Vertex-based finite volume simulation of Liesegang patterns on structureless meshes

Andrew Abi Mansour[1] and Mazen Al-Ghoul[2]
[1]*Department of Chemistry, Indiana University, Bloomington, Indiana, USA*
[2]*Department of Chemistry and Program in Computational Science, American University of Beirut, Beirut, Lebanon*
(Received 24 September 2013; published 10 March 2014)

A computational method is suggested for the simulation of Liesegang patterns in two dimensions on structureless meshes. The method is based on a model that incorporates dynamical equations for the nucleation and growth of solid particles of different sizes into reaction-diffusion equations. We find the model cannot be numerically solved with Galerkin-based finite element methods and cell-centered finite volume methods. Instead, the vertex-based finite volume method is used to correctly reproduce the Liesegang pattern on structureless meshes. The numerical solution is then compared with specially designed experiments on Liesegang patterns in various geometries, and it is shown to be in good agreement.

DOI: 10.1103/PhysRevE.89.033303 PACS number(s): 07.05.Tp, 05.10.−a, 87.18.Hf

I. INTRODUCTION

Chemical reactions coupled to diffusion often produce a spatiotemporal phenomenon that leads to the formation of a localized pattern [1,2]. One classical example of pattern formation in chemistry is periodic precipitation or the so-called Liesegang phenomenon [3–6]. It is a special type of chemical pattern formation that results from periodic precipitation in certain diffusion-driven chemical systems. The pattern is generally considered to be the product of a complex combination of chemical reaction(s), diffusion, nucleation, and growth. Models explaining the Liesegang phenomenon can be generally divided into two groups. The first is based on Ostwald's theory of successive supersaturation where it is assumed that the appearance of precipitation bands is due to a repetitive appearance of a supersaturation wave [7]. Since the system is diffusion controlled, the reaction zone becomes depleted and therefore the time required for precipitation to occur again increases. This in effect leads to an increasing distance of separation between the bands. Theories based on this concept are said to follow prenucleation mechanisms because emphasis is given to the relation among supersaturation, nucleation rate, and growth kinetics. The second group of models is based on postnucleation mechanisms where emphasis is given to the processes occurring after the nucleation of the solid phase. In such theories, it is generally believed that the growth mechanism is the result of the Lifshitz-Slyozov instability [8]: the growth of larger particles at the expense of the neighboring smaller ones, which could dissolve back. In this case the pattern is described as a dissipative structure without any need for supersaturation.

A model that takes both mechanisms into account was introduced in Ref. [9], where it was shown that this model could reproduce the Liesegang pattern on square domains [10]. To our knowledge, there are no simulations of Liesegang patterns in multidimensions on structureless meshes. Moreover, striving to solve the resulting partial differential equations with the Galerkin-based finite element and cell-centered finite volume methods using existing numerical libraries such as the Portable, Extensible Toolkit for Scientific Computation (PETSc) and software such as COMSOL Multiphysics Engineering Simulation Software did not lead to converging solutions. In this paper, a numerical scheme based on the vertex-based finite volume method (vb-FVM) is proposed in order to extend the model applicability to structureless meshes. In specific, it is shown that the Liesegang pattern can be reproduced on circular domains and starting from initial conditions of various geometrical constraints.

This paper is composed as follows. Section II contains the theoretical model describing the Liesegang phenomenon. In Sec. III we present the numerical algorithm to solve the discretized partial differential equations. In Sec. IV we implement the numerical scheme and present the numerical solutions and then compare them to the experiment. Section V concludes the paper.

II. THEORETICAL MODEL

Consider a closed system consisting of two components (A) and (B), of concentration functions $a(\mathbf{x},t)$ and $b(\mathbf{x},t)$, respectively. The vector $\mathbf{x}$ denotes position in real space, and t denotes time. The system Ω is partitioned into two; an interior subdomain is where reactant (A) is homogeneously distributed with concentration a_0 and the reactant (B) is zero, and an exterior part where (A) is zero and (B) is homogeneously distributed with concentration b_0. At time $t > 0$ the two reactants begin to diffuse and react irreversibly according to the following bimolecular chemical reaction

$$A + B \xrightarrow{k_r} C. \tag{1}$$

The reaction constant is k_r such that the rate of the reaction v_r is approximated by the mean-field theory

$$v_r(\mathbf{x},t) = k_r a(\mathbf{x},t) b(\mathbf{x},t). \tag{2}$$

Let $c(\mathbf{x},t)$ denote the concentration function of (C), a weakly soluble electrolyte. Evidently the initial value of (C) is zero everywhere on Ω. Over the boundaries, the fluxes of all components must vanish. The Neumann boundary conditions for the three components are

$$\nabla_{\mathbf{x}} a \cdot \mathbf{n}|_{\partial\Omega} = \nabla_{\mathbf{x}} b \cdot \mathbf{n}|_{\partial\Omega} = \nabla_{\mathbf{x}} c \cdot \mathbf{n}|_{\partial\Omega} = 0, \tag{3}$$

where $\mathbf{n}$ is the normal vector to $\partial\Omega$ and pointing outwards. Taking into account that (A), (B), and (C) undergo diffusion in the absence of convection, the following mass balance

equations can be written as

$$\partial_t a = \nabla_{\mathbf{x}} \cdot \Gamma_a \nabla_{\mathbf{x}} a - v_r, \tag{4}$$

$$\partial_t b = \nabla_{\mathbf{x}} \cdot \Gamma_b \nabla_{\mathbf{x}} b - v_r, \tag{5}$$

$$\partial_t c = \nabla_{\mathbf{x}} \cdot \Gamma_c \nabla_{\mathbf{x}} c + v_r + \mathbf{\Sigma}. \tag{6}$$

The derivative notation ∂_t is defined as $\partial_t \rightarrow \frac{\partial}{\partial_t}$. The scalar function $\mathbf{\Sigma}$ contains terms that are related to the solid phase dynamics. The diffusion coefficients of (A),(B), and (C), are Γ_a, Γ_b, and Γ_c, respectively. The precipitate is assumed to exist in only two forms: small-sized nuclei [represented by the nuclei distribution function $n(\mathbf{x},t)$], which can dissolve back to replenish (C) or transit to large-sized particles [represented by $p(\mathbf{x},t)$], which can grow at the expense of (C). It was shown in Ref. [9] that $\mathbf{\Sigma}$ can take the following simple form

$$\mathbf{\Sigma} = v_d - v_n - v_g, \tag{7}$$

such that v_d is the rate of nuclei dissolution, v_n is the rate of nucleation, and v_g is the rate of growth of the large particles. It is further assumed that the precipitate does not diffuse, which makes the evolution equations of the solid phase free of any terms containing spatial derivatives, and that in turn makes their numerical solution less expensive. This is physically justified on the basis that diffusion in solids is much slower than in liquids, and the nucleation and growth time scales are much less than that of diffusion. This agrees with experimental observations that show that the precipitate formed in Liesegang systems does not diffuse and distort the pattern [11,12]. On the other hand, the particular forms of the nucleation and growth rate functions do not significantly alter the final pattern observed [10,13]. Therefore, the simplest possible shaping functions are chosen such that the equations remain mass balanced. The nucleation, transition, dissolution, and growth rate functions are respectively given by

$$v_n = k_n^{(1)}(c - c_n)H(c - c_n), \tag{8}$$

$$v_t = k_n^{(2)}(c - c_d)H(c - c_d)n, \tag{9}$$

$$v_d = k_d(c_d - c)H(c_d - c)n, \tag{10}$$

$$v_g = k_g(c - c_g)H(c - c_g)p. \tag{11}$$

The parameters $k_n^{(1)}$, $k_n^{(2)}$, k_d and k_g are the nucleation, transition, dissolution and growth rate constants, respectively. The nucleation rate of the smallest particles, v_n, is independent of $n(\mathbf{x},t)$ because otherwise nucleation cannot commence. Moreover, the nucleation rate should vanish when $c(\mathbf{x},t)$ is less than some critical constant, termed c_n, which is why the rate functions have been multiplied by the Heaviside unit-step function. Similarly, the rate of transition from small nuclei to large particles and that of dissolution depend on the critical dissolution constant, c_d. Finally, the growth rate v_g, is a function of the particle distribution function, p and the critical growth constant, c_g. The differential equations describing the precipitate dynamics can be written as

$$\partial_t n = v_n - v_t - v_d, \tag{12}$$

$$\partial_t p = v_t + v_g. \tag{13}$$

The critical constants c_g and c_d are characteristic values of the system such that when $c(\mathbf{x},t)$ exceeds c_g, the growth rate is greater than zero, otherwise this rate vanishes. On the other hand, when $c(\mathbf{x},t)$ is less than c_d the rate of dissolution is positive while that of transition vanishes, otherwise the former goes to zero while the latter is positive. It is assumed that once the particles reach their maximum size, they do not dissolve back. They can grow or remain static in time. In this case, the Lifshitz-Slyozov instability [8] leads to a constraint on the nucleation critical constants: $c_n > c_d \geqslant c_g$. This constraint shall be imposed whenever Liesegang bands are sought after.

III. NUMERICAL ANALYSIS

A. Spatial discretization and time integration

Equations (4)–(6), (12)–(13) are spatially discretized using the vertex-based finite volume method (also known as the control volume finite element method), a general class of numerical methods that are widely employed for solving transport equations. The vertex-based finite volume method was introduced by Baliga and Patankar in 1980 [14]. The method was initially applied to the solution of a two-dimensional convection diffusion equation. Application to the complete Navier-Stokes equations followed later [15]. An attractive feature of this method is that the discretized equations are mass conservative and the nonlinear source or sink terms can be naturally diagonalized. This allows greater flexibility in dealing with systems of nonlinear equations, and leads to a robust method that can handle highly nonorthogonal meshes. This proved to be an integral part of our algorithm as we discuss in Sec. IV A. The numerical scheme used in this work is based on treating nucleation and growth sites at the centers of the control volumes (CVs) whereas diffusion fluxes are computed at the faces of each CV. Equations (4)–(6), (12)–(13) are solved over a two-dimensional mesh partitioned into a finite number of triangles. The numerical solution of these equations is sought on the domain of each CV, which is defined to be a $2m$-sided polygon where m is the number of triangles that contribute to the formation of this CV [16]. Such a polygon should be centered around each node of the mesh so that the total number of CVs is equal to the total number of nodes, N_c. The numerical solution ϕ is defined as a linear interpolation of its values at three nodes of each triangle

$$\phi(\mathbf{x}) = \sum_{i=1}^{3} \zeta_i(\mathbf{x})\phi_i, \tag{14}$$

where $\{\zeta_i\}_{i=1}^3$ is the set of basis functions, taken to be linear functions of space. Furthermore, the union of all CV polygons must form the total mesh. For each triangle, its contributing node belongs to two of its three edges. Thus, the midpoints of each of these two edges are connected to the centroid of the triangle, forming two segments of the CV polygon. The procedure is repeated again for another contributing triangle, and so on. The result is a polygon with $2m$ sides (Fig. 1) and of volume V,

$$V = \frac{1}{3}h \sum_{k=1}^{m} A_k, \tag{15}$$

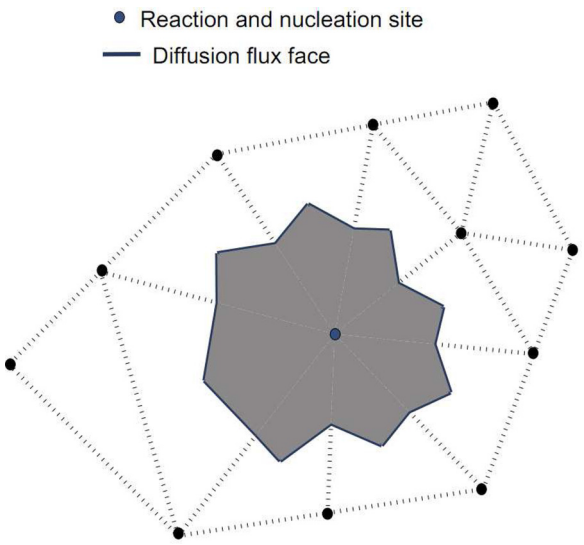

FIG. 1. A hexadecagonal control volume in gray (its centroid is in blue) is generated from eight neighboring triangles that form the region of support.

where h is the depth and A_k the area of triangle k. Equations (4)–(6), (12)–(13) are transformed into their weak form by integrating them over the entire domain. Writing this form at the local (triangular) level yields

$$\int_{\Omega_k} \partial_t \phi dV = \int_{\Omega_k} \nabla_{\mathbf{x}} \cdot \Gamma \nabla_{\mathbf{x}} \phi dV + \int_{\Omega_k} s dV, \tag{16}$$

where Ω_k is the domain of the kth triangle forming the region of support for the CV under consideration, ϕ represents the concentration function of any of the components, and s a nonlinear term that accounts for the reaction, nucleation, or growth. For the nucleation and growth equations, the diffusion term vanishes. The transient and reaction or nucleation integrals are then approximated by a first-order Gauss quadrature, which is evaluated at the centroid of the CV (see Fig. 1)

$$\int_{\Omega_k} \partial_t \phi dV = A_k d_t \phi_c h, \tag{17}$$

where ϕ_c is the value of ϕ at the centroid of the CV. Similarly, the volume integral arising from the nonlinear terms can be evaluated with a first-order Gauss quadrature

$$\int_{\Omega_k} s dV = A_k s(\phi_c) h. \tag{18}$$

On the other hand, by applying the divergence theorem, each diffusion volume integral is transformed into a surface integral, which is split into two line integrals

$$\int \Gamma \nabla_{\mathbf{x}} \phi \cdot \mathbf{n} dS = h \int_{f_1} \Gamma \nabla_{\mathbf{x}} \phi \cdot \mathbf{n} d\ell + h \int_{f_2} \Gamma \nabla_{\mathbf{x}} \phi \cdot \mathbf{n} d\ell. \tag{19}$$

Let the coordinates of the kth triangle be represented by the vectors $\mathbf{x}^{(k_1)}$, $\mathbf{x}^{(k_2)}$, and $\mathbf{x}^{(k_3)}$ (see Fig. 2). Then each line integral is approximated with a first-order Gauss quadrature

$$\begin{aligned}\Gamma \nabla_{\mathbf{x}} \phi \cdot \mathbf{n}_{f_1} \ell\big|_{f_1} \simeq \frac{\Gamma}{2A^{(k)}} \{ &\big[\big(\mathbf{x}_2^{(k_2)} - \mathbf{x}_2^{(k_3)}\big)\big(\phi_{k_1} - \phi_{k_3}\big) - \big(\mathbf{x}_2^{(k_1)} - \mathbf{x}_2^{(k_3)}\big)\big(\phi_{k_2} - \phi_{k_3}\big)\big]\delta\mathbf{x}_2^{(1)} + \big[\big(\mathbf{x}_1^{(k_2)} - \mathbf{x}_1^{(k_3)}\big)\big(\phi_{k_1} - \phi_{k_3}\big) \\ &- \big(\mathbf{x}_1^{(k_1)} - \mathbf{x}_1^{(k_3)}\big)\big(\phi_{k_2} - \phi_{k_3}\big)\big]\delta\mathbf{x}_1^{(1)}\},\end{aligned} \tag{20}$$

$$\begin{aligned}\Gamma \nabla_{\mathbf{x}} \phi \cdot \mathbf{n}_{f_2} \ell\big|_{f_2} \simeq \frac{\Gamma}{2A^{(k)}} \{ &\big[\big(\mathbf{x}_2^{(k_2)} - \mathbf{x}_2^{(k_3)}\big)\big(\phi_{k_1} - \phi_{k_3}\big) - \big(\mathbf{x}_2^{(k_1)} - \mathbf{x}_2^{(k_3)}\big)\big(\phi_{k_2} - \phi_{k_3}\big)\big]\delta\mathbf{x}_2^{(2)} + \big[\big(\mathbf{x}_1^{(k_2)} - \mathbf{x}_1^{(k_3)}\big)\big(\phi_{k_1} - \phi_{k_3}\big) \\ &- \big(\mathbf{x}_1^{(k_1)} - \mathbf{x}_1^{(k_3)}\big)\big(\phi_{k_2} - \phi_{k_3}\big)\big]\delta\mathbf{x}_1^{(2)}\}.\end{aligned} \tag{21}$$

The projection vectors from Fig. 2 can be found [16] to be

$$\delta\mathbf{x}^{(1)} = -\frac{\mathbf{x}^{(k_1)}}{6} + \frac{\mathbf{x}^{(k_2)}}{3} - \frac{\mathbf{x}^{(k_3)}}{6}, \tag{22}$$

$$\delta\mathbf{x}^{(2)} = \frac{\mathbf{x}^{(k_1)}}{6} + \frac{\mathbf{x}^{(k_2)}}{6} - \frac{\mathbf{x}^{(k_3)}}{3}. \tag{23}$$

The diffusion surface integral yields

$$\int \Gamma \nabla_{\mathbf{x}} \phi \cdot \mathbf{n} dS \simeq -h \sum_{i \in \Omega_k} c_{k_1,i} \phi_i, \tag{24}$$

where

$$c_{k_1,k_1} = \frac{\Gamma}{2A_k}[(\mathbf{x}^{(k_3)} - \mathbf{x}^{(k_2)}) \cdot (\delta\mathbf{x}^{(1)} + \delta\mathbf{x}^{(2)})], \tag{25}$$

$$c_{k_1,k_2} = \frac{\Gamma}{2A_k}[(\mathbf{x}^{(k_1)} - \mathbf{x}^{(k_3)}) \cdot (\delta\mathbf{x}^{(1)} + \delta\mathbf{x}^{(2)})], \tag{26}$$

$$c_{k_1,k_3} = \frac{\Gamma}{2A_k}[(\mathbf{x}^{(k_2)} - \mathbf{x}^{(k_1)}) \cdot (\delta\mathbf{x}^{(1)} + \delta\mathbf{x}^{(2)})]. \tag{27}$$

Each of the faces of the polygon has an outward normal vector that is perpendicular to its surface line (see Fig. 2). Since each triangle provides two faces for the control volume, two fluxes are defined with each being parallel to its outward normal vector. The global diffusion matrix is assembled by looping over each node (representing a specific CV) and computing three entries [Eqs. (25)–(27)] for each of the contributing triangles forming the region of support (Fig. 1). The number of triangles in the region of support can be computed from the data structure generated by the open source software TRIANGLE [17], which constructs a set of two-dimensional (2D) arrays (one for the triangular nodes and another for their coordinates) that represent a structureless mesh. For vb-FVM computations, it is necessary that the region of support for each CV is determined. An additional data structure therefore includes a two-dimensional array of length N_c and a variable number of columns.

Let $\mathbf{\Phi}$ represent the discretized unknown functions at all nodes. Then, the system of spatially discretized time-

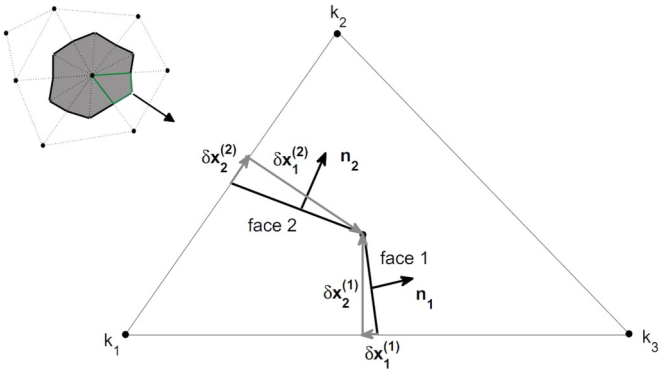

FIG. 2. A contributing triangle to the control volume (centered at node k_1) through its two edges, k_1-k_2 and k_1-k_3. The normal vectors to the faces are projected (in gray) along the triangular axes.

dependent ordinary differential equations for Eqs. (4)–(6), (12)–(13) can be written as

$$d_t \mathbf{\Phi} = -\mathbf{\Psi}\mathbf{\Phi} + \mathbf{s}, \tag{28}$$

where the tensor $\mathbf{\Psi}$ contains the diffusion matrices and the vector $\mathbf{s}$ contains all the nonlinear terms. A variable order and variable step-size memory-based implicit time integrator is used to solve Eqs. (28) using up to fourth-order backward differentiation formulas (BDF) [18]. For a discrete time interval δt at the mth time step, the temporal derivative can be approximated with

$$d_t \mathbf{\Phi} = \sum_{i=0}^{n} \lambda_i \mathbf{\Phi}^{t_{m-i}} + O(\delta t^n). \tag{29}$$

Here $\{\lambda_i\}_{i=0}^{n}$ is a set of BDF coefficients that depend on the current and previous time steps in a complicated manner [19,20]. Since the equations are stiff, the time step was varied (see Fig. 3) such that the minimum number of computations are performed by the solver [21].

B. Algebraic solution

The reactants (A) and (B) form a reaction-diffusion system that is independent of other components. Therefore, the differential equations for (A) and (B) ought to be solved first, and then the solution obtained is used for the nucleation and growth equations. Newton's method is the most commonly used iterative method for finding an approximate solution of a system of nonlinear algebraic equations. For large computational problems, however, building the Jacobian (or an approximation of it) can be quite expensive. Therefore, a fixed point iteration is used that allows decomposition and manipulation of the nonlinear terms in such a way that convergence is rapid for the (A)-(B) reaction-diffusion system. Such a scheme falls under the realm of operator splitting (OS) methods [22–26]. The linearized equations are

$$\left[\mathbf{\Upsilon}_A^{t_m}\right]_j (\mathbf{a}^{t_m})_{j+1} = -\sum_{i=1}^{n} \lambda_i \mathbf{a}^{t_{m-i}}, \tag{30}$$

$$\left[\mathbf{\Upsilon}_B^{t_m}\right]_j (\mathbf{b}^{t_m})_{j+1} = -\sum_{i=1}^{n} \lambda_i \mathbf{b}^{t_{m-i}}, \tag{31}$$

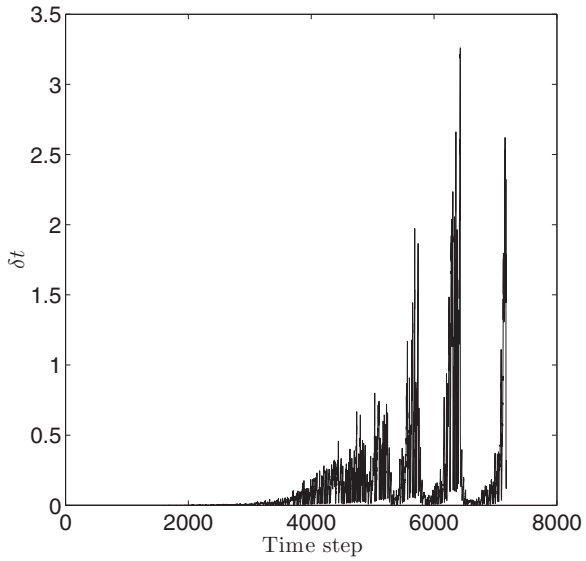

FIG. 3. A plot of the time-step size δt versus the time-step number. In solving Eq. (28), the BDF integrator takes small time steps during the formation of a ring in order to resolve the nonlinear nucleation and growth terms. Between the formation of one ring and another, the system is diffusion controlled, and the solver therefore takes much larger time steps. The parameters used here are the same as those in Fig. 4 for the Liesegang rings.

where the notation $[\mathbf{v}]_j$ denotes that the tensor $\mathbf{v}$ in brackets is being evaluated at the jth iteration. The vectors $\mathbf{a}$ and $\mathbf{b}$ contain the discretized concentrations of (A) and (B), respectively, and

$$\left[\mathbf{\Upsilon}_A^{t_m}\right]_j = \lambda_0 \mathbf{I} + \Gamma_a \mathbf{\Pi} + k_r \left[\mathbf{D}_b^{t_m}\right]_j, \tag{32}$$

$$\left[\mathbf{\Upsilon}_B^{t_m}\right]_j = \lambda_0 \mathbf{I} + \Gamma_b \mathbf{\Pi} + k_r \left[\mathbf{D}_a^{t_m}\right]_j, \tag{33}$$

where $\mathbf{\Pi}$ is the diffusion matrix, $\mathbf{I}$ is the identity matrix, and $\mathbf{D}_v^{t_m}$ is a diagonal matrix with entries equal to $\mathbf{v}$ evaluated at time t_m. The convergence rate of this method is linear. However, since convergence is quite rapid, the method is computationally less expensive than Newton's method. On the other hand, the nucleation, growth, and diffusion equations for (C), (N), and (P) are solved using the trust region algorithm [27], where a modified Newton direction is taken, depending on how well the right-hand side equations are minimized. The unconstrained Newton method leads to the following linearized system

$$[\mathbf{J}]_j \times \begin{pmatrix} \delta\mathbf{c} \\ \delta\mathbf{n} \\ \delta\mathbf{p} \end{pmatrix}_{j+1} = -\begin{pmatrix} \sum_{i=0}^{n} \lambda_i \mathbf{c}^{t_m - i} + \Gamma_c \mathbf{\Pi} c^{t_m} - \mathbf{s}_c^{t_m} \\ \sum_{i=0}^{n} \lambda_i \mathbf{n}^{t_m - i} - \mathbf{s}_n^{t_m} \\ \sum_{i=0}^{n} \lambda_i \mathbf{p}^{t_m - i} - \mathbf{s}_p^{t_m} \end{pmatrix}_j, \tag{34}$$

where $\delta[\mathbf{v}]_{j+1}$ is equal to $[\mathbf{v}]_{j+1} - [\mathbf{v}]_j$ for a vector $\mathbf{v}$, and $\mathbf{s}_c$, $\mathbf{s}_n$, and $\mathbf{s}_p$ are vectors that contain all the discretized nonlinear terms for species (C), (N), and (P), respectively. The Jacobian

J of these equations can be written in the following form:

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} & \mathbf{J}_{13} \\ \mathbf{J}_{21} & \mathbf{J}_{22} & \mathbf{0} \\ \mathbf{J}_{31} & \mathbf{J}_{32} & \mathbf{J}_{33} \end{pmatrix}, \tag{35}$$

where

$$\mathbf{J}_{11} = \lambda_0\mathbf{I} + \Gamma_c\mathbf{\Pi} + k_n^{(1)}\theta^{(n)} + k_g\theta^{(g)}\mathbf{D}_p^{t_m} + k_d(\mathbf{I} - \theta^{(d)})\mathbf{D}_n^{t_m}, \tag{36}$$

$$\mathbf{J}_{12} = -k_d\left(c_d\mathbf{I} - \mathbf{D}_c^{t_m}\right)(\mathbf{I} - \theta^{(d)}), \tag{37}$$

$$\mathbf{J}_{13} = k_g\left(\mathbf{D}_c^{t_m} - c_g\mathbf{I}\right)\theta^{(g)}, \tag{38}$$

$$\mathbf{J}_{21} = -k_n^{(1)}\theta^{(n)} - k_d(\mathbf{I} - \theta^{(d)})\mathbf{D}_n^{t_m} + k_n^{(2)}\theta^{(d)}\mathbf{D}_n^{t_m}, \tag{39}$$

$$\mathbf{J}_{22} = \lambda_0\mathbf{I} + k_d\left(c_d\mathbf{I} - \mathbf{D}_c^{t_m}\right)(\mathbf{I} - \theta^{(d)}) + k_n^{(2)}\theta^{(d)}\left(\mathbf{D}_c^{t_m} - c_d\mathbf{I}\right), \tag{40}$$

$$\mathbf{J}_{31} = -k_n^{(2)}\theta^{(d)}\mathbf{D}_n^{t_m} - k_g\theta^{(g)}\mathbf{D}_p^{t_m}, \tag{41}$$

$$\mathbf{J}_{32} = -k_n^{(2)}\left(\mathbf{D}_c^{t_m} - c_d\mathbf{I}\right)\theta^{(d)}, \tag{42}$$

$$\mathbf{J}_{33} = \lambda_0\mathbf{I} - k_g\left(\mathbf{D}_c^{t_m} - c_g\mathbf{I}\right)\theta^{(g)}, \tag{43}$$

and θ is a diagonal matrix such that its ith entry is

$$\theta_{ii}^{(n)} = H\left(\mathbf{c}_i^{t_m} - c_n\right), \tag{44}$$

$$\theta_{ii}^{(d)} = H\left(\mathbf{c}_i^{t_m} - c_d\right), \tag{45}$$

$$\theta_{ii}^{(g)} = H\left(\mathbf{c}_i^{t_m} - c_g\right). \tag{46}$$

Since the differential equations describing the temporal evolution of the nuclei and the precipitate in Eqs. (12), (13) are free of spatial derivatives ($\mathbf{J}_{22}$ and $\mathbf{J}_{33}$ are diagonal matrices), the Jacobian can be reduced in size from $3N_c \times 3N_c$ to $N_c \times N_c$:

$$[\mathbf{J}_c]_j[\delta\mathbf{c}]_{j+1} = -[\mathbf{r}_c]_j, \tag{47}$$

where

$$\mathbf{J}_c = \mathbf{J}_{11} - \mathbf{J}_{12}\mathbf{J}_{22}^{-1}\mathbf{J}_{21} - \mathbf{J}_{13}\mathbf{J}_{33}^{-1}\mathbf{J}_{31} + \mathbf{J}_{13}\mathbf{J}_{33}^{-1}\mathbf{J}_{32}\mathbf{J}_{22}^{-1}\mathbf{J}_{21}, \tag{48}$$

$$\mathbf{r}_c = \sum_{i=0}^{n}\lambda_i\mathbf{c}^{t_m-i} + \Gamma_c\mathbf{\Pi}\mathbf{c} - \mathbf{s}_c - \mathbf{J}_{12}\mathbf{J}_{22}^{-1}\mathbf{r}_n - \mathbf{J}_{13}\mathbf{J}_{33}^{-1}\mathbf{r}_p + \mathbf{J}_{13}\mathbf{J}_{33}^{-1}\mathbf{J}_{32}\mathbf{J}_{22}^{-1}\mathbf{r}_n, \tag{49}$$

and

$$\mathbf{r}_n = \sum_{i=0}^{n}\lambda_i\mathbf{n}^{t_m-i} - \mathbf{s}_n, \tag{50}$$

$$\mathbf{r}_p = \sum_{i=0}^{n}\lambda_i\mathbf{p}^{t_m-i} - \mathbf{s}_p. \tag{51}$$

From $[\delta\mathbf{c}]_{j+1}$ one can easily compute $[\delta\mathbf{n}]_j$ and $[\delta\mathbf{p}]_j$ as follows

$$[\delta\mathbf{n}]_j = -\left[\mathbf{J}_{22}^{-1}\right]_j \times ([\mathbf{r}_n]_j + [\mathbf{J}_{21}]_j[\delta\mathbf{c}]_j), \tag{52}$$

$$[\delta\mathbf{p}]_j = -\left[\mathbf{J}_{33}^{-1}\right]_j \times ([\mathbf{r}_p]_j + [\mathbf{J}_{32}]_j[\delta\mathbf{n}]_j + [\mathbf{J}_{31}]_j[\delta\mathbf{c}]_j). \tag{53}$$

We note that the reduced Jacobian is not only smaller in size but also symmetric. All of the discretized matrices that result from Eqs. (4)–(6), (12)–(13) are sparse and were therefore assembled in compressed row storage using the CSPARSE library [28]. For each iteration, three linearized systems are to be solved. Since the first two operators [corresponding to Eqs. (32) and (33)] are symmetric positive definite, the preconditioned conjugate gradient (PCCG) method [29] is used to solve the linear systems, with the incomplete Cholesky factorization used for preconditioning. In solving Eq. (47), the biconjugate gradient stabilized method is used since the reduced Jacobian is not always positive definite.

IV. RESULTS AND DISCUSSION

A. Numerical method

We have chosen the vb-FVM as a discretization method because other well-known discretization methods failed to reproduce the Liesegang pattern on complicated geometries. The methods used were the Galerkin-based FEM and the cell-centered FVM (cc-FVM). The former produced unphysical results on circular domains. Decreasing the time step, increasing the number and/or order of the elements, and refining or coarse graining the mesh had no significant impact on the solution. The cell-centered finite volume method, on the other hand, is typically used in commercial CFD codes and performs very well on orthogonal meshes. However, when the mesh is highly distorted, this method is not very robust and problems with convergence might arise [30]. This could be one of the reasons the cc-FVM failed on circular and random geometrical domains but performed well on quasiorthogonal meshes (such as rectangles and squares). Thus, we believe that the vb-FVM correctly reproduced the Liesegang pattern for two reasons: (i) its ability to handle highly nonorthogonal grids with ease, and (ii) its higher numerical stability due to diffusion. The second point follows from the fact that each CV consists of multiple faces (Fig. 1), so each node experiences diffusion from multiple sides, and this in turn reduces numerical instabilities that are prevalent in the other two aforementioned numerical methods.

B. Experimental method

To compare the simulation results with experiment, we pick the cobalt hydroxide-ammonia Liesegang system [31], which displays numerous well-resolved green/blue Liesegang bands in a relatively short period of time. We designed special reactors with different geometries that will allow us to compare the resulting patterns to the simulated ones. The chemistry taking place is summarized as follows. A gel (gelatin) containing the light pink cobalt bromide (source of inner Co^{2+}) displays the green/blue α-$Co(OH)_2$ (solubility product $K_{sp} = 3.00 \times 10^{-16}$) precipitation Liesegang rings upon diffusion of ammonia solution (called outer electrolyte), which consists of ammonia (NH_3), the hydroxide ions (OH^-) and the ammonium ions (NH_4^+). α-$Co(OH)_2$ is one of two polymorphs of cobalt hydroxide [32]. Later, the $Co(OH)_2$ rings near the interface redissolve, forming the orange/yellow $Co(NH_3)_n^{2+}$ complex (for $n = 6$, the formation constant is $K_f = 5.00 \times 10^4$) due to a trailing wave of excess NH_3 [33]. However, in this paper, we only focus on ring formation, which dominates due to a proper choice of experimental parameters, and ignore the dissolution process. Consequently, the effective reaction scheme representing the precipitation scenario in the presence of ammonia is:

$$Co^{+2}(aq) + 2OH^-(aq) \rightarrow \alpha - Co(OH)_2(s) \rightarrow \text{Liesegang rings.} \quad (54)$$

The preparation of the system consists of the following experimental procedures: The required masses of $CoBr_2$ (Aldrich) to prepare a 0.2 M solution and gelatin (5%) (Difco) were weighed to the nearest 0.0001g. These masses are then transferred to a beaker containing doubly distilled water. After dissolution of the solid, gelatin powder is added to make a 5% gelatin solution. The solution is then heated with continuous stirring until all the gelatin dissolves. The resulting pink Co(II)/gel mixture is immediately transferred into a specially designed circular reactor, which consists of a plexiglass Petri-dish-like container above which a plexiglass cover plate is fitted. On the center of this cover, a reservoir tube or multireservoirs are adjusted according to a defined geometry. Each reservoir is fabricated with a cross section of a specific shape. The cover plate is also equipped with spacers to control the thickness of the gel matrix. All experiments were performed with a 0.7 mm spacers. The system is then covered and allowed to stand for about 24 hours in a thermostatic chamber at 18°C. On the next day, the gel is carefully removed from the bottom of the pouring reservoir tube, then a dilute solution of ammonia (1–3 M) is added to the reservoir. As the outer electrolyte solution is being added over the interface, a homogeneous greenish blue precipitate starts to form. Photos of the evolution of the precipitate system are taken with a computer-interfaced digital Canon D450 camera at various time intervals.

C. Qualitative comparison

The numerical algorithm is implemented and tested in different domains of initially separated reactants (A and B) and by choosing suitable parameters that give rise to Liesegang bands. The inner electrolyte (A), which assumes the higher concentration, occupies inner domains (reservoirs) of specific shapes and/or configuration, and the outer electrolyte (B), which has the relatively lower concentration, occupies the remaining domains. The shape of the overall domain in this work is either circular or squared and we choose various shapes and configurations for the inner domains. The choice of the geometrical shapes and configurations are chosen in such a way that can be directly tested experimentally. The spatial distribution function for the precipitate (P) is then plotted at a given time t. In the case of a circular inner domain as shown in Fig. 4, we can see that the system develops into circular Liesegang rings with increasing spacing as expected in such systems [34].

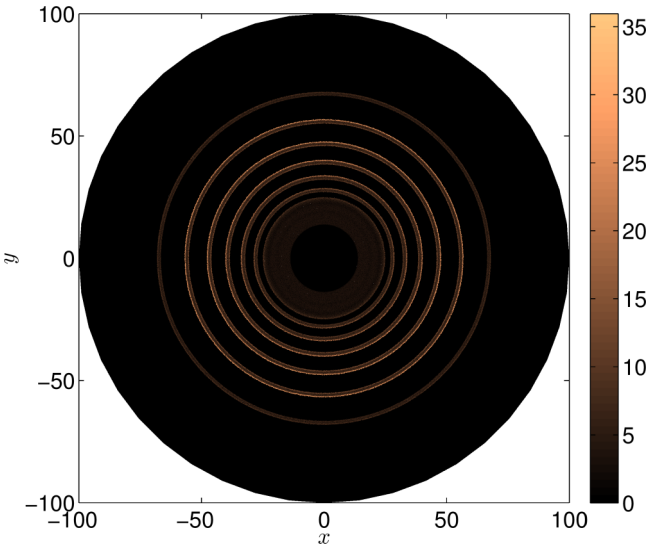

FIG. 4. (Color online) A mesh plot shows the precipitate (P) spatial distribution at $t = 800$ in the form of continuous Liesegang rings in a bounded system with two initially separated reactants; (A) occupied an inner circle of radius $10\sqrt{2}$ while (B) filled the remaining part of the larger circle. Parameters used: $\Gamma_a = 1$, $\Gamma_b = 1$, $\Gamma_c = 0.5$, $a_0 = 500$, $b_0 = 3$, $k_r = 10^{-3}$, $k_n^{(1)} = 2.0$, $k_n^{(2)} = 2.0$, $k_d = 1.0$, $k_g = 1.0$, $c_g = 1$, $c_d = 1.25$, $c_n = 1.5$. Mesh characteristics: 501 873 vertices, 1 001 395 triangles, and maximum elemental area = 0.05. The precipitate global size distribution function has been rescaled by a factor of 0.5.

In the case of Fig. 5, the reactant is initially distributed into four equal circular containers. The resulting pattern is complex especially in the region of intersection of the precipitates. In Fig. 6, we start with a global square domain and the reactant is initially distributed over nine circular domains of equal size. We obtain a precipitation pattern that is symmetric and complex with increasing details at the intersections of the precipitates.

In Fig. 7, we start with an inner domain with a square geometry over a global circular domain. In this case, the rings that form tend to round the edges caused by the inner square geometry. This feature is due to diffusion and will be reproduced experimentally in the next section.

In Fig. 8, we change the inner domain from a square to a triangle. The resulting precipitation pattern is a modulated triangle with round edges (similar to the square situation discussed previously), which lead to a breakup due to strain imposed by the triangular geometry. The breakup persists even after jiggling or refining the mesh. This feature is always

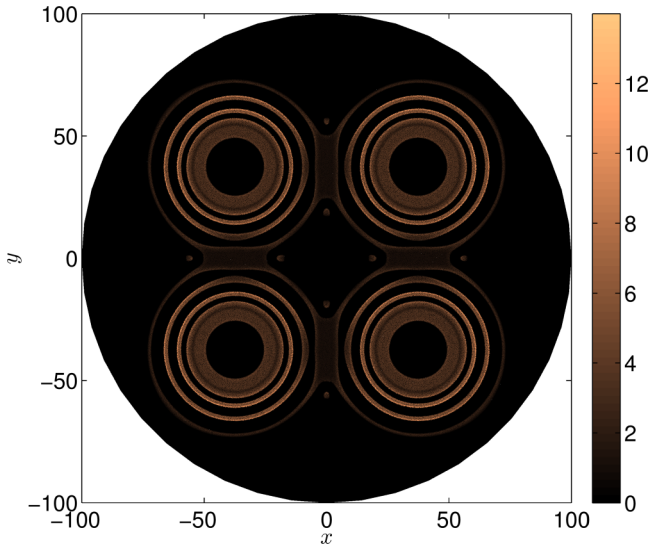

FIG. 5. (Color online) A mesh plot shows the precipitate (P) spatial distribution at $t = 300$ in the form of Liesegang rings that intersect at the center of the circular mesh. The reactant (A) was initially distributed throughout four separated circles, each of radius $10\sqrt{\frac{3}{2}}$, and on the same mesh and with the same parameters used in Fig. 4.

encountered in experiment in the case of geometries with sharp edges and might give rise to dislocations as we will see in the coming section.

The patterns obtained experimentally for the cobalt hydroxide-ammonia Liesegang system are shown in Figs. 12(a)–12(d). In Figs. 12(a)–12(c), the shape of the cross section of the inner reservoir is varied whereas in

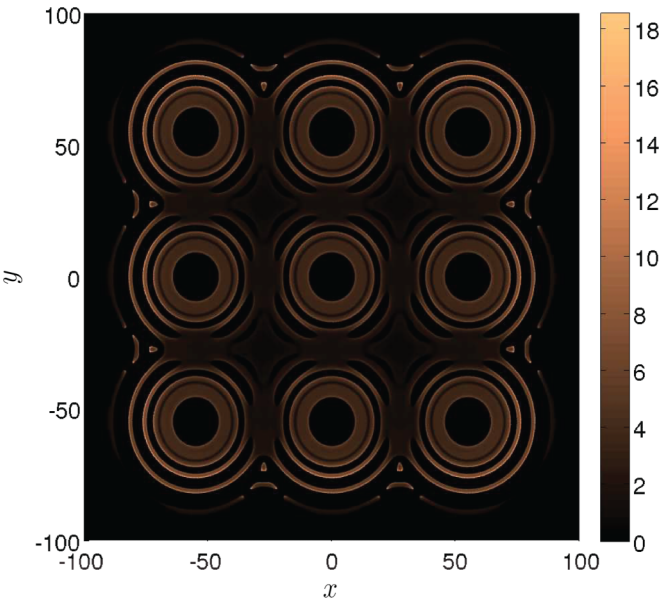

FIG. 6. (Color online) A mesh plot shows the precipitate (P) spatial distribution at $t = 200$ in the form of multiple Liesegang rings intersecting. The reactant (A) was initially distributed throughout nine separated circles, each of radius 10, and with the same parameters used in Fig. 5. Square mesh characteristics: 642 035 vertices, 1 281 543 triangles, and maximum elemental area = 0.05.

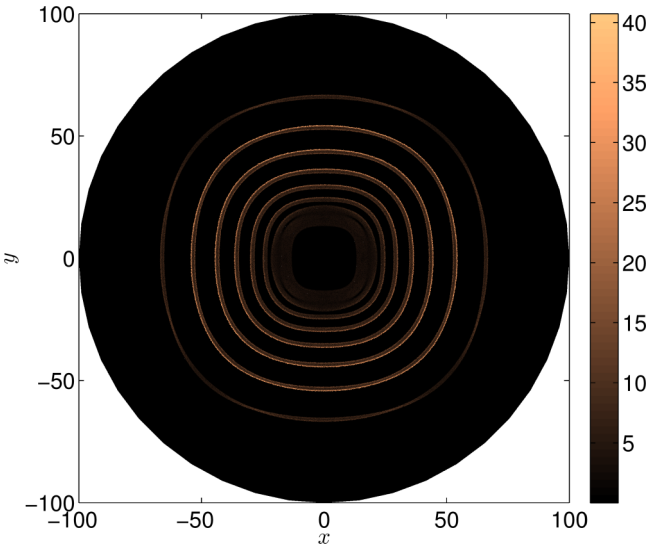

FIG. 7. (Color online) A mesh plot shows the precipitate (P) spatial distribution at $t = 800$ in the form of four Liesegang squares that are eventually (as the front approaches the boundary) perturbed into ellipses due to the circular geometry of the mesh. The reactant (A) was initially distributed throughout a square of length $10\sqrt{2\pi}$ on the same mesh and with the same parameters used in Fig. 5. The precipitate global size distribution function has been rescaled by a factor of 0.5.

Fig. 12(d) more circular reservoirs are placed symmetrically around the center of the domain. It is clear that different patterns emerge as a result of the imposed shape and configurations and their experimental shape is predicted by the

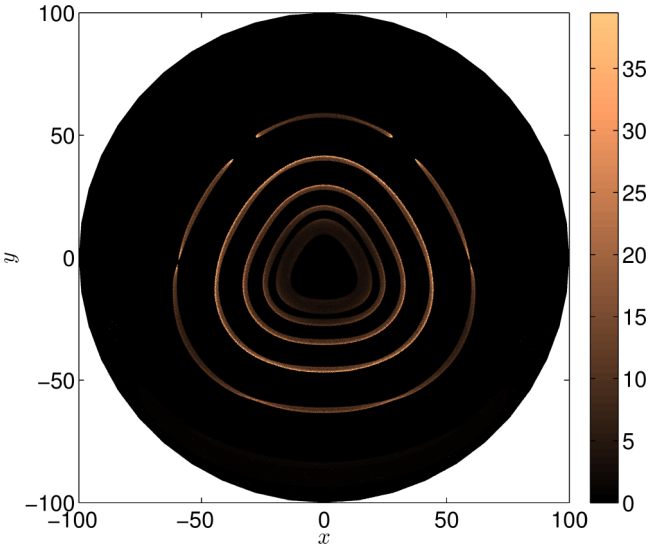

FIG. 8. (Color online) A mesh plot shows the precipitate (P) spatial distribution at $t = 850$ in the form of four Liesegang triangles. The reactant (A) was initially distributed throughout an equilateral triangle of length $10 \times \sqrt{\pi}$ on the same circular mesh used in Fig. 5. Parameters used: $\Gamma_a = 1.0$, $\Gamma_b = 1.0$, $\Gamma_c = 1.0$, $a_0 = 500$, $b_0 = 3$, $k_r = 10^{-3}$, $c_g = 1$, $c_d = 1.25$, $c_n = 1.5$, $k_n^{(1)} = 2.0$, $k_n^{(2)} = 2.0$, $k_d = 1.0$, $k_g = 0.5$. The precipitate global size distribution function has been rescaled by a factor of 0.2.

simulation. For example, Fig. 12(a) represents the experimental realization of Fig. 4; Fig. 12(b) corresponds to simulation in Fig. 8; Fig. 12(c) corresponds to the simulation in Fig. 7; and Fig. 12(d) corresponds to the multireservoir simulation in Fig. 5. These aforementioned comparisons confirm the ability to use the model and simulation as tools for design of precipitation patterns for any geometry. Furthermore, dislocations are also encountered in the case of inner geometries with edges that can be clearly seen in Figs. 12(b), 12(c) for triangular and square geometries.

D. Quantitative comparison

Two essential laws associated with the Liesegang banding phenomenon encountered in a variety of experiments with different electrolytes have also been verified using this model. The first one is the so-called spacing law [35,36] and it indicates that the ratio x_{n+1}/x_n of the positions x_n and x_{n+1} of the consecutive precipitation bands labeled as n and $n+1$ tends to a constant for large n, for a given set of initial conditions, and it can be written as:

$$x_{n+1}/x_n = 1 + p, \tag{55}$$

where p is a spacing coefficient that varies between 0.01 and 0.5, depending on the inner and outer concentrations. Experimentally, this indicates that the spacing between bands increases with the number of bands. Moreover, the time of appearance of the rings also follows a time law [37] also characteristic of Liesegang rings whereby the time t_n it takes to form the nth band increases as a function of n such that the ratio of x_n^2/t_n approaches a constant value as n increases.

In order to verify these laws using the numerical simulation, it is important to define the location of the bands and the time of its formation from simulation and use them systematically. Therefore, we computed the maximum value (in space) of $c(x,t)$ as shown in Fig. 9, which as expected in the case of Liesegang banding exhibits oscillations in time around c_n [Eq. (8)]. The period of formation, t_n, for the nth ring and the distance separating this ring from the initial interface, x_n, can be therefore computed from the temporal evolution of $c_{\max}$. Both aforementioned laws are thereafter verified in this model using Fig. 4 as shown in Figs. 10, 11. These laws can also be verified for all the other simulated patterns yielding almost the same spacing coefficient p at the same inner and outer concentrations, regardless of the geometry of the reservoir. These results are confirmed experimentally using the patterns

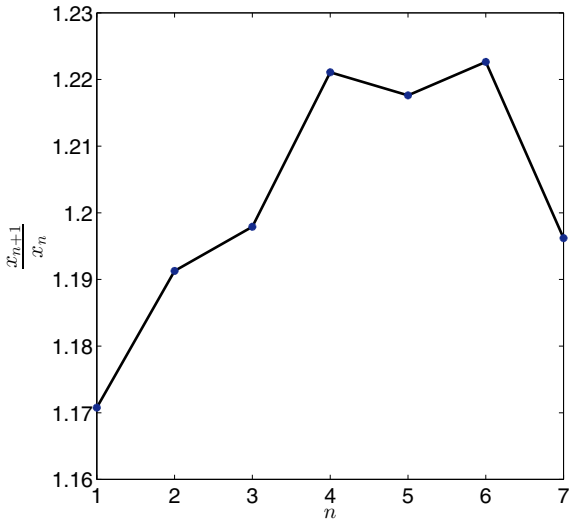

FIG. 10. (Color online) Plot of the ratio x_{n+1}/x_n versus the band number n. It is approximately a constant equal to 1.2. This implies the distance of separation, x_n, follows a geometric progression as encountered in experiments on Liesegang bands.

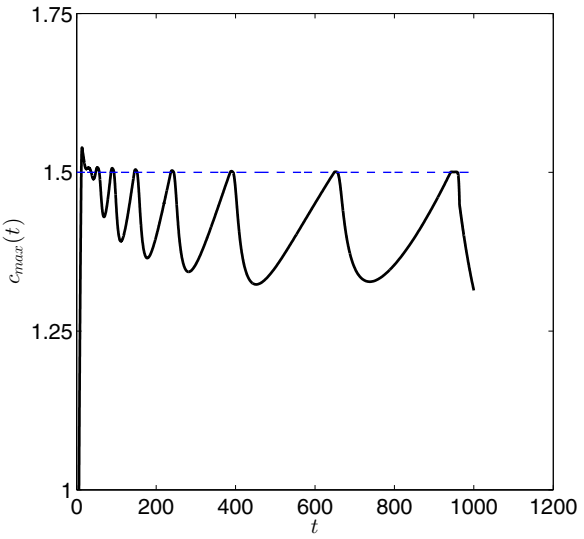

FIG. 9. (Color online) Temporal oscillations of the maximum value (in space) of $c(x,t)$ around c_n (taken to be 1.5 here). The period of formation, t_n, for the n^{th} ring and the distance separating this ring from the initial interface, x_n, are computed from the temporal evolution of $c_{\max}$.

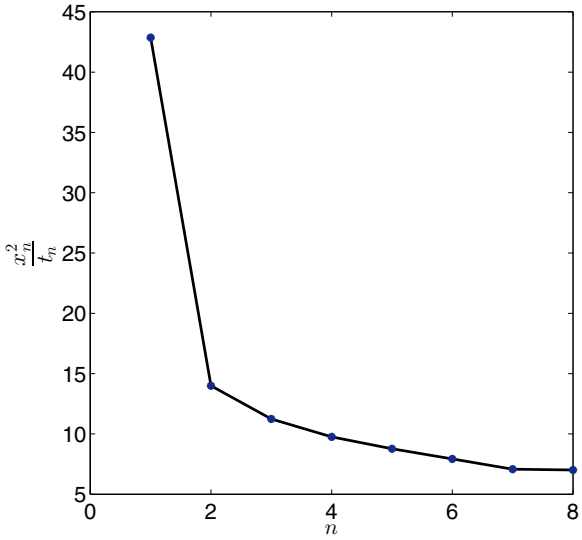

FIG. 11. (Color online) Plot of the ratio x_n^2/t_n versus the band number n. As the number of rings increases, the ratio of x_n^2 to t_n approaches a constant. This indicates the phenomenon is diffusion controlled in the asymptotic limit as encountered in experiments on Liesegang bands.

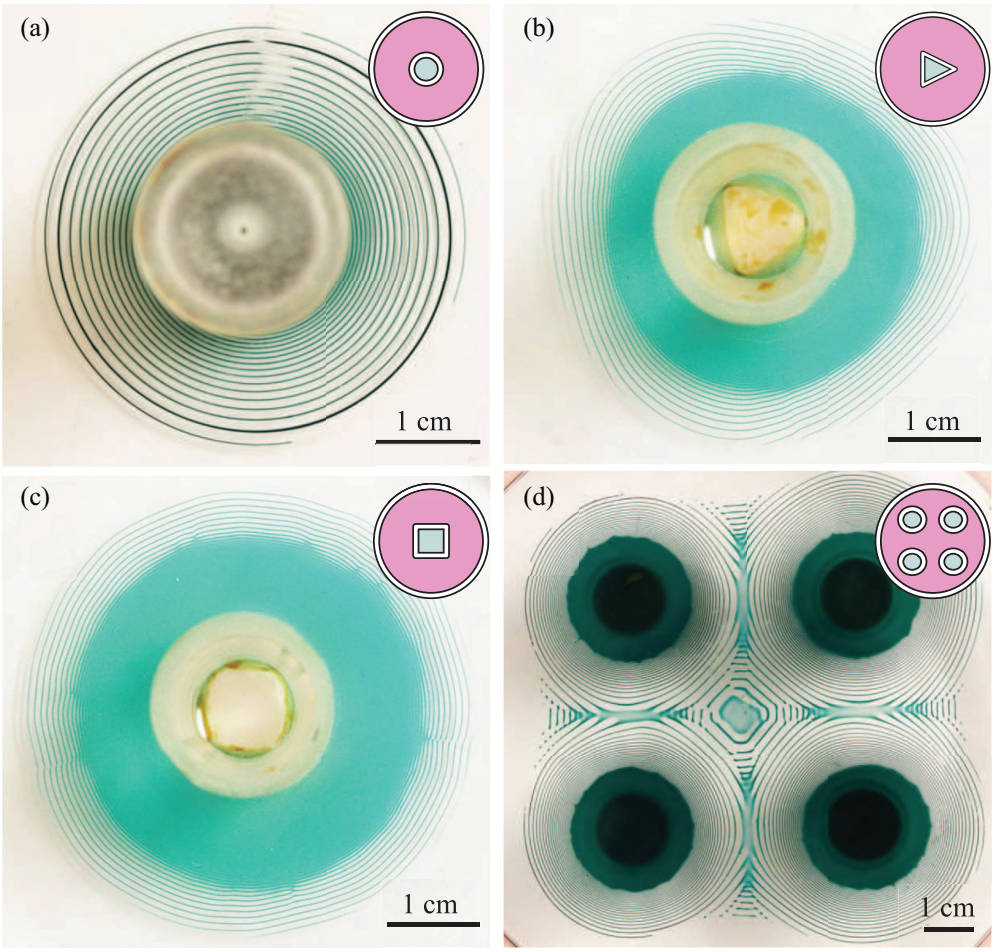

FIG. 12. (Color online) Different precipitation patterns of cobalt hydroxide are obtained with different cross sections of the inner reservoir. The geometry of the setup for each pattern is sketched in the upper corner of each panel. The time at which these patterns are photographed is about 12 hours. (a) Perfect precipitation rings are obtained with a circular cross section. Inner $(Co^{+2}) = 0.2$ M; outer $(NH_3) = 3.7$ M. (b) Elliptic patterns are obtained with a triangular cross section. Inner $(Co^{+2}) = 0.2$ M; outer $(NH_3) = 1$ M. (c) Modulated rings are obtained with a square cross section. Inner $(Co^{+2}) = 0.2$ M; outer $(NH_3) = 1$ M. (d) A more complex pattern is obtained by using 4 circular reservoirs of similar (but not identical) diameters. Inner $(Co^{+2}) = 0.2$ M; outer $(NH_3) = 2$ M.

displayed in Fig. 12, which lead to a spacing coefficient p that varies between 0.05 [Fig. 12(a)], 0.18 [Figs. 12(b), 12(c)], and 0.3 [Fig. 12(d)] depending on the initial concentrations. Furthermore, the value of p was also found to be independent of the geometry for the same initial concentrations, as found in the aforementioned simulations

V. CONCLUSION

We present a numerical algorithm for the simulation of Liesegang patterns on structureless meshes. The method is based on the vertex-based finite volume method. The algorithm is shown to be stable and robust in simulating the Liesegang pattern starting from different initial conditions. The numerical results are shown to be in good agreement with the experiment for various geometrical configurations. Therefore, this method may be used for the prediction and design of complex precipitation patterns.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of grants from the Lebanese Council for Scientific Research (LCNSR) and from the University Research Board, American University of Beirut. They also thank Ms. Manal Ammar for her help on the experimental part.

[1] A. M. Turing, Philos. Trans. R. Soc. London B **237**, 37 (1952).
[2] B. Grzybowski and C. Campbell, Chem. Eng. Sci. **59**, 1667 (2004).
[3] R. L. Liesegang, Photographisches Archiv **37**, 305 (1896).
[4] H. Henisch, *Crystals in Gels and Liesegang Rings* (Cambridge University Press, Cambridge, 1988).
[5] M. Al-Ghoul and R. Sultan, J. Phys. Chem. A **105**, 8053 (2001).
[6] M. Al-Ghoul and R. Sultan, J. Phys. Chem. A **107**, 1095 (2003).

[7] W. Ostwald, *Lehrbuch der Allgemeinen Chemie* (Engelmann, Leipzig, 1897).
[8] I. M. Lifshitz and V. V. Slyozov, J. Phys. Chem. Solid. **19**, 35 (1961).
[9] D. S. Chernavskii, S. C. Muller, and A. A. Polezhaev, Physica D **54**, 160 (1991).
[10] A. A. Polezhaev and S. C. Muller, Chaos **4**, 631 (1994).
[11] S. C. Muller, S. Kai, and J. Ross, J. Phys. Chem. **86**, 4078 (1982).
[12] S. Kai, S. C. Muller, and J. Ross, J. Chem. Phys. **76**, 1392 (1982).
[13] I. L'Heureux, Phys. Lett. A **372**, 3001 (2008).
[14] B. R. Baliga and S. V. Patankar, Numer. Heat Transfer **3**, 393 (1980).
[15] B. R. Baliga and S. V. Patankar, Numer. Heat Transfer **6**, 245 (1983).
[16] Vaughan R. Voller, *Basic Control Volume Finite Element Methods For Fluids And Solids* (World Scientific, Singapore, 2009).
[17] J. R. Shewchuk, Comput. Geom. **22**, 21 (2002).
[18] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice Hall, Upper Saddle River, 1971).
[19] G. D. Byrne and A. C. Hindmarsh, ACM Trans. Math. Softw. **1**, 71 (1975).
[20] K. R. Jackson and R. Sacks-Davis, ACM Trans. Math. Softw. **6**, 295 (1980).
[21] I. Gladwell, L. F. Shampine, and L. F. Thompson, *Solving ODEs with MATLAB* (Cambridge University Press, New York, 2003).
[22] G. Strang, SIAM J. Numer. Anal. **5**, 1 (1968).
[23] P. L. Nash, J. Comput. Phys. **227**, 2073 (2008).
[24] C. T. Kelly, *Iterative Methods for Linear and Nonlinear Equations. Frontiers in Applied Mathematics* (Society for Industrial and Applied Mathematics, Philadelphia, 1995).
[25] J. Geiser, J. Comput. Appl. Math. **217**, 227 (2008).
[26] J. Geiser, in *Numerical Analysis and Scientific Computing Series*, edited by F. Magoules and C. Lai (CRC Press, Boca Raton, 2009).
[27] J. Nocedal and J. S. Wright, *Numerical Optimization* (Springer, New York, 2000).
[28] T. Davis, *Direct Methods for Sparse Linear Systems* (Society for Industrial and Applied Mathematics, Philadelphia, 2006).
[29] Y. Saad, *Iterative Methods for Sparse Linear Systems* (Society for Industrial and Applied Mathematics, Philadelphia, 2003).
[30] Gerard S. B. Lebon, Mayur K. Patel, and Koulis A. Pericleous, J. Algorithm. Comput. Tech. **6**, 129 (2012).
[31] M. Al-Ghoul, H. El-Rassy, T. Coradin, and T. Mokalled, J. Cryst. Growth **312**, 856 (2010).
[32] H. Batlouni, H. El-Rassy, and M. Al-Ghoul, J. Phys. Chem. A Letter **112**, 7755 (2008).
[33] F. E. Lloyd and V. Moravek, Plant Physiol. **3**, 101 (1928).
[34] F. Izsák and I. Lagzi, in *Precipitation Patterns in Reaction-Diffusion Systems*, edited by István Lagzi (Research Signpost, Kerala, India, 2010), pp. 207–217.
[35] R. Matalon and A. Packter, Bull. Soc. Chim. France **4**, 1592 (1923).
[36] M. C. K. Jablczynski, J. Colloid Sci. **10**, 46 (1955).
[37] H. W. Morse and G. W. R. Pierce, Phys. Rev. **17**, 129 (1903).