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

Mohsin A. Mohammed and Miguel A. Dumett

April 9, 2024

Publication Number: CSRCR2024-02

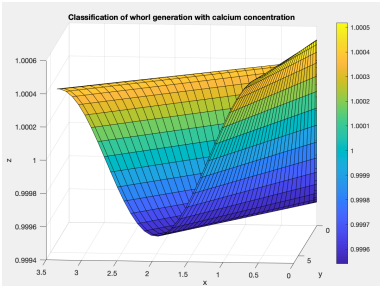
Computational Science & Engineering Faculty and Students Research Articles

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

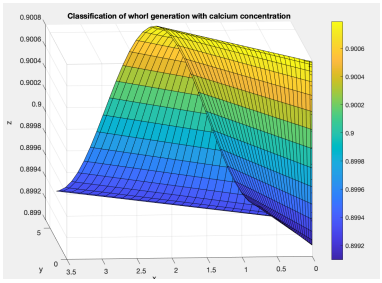
COMPUTATIONAL SCIENCE & ENGINEERING

SAN DIEGO STATE UNIVERSITY

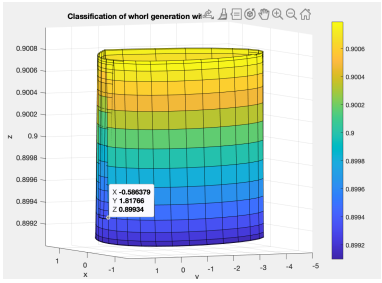
Computational Science Research Center
College of Sciences
5500 Campanile Drive
San Diego, CA 92182-1245
(619) 594-3430



(a) Cartesian plot of U



(b) Cartesian plot of V



(c) Polar plot using surf

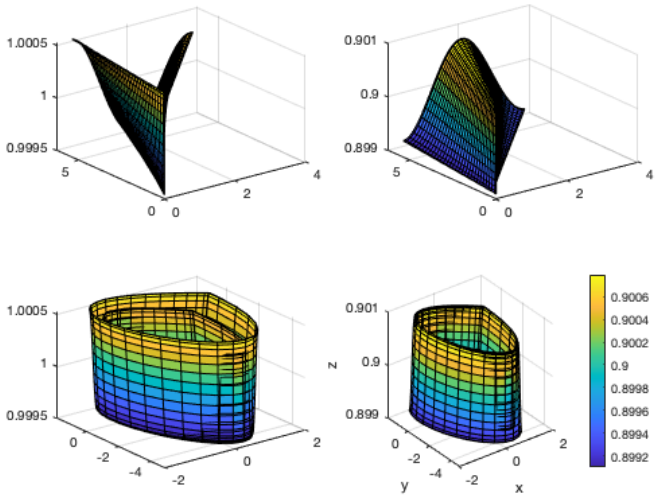


Figure 4: The different variables at time 2.

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.