Spectral Methods for Reaction–Diffusion Systems

V. Rybář

Institute of Mathematics of the Academy of Sciences of the Czech Republic.

Abstract. Although spectral methods proved to be numerical methods that can significantly speed up the computation of solutions of systems of reaction–diffusion equations, finite difference and finite element methods still prevail as the most widespread methods. This contribution offers comparison of the performance of the Fourier spectral method with finite element method for reaction–diffusion system modeling the generation of pigment patterns on the coat of the leopard.

Introduction

Over the last few decades, systems of reaction–diffusion equations have been used in effort to explain and model various phenomena in the field of biology and chemistry. Two species spatial ecological models (e.g. predator–prey models), chemical reaction models and tumor growth can be mentioned as examples. Furthermore, after 1952 when Turing published his now famous and widely cited paper [*Turing*, 1952], it has been put a lot of endeavor into explaining the generation of patters in mammals, fish, gasteropod and many more. Classical book [*Murray*, 2003] offers description and analysis of many models based on reaction–diffusion systems.

General reaction–diffusion system is a system of parabolic partial differential equations and it can be expressed in the following form:

$$\begin{aligned}\frac{\partial u}{\partial t} &= D_1 \Delta u + N_u(u,v) \text{ in } (0,\infty)\times\Omega,\\ \frac{\partial v}{\partial t} &= D_2 \Delta v + N_v(u,v) \text{ in } (0,\infty)\times\Omega,\end{aligned}$$

where $u = u(t,x)$, $v = v(t,x)$, $\Omega \subset \mathbb{R}^2$ is a domain, D_1, D_2 are diffusion coefficients and $N_u(u,v)$, $N_v(u,v)$ are nonlinear reaction terms. Such system is often coupled with zero flux boundary condition,

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

where n stands for the unit outward normal vector to the boundary $\partial\Omega$, as it has in many cases good biological or chemical meaning and random noise as initial condition.

In already mentioned paper [*Turing*, 1952] it was pointed out that if u and v tend to a linearly stable uniform steady state in case $D_1 = D_2 = 0$, then this state can, under certain conditions, become unstable for $D_1 \neq 0, D_2 \neq 0$, and spatial inhomogeneous patterns can evolve. So called *diffusion driven instability* can emerge. Since diffusion term is generally considered to have stabilizing rather then destabilizing effect, Turing's idea may be considered surprising.

Model for generating pigment patterns of leopard and jaguar

Liu, Liaw and Maini present in *Liu et al.* [2006] a simple reaction–diffusion model based on *Barrio et al.* [1999]:

$$\frac{\partial u}{\partial t} = D\delta\Delta u + \alpha u + v - r_2 uv - \alpha r_3 uv^2, \tag{1}$$

$$\frac{\partial v}{\partial t} = \delta\Delta v - \alpha u + \beta v + r_2 uv + \alpha r_3 uv^2. \tag{2}$$

It represents interaction between two morphogens u and v on the square with zero flux boundary condition. With all the coefficients properly set for the particular development stage, this system of equations can model generation of pigment patterns on the coat of leopards and jaguars. Figure 1 shows compiled results presented in *Liu et al.* [2006] in the case of the leopard.

It should be noted that this kind of models is not fully accepted among biologists. There is not clear proof that these patterns emerge as a result of reaction diffusion process of Turing type, i.e. whether at all they exist and if so, what the morphogens u and v are. Moreover, the diffusion coefficient D in (1) needs to be smaller than one to allow the diffusion driven instability to occur. This raises an objection why two chemicals of presumably similar kind should have different physical properties (i.e. the rate of diffusion).

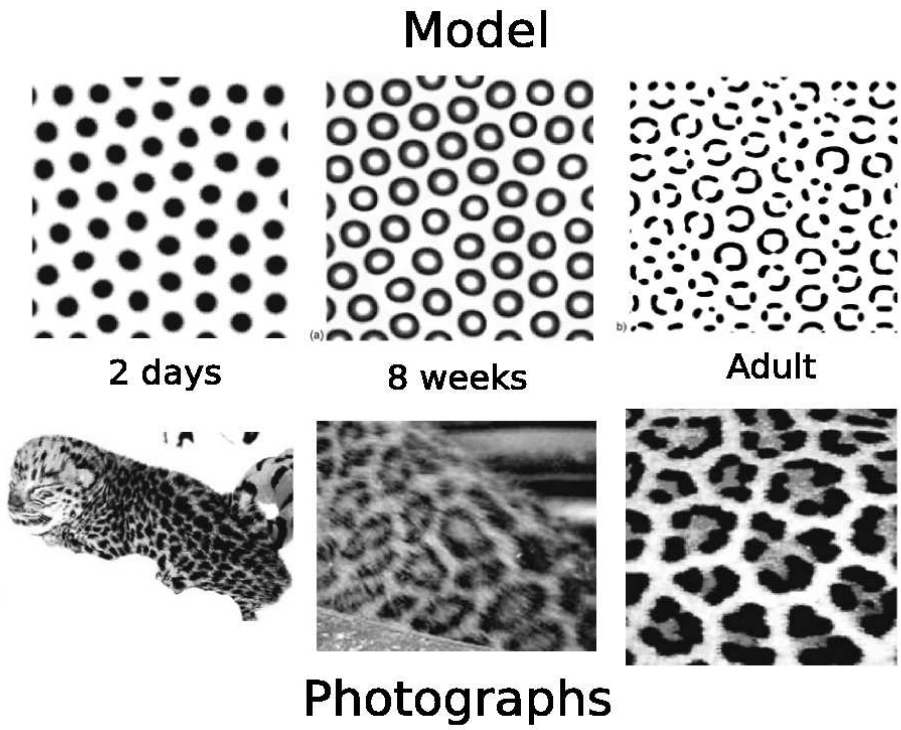


Figure 1. Model and photographs of a coat at various stages of development of a leopard.

Spectral methods

As a suitable first step in our survey of spectral methods we chose the Fourier spectral method for the spatial discretization. This method is considered as one of the fastest [see *Fu et al.*, 2012] among spectral methods and still easy to implement.

To be able to use this method to solve system (1)–(2) we need to replace the zero-flux boundary condition by the periodic one. This replacement is admissible in our experiments, because the steady state solution of the system with zero flux boundary condition is for our purposes qualitatively the same as the system with periodic boundary condition. Both boundary problems in our particular set-up result in the same kind of pattern (spots) that *Liu et al.* [2006] consider as a model of early stage of pattern development of a leopard.

To incorporate zero flux boundary condition, Fourier basis functions need to be replaced by the Chebyshev ones. That is expected to slow down the computations. This problem is addressed in several papers [e.g., *Fu et al.*, 2012], where the authors offer a way how to preserve most of the advantages of the Fourier spectral method while solving system with Neumann boundary condition.

To introduce the Fourier spectral method, we will follow *Trefethen* [2000]. Let us consider a function v sampled on the spatial discretization grid $\{x_1, \ldots, x_N\}$ with $v_j = v(x_j)$. Let v be periodic, i.e. $v_1 = v_N$. The discrete Fourier transform (DFT),$\hat{v}_k = \hat{v}(k)$, is defined by

$$\hat{v}_k = \sum_{j=1}^{N} e^{-\mathrm{i}kx_j} v_j, \quad k = -\frac{N}{2} + 1, \ldots, \frac{N}{2}, \tag{3}$$

where i is the complex unit. The inverse discrete Fourier transform (IDFT) is given by

$$v_j = \frac{1}{2\pi} \sum_{k=-N/2+1}^{N/2} e^{\mathrm{i}kx_j} \hat{v}_k, \quad j = 1, \ldots, N. \tag{4}$$

Using these definitions, we can compute the derivative $w_j = \frac{\mathrm{d}v}{\mathrm{d}x}(x_j),\ j = 1, \ldots, N$ at the nodes of the discretization grid by the following procedure:

1. given $v_j,\ j = 1, \ldots, N$ compute its DFT $\hat{v}_k,\ k = -N/2 + 1, \ldots, N/2$ using (3),
2. define $\hat{w}_k = ik\hat{v}_k,\ k = -N/2 + 1, \ldots, N/2$,
3. compute $w_j,\ j = 1, \ldots, N$ from $\hat{w}_k,\ k = -N/2 + 1, \ldots, N/2$ by (4).

Applying this procedure two times yields second derivatives. Thus the diffusion term in

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + N(u) \tag{5}$$

can be transformed into $-Dk^2\hat{u}_k$ and partial differential equation (5) transforms to a system of ordinary differential equations (ODE)

$$\frac{\mathrm{d}\hat{u}_k}{\mathrm{d}t} = -Dk^2\hat{u}_k + \widehat{N(u)}, \quad k = -N/2 + 1, \ldots, N/2 \tag{6}$$

where $u = (u_1, \ldots, u_N)$ is computed by (4) from $\hat{u}_k$, $k = -N/2 + 1, \ldots, N/2$.

Furthermore, by this transformation system (6) becomes non-stiff. Thus, fast explicit ODE solver can be used. That is a significant advantage compared to the stiff system (5) where one can not usually avoid implicit solver.

Kassam [2003] suggests that Cox and Mathew's fourth order exponential time differencing Runge-Kutta method (ETDRK4) might by particularly suitable for such problems.

Since the nonlinear term $N(u)$ is evaluated in the space domain and then transformed into the Fourier one, steps to deal with aliasing need to be taken [see *Trefethen*, 2000].

The procedure of computation of the derivatives by DFT can be extended to two dimensions by taking advantage of the fact that the two dimensional DFT is separable into sequence of one dimensional transforms. Details can be found in *Kopriva* [2000].

Numerical experiments

Fine-tuning of parameters in systems similar to (1)–(2) is needed to get desired patterns. Consequently,a large number of numerical experiments is required. *Kassam* [2003] shows that Fourier spectral method coupled with ETDRK4 solver can be significantly faster than finite differences with implicit ODE solver.

We used model (1)–(2) from *Liu et al.* [2006] forming spots in the first stage of the pattern development on the coat of leopard as our test case for comparing the performance of finite element and Fourier spectral method. In the first stage, the coefficient in (1)–(2) were set as follows: $D = 0.45,\ \delta = 6,\ \alpha = 0.899,\ \beta = -0.91,\ r_2 = 2$, and $r_3 = 3.5$.

The finite element code we used was a Matlab implementation of P1-conforming finite element on a triangular grid. The built-in Matlab variable time-stepping ODE solver `ode15s` was used for the integration and the code is written in order to exploit fast vector manipulations in Matlab to assembly mass and stiffness matrices.

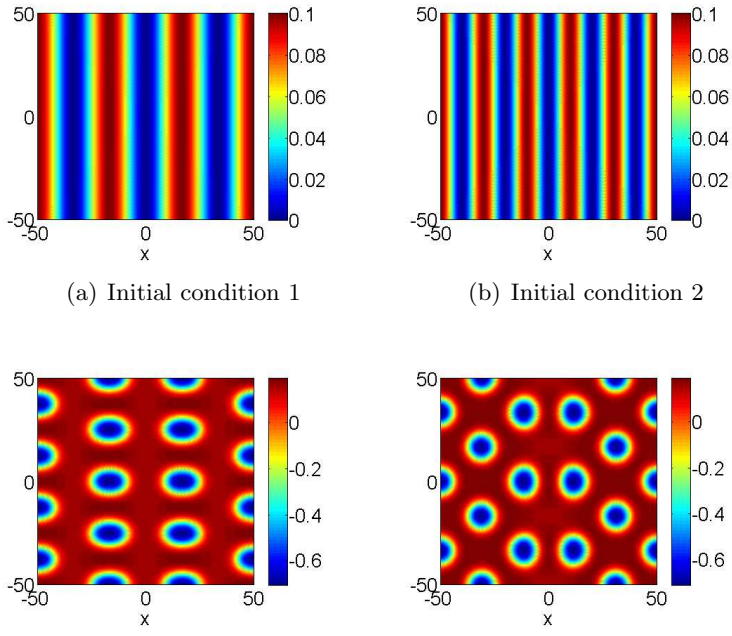
As demonstrated in Figure 2, the steady-state solution strongly depends on the initial condition, therefore we do not require accurate solutions and we are interested mainly in the speed of calculations. This approach is motivated by the nature of biological systems where minor disruptions and oscillations are omnipresent, e.g. no two jaguar coats are alike.

The following experiments were performed in domain $\Omega = (0, 100)^2$ and the computation was terminated at $T_{\text{final}} = 1000$ as the system was close to its steady state at this point. Random noise in $(-0.05, 0.05)$ served as an initial condition. The computational times for the finite element and spectral methods and for various numbers of degrees of freedom are presented in Table 1.

To assess the resolution of the finite element and Fourier spectral method we present Figure 3 that corresponds to $6.6 \cdot 10^3$ degrees of freedom.

Conclusion

The above results proved that the Fourier spectral method is significantly faster compared to finite element method in solving system (1)–(2). For the sake of simplicity we present results only for the first phase of the coat development which is depicted in the left part of Figure 1. The alternation in coefficients for the following phases does not bring any other qualitative



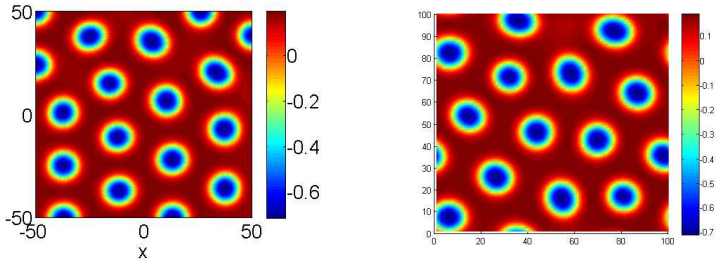
(a) Initial condition 1 (b) Initial condition 2

(c) Solution 1 (d) Solution 2

Figure 2. Dependency of the steady-state solution on the initial condition.

Table 1. The computational times for the finite element and spectral method.

# DOF	Finite element	Spectral method
$1.6 \cdot 10^3$	45 s	15 s
$6.6 \cdot 10^3$	295 s	70 s
$2.6 \cdot 10^5$	1421 s	195 s



(a) FEM (b) Spectral Method

Figure 3. Comparison of finite element solution of reaction–diffusion system (1)–(2) with zero flux b.c. and Fourier spectral method solution with periodic b.c.

difference to the system (1)–(2) than different shape of the resulting pattern [*Liu et al.*, 2006]. Thus we believe that the Fourier spectral method will still remain substantially faster than the finite element method for other phases as well.

The spectral code relies on FFTW, the state of the art package providing Fast Fourier Transform to compute DFT and IDFT [see *Frigo*, 1999, for reference]. That allows us to have the the most computationally expensive operations implemented effectively. Using advanced techniques addressing the bottlenecks of the finite element implementations, e.g. assembly of mass and stiffness matrices as presented by *Rahman et al.* [2013], would likely improve the results of finite element code.

Although the Fourier spectral method is easier to implement, it lacks the universality of finite element methods. Especially the periodic boundary condition may be too limiting for some systems. Other members of the spectral method family, although less easy to implement and more time consuming, can deal with these limitations [e.g., *Fu et al.*, 2012], and seems to have a potential in solving reaction–diffusion systems of mathematical biology.

References

R. A. Barrio, C. Varea, J. L. Aragon, and P. K. Maini, A two-dimensional numerical study of spatial pattern formation in interacting Turing systems, Bull. Math. Biol. 61, 483, 1999.

M. Frigo, A Fast Fourier Transform Compiler, the Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '99), 1999.

H. Fu, and C. Liu, A buffered Fourier spectral method for non periodic ODE, International Journal of Numerical Analysis and Modelling, 9(2), 460, 2012.

A. K. Kassam, Solving reaction–diffusion equations 10 times faster, Oxford University,Numerical Analysis Group Research Report No. 16, 2003.

D. A. Kopriva, Implementing Spectral Methods for Partial Differential Equations, Springer, 2009.

Liu, R. T., S. S. Liaw, and P. K. Maini, Two-stage Turing model for generating pigment patterns on the leopard and the jaguar, Physical Review E 74(1), 011914, 2006.

J.D. Murray, Mathematical Biology II: Spatial Models and Biomedical Applications, Springer-Verlag, 2003.

T. Rahman and J. Valdman, Fast MATLAB assembly of FEM matrices in 2D and 3D: nodal elements, Applied Mathematics and Computation 219, 7151-7158, 2013.

L. N. Trefethen, Spectral methods in MATLAB, Society for Industrial and Applied Mathematics, 2000.

A. M. Turing, The chemical basis of morphogenesis, Phil. Trans. R. Soc. Lond., B237, 37, 1952.