12th International Conference on Domain Decomposition Methods
Editors: Tony Chan, Takashi Kako, Hideo Kawarada, Olivier Pironneau,

39. Numerical computation for some competition-diffusion systems on a parallel computer

R. Ikota[1], M. Mimura[2], T. Nakaki[3]

Introduction

In theoretical biology, spatial segregation of biological species has been investigated by many scientists (see [DHMP99], [IMY98] and the references therein). Among several models explaining such a phenomenon, we deal with the systems of competition-diffusion type.

We consider n kinds of species U_i $(1 \leq i \leq n)$. Let $u_i(x,t)$ be the population density of the species U_i $(1 \leq i \leq n)$ at time $t > 0$ and the position $x \in \Omega$, where Ω is a bounded domain in $\mathbf{R}^N$. Then our model can be described by

$$\frac{\partial u_i}{\partial t} = d_i \Delta u_i + (r_i - \sum_{j=1}^{n} a_{ij} u_j) u_i \qquad (i = 1, 2, \ldots, n), \quad x \in \Omega, \ \ t > 0, \tag{1}$$

where d_i is the diffusion rate, r_i the intrinsic growth rate, a_{ii} the intraspecific competition rate, and a_{ij} $(i \neq j)$ the interspecific competition rate between U_i and U_j. We assume that all these parameters are nonnegative and impose initial and Neumann boundary conditions on (1):

$$u_i(x,0) = u_{i0}(x) \qquad (i = 1, 2, \ldots, n), \quad x \in \Omega, \tag{2}$$

$$\frac{\partial u_i}{\partial \nu} = 0 \qquad (i = 1, 2, \ldots, n), \quad x \in \partial\Omega, \quad t > 0, \tag{3}$$

where ν is the unit outer normal to $\partial\Omega$, and u_{i0} is a nonnegative function.

Specifically we are interested in the case where the competition is extremely strict. In order to treat such situations we rewrite the equations (1) and obtain the following:

$$\frac{\partial u_i}{\partial t} = d_i \Delta u_i + (r_i - a_{ii} u_i) u_i - k \sum_{\substack{j=1 \\ j \neq i}}^{n} b_{ij} u_i u_j \qquad (i = 1, 2, \ldots, n) \quad x \in \Omega, \ \ t > 0. \tag{4}$$

The parameter k represents the magnitude of interspecific competition. We study (4) when k is very large. As $k \to \infty$, we can observe in our numerical computations that the region Ω is divided into each region Ω_i which only a single species U_i occupies. One of our interests is to analyze the behavior of interfaces between $\{\Omega_i\}$. If we use typical numerical methods, we have some difficulties to track the interfaces. That is

[1]University of Tokyo, ikota@ms.u-tokyo.ac.jp
[2]Hiroshima University, mimura@math.sci.hiroshima-u.ac.jp
[3]Kyushu University, nakaki@math.kyushu-u.ac.jp

because they appear in the limit case where $k \to \infty$. In fact, for a fixed large value of k, we can easily perform numerical computations to (4), however, we have no criterion to determine the numerical interfaces by using numerical solutions to u_i.

In this paper we propose a method to analyze (4) when $k \to \infty$, by which we can track the interfaces. Our method is described as follows:

A-Method

Step 1: For $u_i(\cdot, t)$, solve the following PDE:

$$\begin{aligned} \frac{\partial \bar{u}_i}{\partial \tau} &= d_i \Delta \bar{u}_i + (r_i - a_{ii}\bar{u}_i)\bar{u}_i \qquad \text{in } \Omega, \quad 0 < \tau < \Delta t, \\ \frac{\partial \bar{u}_i}{\partial \nu} &= 0 \qquad \text{on } \partial\Omega, \quad 0 < \tau < \Delta t, \\ \bar{u}_i(x,0) &= u_i(x,t) \qquad \text{in } \Omega, \end{aligned}$$

where $\Delta t > 0$ is a given constant $(1 \le i \le n)$.

Step 2: Solve the following ODE until $\tau = \infty$, that is, compute the equilibrium points for $1 \le i \le n$:

$$\begin{aligned} \frac{d\check{u}_i(x,\tau)}{d\tau} &= -\sum_{\substack{j=1 \\ j \ne i}}^{n} b_{ij}\check{u}_i\check{u}_j \qquad \text{in } \Omega, \ \ 0 < \tau < \infty, \\ \check{u}_i(x,0) &= \bar{u}_i(x,\Delta t) \qquad \text{in } \Omega. \end{aligned}$$

Step 3: Put $u_i(x, t+\Delta t) = \check{u}_i(x,\infty)$ $(1 \le i \le n)$.

This method has the advantage that we can determine the interfaces naturally as shown in Fig. 5 without complicated procedure even in the multi-component $(n \ge 2)$ and multi-dimensional $(N \ge 2)$ cases.

The aim of this paper is as follows: We show the mathematical justification of A-Method when $n = 2$ and $d_1 = d_2$ in the second section. The condition $d_1 = d_2$ is imposed by the mathematical reason. We also propose a parallel algorithm to A-Method in the third section. We describe the algorithm and perform numerical simulations for the typical case $n = 3$. When $n \ne 3$, we can similarly treat the problem.

Mathematical justification for the two-component case

Known Results

We consider the following two-component (that is, $n = 2$) system:

$$u_t = d_1\Delta u + f(u)u - kuv \quad \text{in } Q = \Omega \times \mathbf{R}^+, \tag{5}$$

$$v_t = d_2\Delta v + g(v)v - \alpha kuv \quad \text{in } Q = \Omega \times \mathbf{R}^+, \tag{6}$$

$$\frac{\partial u}{\partial \nu} = 0, \quad \frac{\partial v}{\partial \nu} = 0 \quad \text{on } S = \partial\Omega \times \mathbf{R}^+, \tag{7}$$

$$u(x,0) = u_0^k(x), \quad v(x,0) = v_0^k(x) \quad \text{for} \quad x \in \Omega, \tag{8}$$

where $f(u) = r_1 - a_{11}u$ and $g(v) = r_2 - a_{22}v$.

Let $(u^{(k)}, v^{(k)})$ be a solution to (5)–(8) and put $w^{(k)} = u^{(k)} - v^{(k)}/\alpha$. If $u_0^{(k)}$ and $v_0^{(k)}$ converge to u_0 and v_0 respectively, then by Proposition 2.1 in [DHMP99], $w^{(k)}$ converges to a weak solution w of the following problem as $k \to \infty$:

$$w_t = \nabla(d(w)\nabla w) + h(w) \quad \text{in} \quad Q, \tag{9}$$

$$\frac{\partial w}{\partial \nu} = 0 \quad \text{on} \quad S, \tag{10}$$

$$w(x,0) = w_0(x) \equiv u_0(x) - \frac{v_0(x)}{\alpha} \quad \text{for} \quad x \in \Omega, \tag{11}$$

where

$$d(s) = \begin{cases} d_1 & \text{if} \quad s > 0, \\ d_2 & \text{if} \quad s < 0, \end{cases}$$

$$h(s) = \begin{cases} f(s)s & \text{if} \quad s > 0, \\ g(-\alpha s)s & \text{if} \quad s < 0. \end{cases}$$

Under certain conditions, by putting $u = [w]^+$ and $v = \alpha[w]^-$, we observe that the above problem (9)–(11) is equivalent to the following problem (see [DHMP99]):

$$u_t = d_1\Delta u + f(u)u \quad \text{in} \quad Q^{\text{int}}, \tag{12}$$

$$v_t = d_2\Delta v + g(v)v \quad \text{in} \quad Q^{\text{ext}}, \tag{13}$$

$$u = 0 \quad \text{and} \quad v = 0 \quad \text{on} \quad \Gamma, \tag{14}$$

$$d_1\frac{\partial u}{\partial n} = -\frac{d_2}{\alpha}\frac{\partial v}{\partial n} \quad \text{on} \quad \Gamma, \tag{15}$$

$$\frac{\partial v}{\partial n} = 0 \quad \text{on} \quad \partial\Omega \times (0,T], \tag{16}$$

$$u(x,0) = u_0(x), \quad v(x,0) = v_0(x) \quad \text{for} \quad x \in \Omega, \tag{17}$$

where

$$Q^{\text{int}} = \{(x,t) \in \mathbf{R} \times (0,T];\, u(x,t) > 0 \text{ and } v(x,t) = 0\},$$

$$Q^{\text{ext}} = \{(x,t) \in \mathbf{R} \times (0,T];\, u(x,t) = 0 \text{ and } v(x,t) > 0\}.$$

Definition of the Approximation and Results

In this subsection, we show a mathematical justification that A-Method gives an approximation to our problem. In Step 1 of A-Method we solve the following systems:

$$(P_u)\begin{cases} u_t = d_1\Delta u + f(u)u & \text{in} \quad Q, \\ \frac{\partial u}{\partial \nu} = 0 & \text{on} \quad S, \\ u(x,0) = u_0(x) \in C(\bar{\Omega}) & \text{for} \quad x \in \Omega, \end{cases}$$

$$(P_v)\begin{cases} v_t = d_2\Delta v + g(v)v & \text{in} \quad Q, \\ \frac{\partial v}{\partial \nu} = 0 & \text{on} \quad S, \\ v(x,0) = v_0(x) \in C(\bar{\Omega}) & \text{for} \quad x \in \Omega. \end{cases}$$

We denote the solutions to (P_u) and (P_v) by $\mathcal{H}^u(t)u_0$ and $\mathcal{H}^v(t)v_0$, respectively. In Step 2, we solve the following ordinary differential equations:

$$\begin{cases} \dfrac{du}{dt} = -uv, \\ \dfrac{dv}{dt} = -\alpha uv, \\ u(0) = u_0, \quad v(0) = v_0. \end{cases}$$

Recalling

$$\frac{d}{dt}(u - \frac{v}{\alpha}) = 0,$$

then we obtain

$$\lim_{t\to\infty}(u(t), v(t)) = ([u_0 - \frac{v_0}{\alpha}]^+, \alpha[u_0 - \frac{v_0}{\alpha}]^-). \tag{18}$$

Let us define an operator $\mathcal{K}(t)$ parameterized with non-negative number t by

$$\mathcal{K}(t)z_0 \equiv \mathcal{H}^u(t)[z_0]^+ - \frac{1}{\alpha}\mathcal{H}^v(t)(\alpha[z_0]^-). \tag{19}$$

Then we can describe the approximated solution constructed by A-Method as

$$\mathcal{K}(T/n)^n w_0. \tag{20}$$

If $d_1 = d_2$, under certain conditions imposed on w_0 we have proven

$$\begin{aligned} \|\mathcal{K}(T/n)^n w_0 - w(T)\|_{L^2(\Omega)} &\leq C_1(T/n)^{1/2}, \\ \|\mathcal{K}(T/n)^n w_0 - w(T)\|_{L^1(\Omega)} &\leq C_2(T/n). \end{aligned}$$

These inequalities implies that the numerical solutions of A-Method converges as $\Delta t \to 0$. Unfortunately at present we can not prove the convergence when $d_1 \neq d_2$. However our numerical computations suggest that the solution also converges.

Parallel computations for the three-component case

Algorithm

Our algorithm here is shown when $n = 3$. For $n \neq 3$, it is quite easy to extend our algorithm. We describe our algorithm for a computer with three CPUs which are called CPU1, CPU2 and CPU3. To CPUi we assign three arrays, say Array$i$-u, Array$i$-v and Array$i$-w ($i = 1, 2, 3$).

The first step (Fig. 1): First of all, we put the data u, v and w into Array1-u, Array2-v and Array3-w, respectively

The second step (Fig. 1): Then we solve

$$\begin{array}{lll} u_t = d_1\Delta u + (r_1 - a_{11}u)u & \text{on} & \text{Array1-u using CPU1,} \\ v_t = d_2\Delta v + (r_2 - a_{22}v)v & \text{on} & \text{Array2-v using CPU2,} \\ w_t = d_3\Delta w + (r_3 - a_{33}w)w & \text{on} & \text{Array3-w using CPU3.} \end{array}$$

The third step (Fig. 2): We copy Array1-u into Array2-u and Array3-u, Array2-v into Array1-v and Array3-v, Array3-w into Array1-w and Array2-w.

The fourth step (Fig. 3): We compute the ODE system. We separate the region into three parts. We assign each part to CPUi ($i = 1, 2, 3$) respectively.

The fifth step (Fig. 4): Gather data u into Array1-u, v into Array2-v and w into Array2-w.

We note that the second and fourth steps stated above correspond to Steps 1 and 2 of A-Method, respectively.

Numerical experiments

Let us demonstrate our numerical simulations when the region Ω is the two dimensional interval $(0, 1)^2$. We use the workstation Sun Enterprise 450 (4 CPUs, Total memory 2GB). The programs are written in Sun Fortran 77 (Option: `-fast -O5`) and MPI [GLS94].

Numerical parameters we use are 256×256 space mesh and $\Delta t = 0.001$. Computations are halted if one of three species u, v or w becomes extinct.

We obtain the following table which shows the CPU times of the single and parallel computations. We have used 3 CPUs and obtained about 2.3 times speed-up. In our experience, the parallel performance goes up when the nodal points near the interfaces are assigned equally to each CPU.

CPU times

case	Single	Parallel	ratio
a	769sec.	328sec.	2.34
b	2544sec.	1104sec.	2.30
c	2562sec.	1112sec.	2.30
d	2951sec.	1242sec.	2.38
e	3967sec.	1742sec.	2.28

On this table, we remark the following:

- *Single* in the table means the computation using a usual code without MPI.
- *Parallel* means that the computation by our algorithm with 3 CPUs.
- We vary the initial function and parameters $\{r_i\}$ and $\{a_{ij}\}$ in *cases* (a)–(e).

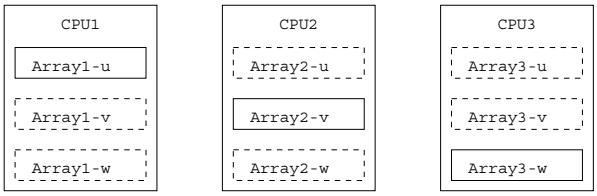

Figure 1: The first and second steps. The data u, v and w are stored in Array1-u, Array2-v and Array3-w, respectively. Then solve the PDE on each CPUs.

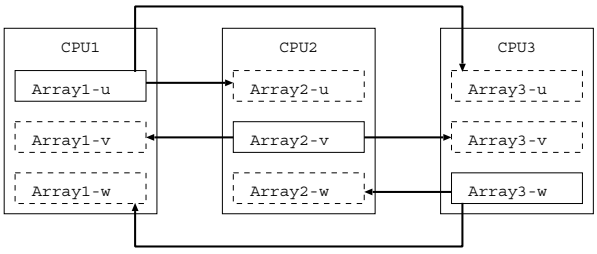

Figure 2: The third step. Message passing between CPUs.

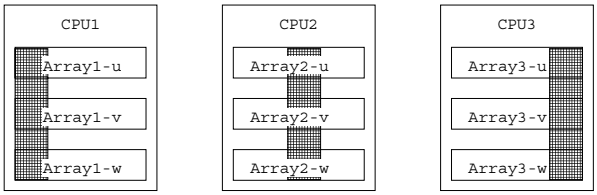

Figure 3: The fourth step. Solve the ODE on each CPUs.

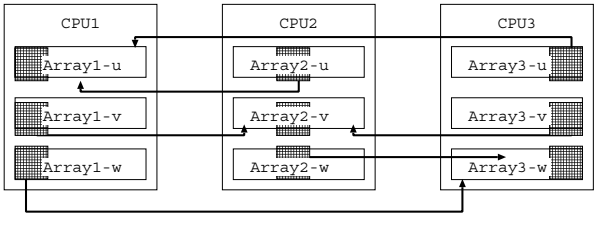

Figure 4: The fifth step. Gather the data.

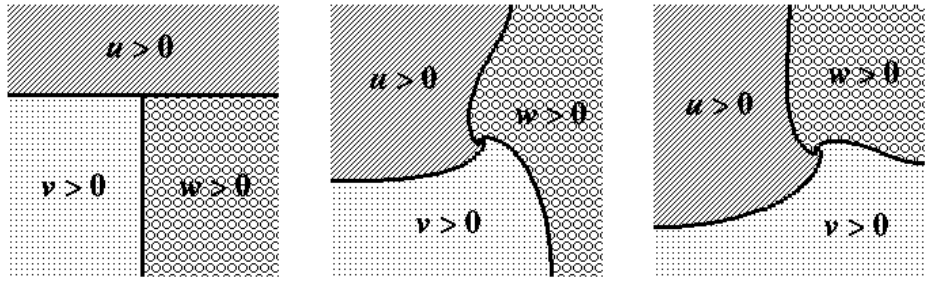

Figure 5: Numerical solutions by the present method for the three-component case in two dimensional space $(0,1)^2$. The solutions are drawn at $t = 0$ (left), $t = 0.5$ (center) and $t = 1$ (right). We can clearly observe the interfaces between regions $\{\Omega_i\}$.

Concluding remarks

A problem in mathematical biology is considered. The method, which we propose in this paper, has the advantages that we can determine the interfaces naturally and clearly as shown in Fig. 5 and that an implementation to the parallel computer can be easily done. We obtained 2.3 times speed-up by using 3 CPUs.

For the two-component case, we justified the method rigorously when $d_1 = d_2$. We can expect that the condition $d_1 = d_2$ is not essential.

Acknowledgments.

We would like to thank Mr. Atsushi Suzuki of Kyushu University for setting us a good computational environment.

References

[DHMP99]E. N. Dancer, D. Hilhorst, M. Mimura, and L. A. Peletier. Spatial segregation limit of a competition-diffusion system. *European J. Appl. Math.*, 10(2):97–115, 1999.

[Eva80]L. C. Evans. A convergence theorem for a chemical diffusion-reaction system. *Houston J. Math.*, 6(2):259–267, 1980.

[GLS94]W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. MIT press, 1994.

[IMY98]M. Iida, M. Mimura, and E. Yanagida. A free boundary problem as a singular limit of a competition-diffusion system. In Y. Nishiura, I. Takagi, and E. Yanagida, editors, *Proceedings of the International Conference on Asymptotics in Nonlinear Diffusive Systems*, pages 217–221, Sendai, 1998. Tohoku Univ.

[MBO94]B. Merriman, J. K. Bence, and S. J. Osher. Motion of multiple junctions: a level set approach. *J. Comput. Phys.*, 112:334–363, 1994.