

A self-organized mesh generator using pattern formation in a reaction-diffusion system

Hirofumi Notsu^{a,*}, Daishin Ueyama^b, Masahiro Yamaguchi^{c,d}

^aWaseda Institute for Advanced Study, Waseda University, 3-4-1, Okubo, Shinjuku-ku, Tokyo 169-8555, Japan

^bMeiji Institute for Advanced Study of Mathematical Sciences (MIMS), Meiji University, 1-1-1, Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan

^cGraduate School of Advanced Mathematical Sciences, Meiji University, 1-1-1, Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan

^dResearch Fellow of the Japan Society for the Promotion of Science

Abstract

A new type of mesh generator is developed by using a self-organized pattern in a reaction-diffusion system. The system is the Gray-Scott model, which creates a spot pattern in a specific parameter region. The spots correspond to nodes of a mesh. The mesh generator has several advantages: the algorithm is simple and processes to improve the mesh, such as smoothing, (locally) addition, and removal of nodes, are automatically performed by the system.

Keywords: Mesh generation, Self-organizing pattern formation, Reaction-diffusion system, The Gray-Scott model, The finite difference method

2000 MSC: 65L50, 35K57, 65L12

1. Introduction

The reaction-diffusion system is one of the well studied mathematical models and includes the so-called self-replication/self-organization mechanism. According to Turing, the combination of diffusion and reaction has the potential to create a pattern in a self-organized manner [1]. Many researchers believe that the mechanism of pattern formation in a reaction-diffusion system mimics that in nature [2, 3]. Indeed, Turing patterns have been experimentally confirmed in chemical systems [4, 5, 6, 7], apart from being observed on the skin of a marine angelfish [8]. The development of a mathematical theory for the reaction-diffusion system in 50 years is stunning. Moreover, many applications have been developed in the real world [9, 3, 8].

Here, by using the self-organizing pattern formation mechanism, we propose a new type of application of the reaction-diffusion system, i.e., a self-organized mesh generator. The proposed mesh generator is a triangular-mesh generator suitable for numerical simulations, especially for the finite element method (FEM). The strategy used in the mesh generator is quite different from previous ones, quadtree/octree [10], Delaunay [11, 12, 13], advancing front [14], bubble [15], and so on. In practical numerical simulations, it is important to obtain a good mesh for a specific domain shape. It is also necessary to optimize the mesh size according to the domain shape and the properties of the solutions. Our strategy is based on the self-organized pattern appearing in a reaction-diffusion system, the Gray-Scott model (GS model) [16, 17, 18, 19], that produces a spot pattern in a suitable parameter region. The spot pattern is obtained from appropriate initial data, and subsequently, the spots fill the domain automatically. We first introduce the concept of the self-organized mesh generator based on the GS model. We then demonstrate the advantages and usefulness of our strategy.

*Corresponding author. Tel./fax: +81 3 52862144

Email address: h.notsu@aoni.waseda.jp (Hirofumi Notsu)

2. Algorithm for a self-organized mesh generator

2.1. The Gray-Scott model

The GS model represents the two reactions $U + 2V \rightarrow 3V$ and $V \rightarrow P$, where U and V are the reacting chemicals and P is an inert product. The reaction-diffusion system, in normalized units, can be written as

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla \cdot (D_u \nabla u) - uv^2 + F(1 - u), \\ \frac{\partial v}{\partial t} = \nabla \cdot (D_v \nabla v) + uv^2 - (F + k)v, \end{cases} \quad (\text{GS})$$

where $u = u(x, t)$ and $v = v(x, t)$ are the concentrations of the chemicals U and V , respectively, as a function of position x and time t ; D_u and D_v are the diffusion coefficients of the chemicals, respectively, and F and k are positive parameters. Details of the GS model can be found in references [16, 17, 18, 19]. In a suitable parameter region, the GS model produces a two-dimensional spot pattern. The features of the obtained spot pattern are as follows. (i) The domain is filled by spots. (ii) For fixed D_u and D_v , the distances between neighbouring spots are almost equal. (iii) The spot pattern exactly fits the domain shape. It is important to note that these features are *automatically* realized when the GS model is used.

We use the spot pattern appearing in the GS model for mesh generation. As expected, the spots correspond to the nodes (nodal points) of a triangular mesh. Although D_u and D_v in (GS) are constants in the whole domain, we change the constants into spatially dependent functions, i.e., $D_u = D_u(x)$ and $D_v = D_v(x)$, to control the local mesh size. We note that no spot pattern appears in the region $\{x; D_u(x) = D_v(x) = 0\}$. Finally, a mesh is generated from a set of nodes by using the Delaunay criterion [20].

2.2. Procedure

Let Ω be a given domain and Γ be the boundary of Ω . For performing computations in Ω , we need a mesh $\mathcal{T}_h = \{K_l\}_{l=1}^{n_e}$, where K_l is the so-called element, i.e., a two-dimensional triangle or a three-dimensional tetrahedron, n_e is the total number of elements, and h is the representative mesh size. Let n_p be the total number of nodes of \mathcal{T}_h and \mathcal{N}_h ($\#\mathcal{N}_h = n_p$) be the set of the nodes. The procedure for the construction of the mesh generator consists of the following five steps and is explained in Figure 1. The five snapshots correspond to the five steps involved in the mesh generation for a complex domain near Tokyo Bay.

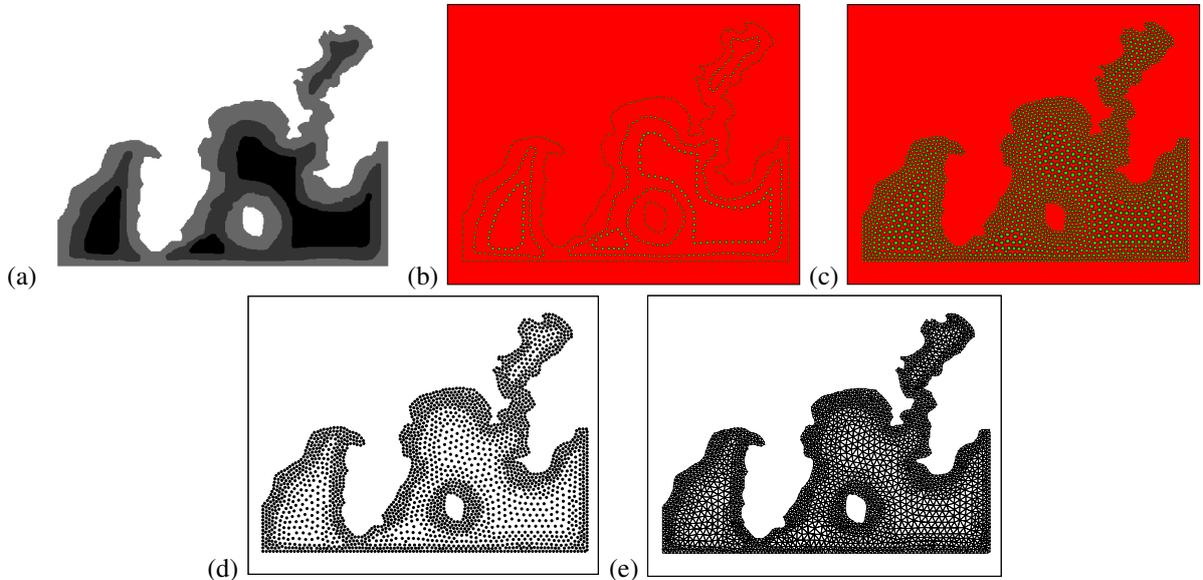


Figure 1: Snapshots of the different steps (steps 1–5) in the self-organized mesh generation.

- Step 1: The domain Ω is given by users. In the grey-scale (bitmap) image shown in Figure 1(a), the non-white region is the domain. We set a fictitious rectangular domain $\Omega_0 (\supset \Omega)$ and consider heterogeneous diffusion coefficients D_u and $D_v : \Omega_0 \rightarrow \mathbf{R}$. The colours in the grey-scale image correspond to the values of D_v , which is zero and positive in the white and non-white regions, respectively.
- Step 2: An artificial spot pattern is set as the initial value in the GS model (see Figure 1(b)); the artificial spots are put on the interfaces between two different colours, including the boundary Γ , and the locations of the artificial spots are fixed.
- Step 3: The system (GS) is numerically solved in Ω_0 by the finite difference method (FDM). Subsequently, the artificial fixed spots on the interfaces produce daughter spots until the domain is filled (Figure 1(c)). The distances of the spots are decided on the basis of the values of the diffusion coefficients.
- Step 4: A set of nodes \mathcal{N}_h is obtained from the final spot pattern (Figure 1(d)).
- Step 5: A mesh \mathcal{T}_h is generated from \mathcal{N}_h by using the Delaunay criterion (Figure 1(e)).

The self-organized mesh generator described above has following advantages: (i) The procedure is quite simple. (ii) Nodes are produced almost automatically by solving the GS model. It does not need complex techniques and criteria, for example, techniques for smoothing, (local) addition, or removal of nodes; such operations are automatically realized.

2.3. Implementation

We describe the details of the implementation in each step of the procedure. First, we fix $F = 0.032$ and $k = 0.063$ and obtain a spot pattern for $D_u/D_v = 2$ in the GS model. Since our implementation is realized through (pixel) image-based computation, we introduce a discrete domain $\Omega = \{x_{i,j}; (i,j) \in \mathcal{S}\}$, where $x_{i,j} \equiv (i\delta x, j\delta x)$ for $i, j \in \mathbf{Z} \cup \{\mathbf{Z} + 1/2\}$, $\delta x \equiv 1$ is the computational mesh size for the FDM, and $\mathcal{S} \subset \mathbf{Z}^2$.

In step 1, we set a discrete rectangular domain $\Omega_0 \equiv \{x_{i,j}; (i,j) \in \mathcal{S}_0\}$ satisfying $\Omega \subset \Omega_0$ ($\mathcal{S} \subset \mathcal{S}_0$) with boundary $\Gamma_0 \equiv \{x_{i,j} \notin \Omega_0; \{x_{i,j \pm 1}, x_{i \pm 1, j}\} \cap \Omega_0 \neq \emptyset\}$. We assume that white colour is used for $\Omega_0 \setminus \Omega$ and that the colour in Ω is not white. Let $\tilde{\Omega}_0 \equiv \Omega_0 \cup \Gamma_0$. In the case of Figure 1, $\Omega_0 = \{1, \dots, 1000\} \times \{1, \dots, 800\} (= \mathcal{S}_0) \subset \mathbf{N}^2$ and $\Gamma_0 = (\{1, \dots, 1000\} \times \{0, 801\}) \cup (\{0, 1001\} \times \{1, \dots, 800\})$. We define spatially dependent diffusion coefficients D_u and $D_v : \tilde{\Omega}_0 \rightarrow \mathbf{R}$ by

$$(D_u, D_v)(x) \equiv (2, 1)d(x), \quad (1)$$

where $d : \tilde{\Omega}_0 \rightarrow \mathbf{R}$ is a function that takes positive values in Ω and zero in $\tilde{\Omega}_0 \setminus \Omega$. Then, the relation $D_u/D_v = 2$ holds in Ω and a spot pattern appears only in Ω . The domain Ω can be recognized from the function d . In Figure 1, $d = 1.5 \times 10^{-1}$, 9.6×10^{-2} , 5.4×10^{-2} , and 0 for the black, dark-grey, light-grey, and white ($\tilde{\Omega}_0 \setminus \Omega$) regions, respectively. When d is a constant in whole domain, the rough relation

$$h \approx c_* \sqrt{d} \quad (2)$$

with $c_* = 48$ holds by our simulations. We can therefore expect the distance between neighbouring spots in the black, dark-grey, and light-grey regions will be about 18.6, 14.9, and 11.2, respectively.

To solve the GS model numerically, we introduce an approximation $(U^n, V^n)(x_{i,j}) \in \mathbf{R}^2$ of $(u, v)(x_{i,j}, t^n)$ for $x_{i,j} \in \tilde{\Omega}_0$ and $n = 0, 1, \dots$; here, $t^n \equiv n\delta t$ and δt is a time increment. Let $V_* \equiv 0.6$. ‘An artificial spot pattern’ in step 2 is

$$U^0(x_{i,j}) = 1 \quad (x_{i,j} \in \tilde{\Omega}_0), \quad \text{and} \quad V^0(x_{i,j}) = \begin{cases} V_* & (x_{i,j} \in \mathcal{A}), \\ 0 & (x_{i,j} \in \tilde{\Omega}_0 \setminus \mathcal{A}), \end{cases} \quad (3)$$

where \mathcal{A} is a set of lattice points on the interfaces between two different colours. We note that the distance between neighbouring spots in \mathcal{A} does not have to be consistent exactly with $c_* \sqrt{d}$ in (2). Indeed, in Figure 1 there are three types of interfaces-between black and dark grey, between dark grey and light grey, and between light grey and white-, and the distance between neighbouring spots in \mathcal{A} at these interfaces is almost 18, 14, and 10, respectively.

In step 3, we use the following explicit finite difference scheme to compute the GS model: find $\{(U^n, V^n)(x_{i,j}); x_{i,j} \in \tilde{\Omega}_0, n = 0, 1, \dots\} \subset \mathbf{R}^2$ such that

$$\begin{cases} \frac{U^{n+1} - U^n}{\delta t}(x_{i,j}) = (L_{\delta x}(u)U^n)(x_{i,j}) - U^n(V^n)^2(x_{i,j}) + F(1 - U^n)(x_{i,j}) & (x_{i,j} \in \Omega_0), \\ \frac{V^{n+1} - V^n}{\delta t}(x_{i,j}) = (L_{\delta x}(v)V^n)(x_{i,j}) + U^n(V^n)^2(x_{i,j}) - (F + k)V^n(x_{i,j}) & (x_{i,j} \in \Omega_0 \setminus \mathcal{A}), \\ V^{n+1}(x_{i,j}) = V_* & (x_{i,j} \in \mathcal{A}), \end{cases} \quad (4)$$

for $n = 0, 1, \dots$, with the initial condition (3) (artificial spot pattern in step 2), and the zero Neumann boundary condition for Γ_0 ; $L_{\delta x}(a) : \{W : \tilde{\Omega}_0 \rightarrow \mathbf{R}\} \rightarrow \{W : \Omega_0 \rightarrow \mathbf{R}\}$ ($a = u, v$) is an operator defined by

$$(L_{\delta x}(a)W)(x_{i,j}) \equiv \sum_{k=1}^2 [\nabla_{\delta x,k} \{D_a(\nabla_{\delta x,k} W)\}](x_{i,j}) \quad (x_{i,j} \in \Omega_0),$$

with $(\nabla_{\delta x,k} W)(x_{i,j}) \equiv \{W(x_{i,j} + (\delta x/2)e_k) - W(x_{i,j} - (\delta x/2)e_k)\}/\delta x$ ($k = 1, 2$), $e_k \equiv (\delta_{k1}, \delta_{k2})$). Further, δ_{pq} ($p, q = 1, 2$) is the Kronecker delta and $D_a(x_{i,j} + (\delta x/2)e_k) \equiv \{D_a(x_{i,j} + \delta x e_k) + D_a(x_{i,j})\}/2$. We can write

$$[\nabla_{\delta x,1} \{D_u(\nabla_{\delta x,1} U^n)\}](x_{i,j}) = \frac{1}{\delta x^2} [D_u(x_{i+1/2,j})\{U^n(x_{i+1,j}) - U^n(x_{i,j})\} - D_u(x_{i-1/2,j})\{U^n(x_{i,j}) - U^n(x_{i-1,j})\}].$$

For sufficiently large n_0 , we obtain the numerical stationary solution $\{(U^{n_0}, V^{n_0})(x_{i,j})\}_{x_{i,j} \in \Omega_0}$. In our computation, we have used the condition

$$\|V^{n_0} - V^{n_0-1}\|_{l^2} / \|V^{n_0-1}\|_{l^2} < \varepsilon \equiv 10^{-5}$$

for every 5,000 time steps, where for $W : \Omega_0 \rightarrow \mathbf{R}$, $\|\cdot\|_{l^2}$ is a discrete L^2 -norm defined by $\|W\|_{l^2} \equiv \{\delta x^2 \sum_{x_{i,j} \in \Omega_0} W(x_{i,j})^2\}^{1/2}$. We note that a highly accurate solution is not required, i.e., a rough computation is enough if spots are created, and that the mesh size h can be controlled by varying d . Because of $L_{\delta x}(a)$ in scheme (4), δt needs to satisfy $\delta t \leq \delta x^2/4 \max\{\|D_u\|_{\infty}, \|D_v\|_{\infty}\}$ ($= \delta x^2/8\|d\|_{\infty}$ when relation (1) is used), where $\|\cdot\|_{\infty}$ is the maximum norm. In Figure 1, $\delta t = 2/3$.

In step 4, \mathcal{N}_h is created from the stationary solution $\{(U^{n_0}, V^{n_0})(x_{i,j})\}_{x_{i,j} \in \Omega_0}$ obtained in step 3, i.e., the local peaks of $\{V^{n_0}(x_{i,j})\}_{x_{i,j} \in \Omega_0}$ become nodes. In our computation, for $x_{i,j} \in \Omega_0$, if the condition

$$V^{n_0}(x_{i,j}) > \max\{V^{n_0}(x_{l,m}); x_{l,m} \in \Omega_0, x_{l,m} \neq x_{i,j}, l \in [i - c_{i,j}, i + c_{i,j}], m \in [j - c_{i,j}, j + c_{i,j}]\} \quad (C_{i,j})$$

is satisfied, we define the point $x_{i,j}$ as a node, where $c_{i,j} \equiv c_0 \sqrt{d(x_{i,j})}$, with $c_0 \equiv 20$. Here, we have set $c_{i,j}$ by using (2). We then obtain \mathcal{N}_h , which is defined by

$$\mathcal{N}_h \equiv \{x_{i,j} \in \Omega_0; x_{i,j} \text{ satisfies } (C_{i,j})\}.$$

The set of points \mathcal{A} used for the ‘artificial spot pattern’ in step 1 will be a subset of \mathcal{N}_h . This means that we have the flexibility to set the nodes at appropriate points.

In step 5, the Delaunay criterion gives a triangulation \mathcal{T}_h from \mathcal{N}_h , where the problem of non-convexity of the domain Ω in the triangulation is solved by using d .

3. Quality of meshes

We examine the quality of meshes generated by the self-organized mesh generator for two sample domains. The quality is confirmed by the distribution of the inner angles and lengths of meshes.

3.1. Example 1

The first sample domain is Figure 1(a) for the generated mesh shown in Figure 1(e), for which $n_p = 2,379$ and $n_e = 4,234$. Theoretical results for the FEM [21] imply that regular triangles are preferred. Figure 2 shows graphs of relative frequency versus angle (left) and length (right). We can see that there are many almost regular triangles and that there are three peaks of length around 10, 14, and 18. The three peaks correspond to the three positive values of d . These results imply that the self-organized mesh generator produces a mesh whose elements are almost regular and whole local mesh sizes are controlled by d .

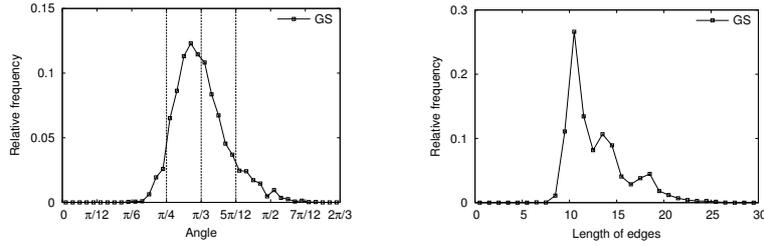


Figure 2: Graphs of relative frequency vs. angle (left) and length (right) for the first sample domain near Tokyo Bay.

3.2. Example 2

The other sample domain is shown in Figure 3 (leftmost figure); the boundary is complex, the mesh size is uniform, and $\Omega_0 \equiv (1, 250) \times (1, 350)$. In our mesh generator, $d = 0.1$ and 0 in the black and white regions, respectively, and $\delta t = 1$. For this example, we compare our mesh with that generated by FreeFem++ [22], which is one of the most famous free FEM softwares. The obtained meshes are shown in the centre (FreeFem++, $n_p = 231$, $n_e = 389$) and rightmost panels (ours, $n_p = 237$, $n_e = 401$) of Figure 3, where we have considered identical boundary nodes. Figure 4 shows plots of relative frequency versus angle (left) and length (right). To see the progress of the self-organized mesh generator, we show two graphs of $n = 5,000$ and $20,000$ in addition to the graphs of the FreeFem++ mesh and our mesh with $n = 55,000$ (final). The red and blue lines indicate a mesh generated by FreeFem++ and our (final) mesh, respectively. The green and magenta lines correspond to $n = 5,000$ and $20,000$ in step 3. We can observe that the distribution of the angles and lengths of our meshes improves as n increases, that the inner angles and lengths of our meshes are concentrated near $\pi/3$ and 15, respectively, and that the final mesh ($n = 55,000$) is better than the mesh generated by FreeFem++.

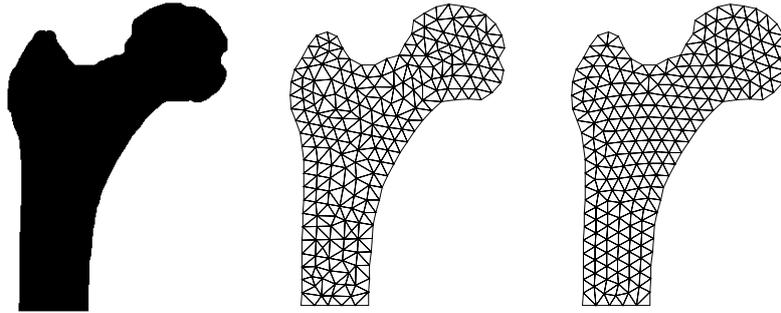


Figure 3: Second sample domain (left), a mesh generated by FreeFem++ (center), and our mesh (right).

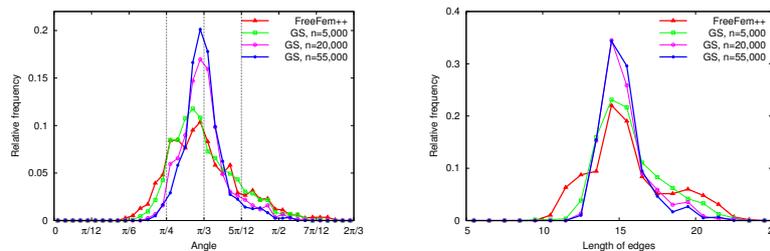


Figure 4: Graphs of relative frequency vs. angle (left) and length (right) for the second sample domain and comparison with the graphs for the FreeFem++ generated mesh.

In order to see the difference of finite element solutions by the two meshes of the example (i.e., the centre

(FreeFem++) and rightmost panels (ours) of Figure 3), we solve a Poisson problem with an inhomogeneous Dirichlet boundary condition, where the external force and the boundary value of the problem are given so that the exact solution is $\phi(x) \equiv \sin(x_1/100)\sin(x_2/100)$. Using the linear (P1) finite element in the FEM, we obtain two finite element solutions $\phi_{h_{FF}}$ and $\phi_{h_{GS}}$ from the mesh generated by FreeFem++ and our mesh, respectively. Let $\|\cdot\|_{H_0^1}$ be the norm of the Sobolev space $H_0^1(\Omega)$. The relative errors between the finite element and exact solutions are $\|\phi_{h_{FF}} - \phi\|_{H_0^1}/\|\phi\|_{H_0^1} = 5.37 \times 10^{-2}$ and $\|\phi_{h_{GS}} - \phi\|_{H_0^1}/\|\phi\|_{H_0^1} = 4.86 \times 10^{-2}$, which imply that $\phi_{h_{GS}}$ is almost 10 percent closer to ϕ than $\phi_{h_{FF}}$ in the sense of $H_0^1(\Omega)$ -norm.

The computational time for our meshes is much longer than that for the meshes generated by FreeFem++. The actual computational times for the second sample domain are almost 85 (ours, $n = 55,000$) and 0.01 s (FreeFem++) on a notebook computer (Intel Core i7 1.8GHz, 4GB memory), i.e., the computational times differ by a factor of about 8,500. This is the main disadvantage of our mesh generator, although the generated meshes are quite good. Step 3 accounts for the largest part of the computational time of our mesh generation (obtaining the solution of the GS model by using the FDM). Since the FDM works well in general-purpose computing on graphics processing units (GPGPU) and shows high performance, the disadvantage relating to the computational time might be considered tolerable.

In general, the computational time of our mesh generation is not so sensitive to the complexity of the domain shape and mainly depends on the size of Ω_0 . Indeed, for a star-shaped domain with the same $\Omega_0 = (0, 250) \times (0, 350)$ it is almost 90 s ($n = 60,000$), and for a twice larger (and similar) domain of Figure 3 (leftmost figure) with $\Omega_0 = (0, 500) \times (0, 700)$ it is almost 455 s ($n = 70,000$).

4. Conclusions

We have proposed a self-organized mesh generator based on a spot pattern of the GS model. It shows the features of a self-replicating system, i.e., it fills the domain with spots, fits the spot pattern to the domain shape, and maintains almost equal distances between neighbouring spots. We have discussed a (pixel) image-based implementation, which works well and is useful for practical digital images. The domain shape and the mesh size are controlled by d in (1), which gives spatially dependent diffusion coefficients D_u and D_v in the GS model. The mesh generator has a unique advantage: the fairly simple procedure does not require additional complex processes and criteria, such as smoothing, (local) addition, and removal of nodes. These additional processes are automatically performed by the non-linear system (GS), i.e., through the self-organization mechanism. Although only a two-dimensional mesh generator has been realized here, the concept of the generator is expected to be useful for the development of a three-dimensional mesh generator. There is a possibility that the generator can control the mesh orientation, if D_u and D_v in (GS) are modified as matrix valued diffusion coefficients and a modified (Delaunay) triangulation technique is employed.

Acknowledgments

This study was supported by the Meiji University Global COE Program ‘Formation and Development of Mathematical Sciences Based on Modeling and Analysis’ and the Meiji University Mathematical Modeling and Analysis Projects for Young Researchers (Development of a Mesh Generator Using a Self-Organizing Mechanism of a Reaction-Diffusion System). The first author was supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan, through the Special Coordination Funds for Promoting Science and Technology (the Tenure Track Program at Waseda Institute for Advanced Study), by the Japan Society for the Promotion of Science (JSPS) through the Grant-in-Aid for Young Scientists (B) (No. 23740090) and the Japanese-German Graduate Externship (Mathematical Fluid Dynamics), and by Waseda University through the Grant for Special Research Projects (No. 2011A-959).

References

- [1] A. M. Turing, The chemical basis of morphogenesis, *Phil. Trans. R. Soc. London B* 237 (1952) 37–72.
- [2] H. Meinhardt, *Models of Biological Pattern Formation*, Academic Press, London, 1982.
- [3] J. D. Murray, *Mathematical Biology I, II*, 3rd ed., Springer, New York, 2002.
- [4] K. J. Lee, H. L. Swinney, Lamellar structures and self-replicating spots in a reaction-diffusion system, *Phys. Rev. E* 51 (1995) 1899–1915.
- [5] K. J. Lee, W. D. McCormick, Q. Ouyang, H. L. Swinney, Pattern formation by interacting chemical fronts, *Science* 261 (1993) 192–194.

- [6] P. De Kepper, V. Castets, E. Dulos, J. Boissonade, Turing-type chemical patterns in the chlorite-iodide-malonic acid reaction, *Physica D* 49 (1991) 161–169.
- [7] P. De Kepper, J.-J. Perraud, B. Rudovics, E. Dulos, Experimental study of stationary Turing patterns and their interaction with traveling waves in a chemical system, *Int. J. Bifurcat. Chaos* 4 (1994) 1215–1231.
- [8] S. Kondo, R. Asai, A reaction-diffusion wave on the skin of the marine angelfish *Pomacanthus*, *Nature* 376 (1995) 765–768.
- [9] A. Adamatzky, B. de Lacy Costello, T. Asai, *Reaction-Diffusion Computers*, Elsevier, Amsterdam, 2005.
- [10] M. A. Yerry, M. S. Shephard, A modified quadtree approach to finite element mesh generation, *IEEE Comput. Graph.* 3 (1983) 39–46.
- [11] P. J. Green, R. Sibson, Computing Dirichlet tessellations in the plane, *Comput. J.* 21 (1978) 168–173.
- [12] S. W. Sloan, A fast algorithm for constructing Delaunay triangulations in the plane, *Adv. Eng. Software* 9 (1987) 34–55.
- [13] T. Taniguchi, Automatic mesh generation for FEM - Application of Delaunay triangulation, Morikita, 1992, (in Japanese).
- [14] R. Löhner, Progress in grid generation via the advancing front technique, *Eng. Comput.* 12 (1996) 186–210.
- [15] K. Shimada, D. Gossard, Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing, *Proc. third ACM symp.* (1995) 409–419.
- [16] P. Gray, S. K. Scott, Autocatalytic reactions in the isothermal continuous stirred tank reactor, *Chem. Eng. Sci.* 38 (1983) 29–43.
- [17] J. E. Pearson, Complex patterns in a simple system, *Science* 261 (1993) 189–192.
- [18] Y. Nishiura, *Far-from-Equilibrium Dynamics*, AMS, 2002.
- [19] Y. Nishiura, D. Ueyama, A skeleton structure of self-replicating dynamics, *Physica D* 130 (1999) 73–104.
- [20] B. Delaunay, Sur la sphère vide, *Bull. Acad. Sci. USSR* 6 (1934) 793–800.
- [21] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, SIAM, Philadelphia, 2002.
- [22] FreeFem++, <http://www.freefem.org/>.