# Modeling Wound Healing With Mimetic Differences

Harshith Das and Miguel A. Dumett

October 10, 2024

**Publication Number: CSRCR2024-08**

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

# Modeling Wound Healing With Mimetic Differences

Harshith Das* Miguel A. Dumett †‡

October 10, 2024

**Abstract**

In this work, wound healing is modeled by the two-dimensional Fisher equation and its discrete analog is achieved by utilizing mimetic differences. This report aims to solve the two-dimensional Fisher-Kolmogorov-Petrovsky-Piskunov equation (or Fisher-KPP equation) in the context of modeling the cell movements as they close a wound made in an epithelial sheet. The family of diffusion-reaction equations, of which the Fisher equation is a part, are commonly used to depict spatial changes in moving cell populations. In particular, the two-dimensional Fisher-KPP equation has been evaluated as suitable for describing cell movement [4] when paired with experimental data. This report uses the experimental data gathered by Habbal et al. [2], a binary image resulting from image captures of a 2d cell sheet made up of Madin-Darby canine kidney (MDCK) cells. The methodology used demonstrates the efficacy of mimetic differences [1] for solving PDEs that span both temporal and spatial domains, with the particular advantages granted by the calculation of the entire image grid simultaneously, rather than traditional methods that rely greatly on solutions from previous time-steps.

## 1 Introduction

Accurate and reliable modeling of cell movement is the first step to establishing a baseline from which advanced predictive applications can be built. With the establishment of normal cell, machine learning models can be trained to spot abnormal cell growth in laboratory settings, such as pharmacological drug development, in order to judge efficacy or detect undesired side effects.

Furthermore, the methodology described in this report demonstrates a generic process of solving image-based partial differential equations using the MOLE library, serving as a proof of concept for applications beyond cell movements and diffusion.

In comparison to explicit Euler method used in the original paper, this report uses a Mimetic Difference Method as defined in the MOLE library [3], making use of a staggered grid that corresponds to the pixels of a given input image.

## 2 Equation and Initial Conditions

### 2.1 The 2D Fisher-KPP Equation

The Fisher-KPP equations is a semilinear parabolic partial differential equation (PDE) which models the interaction of Fickian diffusion with a logistic type growth term. It is given by

$$\frac{\partial u}{\partial t} = D\Delta u + ru(1-u) \tag{1}$$

with Dirichlet and Neumann boundary conditions on the PDE domain (a rectangle containing the bound). Several parameters have been proposed but this work will consider the following values [2]

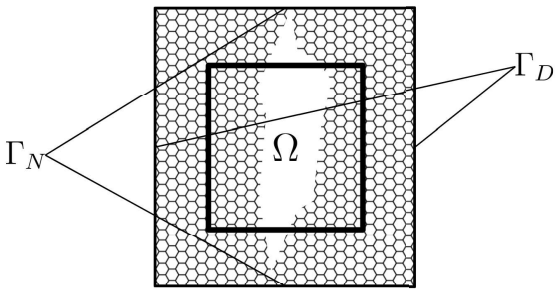
$$\begin{aligned} D &= 2.48 \times 10^{-2}, \\ r &= 0.070, \end{aligned}$$

*Computational Science Master's Program at San Diego State University (hdas0821@sdsu.edu).
†Editor: Jose E. Castillo
‡Computational Science Research Center at San Diego State University (mdumett@sdsu.edu).

chosen from experimental data.

## 2.2 Initial and Boundary Conditions

The domain set by Habbal et al. [2] describes a rectangle $\Omega$ with Dirichlet boundary conditions on its vertical sides and Neumann boundary conditions on horizontal sides as described by the diagram below, where the size of the image is 85x85 pixels.

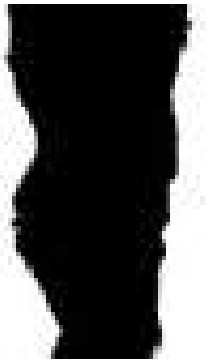



- $\Gamma_N$: top and bottom sides
- $\Gamma_D$: left and right sides

A contiguous cell sheet will always have cells present on the vertical sides, hence the Dirichlet condition. For the sake of the model, no flux is assumed to occur across the horizontal sides on the domain, resulting in the homogeneous Neumann condition.

## 2.3 The dataset

The initial condition of the cell sheet is depicted by the following image:



In this image, white pixels (corresponding to an array value of 1) represent areas where cells are present, while black pixels (valued 0 in the array) are areas where the wound has been simulated by scraping away cells.

### 2.4 The mimetic difference discrete analog of the 2D Fisher-KPP

In order to apply the the Mimetic Difference library to the Fisher equation, it was first transformed into its Mimetic counterpart by the following steps:

1. Original Equation

$$\frac{\partial u}{\partial t} = D\Delta u + ru(1-u)$$

2. Replacing differential operators

$$U_t = DLU + r(1-U)U \tag{2}$$

3. Matching output domains

$$\frac{U^{n+1} - U^n}{dt} = DLU^n + r\text{diag}(I - U^n)U^n \tag{3}$$

4. Performing elementwise products

$$U^{n+1} = [I + dt \cdot DL + r \cdot dt \cdot diag(1 - U_n)]U^n \tag{4}$$

## 3 Image Processing

Without the original binarized images used in the paper's calculations, the figures present were instead processed by thresholding, as the grayscale assay images were in the source publication.

Due to the absence of more images, the following synthesized images were used as initial conditions. Habbal et al [2] notes error in the form of transverse oscillation in places where wound edges meet, as such images 2 and 3 were chosen to examine the effect in Mimetic solution. Image 2 presents an immediate point of meeting, and so tests for oscillations when the edges meet at the beginning of the temporal domain. Image 3 presents two points of meeting, at the top and bottom of the spatial domain, in order to examine the effect of multiple points of meeting early in the time discretization.

Figure 1

(a) Synthesized image 1 (b) Synthesized image 2 (c) Synthesized image 3

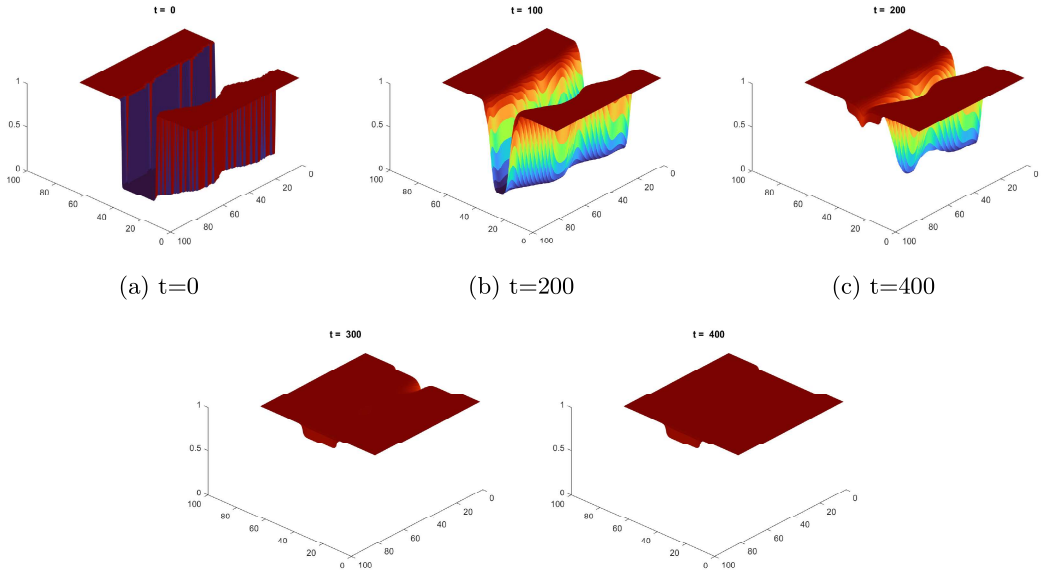
## 4 Results

### 4.1 Solution

Applying the Mimetic method processes the entire grid in one iteration, which avoids using spatial steps. As a result the Mimetic solution does not display the transverse oscillation of wound walls when solved via methods such as Implicit Euler or Crank-Nicholson.

However, black pixels remain along the original wound margins along the top and bottom boundaries due to the Neumann boundary condition enforcing no flux over the border, whereas in an experimental setting the close of cells outside the observed area would converge as well, creating a fully continuous image of only white pixels.

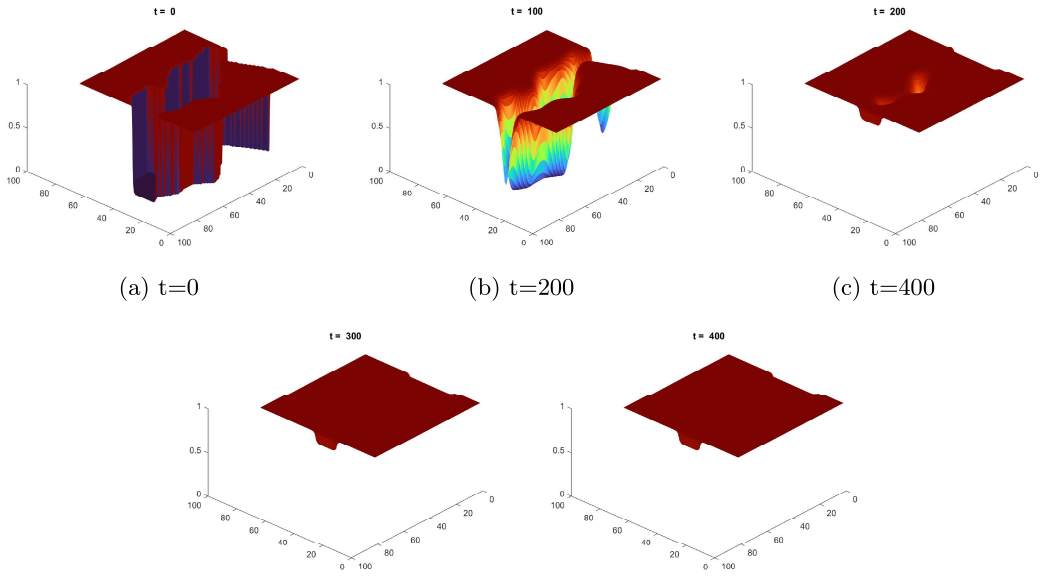
Figure 2: Image 1




(a) t=0 (b) t=200 (c) t=400

(d) t=600 (e) t=800

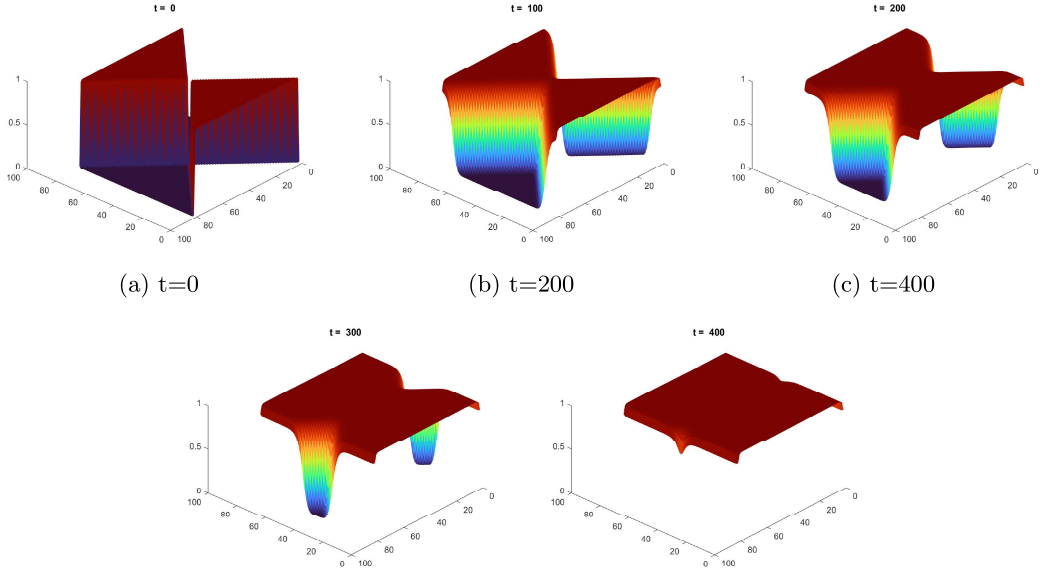
Figure 3: Synthesized Image 1




(a) t=0 (b) t=200 (c) t=400

(d) t=600 (e) t=800

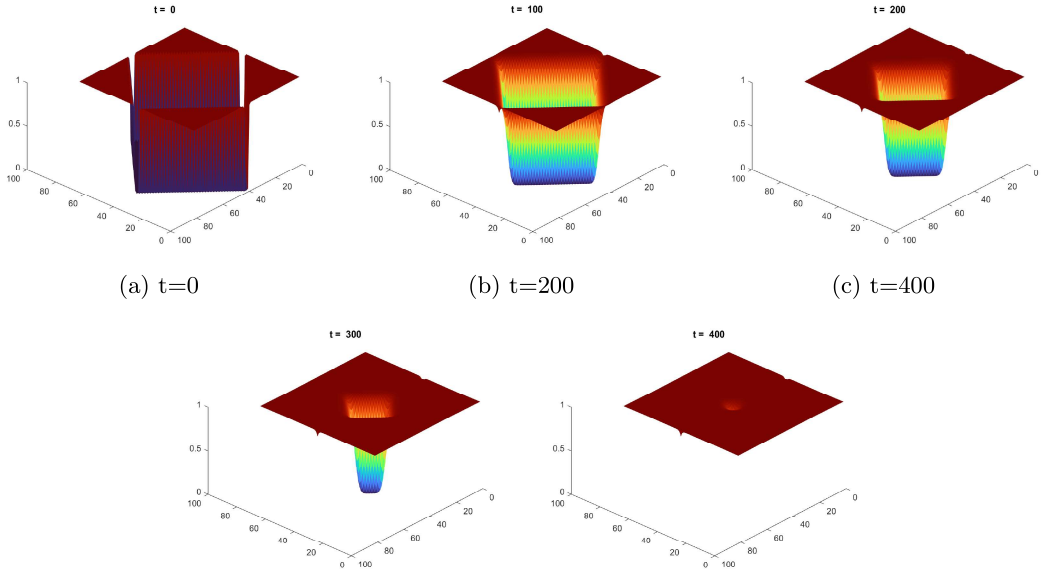
Figure 4: Synthesized Image 2




(a) t=0 (b) t=200 (c) t=400

(d) t=600 (e) t=800

Figure 5: Synthesized image 3




(a) t=0 (b) t=200 (c) t=400

(d) t=600 (e) t=800

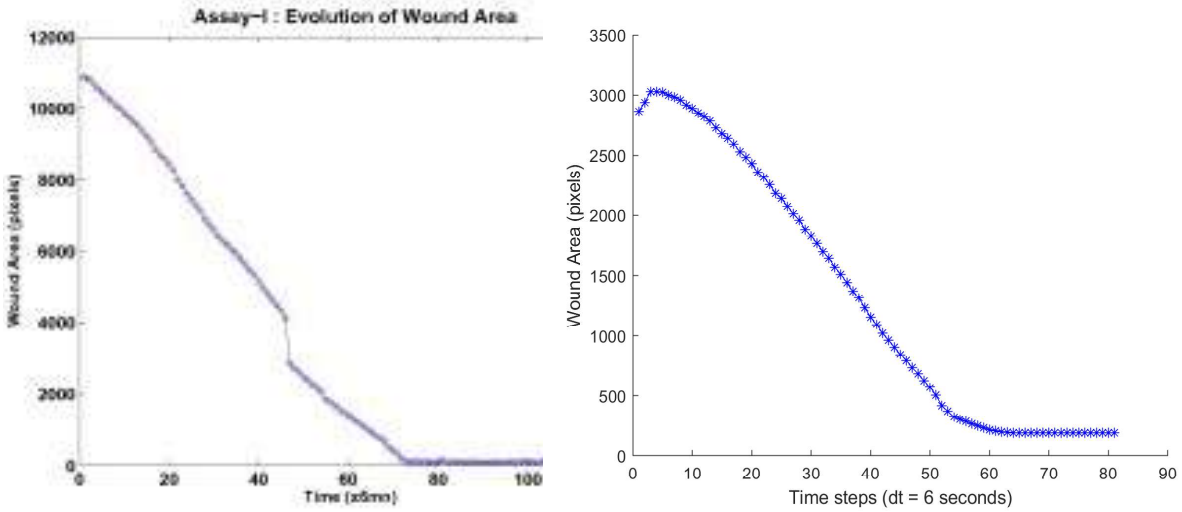
## 4.2 Evaluation

Due to the inclusion of the nonlinear term $ru(1-u)$, finding an analytical solution to compare error can be challenging. Comparing the model to experimental data can serve as a baseline for error calculation. Without access to experimental data, trends in the solution were used to evaluate the solution.

The following figure shows the trends in wound area sourced from Habbal et al. [2] and from the Mimetic solution. Both display a steady decrease. The mimetic approximation displays an anomaly near the beginning which describes an increase in wound size, which stems from the wound area calculation that parses the darker area behind the leading edge as part of the wound itself, and should be disregarded.

## 4.3 Conclusions

As a toolkit, the MOLE library provides a flexible solution to solving PDEs with multi-modal input data. The code attached to this report describes the process of using an image-based initial condition to create image-based solutions over a time discretization. The solution is generated without excessive computational demand, and avoids the oscillating error encountered by iterative methods that base solutions on results from previous time-steps. The MATLAB code also demonstrates a method of capturing the image solution as an animated GIF format for ease of viewing, a process made simple by the Mimetic method of solving that solves entire solution grids in a simple time-steps, resulting in a solution format that can be easily discerned by sight.




(a) Experimental Wound Area

(b) Mimetic Wound Area

Figure 6

# 5 Code

```
% Initialization
close all;
clear all;
clc;
addpath('C:\Users\hkain\Desktop\College\Grad\Spring 2024\COMP 670\mole_MATLAB');
fileN = 'MYGIF.gif';  % This is the animation recorded .gif file name
H=figure('Position', [ 200 200 600 600]);

% Read image and find dimensions - M x N
im1 = imread("pic4.png");
% Initialize variable for later use
m = length(im1(:,1));
n = length(im1(1,:,1));
k=2;
d_param = 2.48 * (10^-2);
r_param = 0.070;
a=0; b=m;
c=0; d=n;
dx = (b-a)/m;
dy = (d-c)/n;
dt = 0.5;
% Threshold image to make B&W, then convert to doubles
im_bw = im2bw(im1);
im_bw = double(im_bw);
im_bw = padarray(im_bw, [1 1], 1, 'both');
im_bw(1,:) = im_bw(2,:);
im_bw(end,:) = im_bw(end-1,:);
iter = 1;
wound_area = zeros(80,1);

% 2D Mimetic operators
L = lap2D(k,m,dx,n,dy);
u = im_bw;
i_vec = eye((m+2)*(n+2));

% Time loop
for t=0:dt:dt*800

    u_vec = reshape(u, [], 1);
    u1_diag = diag(ones(size(u_vec))-u_vec);              % component for next step
    u_next_vec = (i_vec + dt*d_param*L + dt*r_param*u1_diag) * u_vec;
    u_next = reshape(u_next_vec,[m+2,n+2]);

    % impose boundary conditions
    % left and right sides (Dirichlet)
    u_next(:,1) = 1;
    u_next(:,end) = 1;
    % top and bottom sides (Neumann)
    u_next(1,:) = u(1,:);
    u_next(end,:) = u(end,:);
    imnext = u_next;

    % Capture solution as gif every 50 iterations
    if (mod(t, dt*100) ==0)
        figure;
        hold on;
        im_view = imnext(3:end-2, 3:end-2);
        s = surf(im_view);
        s.EdgeColor = 'none';
        title(['t =  ',num2str(t)])
        zlim([0,1])
        colormap turbo;
        clim([0,1]);
        campos([-371,509,7])
        hold off;
        %colorbar;
        pause(0.1);
```

```
68
69
70          % Capture the plot as an image
71           Frame = getframe(gcf);
72           im = frame2im(Frame);
73           [imind, CM] = rgb2ind(im,256);
74
75          % Capture a still frame of the gif for printing
76          rez = [170 170]; %set desired [horizontal vertical] resolution
77          cap = print(gcf,'-RGBImage','-r100'); %capture RGB image (a bit slow)
78
79          % Write the animation to the gif File
80          if t == 0
81              imwrite(imind, CM,fileN,'gif', 'Loopcount',inf);
82          else
83              imwrite(imind, CM,fileN,'gif','WriteMode','append');
84          end
85      end
86
87      % Capture wound area every 10 iterations
88      if (mod(t, dt*10) ==0)
89          unext_bin = imbinarize(u_next,0.9);
90          wound_area_n = sum(sum(unext_bin<0.5));
91          wound_area(iter) = wound_area_n;
92          iter = iter+1;
93      end
94
95      u = (imnext);
96  end
97
98  %%
99  % Plot wound area graph
100 figure;
101 hold on;
102 plot(wound_area,'*-b');
103 xlabel('Time steps (dt = 6 seconds)')
104 ylabel('Wound Area (pixels)')
```

# References


[1] Corbino, J., and Castillo, J.E., High-order mimetic difference operators satisfying the extended Gauss divergence theorem, Journal of Computational and Applied Mathematics, 364 (2020).

[2] Habbal, A., Barelli, H., and Malandain, G., Assessing the ability of the 2D Fisher–KPP equation to model cell-sheet wound closure, Mathematical Biosciences, Vol. 252, 2014, pp 45-59, 0025-5564.

[3] Corbino, J., MOLE: Mimetic Operators Library Enhanced: The Open-Source Library for Solving Partial Differential Equations using Mimetic Methods, 2017. GitHub, `https://github.com/csrc-sdsu/mole`.

[4] Sengers, B.G., Please, C.P., and Oreffo, R.O., Experimental characterization and computational modelling of two-dimensional cell spreading for skeletal regeneration, Journal of the Royal Society, Interface, 4(17), 1107–1117, 2007, `https://doi.org/10.1098/rsif.2007.0233`