.622$, (b) $h = 0.05$, accuracy $= 0.970$, and (c) $h = 0.025$, accuracy $= 0.966$. Surface is the decision boundary classifying data and dots are the data that failed classification.

$(41, 41, 41)$, respectively. Therefore, for this data, grid size $(N_x, N_y, N_z) = (41, 41, 41)$ is guaranteed to be at least 0.995.

4.2.3. *Schoen gyroid*

The Schoen gyroid is an infinitely connected minimal surface and the surface partitions space into two disjoint but congruent regions, hence the volume fraction of each phase is 0.5. The initial points are generated by gyroid surface function as

$$\phi(x, y, z) = \begin{cases} +1 & \text{if } \sin(3\pi x)\cos(3\pi y) + \sin(3\pi z)\cos(3\pi x) + \sin(3\pi y)\cos(3\pi z) \geq 0, \\ -1 & \text{otherwise,} \end{cases}$$

where $x, y, z \in [0.2, 0.8]$. The dataset is selected randomly from $\phi = 1$ and $\phi = -1$ by five hundred each, hence a total of a thousand is used. We add random noise with a Gaussian distribution having standard deviation 0.02 to the coordinated of selected data. Figure 16(a) illustrates the initial data and Fig. 16(b) shows the decision boundary surface with $h = 0.005$ and $N_x = N_y = N_z = 201$. Figures 17(a)–17(d) show the accuracy results of 0.477, 0.729, 0.920 and 0.995 for the number of grid $(N_x, N_y, N_z) = (11, 11, 11), (21, 21, 21), (81, 81, 81)$ and $(321, 321, 321)$, respectively. Therefore, for this data, the grid size $(N_x, N_y, N_z) = (321, 321, 321)$ guarantees the accuracy of the classification to be at least 0.995.

4.2.4. *Performance comparison*

In this section, to validate the effectiveness of our proposed algorithm, we compare the performance of our algorithm and SVM on the two examples. Example 1 is the artificially generated Ripley's synthetic dataset. It contains 250 points and is generated from mixtures of two Gaussians. Figure 18 shows the classification results for

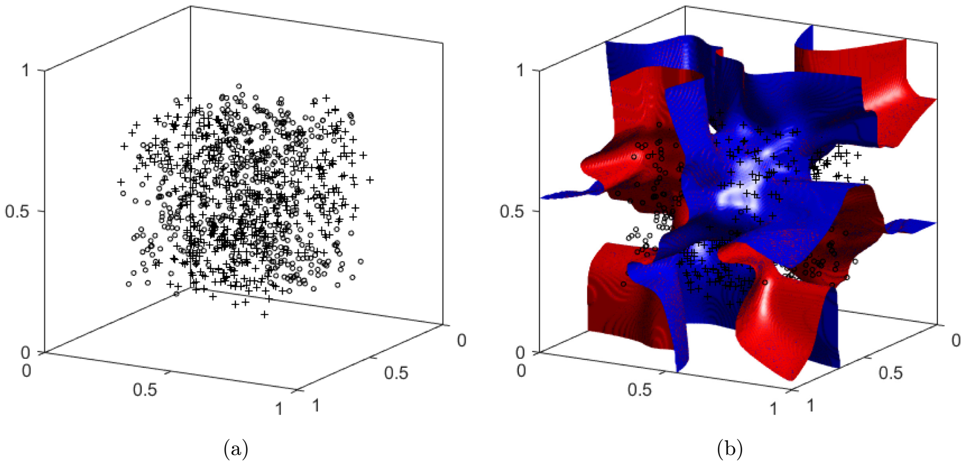
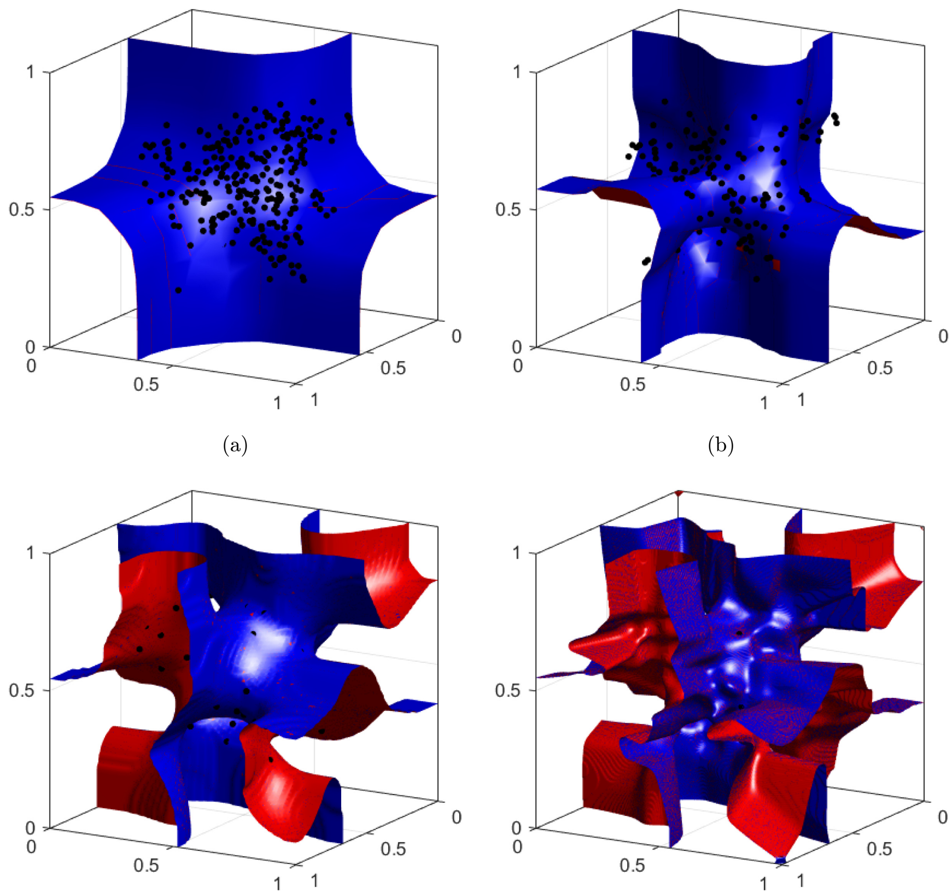


Fig. 16. Gyroid surface point data in three-dimensional space.



(a)

(b)

(c)

(d)

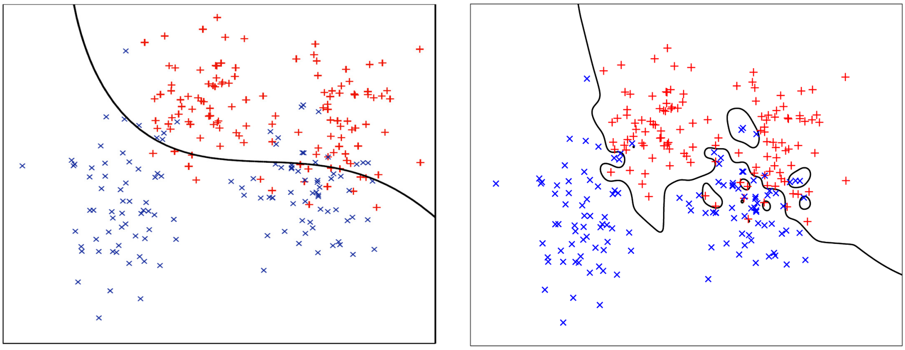
Fig. 17. Classification process for gyroid data with (a) $h = 0.1$, accuracy $= 0.477$, (b) $h = 0.05$, accuracy $= 0.729$, (c) $h = 0.0125$, accuracy $= 0.920$ and (d) $h = 0.003125$, accuracy $= 0.995$. Surface is the decision boundary for classifying data and dots are the data that failed classification.

Table 1. Classification accuracy on examples.

	Example 1 Accuracy (%)	Example 2 Accuracy (%)
SVM	90.6	95.56
Proposed algorithm	98.0	99.70

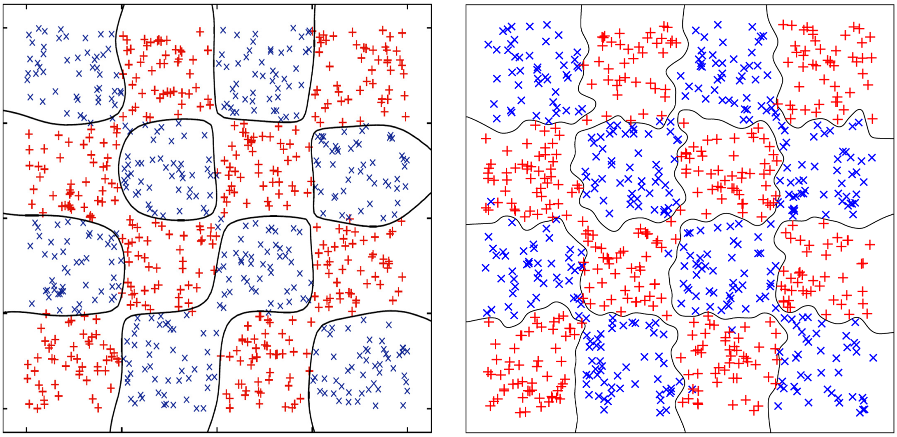
Example 1. Figures 18(a) and 18(b) show the results of the SVM and the proposed algorithm, respectively.

Example 2 is Ho & Kleinbergs checkerboard dataset. It contains two classes of samples produced under uniform distribution and its form is a checkerboard tile of 4×4. Figure 19 shows the classification results for Example 2. Figures 19(a) and 19(b)



(a) (b)

Fig. 18. Performance comparison on Example 1. (a): Reprinted from Peng and Xu[17] with permission from Springer Nature.



(a) (b)

Fig. 19. Performance comparison on Example 2. (a): Reprinted from Peng and Xu[17] with permission from Springer Nature.

show the results of the SVM and the proposed algorithm, respectively. For the results of the SVM, please refer to Peng and Xu.[17]

Table 1 shows the result of performance comparison for the two examples. For Example 1, the accuracy of our algorithm is about 98.0% and the accuracy of SVM is 90.6%. For Example 2, the accuracy of our algorithm is about 99.7% and the accuracy of SVM is 95.56%. It can be seen that accuracy of our algorithm gives better classification.

5. Conclusions

We presented an automatic binary data classification method using the AC equation with a fidelity term. The fidelity term enforces the solution to be the given data.

In the proposed method, we start from a coarse grid and refine the grid until the accuracy of the data classification reaches a given tolerance. Therefore, we can avoid a laborious trial and error procedure. To demonstrate the performance of the proposed algorithm, we carried out several $2D$ and $3D$ computational tests such as two moons, Archimedean spiral, Ho & Kleinbergs checkerboard, two linked tori, two conchospiral and gyroid surface. The computational results confirmed that the proposed algorithm is automatic.

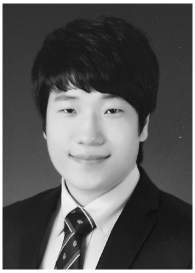
Acknowledgments

The corresponding author (J. S. Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1A2C1003053). The authors thank the editor and the reviewers for their constructive and helpful comments on the revision of this paper.

References

1. G. Alimjan, T. Sun, H. Jumahun, Y. Guan, W. Zhou and H. Sun, A hybrid classification approach based on support vector machine and k-nearest neighbor for remote sensing data, *Int. J. Pattern Recog. Artif. Intell.* **31**(10) (2017) 1750034.
2. G. Alimjan, T. Sun, Y. Liang, H. Jumahun and Y. Guan, A new technique for remote sensing image classification based on combinatorial algorithm of SVM and KNN, *Int. J. Pattern Recog. Artif. Intell.* **32**(7) (2018) 1859012.
3. S. M. Allen and J. W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, *Acta Metall.* **27**(6) (1979) 1085–1095.
4. M. Beneš, V. Chalupecký and K. Mikula, Geometrical image segmentation by the Allen–Cahn equation, *Appl. Numer. Math.* **51**(2–3) (2004) 187–205.
5. A. L. Bertozzi and A. Flenner, Diffuse interface models on graphs for classification of high dimensional data, *SIAM Rev.* **58**(2) (2016) 293–328.
6. L. Breiman, Random forests, *Mach. Learn.* **45**(1) (2001) 5–32.
7. N. Christianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines* (Cambridge University Press, Cambridge, 2002).
8. X. Cui, Y. Liu, Y. Zhang and C. Wang, Tire defects classification with multi-contrast convolutional neural networks, *Int. J. Pattern Recog. Artif. Intell.* **32**(4) (2018) 1850011.
9. C. Garcia-Cardona, E. Merkurjev, A. L. Bertozzi, A. Flenner and A. G. Percus, Multiclass data segmentation using diffuse interface methods on graphs, *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(8) (2014) 1600–1613.
10. S. Haykin, Neural networks and learning machines, *Pearson Upper Saddle River* 3 (2009).
11. D. Jeong and J. Kim, An explicit hybrid finite difference scheme for the Allen–Cahn equation, *J. Comput. Appl. Math.* **340** (2018) 247–255.
12. D. Jeong and J. Kim, Conservative Allen–Cahn–Navier–Stokes system for incompressible two-phase fluid flows, *Comput. Fluids* **156** (2017) 239–246.
13. S. Lacoste-Julien, F. Sha and M. I. Jordan, DiscLDA: Discriminative learning for dimensionality reduction and classification, *Adv. Neural Inf. Process. Syst.* (2009) 897–904.
14. Y. Li and J. Kim, An unconditionally stable hybrid method for image segmentation, *Appl. Numer. Math.* **82** (2014) 32–43.

15. Y. Li, D. Jeong, J. I. Choi, S. Lee and J. Kim, Fast local image inpainting based on the Allen–Cahn model, *Digital Signal Process.* **37** (2015) 65–74.
16. C. Liu and J. Shen, A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method, *Physica D Nonlinear Phenom.* **176**(3–4) (2003) 211–228.
17. X. Peng and D. Xu, Twin support vector hypersphere (TSVH) classifier for pattern recognition, *Neural Comput. Appl.* **24**(5) (2014) 1207–1220.
18. Y. H. Shao, W. J. Chen and N. Y. Deng, Nonparallel hyperplane support vector machine for binary classification problems, *Inf. Sci.* **263** (2014) 22–35.
19. H. Shi, X. Zhao, L. Zhen and L. Jing, Twin bounded support tensor machine for classification, *Int. J. Pattern Recog. Artif. Intell.* **30**(1) (2016) 1650002.
20. K. Teeyapan, N. Theera-Umpon and S. Auephanwiriyakul, A twin-hyperellipsoidal support vector classifier, *J. Intell. Fuzzy Syst.* (2018) 1–12.
21. J. W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*, Vol. 22 (Springer Science & Business Media, New York, 2013).
22. L. L. Wang and Y. Gu, Efficient dual algorithms for image segmentation using TV-Allen-Cahn type models, *Commun. Comput. Phys.* **9**(4) (2011) 859–877.



Sangkwon Kim is a Ph.D. candidate at the Department of Mathematics, Korea University, Korea. He received his M.S. degree in Applied Mathematics and M.S. degree in Mathematics from the Korea University in 2019 and Hanshin University in 2012, respectively. His research interests are in computational finance, computational fluid dynamics, machine learning, and numerical analysis.

Junseok Kim received his Ph.D. in Applied Mathematics from the University of Minnesota, USA in 2002. He also received his B.S. degree from the Department of Mathematics Education, Korea University, Korea in 1995. He joined the faculty of Korea University, Korea in 2008, where he is currently a full professor at Department of Mathematics. His research interests are in computational finance and computational fluid dynamics.