A MESHLESS COLLOCATION METHOD WITH A GLOBAL REFINEMENT STRATEGY FOR REACTION DIFFUSION SYSTEMS ON EVOLVING DOMAINS

SIQING LI [*][†] AND ZHONGHUA QIAO[‡]

Abstract. Turing-type reaction-diffusion systems on evolving domains arising in biology, chemistry and physics are considered in this paper. The evolving domain is transformed into a reference domain, on which we use a second order semi-implicit backward difference formula (SBDF2) for time integration and a meshless collocation method for space discretization. A global refinement strategy is proposed to reduce the computational cost. Numerical experiments are carried out for different evolving domains. The convergence behavior of the proposed algorithm and the effectiveness of the refinement strategy are verified numerically.

Key words. reaction-diffusion systems, evolving domain, meshless method, global refinement, pattern formation

AMS subject classifications. 35K57 65M70

1. Introduction.

In this paper, we consider Turing-type reaction-diffusion system (RDS) on evolving domain

$$\begin{cases} u_t + \bigtriangledown \cdot (\boldsymbol{c}u) = D_u \triangle u + \gamma f(u,v) \\ v_t + \bigtriangledown \cdot (\boldsymbol{c}v) = D_v \triangle v + \gamma g(u,v) \end{cases} \quad \text{on } \Omega_t, \tag{1.1}$$

where Ω_t is a time-dependent domain, u and v denote concentrations of two chemicals, D_u and D_v are diffusion constants, $\boldsymbol{c} = (x_t, y_t)$ is the domain motion velocity field, $f(u,v)$ and $g(u,v)$ represent reaction functions of two chemicals. The RDS model (1.1) governs pattern formation phenomenon in biology. It involves different aspects of embryonic growth and its influence on the actual pattern processing [26], for examples, the skin patterns of angelfish and spatial patters of teeth primordia on alligators. The domain size of these applications usually changes as time increases. In the paper, by assuming the domain growth function $\rho(t)$ spatially linear and isotropic, we have

$$x(t) = \rho(t)\xi, \quad y(t) = \rho(t)\eta, \tag{1.2}$$

with $\rho(0) = 1$ and (ξ, η) on reference domain. Theoretically, the reference domain can be chosen arbitrarily. In our paper, since the coordination relation between the evolving domain Ω_t and the initial domain Ω_0 is known as Eq. (1.2), we suppose that the reference domain is same with the initial domain for convenience in simulations. For the case that the initial domain is different with the reference domain, the computation can still be conducted by using the coordination transformation between the evolving domain and the reference domain. With different forms of reaction functions, various models have been proposed in the literature, such as Schnakenberg model [30], Gierer and Meinhardt model [5] and Gray-Scott model [10]. Different with Turing systems on fixed domains, for both chemicals, the evolving domain Ω_t introduces two additional terms: the advection terms $c \cdot \nabla u$, $c \cdot \nabla v$ and the dilution

[*] Taiyuan Univeristy of Technology, Shanxi, China , (lisq3@sustech.edu.cn)

[†] SUSTech International Center for Mathematics, Southern University of Science and Technology, Shenzhen, China.

[‡] Corresponding author. Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, HongKong (zhonghua.qiao@polyu.edu.hk).

terms. The advection terms come from the moving of elements by the local domain growth and the dilution terms represent the local volume change [5]. Turing-type models on evolving domains have a wide range of applications, for example the skin of angelfish [16, 17], the teeth primordia in the alligator and the pigmentation pattern formation on snakes [26], etc.

The stability analysis and cross-diffusion-driven instability conditions of (1.1) were considered in [24]. The global existence of (1.1) was proved in [31]. Since in general the classical solution of (1.1) cannot be found analytically, numerically study of pattern formations becomes necessary. There have been many works on numerical simulations of Turing-type models on both evolving and fixed domains. A hybrid method which utilizes the merits of a second order implicit integration factor method and the second-order implicit exponential time differencing method were proposed in [28] for stiff reaction-diffusion equations. In [20], a local kernel based meshless method was proposed for both linear and nonlinear reaction diffusion equations. An implicit-explicit time-stepping procedure with finite element spatial approximation was introduced in [18, 31] for solving (1.1). In [18], the authors also proved the stability and convergence of the proposed scheme. The method of line coupled with Gear's method were applied to resolve Turing patterns with uniformly domain growth functions in [5] and non-uniformly domain growth functions in [6]. The implicit backward Euler and moving grid finite element method were used to simulate the doubling splitting behavior of the patterns on evolving domains in [23]. The alternating direction implicit (ADI) method based on an orthogonal spline collocation (OSC) scheme was proposed in [9] to study Turing patterns with different domain growth functions. The same method was also applied to solve two dimensional Riesz space fractional diffusion equations in [32]. In [12, 27], a moving mesh finite element method with a Runge-Kutta time integration was used to solve Turing-type models on fixed domains, where computational costs are significant reduced with a mesh redistribution strategy. For reaction-diffusion equations coupled with moving boundaries, the Krylov implicit integration factor was proved to be a effective time discretization method in [22] by comparing with other methods. Another strategy for parabolic and RDS on the evolving domain is the finite element methods with time dependent meshes [3, 8]. Some theoretical results about these methods were proved in [2, 15]. Although they are not required to map the evolving domain to the reference domain, the mesh generation of these methods leads to additional costs.

In this paper, we will apply the meshless collocation method to solve Turing-type RDS on evolving domains. The Kansa method, proposed by E. J. Kansa in 1990 [13, 14], was a typical meshless method by imposing strong form collocation conditions on partial differential equations. To overcome the solvability problem of the Kansa method [11], the overdetermined Kansa method was applied to solve PDEs in [21]. Convergence results of the overdetermined Kansa method were proved in [4]. The weighted least-squares collocation methods with mixed boundary conditions were developed in [19]. In this work, after reformulating the RDS on the reference domain as [9, 18, 31], we will use the meshless collocation method based on radial basis functions and a second order semi implicit backward difference formula (SBDF2) [29] for spatial and temproral discretization respectively. The meshless method only needs discrete points in the domain [7] and allows us to solve the RDS on domains with more general shapes. Furthermore, considering the domain evolving property of the RDS (1.1) and the simplicity of redistributing discrete sets in meshless method, to save the computation cost, we will propose a global refinement strategy which increases/reduces collocation points as domain evolving. The rest of the paper is organized as follows. In Section 2, the numerical scheme and the global refinement strategy are introduced. Numerical examples to show the effectiveness of the proposed algorithm under different growing functions and domain shapes are contained in Section 3. The paper ends with some conclusions in Section 4.

2. The meshless method for RDS on evolving domains and the global refinement strategy. In this section, we will introduce the numerical scheme to solve Turing-type RDS (1.1), which contains three parts: Firstly, by coordinate transformation, (1.1) can be converted to a RDS on a reference domain; Secondly, using SBDF2 for time discretization and the meshless collocation method for space discretization, we solve the RDS on the reference domain; Finally, we compute the numerical solutions on the reference domain and then transform back to the physical domain. What is more, to save the computation time, we introduce a global refinement strategy in which the discrete set changes along with the volume of the domain.

2.1. The RDS on the reference domain. When solve the RDS on the evolving domain, one strategy is to transform the RDS on the evolving domain Ω_t to a reference domain Ω_c. This strategy has been successfully used to solve Turing-type RDS, see, e.g., [9,18,31]. We employ this strategy in our algorithm. To complete this article, we give a brief introduction of this transformation method in the following. The formulations for $u(x,y,t)$ is shown only since the same method can be applied to $v(x,y,t)$.

Let (ξ,η) be any point in the reference domain Ω_c and (x,y) be any point in the evolving domain Ω_t. They have the relation

$$x := x(\xi,\eta,t), \quad y := y(\xi,\eta,t). \tag{2.1}$$

From (2.1), each point (x,y) can be transformed to a fixed position (ξ,η) in Ω_c and we have $u(x,y,t) := \tilde{u}(\xi,\eta,t)$. We convert the equation for $u(x,y,t)$ in (1.1) to an equation for $\tilde{u}(\xi,\eta,t)$. First of all, the left hand side of the equation for $u(x,y,t)$ can be written as

$$u_t + \nabla\cdot(\boldsymbol{c}u) = \tilde{u}_t + \tilde{u}\nabla\cdot\boldsymbol{c}, \tag{2.2}$$

by applying the representation for $\tilde{u}_t$

$$\tilde{u}_t = u_t + u_x x_t + u_y y_t = u_t + \boldsymbol{c}\cdot\nabla u.$$

Next, the Laplacian of $u(x,y,t)$ can be represented in the (ξ,η) coordinate system as

$$\begin{aligned}\triangle u &= u_{xx} + u_{yy}\\ &= \tilde{u}_{\xi\xi}(\xi_x^2+\xi_y^2) + \tilde{u}_{\eta\eta}(\eta_x^2+\eta_y^2) + \tilde{u}_\xi(\xi_{xx}+\xi_{yy}) + \tilde{u}_\eta(\eta_{xx}+\eta_{yy})\\ &\quad + 2\tilde{u}_{\xi\eta}(\xi_x\eta_x+\xi_y\eta_y).\end{aligned} \tag{2.3}$$

Finally, in order to obtain the relation $\nabla\cdot\boldsymbol{c}$ in (2.2) with the growth function, we suppose that the initial domain coincides with the reference domain Ω_c

$$x(0) = x(\xi,\eta,0) = \xi, \;\; y(0) = y(\xi,\eta,0) = \eta.$$

By (1.2), the divergence of the flow velocity field $\boldsymbol{c}$ can be computed as

$$\nabla\cdot\boldsymbol{c} = x_{tx} + y_{ty} = \frac{2\rho_t(t)}{\rho(t)}, \tag{2.4}$$

by using the transformation in (1.2) and

$$x_t(t) = \rho_t(t)\xi = \frac{\rho_t(t)}{\rho(t)}x(t), \quad y_t(t) = \rho_t(t)\eta = \frac{\rho_t(t)}{\rho(t)}y(t).$$

By substituting results in (2.2), (2.3) and (2.4) to (1.1), we obtain the equivalent RDS on the reference domain Ω_c as

$$\begin{cases} \tilde{u}_t = \dfrac{D_u}{\rho(t)^2} \triangle \tilde{u} + \tilde{f}(\tilde{u}, \tilde{v}) \\ \tilde{v}_t = \dfrac{D_v}{\rho(t)^2} \triangle \tilde{v} + \tilde{g}(\tilde{u}, \tilde{v}) \end{cases} \quad \text{on } \Omega_c, \tag{2.5}$$

with

$$\tilde{f}(\tilde{u}, \tilde{v}) = -\frac{2\rho_t(t)}{\rho(t)}\tilde{u} + \gamma f(\tilde{u}, \tilde{v}), \quad \tilde{g}(\tilde{u}, \tilde{v}) = -\frac{2\rho_t(t)}{\rho(t)}\tilde{v} + \gamma g(\tilde{u}, \tilde{v}).$$

When solve the RDS numerically, boundary conditions and initial conditions are needed. In this paper, we study the RDS (1.1) subject to homogenous Neumann boundary conditions as

$$\frac{\partial u}{\partial n} = \frac{\partial v}{\partial n} = 0 \quad \text{on } \partial\Omega_t,$$

which implies that the RDS is self-organized with no external input. We transform boundary conditions on $\partial\Omega_t$ to the reference domain $\partial\Omega_c$ as

$$\begin{aligned} \frac{\partial u}{\partial n} &= n_x \frac{\partial u}{\partial x} + n_y \frac{\partial u}{\partial y} \\ &= n_\xi \left(\frac{\partial \tilde{u}}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \tilde{u}}{\partial \eta}\frac{\partial \eta}{\partial x} \right) + n_\eta \left(\frac{\partial \tilde{u}}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \tilde{u}}{\partial \eta}\frac{\partial \eta}{\partial y} \right). \end{aligned} \tag{2.6}$$

By applying (1.2) to (2.6), we conclude that the zero flux boundary conditions also holds for $\tilde{u}$ and $\tilde{v}$ as

$$\frac{\partial \tilde{u}}{\partial n} = \rho(t)\frac{\partial u}{\partial n} = 0, \ \frac{\partial \tilde{v}}{\partial n} = \rho(t)\frac{\partial v}{\partial n} = 0 \quad \text{on } \partial\Omega_c. \tag{2.7}$$

For initial conditions, we have

$$\tilde{u}(\xi, \eta, 0) = u(x, y, 0), \quad \tilde{v}(\xi, \eta, 0) = v(x, y, 0) \quad \text{on } \Omega_c \cup \partial\Omega_c. \tag{2.8}$$

In this paper, we will focus on solving the RDS (2.5) by meshless methods with boundary conditions in (2.7) and initial conditions in (2.8). There are two advantages of applying meshless methods to RDS on evolving domain. Firstly, for complicated domain shapes, meshless methods can solve the problems by discrete points instead of meshes by grid-based methods. Secondly, although the RDS on a fixed reference domain Ω_c solves Eq. (2.5) in the paper, the coefficients and reaction functions of RDS on Ω_c are related to time t. Furthermore, numerical solutions are also needed to be mapped back to evolving domain Ω_t. Thus, the discrete points should also be suitable to the evolving domain Ω_t as time t increases. For grid-based methods, more computation cost is required for mesh generation than redistribution of discrete points by meshless methods.

2.2. Discretization formulations. For the time discretization of (2.5), a SBDF2 formula [29] is employed. The diffusion terms are treated implicitly and reaction terms are treated explicitly. Let $\{t^n\}_{n=1}^{I}$ be a division of $[0,T]$ such that $t^n = n\Delta t$ with $\Delta t = T/I$. The semi-discretized formula of (2.5) can be written as

$$\begin{cases} \dfrac{3\tilde{u}^{n+1} - 4\tilde{u}^n + \tilde{u}^{n-1}}{2\Delta t} = \dfrac{D_u}{\rho(t)^2}\triangle \tilde{u}^{n+1} + 2f(\tilde{u}^n, \tilde{v}^n) - f(\tilde{u}^{n-1}, \tilde{v}^{n-1}) \\ \dfrac{3\tilde{v}^{n+1} - 4\tilde{v}^n + \tilde{v}^{n-1}}{2\Delta t} = \dfrac{D_v}{\rho(t)^2}\triangle \tilde{v}^{n+1} + 2g(\tilde{u}^n, \tilde{v}^n) - g(\tilde{u}^{n-1}, \tilde{v}^{n-1}) \end{cases} \text{on } \Omega_c,$$

where $\tilde{u}^n$ and $\tilde{v}^n$ are concentrations of $\tilde{u}$ and $\tilde{v}$ at time t^n. After simplification, we obtain

$$\begin{cases} \dfrac{3}{2}\tilde{u}^{n+1} - \dfrac{\Delta t D_u}{\rho(t)^2}\triangle \tilde{u}^{n+1} = 2\Delta t \tilde{f}(\tilde{u}^n, \tilde{v}^n) - \Delta t \tilde{f}(\tilde{u}^{n-1}, \tilde{v}^{n-1}) + 2\tilde{u}^n - \dfrac{1}{2}\tilde{u}^{n-1} \\ \dfrac{3}{2}\tilde{v}^{n+1} - \dfrac{\Delta t D_v}{\rho(t)^2}\triangle \tilde{v}^{n+1} = 2\Delta t \tilde{g}(\tilde{u}^n, \tilde{v}^n) - \Delta t \tilde{g}(\tilde{u}^{n-1}, \tilde{v}^{n-1}) + 2\tilde{v}^n - \dfrac{1}{2}\tilde{v}^{n-1} \end{cases} \text{on } \Omega_c. \tag{2.9}$$

From (2.9), concentrations of $\tilde{u}$ and $\tilde{v}$ at time t^{n+1} can be evaluated by using their values at time t^n and t^{n-1}.

We then apply the meshless collocation method introduced in [4, 19] for space discretization (2.9) in space. Let $Z = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_N\}$ be discrete trial centers in the domain Ω_c and $\Phi(\cdot,\cdot)$ be a radial basis function whose reproducing kernel Hilbert space is norm equivalent to $H^m(\Omega_c)$. Since collocation points may be different with the trial centers, collocation points in the domain Ω_c and on the boundary $\partial\Omega_c$ are denoted as $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\}$ and $Y = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_K\}$. The fill distance of any discrete set $\mathcal{S}$ is

$$h_{\mathcal{S}} := \sup_{\mathbf{x}\in\Omega} \min_{\chi_i \in \mathcal{S}} \|\boldsymbol{x} - \chi_i\|_{\ell_2(\mathbb{R}^d)}. \tag{2.10}$$

For any $u \in \mathcal{C}(\Omega)$, we define a discrete norm of u on collocation sets $\Upsilon = \{X,\ Y\}$ as

$$\|u\|_{\Upsilon} = \left(\sum_{\epsilon_i \in \Upsilon} u(\epsilon_i)^2\right)^{1/2}. \tag{2.11}$$

Applying formulations as in [19] for the Neumann boundary to system (2.9), the least-squares solution $U^{n+1} \in H^m(\Omega_c),\ \Omega_c \in \mathbb{R}^2$ in the finite dimensional trial space

$$\mathcal{U}_{Z,\Phi_m} := \text{span}\{\Phi(.,\boldsymbol{z}_i), \boldsymbol{z}_i \in Z\},$$

is defined as

$$U^{n+1} := \arg\inf_{w\in\mathcal{U}_{Z,\Phi_m}} \left\| \left(\frac{3}{2}I - \frac{\Delta t D_u}{\rho(t)^2}\Delta\right) w - h(t^{n+1}) \right\|_X^2 + \frac{\mathrm{v}(\partial\Omega_c)}{\mathrm{v}(\Omega_c)^{1/2}} m^{-5} h_X^{-2} \left\| \frac{\partial w}{\partial n} \right\|_Y^2, \tag{2.12}$$

where $\mathrm{v}(\partial\Omega_c)$, $\mathrm{v}(\Omega_c)$ are the size of the domain and boundary of the reference domain Ω_c, respectively, discrete norms on $X,\ Y$ are defined as in (2.11), and $h(\boldsymbol{x}, t^{n+1})$ is given by

$$h(\boldsymbol{x}, t^{n+1}) = 2\Delta t \tilde{f}(\tilde{u}^n(\boldsymbol{x}), \tilde{v}^n(\boldsymbol{x})) - \Delta t \tilde{f}(\tilde{u}^{n-1}(\boldsymbol{x}), \tilde{v}^{n-1}(\boldsymbol{x})) + 2\tilde{u}^n(\boldsymbol{x}) - \frac{1}{2}\tilde{u}^{n-1}(\boldsymbol{x}).$$

The concentration function $U^{n+1} \in \mathcal{U}_{Z,\Phi_m}$ can be approximated as

$$U^{n+1} = \sum_{i=1}^{N} \lambda_{U,i}^{n+1} \Phi(\cdot, \boldsymbol{z}_i), \tag{2.13}$$

with unknown coefficients $\Lambda_U^{n+1} = [\lambda_{U,1}^{n+1}, \ldots, \lambda_{U,N}^{n+1}]^T$. It had been shown in [19] that the numerical solution by (2.12) is convergent with respect to h_Z at the rate of $m-2$. To solve the (2.12), strong form collocation conditions are imposed as

$$\sum_{i=1}^{N} \lambda_{U,i}^{n+1} \left(\frac{3}{2} \sum_{j=1}^{M} \Phi(\boldsymbol{x}_j, \boldsymbol{z}_i) - \frac{\Delta t D_u}{\rho(t)^2} \sum_{j=1}^{M} \triangle \Phi(\boldsymbol{x}_j, \boldsymbol{z}_i) \right) = h(\boldsymbol{x}_j, t^{n+1}) \quad \text{for } \boldsymbol{x}_j \in X, \boldsymbol{z}_i \in Z.$$

On the boundary, zero flux boundary conditions are imposed at set Y as

$$\frac{\partial U^{n+1}}{\partial n} = \sum_{i=1}^{N} \lambda_{U,i}^{n+1} \sum_{j=1}^{N_b} \frac{\partial \Phi(\boldsymbol{y}_j, \boldsymbol{z}_i)}{\partial n} = 0 \quad \text{for } \boldsymbol{y}_j \in Y, \ \boldsymbol{z}_i \in Z.$$

From these $n_X + n_Y$ collocation conditions, we can find numerical solutions at t^{n+1} from (2.12) after computing unknown coefficients Λ_U^{n+1} for U^{n+1} by solving following equation system

$$\begin{bmatrix} \frac{3}{2} \Phi(X, Z) - \frac{\Delta t D_u}{\rho(t)^2} \triangle \Phi(X, Z) \\ \frac{\mathrm{v}(\partial \Omega_c)}{\mathrm{v}(\Omega_c)^{1/2}} m^{-5} h_X^{-2} \frac{\partial \Phi(Y, Z)}{\partial n} \end{bmatrix} \Lambda_U^{n+1} = \begin{bmatrix} h(X, t^{n+1}) \\ 0 \end{bmatrix}. \tag{2.14}$$

The concentration value U^{n+1} is given as (2.13) by using Λ_U^{n+1}. Then we can transform the solution to domain Ω_t by using $u^{n+1}(x, y, t) = U^{n+1}(\xi, \eta, t)$ and coordination relations in (1.2). It should be noted that the method becomes original Kansa method when collocation set satisfies $X \cup Y = Z$. Since the collocation method in (2.14) for the spatially discretization is meshless, it can be applied to more general domains and we will show this property in the simulation part.

2.3. The technique for global refinement strategy. Since we study the reaction-diffusion problems on the evolving domains, it is reasonable to change the number of discrete points along with the domain sizes. For mesh based methods, such as the finite element method [23], the change of data points also leads to the generation of new meshes which may cause additional computation time. For the meshless collocation methods, it is easier to redistribute the discrete points as the volume change of the physical domains. When the volume of the domain is small, we use fewer discrete points, and increase the number of data points as domain grows. The indicator function should depend on the underlying solutions, the domain growing functions and the properties of the domains such as the areas of the domains.

With definition in (2.10), we use h_{Z_0} to denote the fill distance of trial set of the domain at $t = 0$. A formula to decide fill distance h_{Z_t} at time t depends on the domain growth function $\rho(t)$ and initial fill distance h_{Z_0} is considered. To avoid the ill-conditioning problem encontered at small fill distance, we also give a lower bound h_L. Therefore, the fill distance for time t is computed as

$$h_{Z_t} = \max \left\{ \min \left\{ h_{Z_0}, \frac{\nu h_{Z_0}}{\rho(t)} \right\}, h_L \right\}, \tag{2.15}$$

with ν being a positive integer to control the change speed of the fill distance. With fill distance in (2.15), the number of discrete points N_t can be computed as

$$N_t = \left(\left\lfloor \frac{\mathrm{v}(\Omega_c)^{1/d}}{h_t} \right\rfloor \right)^d, \tag{2.16}$$

with d and $\mathrm{v}(\Omega_c)$ being the dimension and volume of the problem domain. In this paper, two kinds of growth functions are considered as

$$\rho(t) = \exp(d_1 t) \quad t \in [0, T], \tag{2.17}$$

$$\rho(t) = 1 + d_2 \sin\left(\frac{\pi t}{d_3}\right) \quad t \in [0, d_3], \tag{2.18}$$

in which $d_1,\ d_2,\ d_3$ are constants. For exponentially growth function in (2.17), the fill distance h_{N_t} firstly decreases as time increases and then fixes to a lower bound h_L. When use the trigonometric growth function in (2.18), the fill distance decreases to the lower bound h_L for $t \leq d_3/2$. And it increases to the initial h_{N_0} when t continuously increases to final time d_3. We will show this behavior numerically in the Section 3. The proposed algorithm is summarized in Algorithm 1.

Algorithm 1 The meshless collocation method with a global refinement strategy for RDSs

```
Input:
The parameters $\gamma, D_u, D_v$ , functions $f(u,v),\ g(u,v)$, time step $\Delta t$ and final time $T$;
Initial discrete sets with fill distance $h_{Z_0}$ on domain $\Omega_c$ and the lower bound $h_L$;
Iteration process:
Initialization: Generate initial value $U^0,\ V^0$; compute coefficient vectors $\Lambda_U^1,\ \Lambda_V^1$ for $U^1,\ V^1$ by one step forward Euler; Compute $\Lambda_U^2,\ \Lambda_V^2$ for $U^2,\ V^2$ by scheme in Section 2.2; And save coefficient vector $\Lambda_U^1,\ \Lambda_V^1 \in \mathbb{R}^{N_{t^1}}$ and $\Lambda_U^2,\ \Lambda_V^2 \in \mathbb{R}^{N_{t^2}}$;
For $n = 3, \cdots, I$,
Compute $h_{Z_{t^n}}$ by (2.15) and $N_{t^n}$ by (2.16);
if $h_{Z_{t^n}} = h_{Z_{t^{n-1}}}$,
Apply scheme in Section 2.2 to get $\Lambda_U^n,\ \Lambda_V^n$ and $U^n, V^n$; Save coefficients vector $\Lambda_U^{n-1},\ \Lambda_V^{n-1} \in \mathbb{R}^{N_{t^{n-1}}}$ and $\Lambda_U^n,\ \Lambda_V^n \in \mathbb{R}^{N_{t^n}}$;
else Firstly, use $\Lambda_U^{n-2},\ \Lambda_V^{n-2} \in \mathbb{R}^{N_{t^{n-2}}}$ and $\Lambda_U^{n-1},\ \Lambda_V^{n-1} \in \mathbb{R}^{N_{t^{n-1}}}$ to interpolate the updated $U^{n-2}, V^{n-2} \in \mathbb{R}^{N_{t^n}}$ and $U^{n-1}, V^{n-1} \in \mathbb{R}^{N_{t^n}}$. Then apply the scheme introduced in Section 2.2 to get $\Lambda_U^n,\ \Lambda_V^n$ and $U^n, V^n$; Save coefficient vectors $\Lambda_U^{n-1},\ \Lambda_V^{n-1} \in \mathbb{R}^{N_{t^{n-1}}}$ and $\Lambda_U^n,\ \Lambda_V^n \in \mathbb{R}^{N_{t^n}}$.
endif
endfor
```

3. Numerical simulations. In this section, we will numerically show the behavior of our proposed method. The functions $f(u,v)$ and $g(u,v)$ in RDS (1.1) have different representations in different models. In our numerical tests, we consider the Schnakenberg mode (SH model) with

$$\begin{cases} f(u,v) = a - u + u^2 v, \\ g(u,v) = b - u^2 v. \end{cases} \tag{3.1}$$

For all numerical tests, the Whittle-Matérn-Sobolev kernel is used

$$\Phi_m(x) := \|\epsilon x\|_2^{m-d/2}\mathcal{K}_{m-d/2}(\|\epsilon x\|_2) \quad \text{for } x \in \mathbb{R}^d,$$

where K_ν is the Bessel functions of the second kind. We will use unscaled kernel with $\epsilon = 1$ unless specificed otherwise.

3.1. Example 1. The reference domain sets as $\Omega_c = [0,1]^2$. The exponetially growth function in (2.17) is applied with $d_1 = 0.001$, i.e. $\rho(t) = \exp(0.001t)$. We use the parameters $D_u = 0.01,\ D_v = 1,\ a = 0.1,\ b = 0.9,\ \gamma = 1$ which is same with in [9]. To show the accuracy and effectiveness of our method, same setting with [9, EX1] is used with the exact solutions on Ω_t

$$u^*(x,y,t) = \cos(t)\cos\left(\frac{2\pi x}{\rho(t)}\right)\cos\left(\frac{\pi y}{\rho(t)}\right), \quad v^*(x,y,t) = \cos(t)\cos\left(\frac{\pi x}{\rho(t)}\right)\cos\left(\frac{2\pi y}{\rho(t)}\right). \tag{3.2}$$

The initial conditions are generated from the exact solutions as

$$u^*(x,y,0) = \cos\left(\frac{2\pi x}{\rho(t)}\right)\cos\left(\frac{\pi y}{\rho(t)}\right), \quad v^*(x,y,0) = \cos\left(\frac{\pi x}{\rho(t)}\right)\cos\left(\frac{2\pi y}{\rho(t)}\right). \tag{3.3}$$

From the transformation relation (1.2), the exact solutions in the reference domain Ω_c are

$$\tilde{u}^*(\xi,\eta,t) = \cos(t)\cos(2\pi\xi)\cos(\pi\eta), \quad \tilde{v}^*(\xi,\eta,t) = \cos(t)\cos(\pi\xi)\cos(2\pi\eta). \tag{3.4}$$

In order to ensure the exact solutions (3.4) for RDS, instead of the original reaction functions $\tilde{f}(\tilde{u},\tilde{v})$ and $\tilde{g}(\tilde{u},\tilde{v})$, we construct the new reaction functions $F(\tilde{u},\tilde{v}),\ G(\tilde{u},\tilde{v})$ as

$$F(\tilde{u},\tilde{v}) = \overline{f}(\xi,\eta,t) + \tilde{f}(\tilde{u},\tilde{v}), \quad G(\tilde{u},\tilde{v}) = \overline{g}(\xi,\eta,t) + \tilde{g}(\tilde{u},\tilde{v}),$$

with

$$\begin{aligned}\overline{f}(\xi,\eta,t) &= \frac{\partial\tilde{u}^*(\xi,\eta,t)}{\partial t} - \frac{D_u}{\rho(t)^2}\triangle\tilde{u}^*(\xi,\eta,t) - \tilde{f}(\tilde{u}^*,\tilde{v}^*),\\ \overline{g}(\xi,\eta,t) &= \frac{\partial\tilde{v}^*(\xi,\eta,t)}{\partial t} - \frac{D_v}{\rho(t)^2}\triangle\tilde{v}^*(\xi,\eta,t) - \tilde{g}(\tilde{u}^*,\tilde{v}^*).\end{aligned}$$

We compute the $L^2(\Omega)$ error for numerical solutions as

$$e_h = \sqrt{e_{h,u}^2 + e_{h,v}^2}$$

with

$$e_{h,u} = \|u^n - u^*\|_{L^2(\Omega)} \approx \frac{1}{n_E}\sqrt{\sum_{n=1}^{n_E}(u_i^n - u_i^*)^2}, \quad e_{h,v} \approx \frac{1}{n_E}\sqrt{\sum_{n=1}^{n_E}(v_i^n - v_i^*)^2}.$$

with n_E being the number of evaluation points in the domain.

We firstly consider the convergence behavior of the method. By $m = 4$ and time step $\Delta t = 0.01,\ T = 10$, Figure 3.1 (a) shows the convergence of the solutions to uniformly distributed discrete set $n_Z = [10^2, 15^2, 20^2, 25^2, 30^2]$ under different collocation sets $n_X + n_Y = kn_Z, k \in \{1,2,3\}$. It can be seen that similar accuracy and convergence behaviors are shown for both Kansa method and different overdetermined settings. The singularity which may appear in original Kansa method does not appear in our cases. Figure 3.1 (b) shows the $L^2(\Omega)$ error under different kernel smoothness by $k = 1$. It can be seen that higher kernel smoothness leads to higher accuracy solutions. However, the more computation cost is needed for higher m. The ill-condition problem may also appear when high kernel smoothness and small fill distance are used. Considering these factors, in our simulations, if not specified, we will use $k = 1$ and $m = 4$.

In Algorithm 1, for the initialization, we compute U^1, V^1 with initial data U^0, V^0 by one step forward Euler method. To show the influence of first step approximation, we test the convergence of the solutions with respect to t by both the first order forward Euler method and second order explicit Runge-Kutta (RK2) method. For RDS

$$\begin{cases} \tilde{u}_t = \dfrac{D_u}{\rho(t)^2} \triangle \tilde{u} + \tilde{f}(\tilde{u}, \tilde{v}) := R_1(\tilde{u}, \tilde{v}, t) \\ \tilde{v}_t = \dfrac{D_v}{\rho(t)^2} \triangle \tilde{v} + \tilde{g}(\tilde{u}, \tilde{v}) := R_2(\tilde{u}, \tilde{v}, t) \end{cases} \quad \text{on } \Omega_c,$$

the one-step forward Euler method to compute $\tilde{u}^1,\ \tilde{v}^1$ can be written as

$$\begin{cases} \tilde{u}^1 = \tilde{u}^0 + \Delta t R_1(u^0, v^0, 0), \\ \tilde{v}^1 = \tilde{v}^0 + \Delta t R_2(u^0, v^0, 0), \end{cases} \tag{3.5}$$

and RK2 method for the $u^1,\ v^1$ can be obtained by solving

$$\begin{cases} \tilde{u}^1 = \tilde{u}^0 + \dfrac{\Delta t}{2}\Big(R_1(\tilde{u}^0, \tilde{v}^0, 0) + R_1\big(\tilde{u}^0 + \Delta t R_1(\tilde{u}^0, \tilde{v}^0), \tilde{v}^0 + \Delta t R_2(\tilde{u}^0, \tilde{v}^0), \Delta t\big)\Big), \\ \tilde{v}^1 = \tilde{v}^0 + \dfrac{\Delta t}{2}\Big(R_2(\tilde{u}^0, \tilde{v}^0, 0) + R_2\big(\tilde{u}^0 + \Delta t R_1(\tilde{u}^0, \tilde{v}^0), \tilde{v}^0 + \Delta t R_2(\tilde{u}^0, \tilde{v}^0), \Delta t\big)\Big). \end{cases} \tag{3.6}$$

Figure 3.1 (c) shows the numerical results by parameters $\Delta t = [5E-1, 1E-1, 5E-2, 2E-2]$, $m = 4$, $T = 1$, $n_Z = 55^2$, $k = 1$, $n_X = n_Z$. It can be seen that second-order accuracy in time is obtained by both methods. We will use the forward Euler method for first step approximation in other simulations.

We test the performance of our global refinement strategy in this part. When using $m = 4$ and time step $\Delta t = 0.005$, Figure 3.2 shows results under fixed discrete set $N_Z = 25^2$ and different global refinement strategies with $N_0 = 15^2, \nu \in \{1, 2, 3\}$ in (2.15) and (2.16). Figure 3.2 (a) shows the $L^2(\Omega)$ error of the solutions as time increases. Figure 3.2 (b) and (c) plot discrete sets N_t as time step n_t increases and CPU time as time t increases. The $L^2(\Omega)$ error keeps at the same order under all four cases. However, since the initial set $N_0 = 15^2$ increases to the discrete set $N_T = 25^2$ under different ν, the CPU time by global refinement strategies is obviously smaller than the case by the fixed set $N_Z = 25^2$. It is noted that CPU time for $\nu = 3$ is about half of that used by fixed point set.

The same example was also considered in [9, Example 1]. The ADI extrapolated Crank-Nicolson orthogonal spline collocation method was used in [9]. Under the same settings, Table 3.1 shows the $L^2(\Omega)$ error and convergence rate of the solution under $\epsilon = 5,\ k = 1,\ \Delta t = 0.01,\ T = 750,\ m = 6$. The $\epsilon = 5$ is used to avoid the ill conditioning problem caused by high kernel smoothness $m = 6$. The convergence rate

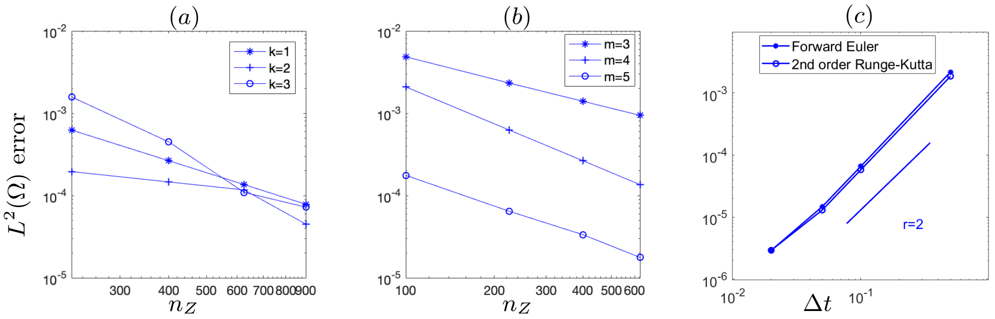

FIG. 3.1. *For the SH model with the unit square domain as reference domain, the $L^2(\Omega)$ error (a) under different overdetermined setting $n_X = kn_Z$ with $\Delta t = 0.01,\ T = 10$; (b) under different kernel smoothness with $k = 1,\ \Delta t = 0.01,\ T = 10$; (c) comparison between forward Euler method and RK2 with parameters $\Delta t = [5E-1, 1E-1, 5E-2, 2E-2],\ m = 4,\ T = 1,\ n_Z = 55^2,\ k = 1,\ n_X = n_Z$.*

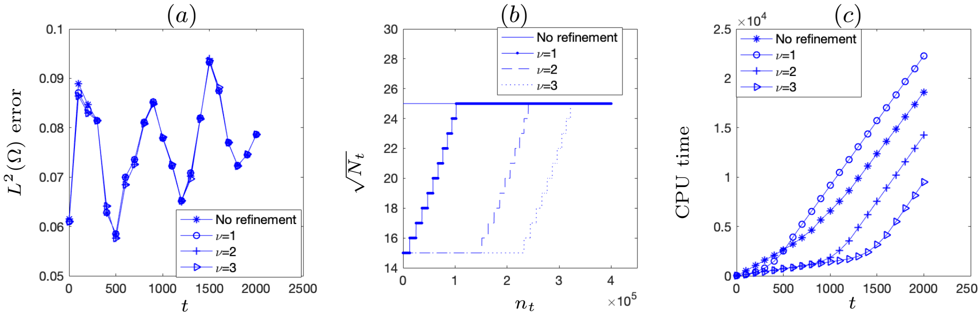

FIG. 3.2. *For the SH model with the unit square domain as reference domain, when $m = 4$, time step $\Delta t = 0.005,\ T = 2001$, the results under fixed discrete sets $n_Z = 25^2$ and different global refinement strategies with $\nu = 1, 2, 3$ in (2.15), $N_0 = 15^2$ in (2.16): (a) the $L^2(\Omega)$ error; (b) the discrete sets N_t; (c) CPU time.*

is computed as

$$\text{Rate} = \frac{\log(e_{h_1}/e_{h_2})}{\log(h_1/h_2)}.$$

In [9, Example 1], we can see that the order of convergence rate to discrete sets is $O(h^{r+1})$ in $L^2(\Omega)$ error with r being the degree of polynomial. From Table 3.1, the accuracy order of our method at $m = 6$ is the same with that in [9, Example 1] at $r = 3$ and similar convergence rates are also obtained by two methods.

3.2. Example 2. We consider the same example as in [9, Example 2] and [1] in this example. The reference domain Ω_c sets as the initial domain $\Omega_0 = [0, 1]^2$. The domain growth function is $\rho(t) = \exp(0.001t),\ t \in (0, 1950]$. Same with [1], we fix the parameters $D_u = 1,\ D_v = 10,\ a = 0.1,\ b = 0.9,\ \gamma = 10$. Same with [9, Example 2], the initial conditions are set as a small perturbation on steady state (u_∞, v_∞) of SH model without considering domain evolving

$$u(x, y, 0) = u_\infty + 10^{-3}\sum_{l=1}^{10}\cos(2\pi lx)/l, \quad v(x, y, 0) = v_\infty + 10^{-3}\sum_{l=1}^{10}\cos(2\pi lx)/l.$$

TABLE 3.1

When $\Delta t = 0.005$, $T = 750$ and $m = 6$, $\epsilon = 5$ in SH model, $L^2(\Omega)$ errors and convergence rates comparison between our scheme and [9, Example 1]

N	e_h error	Rate	e_h ($r = 3$ in [9, Example 1])	Rate
10^2	$0.807 * 10^{-3}$		$0.171 * 10^{-3}$	
20^2	$0.489 * 10^{-4}$	4.042	$0.106 * 10^{-4}$	4.001
30^2	$0.807 * 10^{-5}$	4.443	$0.209 * 10^{-5}$	4.002

And (u_∞, v_∞) obtained by solving $f(u_\infty, v_\infty) = 0$ and $g(u_\infty, v_\infty) = 0$ with $f,\ g$ in (3.1) is given as

$$u_\infty = a + b \quad \text{and} \quad v_\infty = \frac{b}{(a+b)^2}. \tag{3.7}$$

By using $\Delta t = 0.01, m = 4$, Figures 3.3 and 3.4 show patterns at $t \in \{750,\ 950,\ 1050,\ 1100,\ 1700,\ 1950\}$ by uniformly distributed point $n_Z = 30^2$ and by global refinement as shown in Figure 3.8 (a) respectively. It can be seen that the stable changing patterns about strips and spots are observed under both cases. In the global refinement setting, the parameter $\nu = 1$ in (2.15) and the discrete set increases from $N_t = 18^2$ to $N_t = 35^2$ in (2.16). In Figure 3.3, the one vertical stripe pattern first forms at $t = 750$. After semi-circular patterns at $t = 950, 1050$, a circlular pattern forms at $t = 1100$. One and half strips generates at $t = 1700$. The two horizon strips are formed at final time $t = 1950$. In Figure 3.4, the patterns by global refinement strategy are same with those by fixed points in Figure 3.3 except the location of strips at $t = 1700$, which again shows the effectiveness of our refinement strategy. Compared with results in [9, Figure 1b] and [1, Figure 2], exact same patterns are formed at time $t = 1100$ and final time $t = 1950$. At other times, similar patterns are also generated with different directions and locations. This might be caused by the difference of the discrete points in the domain.

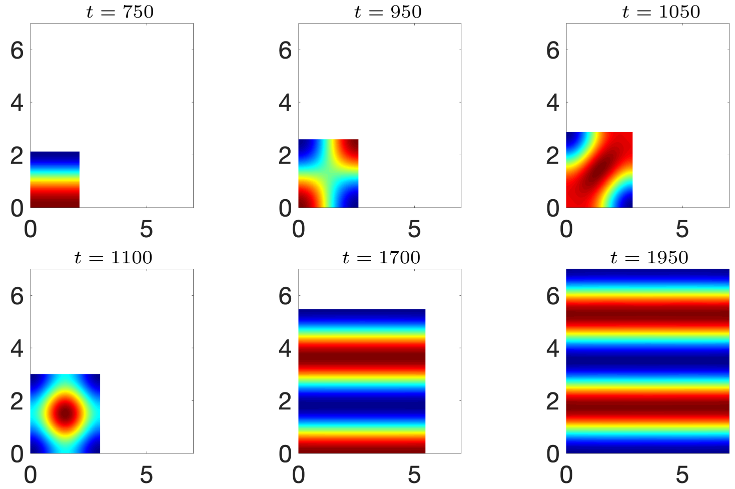

FIG. 3.3. *In Example 2, when use domain growth function $\rho(t) = \exp(0.001t)$ and parameters $D_u = 1,\ D_v = 10,\ a = 0.1,\ b = 0.9,\ \gamma = 10,\ m = 4,\ \Delta t = 0.01$, the patterns at different time t of SH model under fixed discrete points $n_Z = 30^2$.*

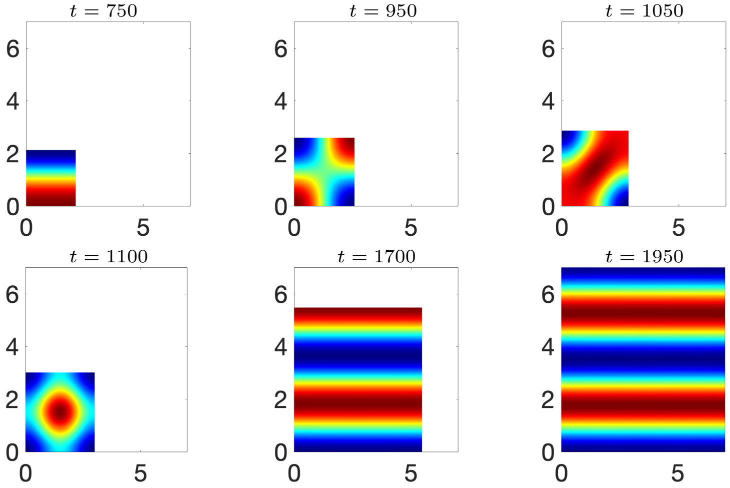

FIG. 3.4. *In Example 2, under same setting with Figure 3.3, the patterns at different time t of SH model under global refinement with the parameter $\nu = 1$ in* (2.15) *and the discrete set increasing from $n_Z = 18^2$ to $n_Z = 35^2$ in* (2.16) *as show in Figure 3.8 (a).*

3.3. Example 3. In this example, we consider the domain growth function (2.18) by using $d_2 = 9,\ d_3 = 1000$ as

$$\rho(t) = 1 + 9\sin\left(\frac{\pi t}{1000}\right), \quad t \in [0, 1000].$$

The reference domain set as $\Omega_0 = \Omega_c = [-0.25, 0.25]^2$. From the growth function, we can see that the initial domain Ω_0 grows to the $\Omega_{500} = [-2.5, 2.5]^2$ and then decreases back to the initial domain at final time $T = 1000$. We use parameters' values in the RDS (1.1) as $D_u = 0.01,\ D_v = 1,\ a = 0.1,\ b = 0.9,\ \gamma = 1,\ m = 4$ and $\Delta t = 0.005$. Same initial conditions with Example 2 are used. By observing patterns from global refinement strategy with different ν and considering the computation time, we find that $\nu = 5$ in (2.15) and N_t from 20^2 to 35^2 in (2.16) can result similar patterns in [31]. Figure 3.5 shows numerical simulations at time $t \in \{0,\ 80,\ 120,\ 150,\ 220,\ 380,\ 500,\ 740,\ 820,\ 910,\ 960,\ 1000\}$. The stable changing patterns, which contain different number of spots, are generated. As domain grows, we can observe the spot mode doubling behavoir at time $t \in \{150,\ 220,\ 380\}$ which are consistence with results in [9,31]. When domain reaches its largest size at $t = 500$, the number of spots pattern is 10 which is same as in [31]. When the domain size decreases for $t > 500$, same with [9,31], we can observe the mode merging behavior. The pattern merges to a single spot when the domain is close to the initial size as in [31].

3.4. Example 4. In this part, we consider the initial domain being a hexagon with vertexes as

$$\left[(1,0);\ \left(\frac{1}{2},\frac{\sqrt{3}}{2}\right);\ \left(-\frac{1}{2},\frac{\sqrt{3}}{2}\right);\ (-1,0);\ \left(-\frac{1}{2},-\frac{\sqrt{3}}{2}\right);\ \left(\frac{1}{2},-\frac{\sqrt{3}}{2}\right);\ (1,0);\right]$$

We use the domain growth function as $\rho(t) = \exp(0.001t)$ and parameters in RDS (1.1) as $D_u = 1,\ D_v = 10,\ a = 0.1,\ b = 0.9,\ \gamma = 114$ which are same as in [25, Sec. 4.1.4]. Initial conditions are same with Example 2. The uniformly distributed boundary

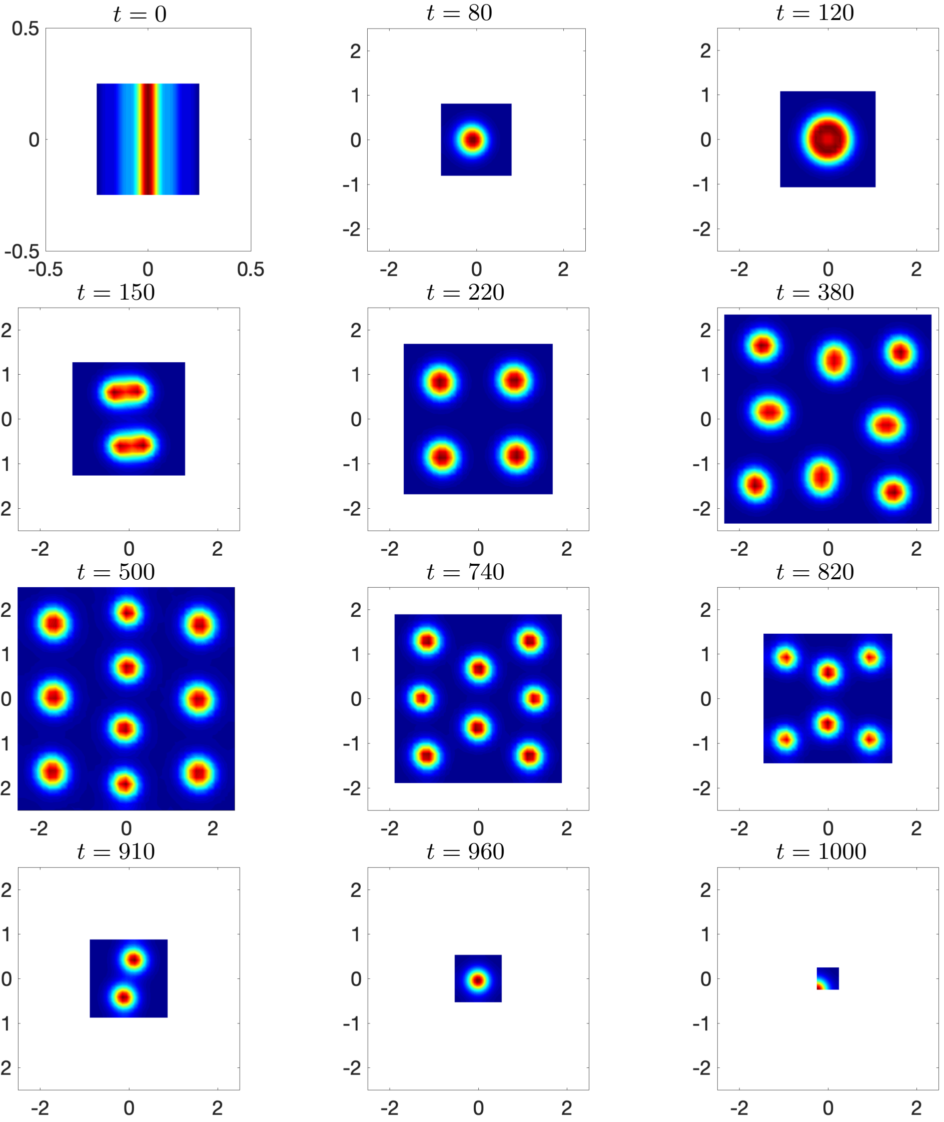

FIG. 3.5. *In Example 3, when use global refinement strategy with discrete sets in the domain as in Figure 3.8 (b) and the domain growth function as $\rho(t) = 1 + 9\sin(\pi t/1000)$, patterns generated by $D_u = 1,\ D_v = 10,\ a = 0.1,\ b = 0.9,\ \gamma = 10,\ m = 4$ and $\Delta t = 0.005$.*

discrete set with $N_b = 60$ is used. When the global refinement strategy with discrete set from $N_0 = 240$ to $N_T = 801$ and $\nu = 1$ in (2.15) as Figure 3.8 (c) , Figure (3.6) plots numerical sresults at time $t \in \{20,\ 200,\ 400,\ 470,\ 600,\ 860,\ 880,\ 950,\ 1000\}$ with $\Delta t = 0.005$. The numerical example is also considered in [25]. As domain grows, we can observe the stable changing patterns, which switch between the strips and spots. And the number of spots and strips increase as domain grows. We see the similar patterns by our methods at $t \in \{20,\ 200,\ 470,\ 880,\ 1000\}$ also appear in [25]. However, we observe spot patterns at $t = \{860,\ 950\}$ which are not shown in [25].

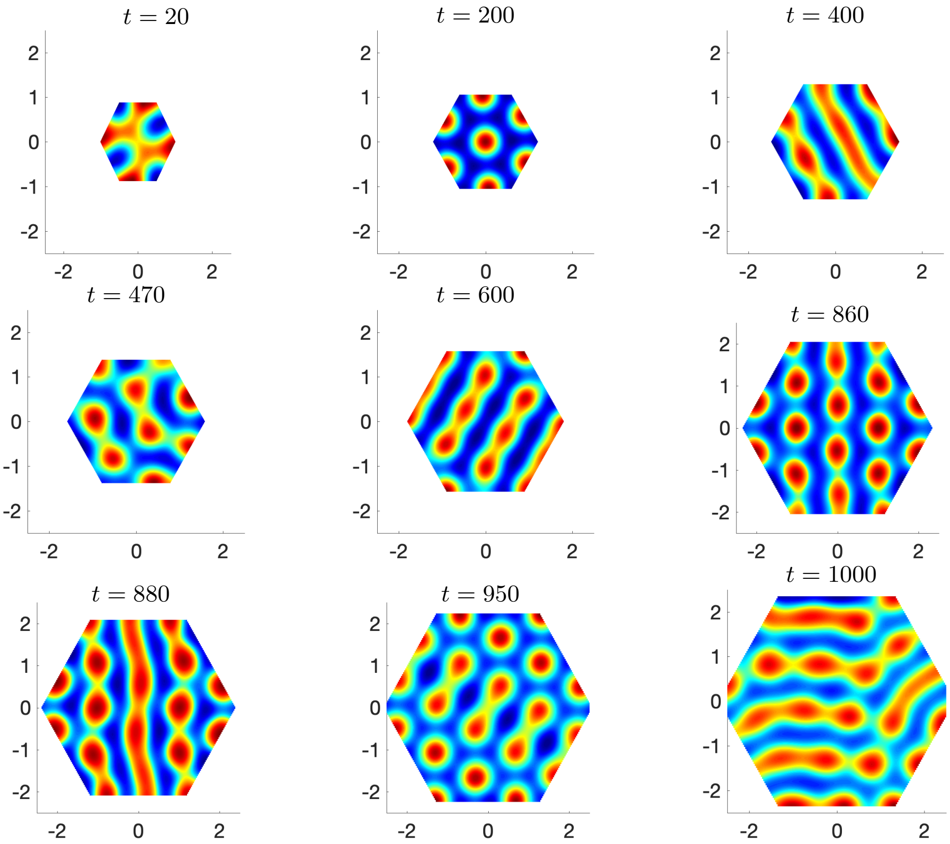

FIG. 3.6. *In Example 4, in the hexagon domain, when use global refinement strategy with discrete sets in the domain as in Figure 3.8 (c) and the domain growth function as $\rho(t) = \exp(0.001t)$, patterns generated by $D_u = 1,\ D_v = 10,\ a = 0.1,\ b = 0.9,\ \gamma = 114,\ m = 4$ and $\Delta t = 0.005$.*

3.5. Example 5. In this example, we solve the RDS on a star-shape domain. The initial domain denotes in polar coordinate system (θ, r) as

$$r(\theta) = \left(\cos(4\theta) + \sqrt{\frac{18}{5} - \sin^2(4\theta)}\right)^{\frac{1}{3}}, \quad \theta \in [0, 2\pi].$$

The domain grows exponentially as $\rho(t) = \exp(0.001t)$. We use parameters in the RDS (1.1) as $D_u = 1,\ D_v = 10,\ a = 0.1,\ b = 0.9,\ \gamma = 10,\ \Delta t = 0.005$. Same initial conditions with Example 2 are used. By global refinement strategy with $\nu = 2$ and uniformly distributed set from $N_0 = 154$ to $N_t = 665$ as in Figure 3.8 (d), Figure 3.7 shows numerical simulations at time $t \in \{100,\ 1000,\ 1300,\ 1500,\ 1700,\ 2000\}$. Similar with results in the hexagon domain, we can observe that the stable changing patterns switch between strips and spots again. The number of strips and spots also increases as the domain grows.

4. Conclusion. We applied the meshless collocation method to the Turing-type RDS on evolving domains. Since only the discrete points are needed, a simple global refinement strategy is proposed in which the trial centers are increased as domain evolving. The refinement strategy can save the CPU time compared with the results by fixed trial set. We consider both regular and irregular domain shapes. We can observe

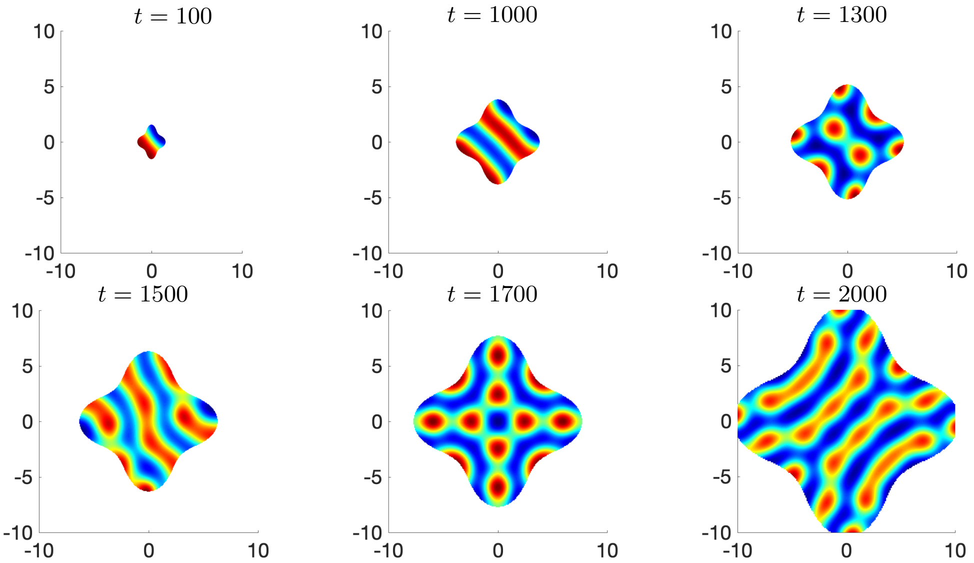

FIG. 3.7. *In Example 5, in the star-shape domain, when use global refinement strategy with discrete sets in the domain as in Figure 3.8 (d) and the domain growth function as* $\rho(t) = \exp(0.001t)$*, patterns generated by* $D_u = 1,\ D_v = 10,\ a = 0.1,\ b = 0.9,\ \gamma = 10,\ m = 4$ *and* $\Delta t = 0.005$.

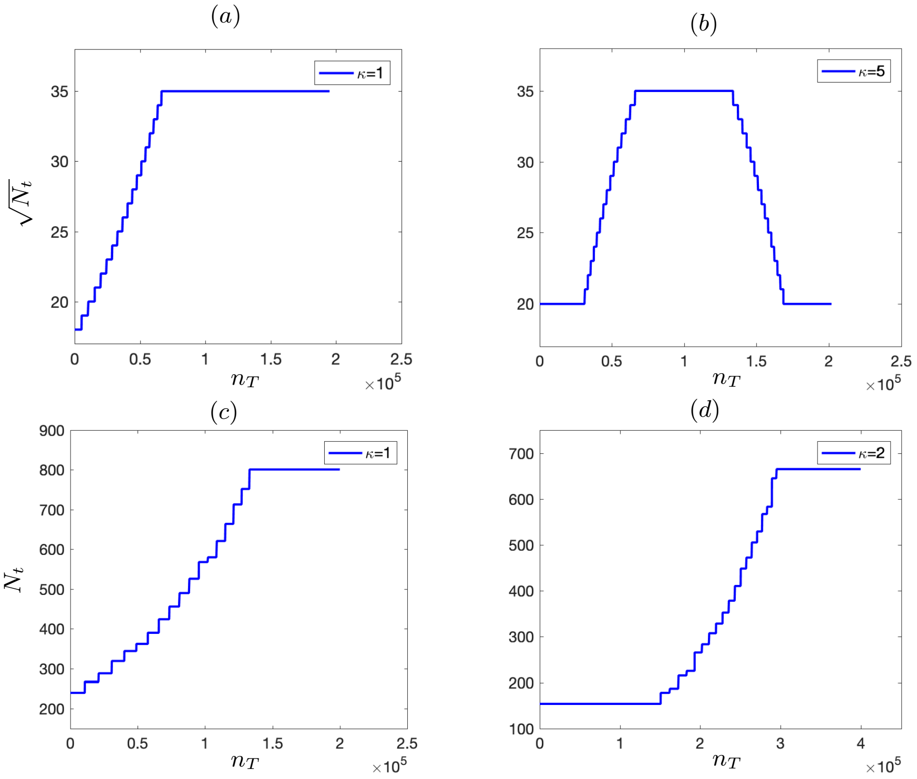

FIG. 3.8. *For the SH model , the change of trial centers* N_t *or* $\sqrt{N_t}$ *as number of time steps* n_T *increases under different global refinement settings for different examples: (a) for EX2, in the square domain the* $\nu = 1$ *in* (2.15)*; (b) for EX3, in the square domain,* $\nu = 5$ *in* (2.15)*; (c) for EX4, in the hexagon domain,* $\nu = 1$ *in* (2.15)*; (d) for EX5, in the star-shape domain,* $\nu = 2$ *in* (2.15)

the mode merging and splitting behavior as domain sizes decrease and increase. This phenomenon is consistent with patterns appear in nature, like patterns on fishes. Different isotropic growth functions are considered in the paper. For growth functions which are not isotropic and spatially linear, in the equivalent RDS on the reference domain (Eq. (2.5)), besides the Laplacian $\Delta\tilde{u}$, the mixed partial derivative term $\tilde{u}_{\xi\eta}$ and first order derivative terms $\tilde{u}_{\xi}, \tilde{u}_{\eta}$ will also appear. These terms can also be approximated by our meshless method. We will consider these problems in our future work. Further work will also apply the method to problems in chemotaxis systems.

Acknowledgments. We would like to thank the reviewers for their valuable comments which helped to improve the manuscript significantly. This work is supported in part by a China Postdoctoral Science Foundation 2019M661059 (S. Li) and Hong Kong Research Council GRF grants 15300417 and 15302919 (Z. Qiao).

REFERENCES

[1] L. A González, J. C Vanegas, and D. A Garzón, *Formación de patrones en sistemas de reacción-difusión en dominios crecientes*, Revista Internacional de Métodos Numéricos, 25 (2009), pp. 145–161.

[2] Amal Alphonse, Charles M Elliott, and Björn Stinner, *An abstract framework for parabolic pdes on evolving spaces*, arXiv preprint arXiv:1403.4500, (2014).

[3] Stefano Bonaccorsi and Giuseppina Guatteri, *A variational approach to evolution problems with variable domains*, Journal of Differential Equations, 175 (2001), pp. 51–70.

[4] K. C. Cheung, L. Ling, and R. Schaback, *h^2-convergence of least-squares kernel collocation methods*, SIAM Journal on Numerical Analysis, 56 (2018), pp. 614–633.

[5] E. J. Crampin, Eamonn A. G., and Philip K. M., *Reaction and diffusion on growing domains: Scenarios for robust pattern formation*, Bulletin of Mathematical Biology, 61 (1999), pp. 1093–1120.

[6] E. J. Crampin, W.W. Hackborn, and P.K. Maini, *Pattern formation in reaction-diffusion models with nonuniform domain growth*, Bulletin of Mathematical Biology, 64 (2002), pp. 747–769.

[7] M. Dehghan, M. Abbaszadeh, and A. Mohebbi, *A meshless technique based on the local radial basis functions collocation method for solving parabolic–parabolic patlak–keller–segel chemotaxis model*, Engineering Analysis with Boundary Elements, 56 (2015), pp. 129–144.

[8] Dominik Edelmann, *Finite element analysis for a diffusion equation on a harmonically evolving domain*, arXiv preprint arXiv:2009.11105, (2020).

[9] R. I. Fernandes, B. Bialecki, and G. Fairweather, *An ADI extrapolated crank–nicolson orthogonal spline collocation method for nonlinear reaction–diffusion systems on evolving domains*, Journal of Computational Physics, 299 (2015), pp. 561–580.

[10] P. Gray and S. K. Scott, *Sustained oscillations and other exotic patterns of behavior in isothermal reactions*, Journal of Physical Chemistry, 89 (1985), pp. 22–32.

[11] Y. C. Hon and R. Schaback, *On unsymmetric collocation by radial basis functions*, Applied Mathematics and Computation, 119 (2001), pp. 177–186.

[12] G. Hu, Z. Qiao, and T. Tang, *Moving finite element simulations for reaction-diffusion systems*, Advances in Applied Mathematics & Mechanics, 4 (2012).

[13] E. J. Kansa, *Multiquadricsa scattered data approximation scheme with applications to computational fluid-dynamicsi surface approximations and partial derivative estimates*, Computers & Mathematics with Applications, 19 (1990), pp. 127–145.

[14] ———, *Multiquadricsa scattered data approximation scheme with applications to computational fluid-dynamicsii solutions to parabolic, hyperbolic and elliptic partial differential equations*, Computers & Mathematics with Applications, 19 (1990), pp. 147–161.

[15] Peter E Kloeden, José Real, and Chunyou Sun, *Pullback attractors for a semilinear heat equation on time-varying domains*, Journal of Differential Equations, 246 (2009), pp. 4702–4730.

[16] S. Kondo and R. Asai, *A reaction–diffusion wave on the skin of the marine angelfish pomacanthus*, Nature, 376 (1995), pp. 765–768.

[17] S. Kondo and T. Miura, *Reaction-diffusion model as a framework for understanding biological pattern formation*, science, 329 (2010), pp. 1616–1620.

[18] O. Lakkis, A. Madzvamuse, and C. Venkataraman, *Implicit–explicit timestepping with finite*

element approximation of reaction–diffusion systems on evolving domains, SIAM Journal on Numerical Analysis, 51 (2013), pp. 2309–2330.

[19] S. Li and L. Ling, *Weighted least-squares collocation methods for elliptic pdes with mixed boundary conditions*, Engineering Analysis with Boundary Elements, 105 (2019), pp. 146–154.

[20] W. Li, K. Rubasinghe, G. Yao, and L. Kuo, *The modified localized method of approximated particular solutions for linear and nonlinear convection-diffusion-reaction pdes*, Advances in Applied Mathematics and Mechanics, 12 (2020), pp. 1113–1136.

[21] L. Ling, R. Opfer, and R. Schaback, *Results on meshless collocation techniques*, Engineering Analysis with Boundary Elements, 30 (2006), pp. 247–253.

[22] S. Liu and X. Liu, *Krylov implicit integration factor method for a class of stiff reaction-diffusion systems with moving boundaries*, Discrete & Continuous Dynamical Systems - B, 1 (2020), pp. 141–159.

[23] A. Madzvamuse, P. K Maini, and A. J. Wathen, *A moving grid finite element method for the simulation of pattern generation by turing models on growing domains*, Journal of Scientific Computing, 24 (2005), pp. 247–262.

[24] A. Madzvamuse, H. S. Ndakwo, and R. Barreira, *Stability analysis of reaction-diffusion models on evolving domains: the effects of cross-diffusion*, Discrete and Continuous Dynamical Systems-Series A, 36 (2016), pp. 2133–2170.

[25] A. Madzvamuse, A. J. Wathen, and P. K. Maini, *A moving grid finite element method applied to a model biological pattern generator*, Journal of computational physics, 190 (2003), pp. 478–500.

[26] J. D. Murray, *Mathematical biology, vol. 19 of biomathematics*, 1989.

[27] Z. Qiao, *Numerical investigations of the dynamical behaviors and instabilities for the gierer-meinhardt system*, Communications in Computational Physics, 3 (2008), pp. 406–426.

[28] Y. Qiu, W. Chen, and Q Nie, *A hybrid method for stiff reactiondiffusion equations*, Discrete & Continuous Dynamical Systems - B, 12 (2019), pp. 6387–6417.

[29] S. J. Ruuth, *Implicit-explicit methods for reaction-diffusion problems in pattern formation*, Journal of Mathematical Biology, 34 (1995), pp. 148–176.

[30] J Schnakenberg, *Simple chemical reaction systems with limit cycle behaviour*, Journal of theoretical biology, 81 (1979), pp. 389–400.

[31] C. Venkataraman, O. Lakkis, and A. Madzvamuse, *Global existence for semilinear reaction–diffusion systems on evolving domains*, Journal of mathematical biology, 64 (2012), pp. 41–67.

[32] Z. Y. Xing and L. P. Wen, *The fast implementation of the adi-cn method for a class of two dimensional riesz space fractional diffusion equations*, Advances in Applied Mathematics and Mechanics, 4 (2019), pp. 942–956.