Turing patterns in a self-replicating mechanism with a self-complementary template

Leo L. Tsai, Geoffrey R. Hutchison, and Enrique Peacock-López[a)]
Department of Chemistry, Williams College, Williamstown, Massachusetts 01267

(Received 13 March 2000; accepted 1 May 2000)

A variety of nonlinear chemical models, such as the Selkov–Schnakenberg, exhibit Turing patterns. The Templator, which is based on a minimal autocatalytic monomer–dimer system, is a newer two-variable model also able to show Turing patterns. Here we find that the dynamic behavior of the Templator is quite similar to other models with cubic nonlinearities. This is demonstrated through a series of computer simulations in two dimensions utilizing the cellular automata approach. The selection of parameter values is based on linear stability analysis, which provides a relatively simple method of predicting Turing pattern formation. The simulations reveal spot, labyrinth, and striped patterns, in agreement with the predictions of the analysis. Other behaviors, such as honeycomb patterns, are also observed. For some parameter values, we study transient spot replication. Our findings strongly suggest that the Templator may belong to the same class of models previously studied by Pearson. © *2000 American Institute of Physics.* [S0021-9606(00)70229-7]

I. INTRODUCTION

In recent years, several theoretical models have been introduced to provide insight into the development of spatial patterns in nature. Since the work of Turing in the 1950s,[1] theoretical work regarding spatial patterns has led to a better understanding of how these patterns arise from certain mechanisms.[2,3] Pattern generation is seen everywhere, from stripes on zebras[4] to chemical reactions.[5] Current explorations of pattern formation range from cell differentiation in developing embryos[6] to the macroscopic behavior of ameba clusters.

Improvements in technology have expanded the freedom to study the dynamic behavior of models subjected to different parameter values. Increases in computational speed have aided the simulation of models of partial differential equations (PDE's); computational methods such as the cellular automata have also facilitated the exploration of such models. Also mathematical analyses, from Turing's linear stability method to more elaborate schemes,[7–10] have facilitated parameter selection for the simulations.

Turing patterns, which are temporally stable, have been studied extensively for a variety of mechanisms;[11–13] generalized experimental methods for creating chemical systems which exhibit such behavior now exist.[14] Linear stability analysis[15] provides a simple and effective method of predicting the behavior of a stable system subjected to a small perturbation. In chemical models, which describe the formation of patterns, such perturbations arise from differences in diffusion rates of the reactants. This class of ''reaction-diffusion'' (RD) systems is particularly prominent in the study of enzyme kinetics, as evidences by the large number of well-studied models which exist today.

The Templator mechanism proposed by Peacock-López *et al.*,[16] is a relatively new model based on experiments involving autocatalytic biochemical reactions.[17] A description of this mechanism is discussed in Sec. II. Confirmation of Turing patterns, in the form of stripes, was achieved through use of computer simulations utilizing the cellular automata approach; this is discussed in Sec. III. The results of our simulations are presented in Sec. IV and the conclusion in Sec. V.

II. THE TEMPLATOR MODEL

The Templator is a minimal model based on autocatalytic reactions observed in self-replicating molecules.[17–27] Self-replicating molecular systems have been synthesized in the laboratory by Rebek *et al.*[17–22] One of Rebek's self-replicating system is represented schematically below,

$$R+S \xrightarrow{k_{\mathrm{uncat}}} T, \qquad (1)$$

$$R+S+T \xrightarrow{k_{\mathrm{templ}}} T+T, \qquad (2)$$

where R can stand for adenine ribose, (AR), diaminotriazine xanthene, (DIX), adenine ribose-Z, (ZAR), adenine ribose-Z-N_2, (ZNAR), where Z is a blocking group like benzyloxycarbonyl. The other molecule, T, can be naphthalene imide, (NI), biphenyl imide, (BI), thymine, (T). We notice that these molecules are self-complementary and when bound covalently form a product that can work as a template for the formation of more product except for DIXBI, which cannot self-replicate.

In formulating the Templator, Peacock-López *et al.*,[16] have used a simplified monomer–dimer system, whereby the dimer serves as a template.

The mechanism for the Templator is as follows:

$$A_0 \xrightarrow{k_0} A, \qquad (3)$$

[a)] Author to whom correspondence should be addressed. Electronic mail: epeacock@tonatiuh.williams.edu

$$A+A\xrightarrow{k_1}B, \tag{4}$$

$$2A+B\xrightarrow{k_2}2B, \tag{5}$$

$$B\xrightarrow{E}P. \tag{6}$$

The monomer, A, is continuously pumped from a source and combines with itself to form a dimer, B. This dimer then catalyzes its own production. Continual removal via some enzymatic path is also assumed.

By using mass actions laws and by rescaling parameters of the system to produce dimensionless quantities,[16] the Templator is reduced to a set of differential equations describing the change in concentration of the monomer and the dimer,

$$\dot{a}=r_0-2a^2b-2k_ua^2, \tag{7a}$$

$$\dot{b}=a^2b+k_ua^2b-\frac{b}{(K+b)}, \tag{7b}$$

where a and b refer to the monomer and dimer, respectively, r_0 represents the rate of input of the monomer, k_u refers to the rate of uncatalyzed formation of the dimer, and the last term refers to an enzymatic removal of the dimer. This last term is necessary for the existence of stable limit cycles.[16]

III. COMPUTER SIMULATIONS

Several methods exist for solving reaction-diffusion problems. Cellular automata (CA) simplifies calculations by using discrete time, space, and state values, which are easily processed by a computer. Several CA methods for solving reaction-diffusion systems rely on the qualitative behavior of the underlying partial differential equations.[28,30] A class of automata introduced by Weimer and Boon uses careful discretization of values to minimize errors and provides very fast quantitative solutions for RD systems.[29–31] The use of a lookup table for the reaction term and fixed-point arithmetic increases calculating speeds significantly.

A reaction-diffusion system is essentially a partial differential equation,

$$\frac{\partial \mathbf{n}(\mathbf{r},t)}{\partial t}=D\nabla^2\mathbf{n}(\mathbf{r},t)+f(\mathbf{n}(\mathbf{r},t)), \tag{8}$$

where $\mathbf{r}$ gives the spatial location and $f(\mathbf{n})$ is the rate law of the reaction system in question. In general, with multispecies reactions, the value $\mathbf{n}$ is a concentration vector for each species. The diffusion operator is easily approximated by many techniques, including a very efficient method using square masks.[28] The reaction operator, $f(\mathbf{n})$, is generally a nonlinear function. These equations can be solved by any method, but, in our experience, a first-order approximation is simple and most rapid.

While the operator for diffusion does not need to consider discretization, by nature the nonlinear reaction operator must provide for some sort of operator $\Phi_T(\Delta t f(x))$ to truncate the reaction term. One solution which preserves accuracy *on average* uses a probabilistic rule. This introduces noise into the system, but this can be minimized by increasing the accuracy of the fixed-point discretization in time or space.[31] It is relatively simple to create fixed-point arithmetic and modular grids for CA by using the programming language Ada. The states are discretized by fixing the range of values and the precision of the implementation.

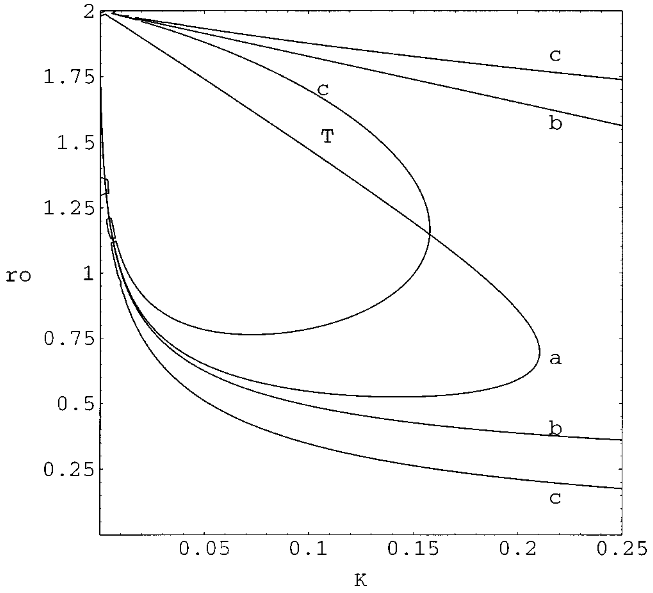


FIG. 1. Diagram in parameter space. Using linear stability analysis, we obtained a set of three conditions required for stable Turing patterns. Lines a, b, and c depict the null sets of these conditions. Region T represent the linear analysis prediction of parameter values able to generate Turing patterns.

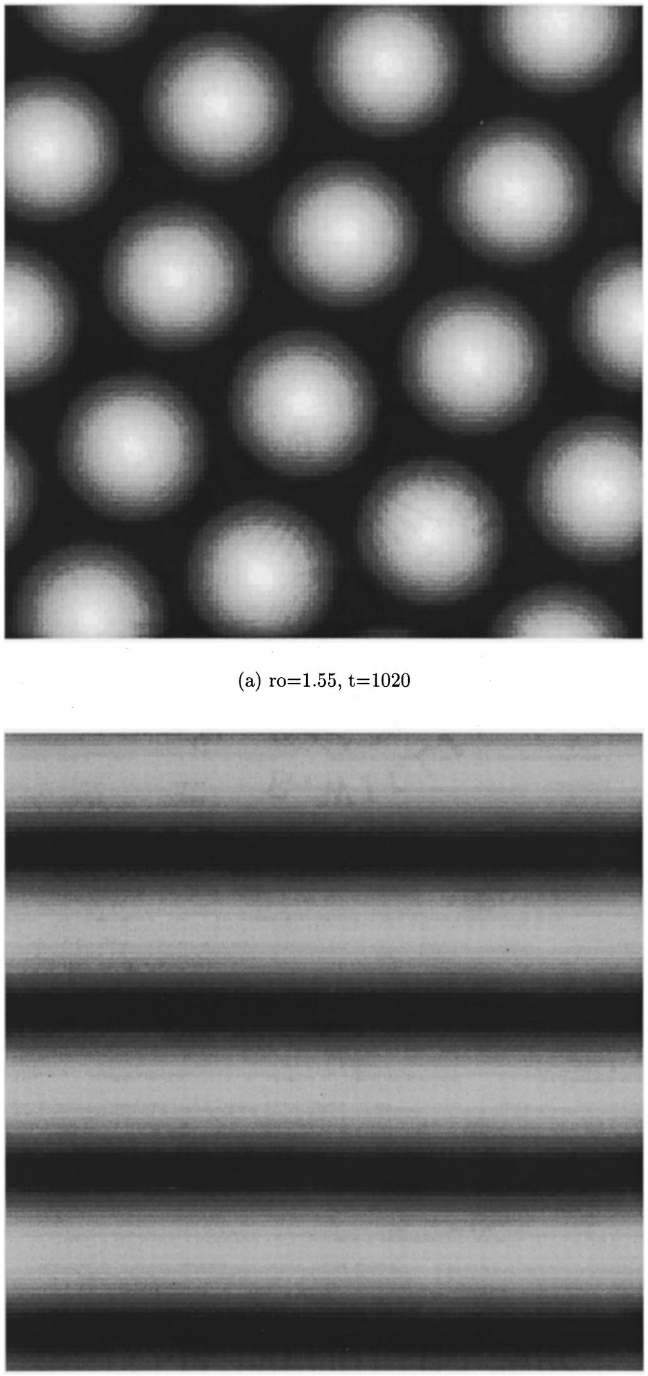
The range of concentration values for Eq. (8) is set at approximately the same concentration range observed for the reaction ODE's. This requires that the diffusion operator remains within the bounds of this range. Normally the bounds are made much larger than necessary and capped, preventing the system from diverging while running a simulation. However, the limits are usually unnecessary for most parameters.

For simplicity, the operators are also applied independently using higher precision calculation, then combined while applying the probabilistic rounding. This produces a solution to the partial differential equation consistent with the necessary discretization of time, space, and state needed in CA. The approach differs only slightly from finite-difference methods adding the restriction of discrete states.

IV. TURING PATTERNS

Simulations were run on the Templator model fixing $k_u=0.01$ and using r_0 and K values within the Turing, T, zone depicted in Fig. 1. The results of the simulations agree with the prediction that only the lowest modes are exhibited in this region.

As the pumping of the monomer (r_0) is decreased the system approaches the limit of spatial stability. Near this border, the model passes through a transient striped stage before breaking into spots with a hexagonal symmetry, H_0. As r_0 is further decreased, the spot patterns become higher-ordered. In Fig. 2 we compare two stable Turing patterns for $K=0.10$ and different values of r_0. The level of shading represents the concentration of the dimer, with lighter shades



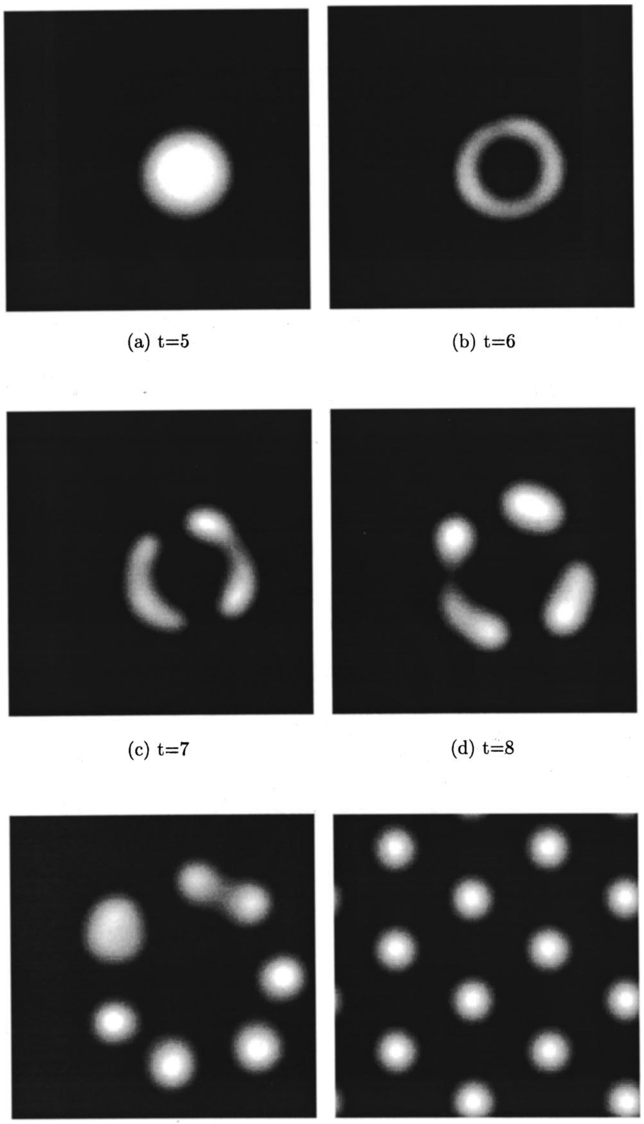
(a) ro=1.55, t=1020

(b) ro=1.60, t=289

FIG. 2. Transition from spots to stripes as we vary r_0 for homogeneous initial condition and $K=0.10$.

corresponding to higher concentrations. Periodic boundary conditions were implemented in all simulations.

The Templator shows a similar series of stable H_0 patterns under different values of r_0 and $K=0.05$. The evolution of the model, however, is different from simulations seen in Fig. 2. Here the simulation initially evolves with the spontaneous formation of a spot, followed by the growth of the spot, the formation of a ring from the spot, and the fragmentation of the ring into four spots. These spots then multiply until a stable hexagonal pattern H_0 is formed. Figure 3 provides an example of this behavior, using the homogeneous initial condition. Evidence of this evolution may suggest the capacity of the Templator to exhibit self-replicating spots.[32]



(a) t=5

(b) t=6

(c) t=7

(d) t=8

(e) t=9

(f) t=1320

FIG. 3. Time evolution of a spontaneous generated spot from a homogeneous initial condition in a 128×128 grid for $K=0.05$ and $r_0=0.65$.

Figures 4 show the initial evolutionary stages of the Templator, with $K=0.05$ and $r_0=0.65$. This simulation is the same as the one seen in Fig. 3, except for a larger grid (256×256) and the use of a central square for the initial condition. Here the model also breaks into four spots, but without the transient ring as in Fig. 3. These spots then undergo multiple replications before settling to a hexagonal H_0 pattern as in Fig. 2. In Fig. 4(c) we notice spontaneous appearance of spots similar to those in Fig. 3.

Finally for the Templator, we have observed the evolution of other patterns at high values of K. For example an inverted spot, or ''honeycomb'' pattern, H_π, is observed for $K=0.25$ and depicted in Fig. 5.

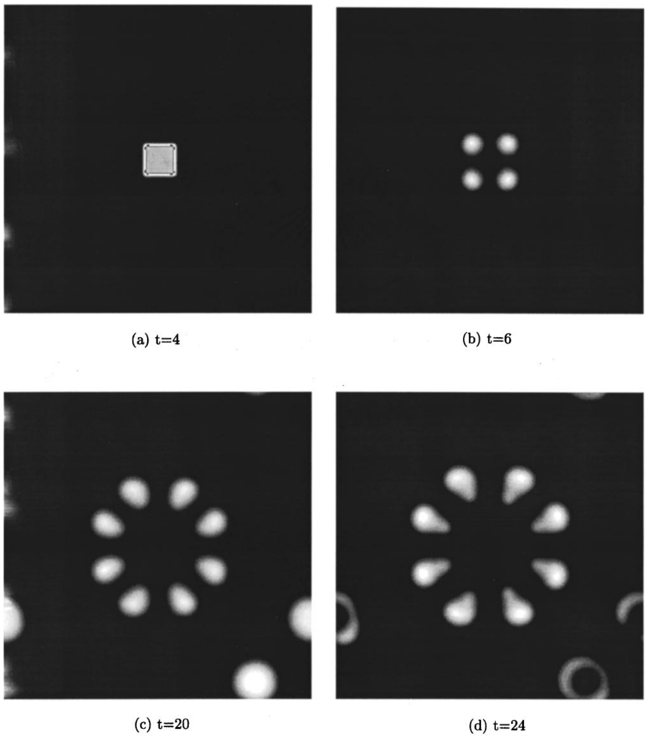


FIG. 4. Time evolution of a central square initial condition in a 256×256 grid for $K=0.05$ and $r_0=0.65$.

V. CONCLUSIONS

For the new templator model we have studied numerically and analytically the Turing patterns associated to a self-replicating dimer with a self-complementary template. Using the cellular automata approach, we observed spots with a hexagonal symmetry, stripes and other patterns similar to those observed in the Selkov–Schnakenberg–Gray–Scott (SSGS) (Refs. 33,34) type models and the Brusselator.[10] The major difference between the templator model and the other models is in the character of the nonlinearities. The templator model has a cubic nonlinear term, which is quadratic in the inhibitor and linear in the activator. In contrast, the SSGS type models and the Brusselator have a cubic nonlinearity which is linear in the inhibitor and quadratic in the activator. Also, the degradation term, which is linear in the SSGS type models and the Brusselator, is nonlinear in the templator and shows saturation for large values of the activator. Despite these differences, the templator shows similar Turing patterns as the other models. Thus it is possible that the templator model belongs to the same class of two variable models, like the SSGS, studied by Peason *et al.*[11]

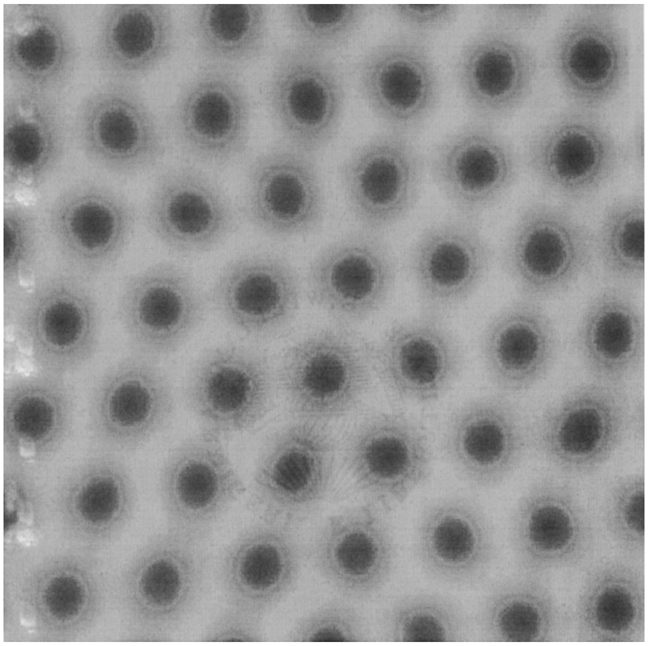


FIG. 5. Final honeycomb H_π pattern in a 256×256 grid for $K=0.25$ and $r_0=1.56$.

ACKNOWLEDGMENTS

The authors thank the NSF (CHE 9312160) and the Bronfman Science Center for their support.

[1] A. M. Turing, Philos. Trans. R. Soc. London, Ser. B **327**, 37 (1952).
[2] I. R. Epstein and K. Showalter, J. Phys. Chem. **100**, 13132 (1996).
[3] P. K. Maini, K. J. Painter, and H. N. P. Chau, J. Chem. Soc., Faraday Trans. **93**, 93 (1997).
[4] J. D. Murray, J. Theor. Biol. **88**, 161 (1981).
[5] K. J. Lee, W. D. McCormick, J. E. Pearson, and H. L. Swinney, Nature (London) **369**, 215 (1994).
[6] J. J. Tyson and J. D. Murray, Development (Cambridge, U.K.) **106**, 421 (1989).
[7] M. C. Cross and P. C. Hohenberg, Rev. Mod. Phys. **65**, 851 (1993).
[8] V. Dufiet and J. Boissonade, J. Chem. Phys. **96**, 664 (1992).
[9] J. Verdasca, A. de Wit, G. Dewel, and P. Borckmans, Phys. Lett. A **168**, 194 (1992).
[10] P. Borckmans, G. Dewel, A. DeWitt, and D. Walgraef, in *Chemical Waves and Patterns*, edited by R. Kapral and K. Showalter (Kluwer, Dordrecht, 1995).
[11] J. E. Pearson, Science **261**, 189 (1993).
[12] I. Lengyel and I. R. Epstein, Science **251**, 650 (1991).
[13] K. J. Lee, W. D. McCormick, Q. Ouyang, and H. L. Swinney, Science **261**, 192 (1993).
[14] I. Lengyel and I. R. Epstein, Proc. Natl. Acad. Sci. USA **89**, 3977 (1992).
[15] J. D. Murray, *Mathematical Biology*, 2nd ed. (Springer, Heidelberg, 1993).
[16] E. Peacock-López, D. Radov, and C. Flesner, Biophys. Chem. **65**, 171 (1997).
[17] J. Rebek, Jr., Science **255**, 848 (1990).
[18] T. Tjivikua, P. Ballester, and J. Rebek, Jr., J. Am. Chem. Soc. **112**, 1249 (1990).
[19] E. A. Wintner, M. Morgan Conn, and J. Rebek, Jr., Acc. Chem. Res. **27**, 198 (1994).
[20] E. A. Wintner, M. Morgan Conn, and J. Rebek, Jr., J. Am. Chem. Soc. **116**, 8877 (1994).
[21] J. Rebek, Jr., Chem. Br. **30**, 286 (1994).
[22] J. Rebek, Jr., Sci. Am. **271**, 48 (1994).
[23] T. Li and K. C. Nicolaou, Nature (London) **369**, 218 (1994).
[24] D. Sievers and G. von Kiedrowski, Nature (London) **369**, 221 (1994).
[25] S. Yao, I. Gossh, R. Zutshi, and J. Chmielewski, J. Am. Chem. Soc. **119**, 10559 (1997).
[26] L. J. Maher, P. B. Dervan, and B. Wold, J. Biochem. Biophys. Methods **29**, 8820 (1990).
[27] K. M. Vasquez and J. H. Wilson, Trends Biochem. Sci. **23**, 1 (1998).
[28] J. R. Weimar, J. J. Tyson, and L. T. Watson, Physica D **55**, 309 (1992).
[29] J. R. Weimar and J-P. Boon, Phys. Rev. E **49**, 1749 (1994).
[30] J. R. Weimar and J.-P. Boon, in *Lattice Gas Automata and Pattern Formation*, edited by L. Lawniczak and R. Kapral (Field Institute, Canada, 1994).
[31] J. R. Weimer, *Simulation with Cellular Automata* (Logos, Berlin, 1997).
[32] C. B. Muratov, Phys. Rev. E **54**, 3369 (1996).
[33] J. Schnakenberg, J. Theor. Biol. **81**, 389 (1979).
[34] P. Gray and S. Scott, Chem. Eng. Sci. **38**, 29 (1983).