Marine Algae Stalk Ca^{2+} Regeneration Using Mimetic Differences

Mohsin A. Mohammed and Miguel A. Dumett

April 9, 2024

Publication Number: CSRCR2024-02

Computational Science & Engineering Faculty and Students Research Articles

Database Powered by the Computational Science Research Center Computing Group & Visualization Lab

COMPUTATIONAL SCIENCE & ENGINEERING

SAN DIEGO STATE UNIVERSITY

Computational Science Research Center
College of Sciences
5500 Campanile Drive
San Diego, CA 92182-1245
(619) 594-3430

Marine Algae Stalk Ca^{2+} Regeneration Using Mimetic Differences

Mohsin A. Mohammed * and Miguel A. Dumett †‡

April 9, 2024

Abstract

In this study, we simulate a model for the whorl formation in marine algae, focusing specifically on the influence of external calcium concentrations. At the heart of our investigation is the application of the mimetic operators. Mimetic Operators Library Enhanced library, a sophisticated numerical tool designed for solving partial differential equations with high accuracy and efficiency.

1 Introduction

Marine algae, like Acetabularia acetabulum, are single-celled organisms that can grow quite large and have complex shapes.

These algae are able to regenerate, or grow back parts that are removed. If the top "umbrella" is taken off, the algae can completely regrow it thanks to a special part in its base, called the nucleus, where all its genetic information is stored. Even if a piece from its stalk is cut, it can grow a new top again.

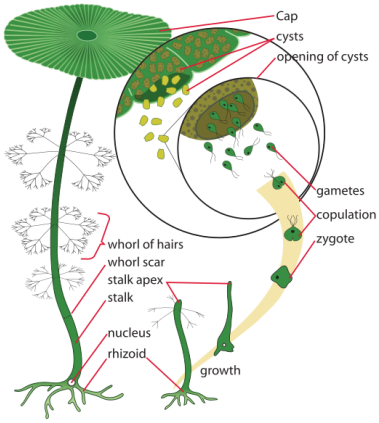


Figure 1: Algae Structure and it's body parts (taken from Wikipedia)

This study focuses on reproducing with the Corbino-Castillo mimetic operators [2] via the utilization of the Mimetic Operators Library Enhanced (MOLE) library [1], a particular aspect of the algae's growth: the formation of ring-like structures along its stalk, influenced by the amount of calcium in the water [4].

2 Related Work

The marine alga Acetabularia acetabulum is a good example for understanding how morphogenesis occurs when a single-cell organism develops. Differential equation models that exclude genetics and intracellular signaling have been proposed. The pattern development is caused by the system's Turing instability (see

*Computational Science Master's Program at San Diego State University (mmohammed5956@sdsu.edu).
†Editor: Jose E. Castillo
‡Computational Science Research Center at San Diego State University (mdumett@sdsu.edu).

[4]), applies Murray's model, which is a purely chemical model and reaction-diffusion equation involving two reactants.

The effect of calcium concentration on the start of the whorl pattern requires a minimal amount of calcium concentration [3]. In addition, whorls will stop developing once the external calcium concentrations reach certain upper bound. These facts can be clearly seen in the proposed model. More specifically, it is possible to obtain a specific interval of calcium concentration for the whorl pattern to occur, were several transitions type can occur though in the case of thin stem wall only continuous type and catastrophic type can happen [5]. Different interesting dynamical behavior are described by the theory given in [5] and found numerically by [3].

Our purpose is to be able to replicate numerically all findings in [3] but utilizing mimetic differences and the MOLE library developed at San Diego State University [1].

3 Model Equation

This is a model proposed by Murray [4], but also numerically investigated by [3], which is an adaptation of a simple two-species Turing mechanism. The equations are:

$$\begin{aligned} \frac{\partial A}{\partial t} &= D_A\,\Delta A + k_1 - k_2 A + k_3 A^2 B, \\ \frac{\partial B}{\partial t} &= D_B\,\Delta B + k_4 - k_3 A^2 B, \end{aligned}$$

where k_i, $i = 1, 2, 3, 4$, $D_A, D_B > 0$ and A and B are functions of r, θ, and t with the annulus domain defined by

$$R_i \le r \le R_0, \qquad 0 \le \theta \le 2\pi, \qquad R_i = 1, \qquad R_0 = 1.5.$$

A and B define the density of two substances inside the annular growth region at the top mature Acetabularia.

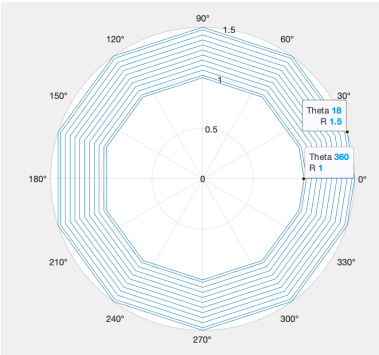


Figure 2: This is our model domain defined by R and θ in the blue color region

Here the assumption is that a reaction $2\,\mathrm{A} + \mathrm{B} \xrightarrow{\mathrm{k_3}} 3\,\mathrm{A}$ took place.

Basically, B is calcium, and A could be a molecule that is increased in the calcium presence but are constantly transformed to other substances at a rate of k_2. And they are generated by constant rate k_1, k_4, respectively, for A and B.

The model is non-dimensionalized via

$$u = A\left(\frac{k_3}{k_2}\right)^{1/2},\ v = B\left(\frac{k_3}{k_2}\right)^{1/2},\ t^* = \frac{D_A t}{R_i^2},\ x^* = \frac{x}{R_i},\ d = \frac{D_B}{D_A},\ a = \frac{k_1}{k_2}\left(\frac{k_3}{k_2}\right)^{1/2},\ \lambda = \frac{k_4}{k_2}\left(\frac{k_3}{k_2}\right)^{1/2},\ R^2 = \frac{R_i^2 k_2}{D_A}.$$

and becomes (where all stars are omitted to facilitate the formulas manipulation),

$$\begin{aligned} \frac{\partial u}{\partial t} &= D\,u\,\Delta A + R^2(a - u - u^2 v) \quad \text{in } \Omega, \\ \frac{\partial v}{\partial t} &= d\,D\,v\,\Delta B + R^2(\lambda - u^2 v) \quad \text{in } \Omega, \\ u_r &= v_r = 0, \quad \text{on } \partial\Omega, \end{aligned}$$

with

$$\Delta u = \frac{\partial^2 u}{\partial r^2} + r^{-1}\frac{\partial u}{\partial r} + r^{-2}\frac{\partial^2 u}{\partial \theta^2}$$

After substituting the Laplacian in polar coordinates, the partial differential equation (PDE) system is given by,

$$\begin{aligned} u_t &= R^2(a - u + vu^2) + \frac{\partial^2 u}{\partial r^2} + r^{-1}\frac{\partial u}{\partial r} + r^{-2}\frac{\partial^2 u}{\partial \theta^2} \\ v_t &= R^2(b - vu^2) + d\frac{\partial^2 v}{\partial r^2} + r^{-1}\frac{\partial v}{\partial r} + r^{-2}\frac{\partial^2 v}{\partial \theta^2} \end{aligned}$$

4 Solving the PDE

The MOLE library directly approximates the Cartesian Laplacian operator with single command but in our case, we have to slice the Laplacian in both r and θ because the second-order derivatives with respect to r and θ appear separately in polar coordinates and also because of the presence of r^{-2}, r^{-1} terms being multiplied. Since the matrix representation of the mimetic Laplacian operator is defined as the product of the matrix representations of both the divergence and the gradient operators, we need to exhibit the latter.

4.1 Divergence

The mimetic divergence is given by

$$\begin{aligned} D \cdot \mathbf{F} &= \nabla \cdot \mathbf{F} \\ D \cdot \mathbf{F} &= \nabla \cdot \mathbf{F} = \frac{\partial F_r}{\partial r} + \frac{\partial F_\theta}{\partial \theta} \\ D \cdot \mathbf{F} &= \begin{bmatrix} | & | \\ D_r & D_\theta \\ | & | \end{bmatrix} \begin{bmatrix} — & F_r & — \\ — & F_\theta & — \end{bmatrix} \end{aligned}$$

4.2 Gradient

The mimetic gradient is given by

$$\begin{aligned} G &= \nabla F(r,\theta) = \frac{\partial F}{\partial r}\hat{i} + \frac{\partial F}{\partial \theta}\hat{j} \\ G &= \begin{bmatrix} — & G_r & — \\ — & G_\theta & — \end{bmatrix} \end{aligned}$$

4.3 Laplacian

$$\begin{aligned} L &= DG \\ L &= \nabla^2 = \nabla \cdot \nabla \\ L &= \nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{\partial^2}{\partial \theta^2} \end{aligned}$$

4.4 Initial Conditions

The initial conditions are given in terms of Bessel functions:

$$\begin{aligned} u_0 &= a + b + \epsilon(J_1(r)Y_1^{'}(1) - J_1^{'}(\delta)Y_1(\delta))\cos\theta, \\ v_0 &= \frac{b}{(a+b)^2} + \epsilon(J_1(r)Y_1^{'}(1) - J_1^{'}(\delta)Y_1(\delta))\cos\theta, \\ J_\alpha(x) &= \sum_{m=0}^{\infty} \frac{(-1)^m}{m!\,\Gamma(m+\alpha+1)}\left(\frac{x}{2}\right)^{2m+\alpha}, \\ Y_\alpha(x) &= \frac{J_\alpha(x)\cos(\alpha\pi) - J_{-\alpha}(x)}{\sin(\alpha\pi)}. \end{aligned}$$

4.5 Mimetic discrete boundary and boundary conditions

The mimetic discrete analog of the spatial derivatives of

$$\begin{aligned} u_t &= R^2(a-u+vu^2)+\frac{\partial^2 u}{\partial r^2}+r^{-1}\frac{\partial u}{\partial r}+r^{-2}\frac{\partial^2 u}{\partial \theta^2} \\ v_t &= R^2(b-vu^2)+d\left(\frac{\partial^2 v}{\partial r^2}+r^{-1}\frac{\partial v}{\partial r}+r^{-2}\frac{\partial^2 v}{\partial \theta^2}\right) \end{aligned}$$

utilizing the mentioned discrete analogs as well as convenient interpolation operators together with the time-derivative discretization for u_t produce

$$\begin{aligned} u_t &= R^2(diag(a)-u+vu^2)+D_rG_ru+r^{-1}I^GG_ru+r^{-2}D_\theta G_\theta u \\ \frac{u_{new}-u_{old}}{dt} &= R^2(diag(a)-u_{new}+diag(v_{old})diag(u_{old})u_{new}) \\ &+ D_rG_ru_{new}+diag(r^{-1})I^GG_ru_{new}+diag(r^{-2})D_\theta G_\theta u_{new} \\ u_{new}-u_{old} &= dtR^2diag(a)+dt(-R^2I+R^2diag(v_{old})diag(u_{old}) \\ &+ D_rG_r+diag(r^{-1})I^GG_r+diag(r^{-2})D_\theta G_\theta)u_{new} \end{aligned}$$

Let

$$X=-R^2I+R^2diag(v_{old})diag(u_{old})+D_rG_r+diag(r^{-1})I^GG_r+diag(r^{-2})D_\theta G_\theta$$

$$\begin{aligned} u_{new}-u_{old} &= dtR^2diag(a)+dtXu_{new} \\ (I-dtX)u_{new} &= dtR^2diag(a)+u_{old} \\ (I-dtX+BC)u_{new} &= dtR^2diag(a)+u_{old} \end{aligned}$$

where BC stands for boundary conditions.

Similarly for v_t,

$$\begin{aligned} v_t &= R^2(b-vu^2)+d(D_rG_rv+r^{-1}I^GG_rv+r^{-2}D_\theta G_\theta v) \\ \frac{v_{new}-v_{old}}{dt} &= R^2(diag(b)-diag(u_{new}^2))v_{new}+d\left(D_rG_rv_{new}+diag(r^{-1})I^GG_rv_{new}+diag(r^{-2})D_\theta G_\theta v_{new}\right) \\ v_{new}-v_{old} &= dt\,R^2diag(b)+dt(-R^2diag(u_{new}^2)+d(D_rG_r+diag(r^{-1})I^GG_r+diag(r^{-2})D_\theta G_\theta))v_{new} \end{aligned}$$

Let

$$Y=-R^2diag(u_{new}^2)+d(D_rG_r+diag(r^{-1})I^GG_r+diag(r^{-2})D_\theta G_\theta)$$

$$\begin{aligned} v_{new}-v_{old} &= dt\,R^2diag(b)+dt\,Yv_{new} \\ (I-dtY)v_{new} &= dt\,R^2diag(b)+v_{old} \\ (I-dtY+BC)v_{new} &= dt\,R^2diag(b)+v_{old} \end{aligned}$$

We are using the Neumann boundary conditions.

We apply the MATLAB inverse matrix command in the above functions. We are using interpolation because directly applying gradient will make the calculations at faces instead of centers but we need it to centers so one can apply the divergence later.

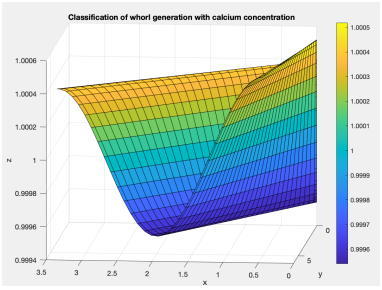
After factorising and applying boundary conditions, the code is ran a time loop from 0 to 2 with a time step of 1e-3.

5 Results

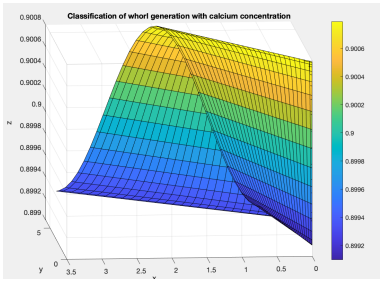
The first two panels of the figure on top, exhibit a snapshot of the evolution of concentrations of chemical species A and B (non-dimensionalized) in Cartesian coordinates. The last panel displays a plot in polar coordinates of them.

On the other hand, Figures 4

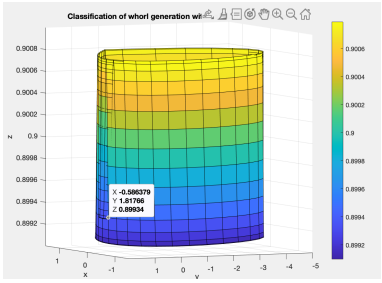
In addition, Figure 5 depicts, utilizing MATLAB command surf, the growing of the whorl hair.



(a) Cartesian plot of U



(b) Cartesian plot of V



(c) Polar plot using surf

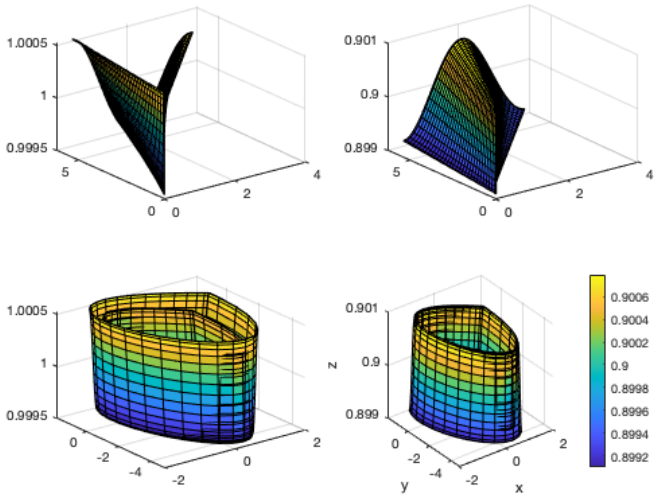


Figure 4: The different variables at time 2.

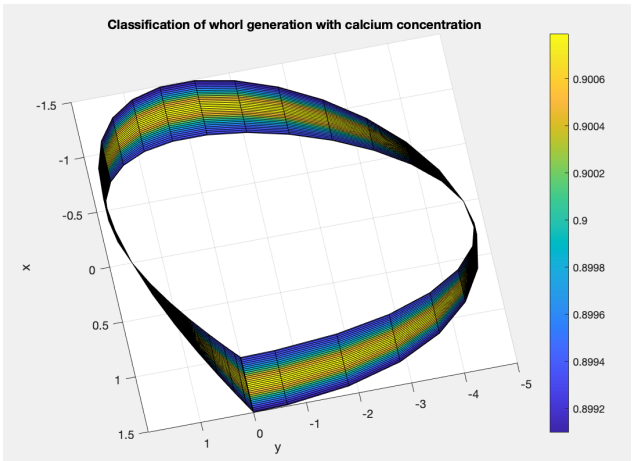


Figure 5: Polar plot using surf, yellow color regions depicts that whorl hair growth when reacted with outside calcium concentration

6 Conclusion

In this project a system of chemical reaction-diffusion equations is solved using the MOLE library. Analyzed the two reactants dynamic transitions with respect to the whorl formation of marine algae when it is caused by outside calcium concentration on its thin annulus.

We share our MATLAB code to facilitate the readers to reproduce our findings.

7 Code

```
clc
close all
clear

addpath('../mole_MATLAB')

%% Spatial discretization
k = 2;          % Order of accuracy (spatial)
m = 30;         % Number of cells in r direction
n = 30;         % Number of cells in theta direction
r1 = 1;
delta = 1.5;
r2 = delta;
theta1 = 0;
theta2 = 2*pi;
dr = (r2-r1)/m;  % Step length in r direction
dtheta = (theta2-theta1)/n; % Step length in theta direction

%% 2D Staggered grid
r_grid = [r1 r1+dr/2 : dr : r2-dr/2 r2];
theta_grid = [theta1 theta1+dtheta/2 : dtheta : theta2-dtheta/2 theta2];

[r_pts, theta_pts] = meshgrid(r_grid, theta_grid);
x = r_pts.*cos(theta_pts);
y = theta_pts.*sin(theta_pts);

[x1, y1] = pol2cart(r_pts, theta_pts);

%% Parameters
a = 0.1;
b = 0.9;
d = 9;
R = 3.45;
epsilon = 1e-2;

%% Simulation time
t_end = 0.5;
dt = 0.01;

%% Bessel functions calculation
Jn = besselj(1, r_grid);
Yn = besselj(2, r_grid);
J1 = besselj(1, r_pts);
Y1 = Yn(end);
DJn = diff(Jn) / dr;
DYn = diff(Yn) / dr;
DJ_delta = DJn(end);
DY1 = DYn(1);


```

```
%% IC
U0 = a + b + epsilon*((J1*DY1)-(DJ_delta*DY1)).*cos(theta_grid);
V0 = (b/(a+b)^2) + epsilon*((J1*DY1)-(DJ_delta*DY1)).*cos(theta_grid);

U_old = U0(:);
V_old = V0(:);

%% Equation discretization
D = div2D(k, m, dr, n, dtheta);
G = grad2D(k, m, dr, n, dtheta);

Dr = D(:,1:n*(m+1));
D_theta = D(:,n*(m+1)+1:end);

Gr = G(1:n*(m+1), :);
G_theta = G(n*(m+1)+1: end, :);

r_pts = r_pts(:);
theta_pts = theta_pts(:);

IG = interpolFacesToCentersG2D(k, m, n); % interpGMat2D(k, m, n);
I_Gr = IG(1:size(G,2),1:n*(m+1)); % IG(:,1:n*(m+1));

%% BC
BC_term_eq = robinBC2D(k, m, dr, n, dtheta, 0, 1);

%% Time loop
for t = 0 : dt : t_end
    X = R^2 * diag(V_old) * diag(U_old)                ...
       + (Dr * Gr) + (diag(1./r_pts) * I_Gr * Gr)  ...
       + (diag(1./r_pts.^2) * D_theta * G_theta)   ...
       - R^2*diag(length(U_old));
    X = eye(length(U_old), length(U_old)) - dt*X + BC_term_eq;
    U_new = X\((dt*R^2*a*ones(length(U_old),1))+U_old);

    Y = (d*(Dr*Gr + diag(1./r_pts) * I_Gr * Gr)  ...
        + (diag(1./r_pts.^2) * D_theta * G_theta))  ...
        - R^2* diag(U_new) * diag(U_old);
    Y = eye(length(V_old), length(V_old)) - dt * Y + BC_term_eq;
    V_new = Y\(dt*R^2*b*ones(length(V_old),1)+V_old);

    U_old = U_new;
    V_old = V_new;

    subplot(2,2,1);
    surf(x1,y1,reshape(U_new, m+2, n+2));

    subplot(2,2,2);
    surf(x1,y1,reshape(V_new, m+2, n+2));

    subplot(2,2,3);
    surf(x,y,reshape(U_new, m+2, n+2));
    subplot(2,2,4);
    surf(x,y,reshape(V_new, m+2, n+2));

```

```
106     xlabel('x')
107     ylabel('y')
108     zlabel('z')
109
110     colorbar
111     drawnow
112
113     pause(0.1);
114 end
```

References

[1] Corbino, J., and Castillo, J.E., MOLE: Mimetic Operators Library Enhanced: The Open-Source Library for Solving Partial Differential Equations using Mimetic Methods, 2017.

[2] Corbino, J., and Castillo, J.E. High-order mimetic finite-difference operators satisfying the extended Gauss divergence theorem. In: Journal of Computational and Applied Mathematics 364 (2020), p. 112326. issn: 0377-0427. doi: https://doi.org/10.1016/j.cam.2019.06.042. url:

https://www.sciencedirect.com/science/article/pii/S0377042719303231.8`\item` command.

[3] Mao, Y., Yan, D., and Lu, C.H., Dynamic transitions and stability for the acetabularia whorl formation. arXiv:1810.10120 [math.AP]

[4] Murray, J.D., Mathematical biology II: Spatial models and biomedical applications, 3rd edition, Springer-Verlag, New York, 2001.

[5] Ma, T., and Wang, S., Phase transition dynamics, Springer, 2014.