Reaction-Diffusion Models for the Analysis of Molecular Biological Interactions

April 29, 2011
Marzuola MATH 564

Timothy Palpant
PID: 714225766

Contents

1 Background

Cells synthesize proteins, which in turn comprise much of the cell's structure, catalyze reactions, and effect transcriptional changes within the nucleus. Understanding the role that proteins play in regulating gene expression is of utmost importance to understanding cellular behavior in response to the environment and during development, as well as gene mis-regulation during diseases such as cancer. For this reason, several techniques have been developed to experimentally measure and quantify the interactions of proteins with DNA in the nucleus. Oftentimes, raw data is collected through fluorescence microscopy, and mathematical models are required to parse the results into meaningful biological information. Here, we present an overview of one such technique, Fluorescence Recovery After Photobleaching (FRAP), and implement several related incarnations of FRAP models. We then fit these numerical simulations to an example FRAP data set.

1.1 Kinetics of Nuclear Binding Proteins

At the level of the cell nucleus (typically $10-100\mu m$), transport of proteins is primarily accomplished through diffusion at no cost to the cell. Therefore, diffusion models are often used when seeking to model the kinetics of intranuclear proteins. As DNA-binding proteins diffuse through the nucleus, their movement is retarded by binding to relatively immobile DNA. This reaction can be thought of as a first order reaction of a single species between two states:

$$unbound \rightleftharpoons bound \tag{1}$$

In the unbound state, proteins diffuse freely at a rate governed by the viscosity of the nuclear fluid and the molecular weight of the protein. In the bound state, proteins are immobile. Therefore, the kinetics of binding can be understood by quantifying the degree to which observed diffusion is restricted for a given protein. This observed diffusion is often called the "effective diffusion" of a protein.

1.2 FRAP - Fluorescence Recovery After Photobleaching

In FRAP experiments, a DNA-binding protein of interest is fluorescently tagged so that its distribution throughout the nucleus can be observed by microscopy. At biological concentrations, fluorescence (f) is approximately directly proportional to the concentration (c) of protein

$$f \propto c \tag{2}$$

and is therefore a useful proxy for determining protein concentrations. To record diffusion of a protein throughout the nucleus, a time series of images is recorded. The equilibrium distribution of fluorescence is then disrupted by photobleaching a certain region of the image with a high-intensity laser. Photobleaching destroys the fluorescent ability of proteins within this region of interest (ROI), resulting in a local perturbation where fluorescence equals 0. As proteins continue to diffuse after photobleaching, fluorescence is "recovered" by proteins that diffuse into the region from the rest of the nucleus, eventually reaching a new equilibrium (Figure 1). The rate at which this fluorescence recovers is a function of the effective rate of diffusion of the protein.

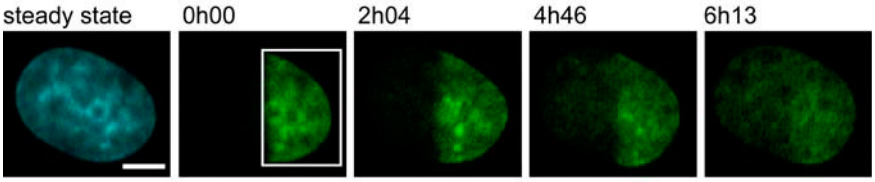

Figure 1: Example of a FRAP imaging experiment (Figure adapted from [1])

FRAP has been used previously to study the kinetics of a wide variety of nuclear proteins, including the cancer factor p53, actin, and histone proteins [2, 3, 4].

1.3 Modeling Choices

After extracting intensities from the series of images, the data must be fit to a model to extract information about the kinetics of the protein. FRAP kinetic models often make assumptions about the nature of the problem to make numerical simulation and curve fitting possible [5, 6]. Some of the modeling choices include (adapted from [7]):

1. Nuclear geometry: Cell nuclei are often irregularly shaped, and it is difficult to precisely define the geometry in three dimensions. FRAP models often reduce the problem to two dimensions since microscopy images are recorded in two dimensions, although three-dimensional models have been used as well [1]. For analytical solutions, the nucleus is often assumed to be a rectangle, square, or circle.

2. Photobleach: Depending on the assumed nuclear geometry, photobleaching can be performed in a narrow strip, or in a circular spot.

3. Diffusion: Fickian diffusion is often assumed, although more complex models of anomalous diffusion have been used as well [8, 9, 10, 11].

4. Distribution of binding sites: DNA binding sites are not always uniformly distributed throughout the nucleus, but can be localized (for example near the periphery) or otherwise heterogenously distributed.

5. Binding Reaction: Binding can be modeled as effective diffusion (in which binding decreases the diffusion constant of a protein), a simple first-order reaction (with kinetics governed by rate constants), or as a more complex series of reactions.

In this work, we present three models of FRAP kinetics. In the first, only diffusion is considered, and the experimental setup is designed to reduce the modeling problem to one dimension, leading to analytical solutions of the diffusion (or heat) equation. The second model extends this to two dimensions, and models the nucleus as a square. The third model incorporates the binding reaction, and tracks the bound and unbound populations of protein.

2 Model Setup and Analysis

We will model diffusive fluorescence recovery in two ways. In the first type of model, termed "Free Diffusion Models," only diffusion is considered, first in 1D and then in 2D. The protein binding kinetics are represented by the diffusion constant D_{eff} [12]. In the second type of model, reaction and diffusion are considered. Here, the binding reaction kinetics are separated from diffusion, at the expense of being more difficult to fit to experimental data without additional information or assumptions. In all three cases, we assume no flux of proteins through the nuclear membrane, a homogeneous initial distribution of fluorescence, and a homogeneous distribution of potential binding sites.

2.1 Free Diffusion Models

2.1.1 Fick's Second Law

The diffusion equation, also known as Fick's Second Law, takes the same form as the heat equation, and states that the rate of change of the concentration c at any given point is given by

$$\frac{\partial c}{\partial t} = \nabla \cdot (D \nabla c) \tag{3}$$

If we assume that D is constant for a given molecule in a given solvent, this can be written using the Laplacian operator

$$\frac{\partial c}{\partial t} = D \nabla \cdot \nabla c = D \Delta c \tag{4}$$

where D is called the diffusion constant. Steady-state solutions occur when

$$\frac{\partial c}{\partial t} = D \Delta c = 0 \tag{5}$$

which is Laplace's equation. The diffusion equation can be solved analytically in one dimension by separation of variables. Assume the solution u can be written $u(x,t) = DA(x)B(t)$. Then

$$A(x)B'(t) = DA''(x)B(t) \tag{6}$$

which requires that

$$\frac{A''(x)}{A(x)} = \frac{1}{D}\frac{B'(t)}{B(t)} = -\lambda \tag{7}$$

for some constant $\lambda > 0$. This reduces the PDE into a system of two ODEs. Solving both of these ODEs independently gives

$$\begin{aligned} A(x) &= c_1 \sin\sqrt{\lambda}x + c_2 \cos\sqrt{\lambda}x \\ B(t) &= c_3 e^{-D\lambda t} \end{aligned} \tag{8}$$

which means that

$$u(x,t) = A(x)B(t) = c_1' e^{-D\lambda t} \sin\sqrt{\lambda}x + c_2' e^{-D\lambda t} \cos\sqrt{\lambda}x \tag{9}$$

solves the diffusion equation for any λ.

2.1.2 Neumann Boundary Conditions for FRAP Modeling

The experimental setup determines the boundary and initial conditions. In free-diffusion models, restricted diffusion is accounted for by the "effective diffusion constant" D_{eff}. That is, the fluorescence intensity u at a point $\mathbf{x}$ is modeled by:

$$\frac{\partial}{\partial t} u(\mathbf{x}, t) = D_{\text{eff}} \Delta u(\mathbf{x}, t), \quad t > 0 \tag{10}$$

with an initial condition given by the fluorescence data immediately after photobleaching (at time $t = 0$):

$$u(\mathbf{x}, 0) = f(\mathbf{x}) \tag{11}$$

Complete photobleaching within the region of interest (ROI) is often assumed and a homogeneous distribution of protein elsewhere, meaning that the initial conditions are

$$f(\mathbf{x}) = \begin{cases} 0, & x \in ROI \\ u_0, & x \notin ROI \end{cases} \tag{12}$$

The nuclear membrane is the physical boundary. We assume no flux of fluorescent biomolecules through the membrane, i.e. Neumann boundary conditions. Thus, the complete free-diffusion FRAP model is

$$\begin{aligned} &\frac{\partial}{\partial t}u(\mathbf{x},t) = D_{\text{eff}}\Delta u(\mathbf{x},t), \quad \mathbf{x} \in \Omega, \quad t > 0, \\ &\frac{\partial u}{\partial \eta} = 0, \quad \mathbf{x} \in \partial\Omega, \quad t > 0 \\ &u(\mathbf{x},0) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \end{aligned} \tag{13}$$

where Ω represents the cellular membrane-bound compartment, $\partial\Omega$ the membrane (i.e. the boundary of the compartment), and η is the outer unit vector normal to the boundary.

2.1.3 Narrow-Strip Photobleaching Reduces the FRAP Experiment to 1D

By designing the FRAP experiment carefully, the diffusion model can be considered in a single dimension. Following the lead of [5], we model the nucleus as a rectangle of length l. The photobleached region is a narrow, rectangular band within this rectangle. Let the photobleached band have width $2h$ and be centered on the x-axis at c. The problem can then be reduced to one dimension:

$$\begin{aligned} &u_t(x,t) = D_{\text{eff}}u_{xx}(x,t), \quad x \in (0,l), \quad t > 0, \\ &u(0,t) = 0, \quad u(l,t) = 0, \quad t > 0 \\ &u(x,0) = f(x), \qquad x \in (0,l), \end{aligned} \tag{14}$$

with the initial condition

$$f(x) = \begin{cases} 0, & |x-c| \leq 0 \\ u_0, & |x-c| > 0 \end{cases} \tag{15}$$

In physical terms, this corresponds to complete photobleaching (no fluorescence) within the narrow band of width $2h$ centered at c. Outside of this band, the fluorescence is assumed to be unchanged, i.e. u_0 is the initial uniform steady-state concentration before photobleaching. From the general solution to the 1D diffusion equation in (9), the Neumann boundary conditions eliminate the sines and allow only λ's of the form $\lambda = (n\pi/l)^2$. Thus the Fourier series solution to (15) is

$$u(x,t) = \sum_{n=1}^{\infty} \frac{1}{n\pi} e^{-(n\pi/l)^2 D_{\text{eff}} t} \cos\frac{n\pi x}{l} \tag{16}$$

To derive an expression for the fluorescence recovery after photobleaching, integrate over the photobleached area $x = [c-h, c+h]$ and divide by the original fluorescence immediately prior to photobleaching $2hu_0$ to

get

$$f_R(t) = \frac{1}{2hu_0}\int_{c-h}^{c+h} u(x,t)\,dx \tag{17}$$

$$= \frac{l-2h}{l} - \frac{l}{h\pi^2}\sum_{n=1}^{\infty}\frac{1}{n^2}e^{-(n\pi/l)^2 D_{\text{eff}}t}\left[\sin\left(\frac{n\pi(c-h)}{l}\right) - \sin\left(\frac{n\pi(c+h)}{l}\right)\right] \tag{18}$$

2.2 Small Spot Photobleaching Can Be Considered an Infinite Domain

In an alternative kind of FRAP setup, photobleaching is performed on a very small spot near the center of the nucleus. In this situation, the relative scale of the nucleus to the photobleached spot is large enough that the domain can be considered infinite in $\mathbb{R}^2$, and the diffusion equation solved by first finding the Fundamental solution [adapted from class notes / Strang CSE]. On an infinite domain, boundary conditions no longer constrain the solution given in (9), so the 1D general solution is given by the integral over all $k = \sqrt{\lambda}$

$$u(x,t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\hat{u}_0(k)e^{ikx}e^{-Dk^2t} \tag{19}$$

Now consider the initial condition of a point source $G(x,0) = \delta(x)$ which has Fourier transform $\hat{u}_0 = 1$. Then the solution given by (19) is

$$G(x,t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ikx}e^{-Dk^2t}\,dk \tag{20}$$

$$= \frac{1}{\sqrt{4\pi Dt}}e^{-x^2/4Dt} \tag{21}$$

In 2D, we consider $u(\mathbf{x},0) = \delta^2(x,y) = \delta(x)\delta(y)$, but we can separate x from y to give

$$G(x,y,t) = \frac{1}{4\pi Dt}e^{-x^2/4Dt}e^{-y^2/4Dt} \tag{22}$$

Now the solution for any initial conditions given by $u(x,y,0)$ can be found by convolving with the Green's function $G(x,y,t)$:

$$\begin{aligned} u(x,y,t) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(v,w,0)G(x-v,y-w,t)\,dv\,dw \\ &= \frac{1}{4\pi Dt}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(v,w,0)e^{-(x-v)^2/4Dt}e^{-(y-w)^2/4Dt}\,dv\,dw \end{aligned} \tag{23}$$

2.3 Reaction-Diffusion Model

It is often desirable to extract more meaningful biological information from the kinetic model. To more accurately model the restricted diffusion of proteins due to binding immobile DNA-structures, a system of PDEs is designed in which the population of unbound molecules u_r undergo diffusion and reaction, while the population of bound molecules u_b are immobile and only undergo reaction:

$$\begin{aligned} \frac{\partial}{\partial t}u_r &= D\frac{\partial^2}{\partial x^2}u_f - k_a u_f + k_d u_b, \\ \frac{\partial}{\partial t}u_b &= k_a u_f - k_d u_b \end{aligned} \tag{24}$$

D is now the actual the diffusion coefficient (not the "effective" coefficient), and k_a and k_d represent the

binding and unbinding rate constants, respectively. In this model, the effect of restricted diffusion is more accurately represented by the binding constants rather than being lumped into the diffusion constant.

Unfortunately, this means that there are three unknown parameters that must be fit to the experimental data. In order to simplify the problem somewhat, the diffusion constant is often estimated based on the molecular weight of the protein and the viscosity of the nuclear fluid. In some cases, it is possible to disrupt the DNA-binding ability of the protein through genetic approaches, fit the diffusion constant using a free-diffusion model, and then incorporate this parameter into the reaction-diffusion model.

2.4 Fitting Experimental Data

FRAP kinetic models are generally fit to experimental data with a nonlinear least-squares optimization algorithm. When multiple parameters are being fit simultaneously, there are often multiple "best fit" configurations, so testable parameters are often experimentally confirmed when possible after fitting [12].

3 Numerical Solution of FRAP Models

We numerically simulated three variations of the models (15) and (24) using explicit finite differences. In all three cases, we use a second-difference approximation of diffusion with Neumann boundary conditions (i.e. the B matrix) and an explicit forward-Euler approximation in time. This results in approximating $u_t = D\Delta u$ as

$$\frac{U(x,t+\Delta t)-U(x,t)}{\Delta t} = -D\frac{BU(x,t)}{(\Delta x)^2} \tag{25}$$

in 1D and

$$\frac{U(x,t+\Delta t)-U(x,t)}{\Delta t} = -D\frac{(B2D)U(x,t)}{(\Delta x)^4} \tag{26}$$

in 2D, where B2D is the appropriate analog. The accuracy of the numerical scheme is first-order with respect to the time step and second-order in space:

$$\Delta u = O(\Delta t) + O((\Delta x)^2) \tag{27}$$

If we rearrange, we see that

$$U(x,t+\Delta t) = U(x,t) + \frac{D\Delta t}{(\Delta x)^2}BU(x,t) \tag{28}$$

Let $R = D\Delta t/(\Delta x)^2$. Stability is achieved for the 1D case when

$$R \leq \frac{1}{2} \tag{29}$$

and for the 2D case when

$$R \leq \frac{1}{4} \tag{30}$$

Thus, we have a condition for Δt that

$$\Delta t \leq \frac{D}{2}(\Delta x)^2 \tag{31}$$

for 1D and

$$\Delta t \leq \frac{D}{4}(\Delta x)^2 \tag{32}$$

for 2D.

When integration is performed, we approximate with Simpson's rule for integration, with our Matlab implementation given in Listing 1.

Listing 1: MATLAB Implementation of Simpson Approximate Integration

```
function [ I ] = simpson( x, y )
%SIMPSON Evaluate "integrals" with Simpson approximation
%   Approximate the integral of the points in (x,y)
%   x should be evenly spaced (i.e. step size h)

h = x(2) - x(1);
I = h*(y(1) + y(end) + 2*sum(y(3:2:end-2)) + 4*sum(y(2:2:end-1))) / 3;

end
```

Furthermore, in all cases, fluorescence can be scaled to the original uniform distribution such that fluorescence occurs on the range $[0, 1]$. All second-difference matrices were stored as sparse matrices for computational efficiency.

3.1 Solving and Fitting Free Diffusion Models

Our first two models consider only diffusion, first in 1D and then in 2D. These models contain a single parameter D_{eff} that can easily be fit to experimental data.

3.1.1 Model 1 - 1D Diffusion Equation

For our first numerical model, we consider the case of 1D diffusion, corresponding to the FRAP experiment outlined in Section 2.1.3. Photobleaching is performed on a narrow band from $[c-w, c+w]$ in the center of the domain. The initial conditions are thus simply established by the code in Listing 2.

Listing 2: Initial Conditions for 1D Model with Narrow Band Photobleaching

```
function [ y ] = initialConditions1D(x, c, w)
%INITIALCONDITIONS1D Initial conditions for FRAP model 1
%   Take a list of grid points x and return the initial conditions for a
%   narrow band photobleach strip of width 2w, centered at c
%   Fluorescence is assumed to be at steady-state equilibrium prior to
%   photobleaching

% Validate that the setup makes sense
if c-w < min(x) || c+w > max(x)
    error('Photobleaching band must be within the simulated region [0,L]!');
end

% Within the photobleach strip x \in [c-h, c+h], y = 0
% Elsewhere y = 1
y = (x <= c-w | x >= c+w);

end
```

We construct the B matrix with Neumann boundary conditions in Listing 3.

Listing 3: Constructing Forward-Difference Matrix for 1D Model with Neumann Boundary Conditions

```
function [ D ] = diffNeumann1D( N )
%DIFFNEUMANN1D Construct the diff. matrix for 1D diffusion eq. w/no flux
%   N = number of grid points (including boundaries since they're unknown)
%   with Neumann boundary conditions
%   This is B_n from Strang CSE

e = ones(N,1);
D = spdiags([-e 2*e -e], -1:1, N, N);

% Neumann BC's on both ends
D(1,1) = 1;
D(end,end) = 1;

end
```

We then solve the model with our finite difference scheme given in Listing 4.

Listing 4: MATLAB Implementation of FRAP Model 1 Solver

```
function [ f, t ] = frapmodel1( L, c, w, N, tf, deltaT, D )
%FRAPMODEL1 Numerically simulate a FRAP experiment with a 1D model
%   L = length of the nucleus (fluorescent membrane region) [0, L]
%   c = center of photobleaching band (c \in (0, L))
%   w = width of photobleaching band (should be << L and \in ((w, L-w))
%   N = number of grid points (should have L/N < w to observe photobleach)
%   tf = final time point (initial time point t0 = 0)
%   deltaT = timestep to use for forward-differences (ensure stability!)
%   D = diffusion coefficient
% RETURN:
%   f = mean fluorescence within the photobleached band [c-h, c+h],
%   i.e. the simulated FRAP recovery curve at each timestep
%   t = actual time points corresponding to the values in f

%% Divide the region L into N grid points
x = linspace(0, L, N)';
% Step size (should = L/N)
h = x(2) - x(1);

%% Check the stability based on the current grid resolution and timestep
CFL = D * deltaT / h^2;
% Quit if solution unstable
if CFL >= 0.5
    fprintf('CFL = %0.02f\n', CFL);
    error('CFL condition is not met! Solution will exhibit instability!');
end

%% Get the initial condition vector for the current grid
u0 = initialConditions1D(x, c, w);
bleached = (u0 == 0);

%% Construct the difference matrix with Neumann BC's
B = diffNeumann1D(N);
```

```
%% Solve the diffusion equation with finite difference approximation
u = u0;
numtsteps = ceil(tf / deltaT);
f = zeros(numtsteps+1, 1);
t = zeros(numtsteps+1, 1);

for n = 1:numtsteps
   % Fluorescence recovery  within the photobleached region [c-w, c+w]
   % using the calculated grid points
   t(n) = (n-1)*deltaT;
   f(n) = simpson(x(bleached), u(bleached)) / (2*w);

   % Approximate the next time step u_{n+1} using the difference matrix
   % This is a forward-difference with respect to time O(dt)
   % and a centered-difference with respect to space O(h^2)
   % Solve for u_{n+1}
   u = u - (D*deltaT/h^2)*B*u;
end

% Mean fluorescence within the photobleached region [c-w, c+w]
% using the calculated grid points
t(n+1) = tf;
f(n+1) = simpson(x(bleached), u(bleached)) / (2*w);

end
```

An example of the evolution of the model is shown in Figure 2. The photobleached band from $x = 0.75$ to $x = 1.25$ quickly recovers fluorescence as diffusion occurs. The system evolves to a uniform equilibrium, although the resulting uniform distribution is reduced from pre-photobleaching conditions because many of the fluorescent proteins no longer fluoresce, a result observed *in vivo*.

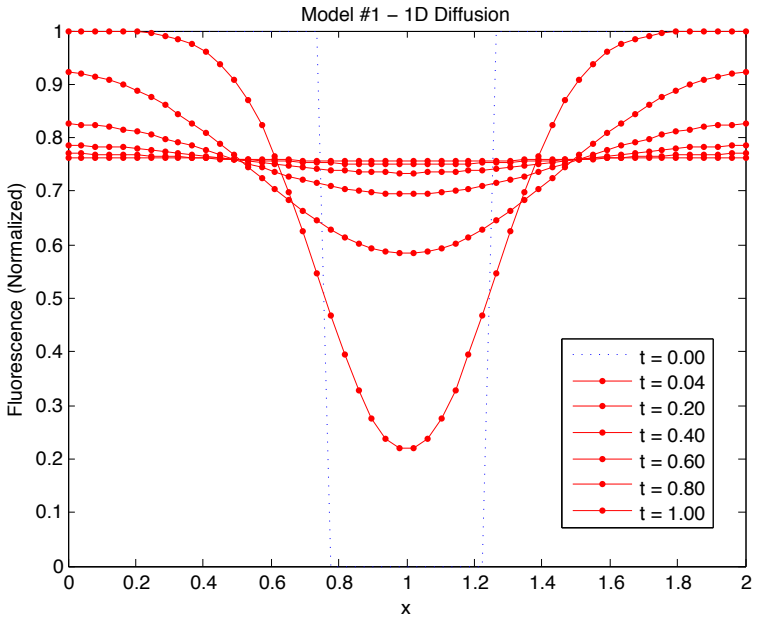

Figure 2: Evolution of Model 1 over time with $L = 2$, $w = 0.25$, $D = 0.2$. Diffusion was simulated with timesteps $\Delta t = 1 \times 10^{-3}$s with $N = 50$ grid points ($h = 0.04$). Initial conditions are shown in blue and simulated timepoints in red.

We then sought to fit this model to a data set of experimental FRAP fluorescence data (Supplementary Tables 1 and 2). Since the model contains only a single parameter, fitting was performed in two steps. First, a range of biologically relevant D_{eff} were sampled. Then optimization was performed with Matlab's lsqcurvefit command. The algorithm is shown in Listing 5.

Listing 5: Least Squares Fitting of Experimental FRAP Data to Model 1

```
function [ Deff, rmsd, residuals ] = fitmodel1(fExp, L, w, tStep)
%FITMODEL1 Fit the experimental FRAP data to the 1D model
%    fExp = experimental FRAP recovery data
%    f0 = initial uniform fluorescence immediately prior to photobleaching
%    L = length of the cellular membrane-bound organelle
%    w = width of the photobleached region
%    tStep = length of time between experimental data points

numImages = length(fExp);
tFinal = tStep*(numImages-1);
% Use enough grid points to have at least 5 within the photobleached region
N = ceil(5*L/w);
h = L / N;

% Sample uniformly to avoid local minima during optimization
Ds = 10:5:100;
rmsds = zeros(size(Ds));
for i = 1:length(Ds)
    rmsds(i) = sqrt(sum((f(Ds(i), 0:tStep:tFinal)-fExp).^2) / numImages);
end
[m,i] = min(rmsds);
```

```
Di = Ds(i);

% Fit the 1D model with least squares optimization
options = optimset('TolFun', 1e-6);
[Deff, rmsd, residuals] = lsqcurvefit(@f, Di, 0:tStep:tFinal, fExp, [], [], options);

function simulation = f(D, T)
    % Crank down dt until stability is reached
    dt = tStep;
    if D*dt/h^2 >= 0.5
        dt = 0.5*0.5*h^2 / D;
    end

    % Run the simulation
    [f,t] = frapmodel1(L, L/2, w, N, tFinal, dt, D);

    % Interpolate the results back to the experimental time points
    % and compute the residuals for lsqnonlin optimization
    simulation = interp1(t, f, T, 'linear')';
end

end
```

The resulting best fit gave a diffusion coefficient of $D_{eff} = 29.704 \mu m^2 s^{-1}$, and the simulated and experimental fluorescence recovery profiles are shown in Figure 3.

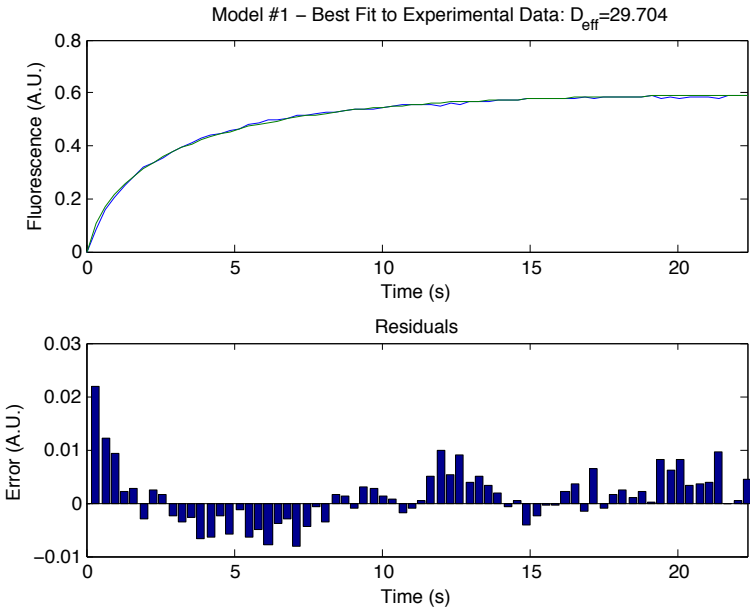

Figure 3: Plot of best fit to experimental FRAP data using Model 1 and the residuals.

3.1.2 Model 2 - 2D Diffusion Equation

We then extended this model to consider the 2D FRAP setup in which the nucleus is modeled as a square, and photobleaching is performed on a small, circular spot at any point within the nucleus. Initial conditions were

computed as in the 1D case, with 0 units fluorescence within the photobleached circle, and 1 (normalized) units of fluorescence elsewhere (Listing 6).

Listing 6: Initial Conditions for 2D Model with Spot Photobleaching

```
function [ y ] = initialConditions2D(L, N, c, r)
%INITIALCONDITIONS2D Initial conditions for 2D FRAP models
%   Take a grid of points G and return the initial conditions for a
%   spot (circle) photobleach of radius r, centered at c
%   Fluorescence is assumed to be at steady-state equilibrium prior to
%   photobleaching


% Validate that the setup makes sense
if any(c-r < 0) || any(c+r > L)
    error('Photobleaching spot must be within the simulated region [0,L]x[0,L]!');
end

% Divide the region [0, L] x [0, L] into NxN grid points
X = repmat(linspace(0,L,N), N, 1);
Y = repmat(linspace(0,L,N)', 1, N);

% Within the photobleach spot x \in, y = 0
% Elsewhere y = 1
y = (((X-c(1)).^2 + (Y-c(2)).^2) > r^2);

end
```

The two-dimensional analog of the second-difference matrix with Neumann boundary conditions was constructed as the sum of the Kronecker product of the one-dimensional B matrix as shown in Listing 7.

Listing 7: Constructing Forward-Difference Matrix for 2D Model with Neumann Boundary Conditions

```
function [ D ] = diffNeumann2D( N )
%DIFFNEUMANN2D Construct the diff. matrix for 2D diffusion eq. w/no flux
%   N^2 = number of grid points (including boundaries since they're unknown)
%   with Neumann boundary conditions
%   This is B2D for a unit square [0,1]x[0,1] from Strang CSE
%   Using the 5-point Laplacian molecule

% Construct using the Kronecker tensor product
% as outlined in Strang CSE p. 291
B = diffNeumann1D(N);
M = diag([0.5 ones(1,N-2) 0.5]);
D = kron(M,B) + kron(B,M);

end
```

The model was then simulated in a very similar way to the 1D model, as shown in Listing 8.

Listing 8: MATLAB Implementation of FRAP Model 2 Solver

```
function [ f, t ] = frapmodel2( L, c, r, N, tf, deltaT, D )
%FRAPMODEL2 Numerically simulate a FRAP experiment with a 2D square model
```

```
%   L = side length of the imaged fluorescence region [0, L] x [0, L]
%   c = center of photobleaching spot (c \in [0, L] x [0, L])
%   r = radius of photobleaching spot (should be << L and \in [r, L-r] x [r, L-r])
%   N = number of grid points (should have L/N < w to observe photobleach)
%   tf = final time point (initial time point t0 = 0)
%   deltaT = timestep to use for forward-differences (ensure stability!)
%   D = diffusion coefficient
% RETURN:
%   f = mean fluorescence within the photobleached spot
%   i.e. the simulated FRAP recovery curve at each timestep
%   t = actual time points corresponding to the values in f


%% Check the stability based on the current grid resolution and timestep
% Step size
h = L / N;
% Stability condition
CFL = D * deltaT / h^2;
% Quit if solution unstable
if CFL >= 0.25
    fprintf('CFL = %0.02f\n', CFL);
    error('CFL condition not met! Solution will exhibit instability!');
end

%% Get the initial condition matrix for the current grid
u0 = initialConditions2D(L, N, c, r);
bleached = find(u0 == 0);
numBleached = numel(bleached);

%% Construct the difference matrix with Neumann BC's
B = diffNeumann2D(N);

%% Solve the diffusion equation with finite difference approximation
u = u0(:);
numtsteps = ceil(tf / deltaT);
t = zeros(numtsteps+1,1);
f = zeros(numtsteps+1,1);

for n = 1:numtsteps
  % Calculate fluorescence recovery within the photobleached spot
    % using the calculated grid points
    t(n) = (n-1)*deltaT;
    f(n) = sum(u(bleached)) / numBleached;

    % Approximate the next time step u_{n+1} using the difference matrix
    % This is a forward-difference with respect to time O(dt)
    % and a centered-difference with respect to space O(h^2)
    % Solve for u_{n+1}
    u = u - (D*deltaT/h^2)*B*u;
end

% Final timestep
```

```
55  t(n+1) = tf;
    f(n+1) = sum(u(bleached)) / numBleached;

    end
```

An example showing the evolution of the free-diffusion model in 2D is shown in Figure 4. Again, the system approaches uniform equilibrium over time, and quickly smooths the perturbation caused by photobleaching.

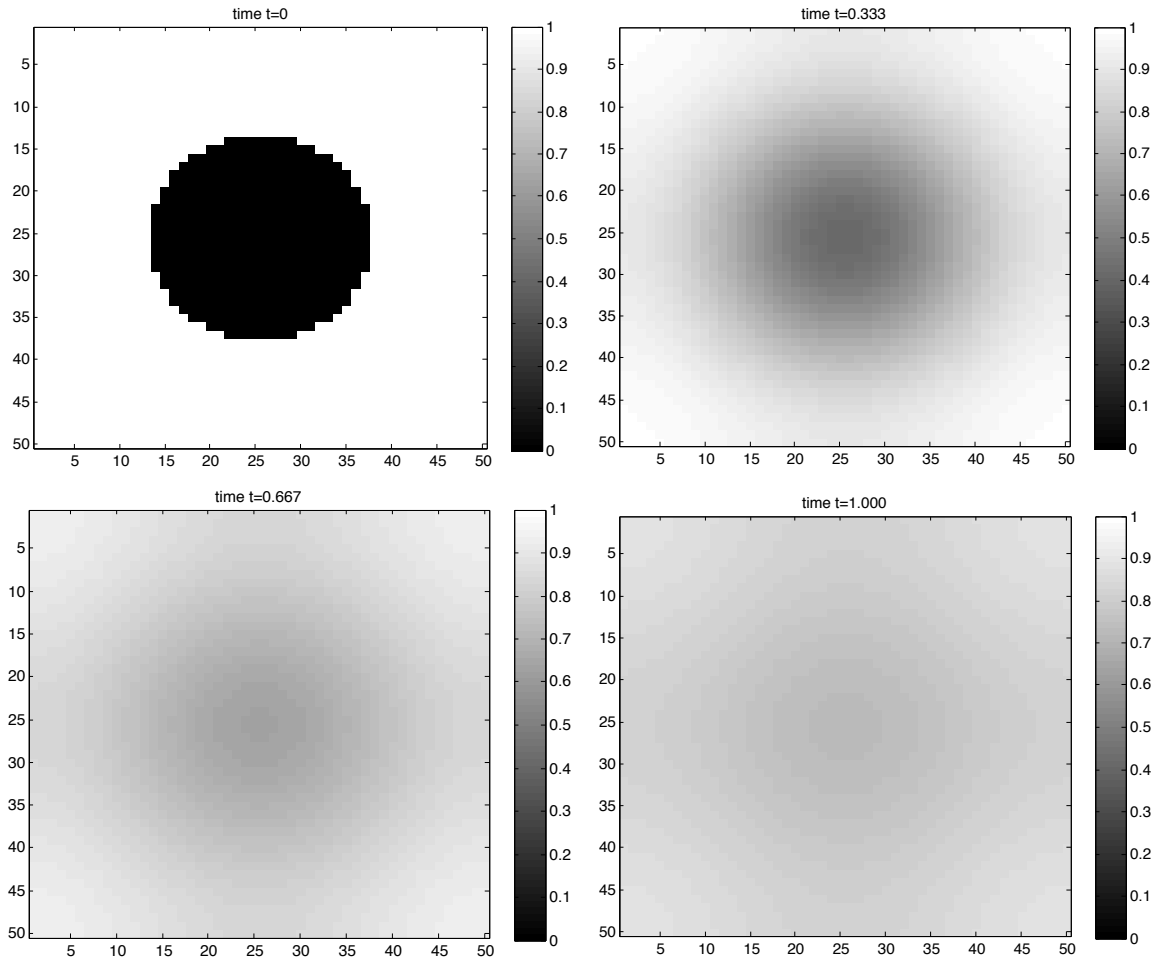

Figure 4: Evolution of Model 2 over time with $L = 2$, $c = (1, 1)$, $r = 0.5$, $D = 0.2$. Diffusion was simulated with timesteps $\Delta t = 1.9 \times 10^{-3}$s with $N = 50^2$ grid points ($h = 0.04$).

Curve fitting was performed using code analogous to that used for Model 1 (not shown for brevity), and the resulting best fit gave $D_{eff} = 31.704 \mu m^2 s^{-1}$ (Figure 5). Interestingly, the 2D model resulted in a slightly higher estimation of D_{eff} with slightly higher root-mean-squared deviation (RMSD), although the differences were fairly insignificant.

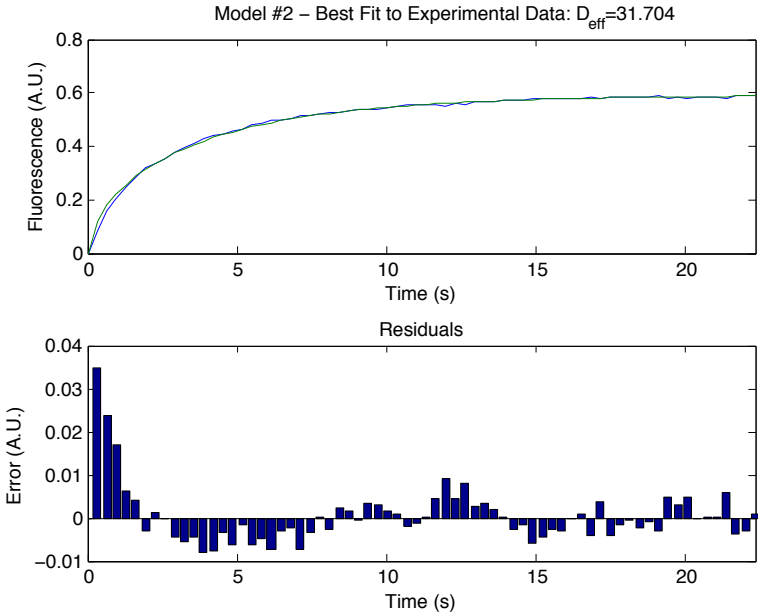

Figure 5: Plot of best fit to experimental FRAP data using Model 2 and the residuals.

3.2 Solving and Fitting the Reaction-Diffusion Model

Lastly, we constructed a simulation of the reaction-diffusion (R-D) model presented in Section 2.3. In this model, the 2D diffusion model was extended to incorporate bound (immobile) and unbound (mobile) populations. The reaction system was assumed to be at dynamic equilibrium prior to photobleaching. Thus, if the binding reaction was governed by the rate constant k_{on} and unbinding by the rate constant k_{off}, the equilibrium constant for the entire system is given by $K = k_{on}/k_{off}$. Furthermore, the no flux boundary conditions assure that the bound and unbound populations (u_b and u_f, respectively) total a constant level u_0

$$u_b + u_f = u_0 \tag{33}$$

Thus, the initial concentrations (at equilibrium) of protein in each state are given by

$$u_f = \frac{1}{1+K} \tag{34}$$

$$u_b = u_0 - u_f \tag{35}$$

These initial conditions were implemented in Listing 9.

Listing 9: Initial Conditions for R-D Model with Spot Photobleaching

```
function [ b, u ] = initialConditionsRD(L, N, c, r, K)
%INITIALCONDITIONSBD Initial conditions for R-D FRAP model #3
%   Take a grid of points G and return the initial conditions for a
%   spot (circle) photobleach of radius r, centered at c
%   Fluorescence is assumed to be at steady-state equilibrium prior to
%   photobleaching with uniform fluorescence y0


% Validate that the setup makes sense
```

```
if any(c-r < 0) || any(c+r > L)
    error('Photobleaching spot must be within the simulated region [0,L]x[0,L]!');
end

% Divide the region [0, L] x [0, L] into NxN grid points
X = repmat(linspace(0,L,N), N, 1);
Y = repmat(linspace(0,L,N)', 1, N);

% Within the photobleach spot x \in, y = 0
% Elsewhere set bound and unbound populations to steady-state distributions
% based on the reaction equilibrium constant K
uEquil = 1 / (1+K);
bEquil = 1 - uEquil;
mask = (((X-c(1)).^2 + (Y-c(2)).^2) > r^2);

u = uEquil * mask;
b = bEquil * mask;

end
```

We then solved this new R-D model by extending the 2D diffusion model to keep track of and incorporate bound and unbound populations. The system of ODEs were solved in parallel, with diffusion being applied only to the unbound population (Listing 10).

Listing 10: MATLAB Implementation of FRAP Model 3 Solver

```
function [ f, t ] = frapmodel3( L, c, r, N, tf, deltaT, D, kOn, kOff )
%FRAPMODEL3 Numerically simulate a FRAP experiment with a 2D R-D model
%   L = side length of the imaged fluorescence region [0, L] x [0, L]
%   c = center of photobleaching spot (c \in [0, L] x [0, L])
%   r = radius of photobleaching spot (should be << L and \in [r, L-r] x [r, L-r])
%   N = number of grid points (should have L/N < w to observe photobleach)
%   tf = final time point (initial time point t0 = 0)
%   deltaT = timestep to use for forward-differences (ensure stability!)
%   D = diffusion coefficient
%   kOn = binding constant
%   kOff = dissociation constant
% RETURN:
%   f = mean fluorescence within the photobleached spot
%   i.e. the simulated FRAP recovery curve at each timestep
%   t = actual time points corresponding to the values in f


%% Check the stability based on the current grid resolution and timestep
% Step size
h = L / N;
% Stability condition
CFL = D * deltaT / h^2;
% Quit if solution unstable
if CFL >= 0.25
    fprintf('CFL = %0.02f\n', CFL);
    error('CFL condition not met! Solution will exhibit instability!');
end
```

```
%% Get the initial condition matrix for the current grid
% Equilibrium constant for the binding/dissociation reaction
K = kOn / kOff;
[b0, u0] = initialConditionsRD(L, N, c, r, K);
bleached = find(b0 == 0);
numBleached = numel(bleached);


%% Construct the difference matrix with Neumann BC's
B = diffNeumann2D(N);


%% Solve the diffusion equation with finite difference approximation
b = b0(:);
u = u0(:);
numtsteps = ceil(tf / deltaT);
t = zeros(numtsteps+1,1);
f = zeros(numtsteps+1,1);

for n = 1:numtsteps
  % Calculate fluorescence recovery within the photobleached spot
    % using the calculated grid points
    % Both bound AND unbound molecules fluoresce, even though only unbound
    % (free) molecules are diffusing
    t(n) = (n-1)*deltaT;
    f(n) = sum(b(bleached)+u(bleached)) / numBleached;

    % Approximate the next time step u_{n+1} using finite differences
    % This is a forward-difference with respect to time O(dt)
    % and centered-difference with respect to space O(h^2)
    % Unbound/free population undergoes diffusion and reaction
    tmpU = u;
    u = u - (D*deltaT/h^2)*B*u - kOn*u + kOff*b;
    % Bound population only undergoes reaction
    b = b + kOn*tmpU - kOff*b;
end

% Final timestep
t(n+1) = tf;
f(n+1) = sum(b(bleached) + u(bleached)) / numBleached;

end
```

When solved, the model displays dynamics similar in nature to the 2D diffusion model, suggesting that retarded diffusion is being captured by the presence of bound and unbound populations. An example is shown in Figure 6.

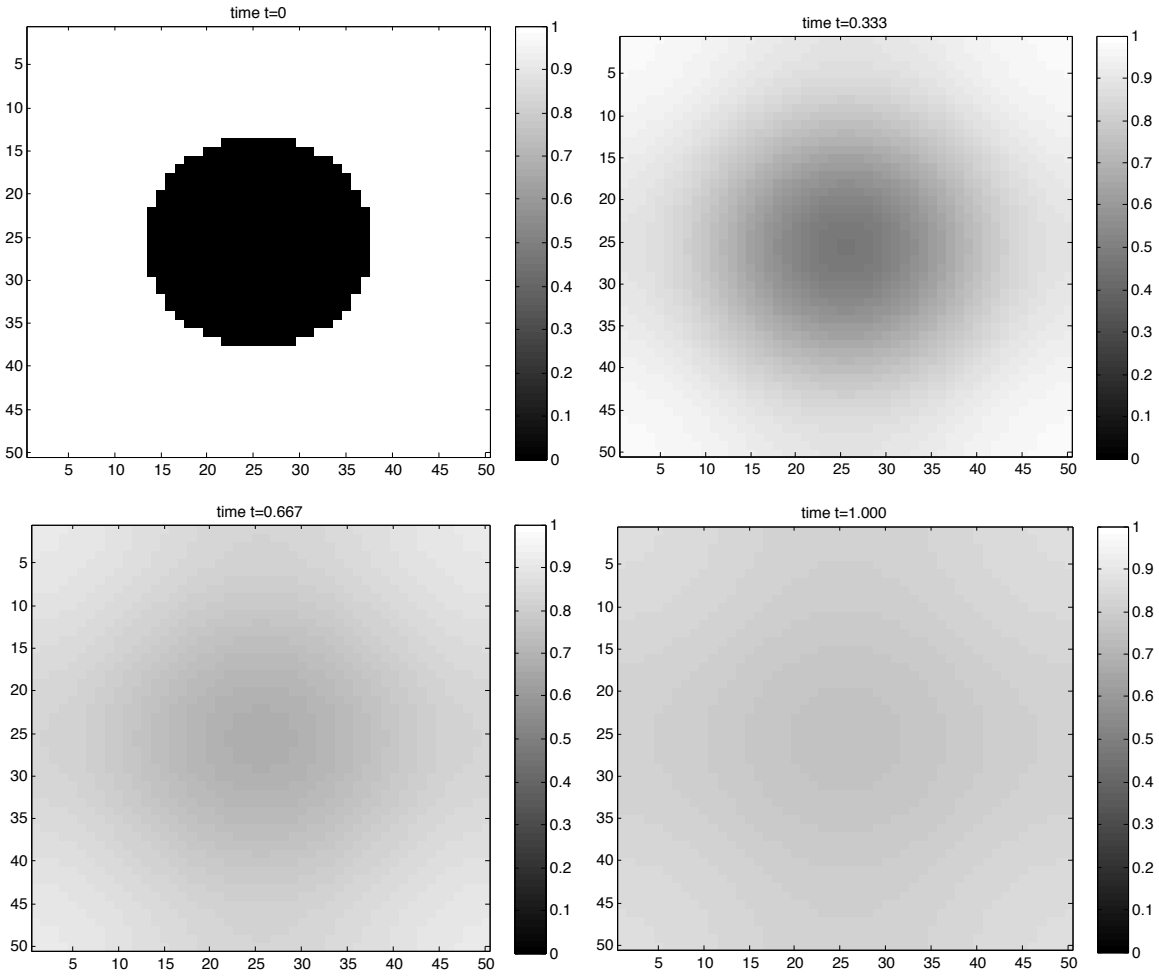

Figure 6: Evolution of Model 3 over time with $L = 2$, $c = (1, 1)$, $r = 0.5$, $D = 0.5$, $k_b = 0.02$, $k_u = 0.02$. Diffusion was simulated with timesteps $\Delta t = 7 \times 10^{-4}$s with $N = 50^2$ grid points ($h = 0.04$).

Because 3 parameters are involved in fitting the R-D model, initially sampling the parameter space was unfeasible. We therefore first fit the experimental data with Model 2 and used the resulting diffusion coefficient as an initial guess for optimization (Listing 11).

Listing 11: Least Squares Fitting of Experimental FRAP Data to Model 3

```
function [ Cfit, rmsd, residuals ] = fitmodel3(fExp, L, r, tStep)
%FITMODEL3 Fit the experimental FRAP data to the R-D model
%   fExp = experimental FRAP recovery data
%   L = length of the cellular membrane-bound organelle
%   r = radius of the photobleached region
%   tStep = length of time between experimental data points

numImages = length(fExp);
tFinal = tStep*(numImages-1);
% Use enough grid points to have at least 5 within the photobleached region
N = ceil(5*L/r);
h = L / N;
```

```
% Initial parameters guess
Ci = [50, 0.03, 0.03];

% Fit the R-D model with least squares optimization
options = optimset('TolFun', 1e-3);
[Cfit, rmsd, residuals] = lsqcurvefit(@f, Ci, 0:tStep:tFinal, fExp, [0, 0, 0], [], options);

function simulation = f(C,T)
    % Extract the 3 parameters
    D = C(1);
    kOn = C(2);
    kOff = C(3);

    % Crank down dt until stability is reached
    dt = tStep;
    if D*dt/h^2 >= 0.25
        dt = 0.5*0.25*h^2 / D;
    end

    % Run the simulation
    [f,t] = frapmodel3(L, [L/2 L/2], r, N, tFinal, dt, D, kOn, kOff);

    % Interpolate the results back to the experimental time points
    % and compute the residuals for lsqnonlin optimization
    simulation = interp1(t, f, T, 'linear')';
end

end
```

As expected, the model results in a much higher diffusion coefficient D because binding events are no longer lumped into the diffusion (Figure 7. In this way, this more complicated model allows for more meaningful parameter fitting. In addition to more accurately representing the diffusion coefficient, additional information is given about the relative rates of binding in the rate coefficients k_{on} and k_{off}. This information can be used to determine the distribution of expected residency time for a given protein.

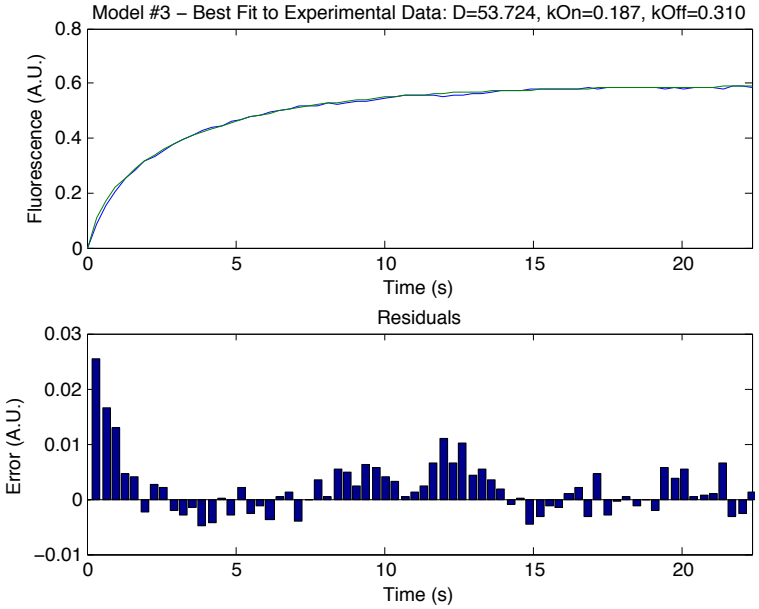

Figure 7: Plot of best fit to experimental FRAP data using Model 3 and the residuals.

Unfortunately, by sampling the parameter space of k, we found that equally good fits occurred along a ratio given by k_{on}/k_{off}, but that a unique fit was not possible (Figure 8). Thus, the reaction-diffusion model is only helpful when the diffusion constant can be measured or estimated by other means, therefore pinning down specific values for the binding constants.

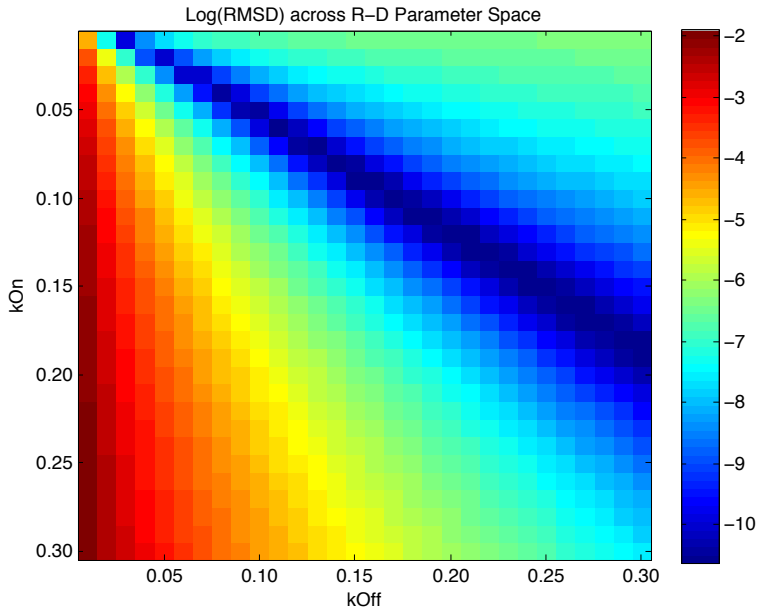

Figure 8: Heatmap of Log(RMSD) vs. experimental data using Model 3 with $D = 53.47$. Multiple equally good fits arise along the ratio $K_{best} = k_{on}/k_{off}$.

4 Conclusions

We demonstrated two classes of kinetic models used to interpret FRAP data and implemented solvers for 1D free diffusion, 2D free diffusion, and 2D reaction-diffusion. Models of increasing complexity are continually being developed as computational power is no longer limiting, however, additional parameters often require additional assumptions in order to procure a "best fit." An additional outstanding question is how well different models actually interpret the data. Because experiments are often designed with a given model in mind, it is difficult to compare the results.

FRAP models illustrate how efficient numerical solvers can be used to simulate models for curve fitting to experimental data. Efficient storage using sparse, banded matrices allows for simulations to be run quickly on a wide range of parameter space, and for least-squares optimization to be performed against experimental data. In this way, FRAP modeling allows for meaningful biological information to be extracted from fluorescence microscopy images, and further development of kinetic models and numerical methods to simulate them efficiently will provide increasingly accurate understanding of biological phenomena.

5 Supplemental Tables

5.1 Table 1 - Experimental FRAP Parameters

FRAP image series	
x,y scale	0.44 µm/pixel
time scale	0.324 s/image
Bleach after 5th image in series	

Bleached ROI	Pixels		µm			
	x	y	x		y	
Coordinates	168	88	73.92	85.36	38.72	47.08
Size	26	19	11.44		8.36	

5.2 Table 2 - Experimental FRAP Data Used for Fitting

Image	Bleached Region Intensity	Image	Bleached Region Intensity
1	150.486	45	122.909
2	149.391	46	123.807
3	149.416	47	123.881
4	149.086	48	124.278
5	148.831	49	124.665
6	53.255	50	125.179
7	63.786	51	125.183
8	72.749	52	125.901
9	79.039	53	125.848
10	84.749	54	125.722
11	88.745	55	125.879
12	92.872	56	125.658
13	95.187	57	125.605
14	97.879	58	126.348
15	100.704	59	125.461
16	102.899	60	126.496
17	104.617	61	126.259
18	106.761	62	126.208
19	108.257	63	126.494
20	109.132	64	126.420
21	110.842	65	126.722
22	111.436	66	125.786
23	113.134	67	126.095
24	113.973	68	125.903
25	115.247	69	126.580
26	115.605	70	126.578
27	116.305	71	126.601
28	117.702	72	125.932
29	117.918	73	127.198
30	118.099	74	127.144
31	119.086	75	126.685
32	119.010		
33	119.597		
34	120.370		
35	120.335		
36	120.819		
37	121.385		
38	121.850		
39	122.553		
40	122.759		
41	122.899		
42	122.648		
43	122.325		
44	123.140		

References

[1] Beaudouin J, Mora-Bermúdez F, Klee T, Daigle N, Ellenberg J: **Dissecting the contribution of diffusion and interactions to the mobility of nuclear proteins**. *Biophysical Journal* 2006, **90**(6):1878–94.

[2] Hinow P, Rogers CE, Barbieri CE, Pietenpol JA, Kenworthy AK, Dibenedetto E: **The DNA binding activity of p53 displays reaction-diffusion kinetics**. *Biophysical Journal* 2006, **91**:330–42.

[3] Mueller F, Wach P, Mcnally JG: **Evidence for a common mode of transcription factor interaction with chromatin as revealed by improved quantitative fluorescence recovery after photobleaching**. *Biophysical Journal* 2008, **94**(8):3323–39.

[4] Axelrod D, Koppel DE, Schlessinger J, Elson E, Webb WW: **Mobility measurement by analysis of fluorescence photobleaching recovery kinetics**. *Biophysical journal* 1976, **16**(9):1055–69.

[5] Carrero G, McDonald D, Crawford E, de Vries G, Hendzel MJ: **Using FRAP and mathematical modeling to determine the in vivo kinetics of nuclear proteins**. *Methods* 2003, **29**:14–28.

[6] van Royen ME, Farla P, Mattern KA, Geverts B, Trapman J, Houtsmuller AB: **Fluorescence recovery after photobleaching (FRAP) to study nuclear protein dynamics in living cells**. *Methods Mol Biol* 2009, **464**:363–85.

[7] Mueller F, Mazza D, Stasevich TJ, Mcnally JG: **FRAP and kinetic modeling in the analysis of nuclear protein dynamics: what do we really know?** *Current Opinion in Cell Biology* 2010, **22**(3):403–11.

[8] Periasamy N, Verkman AS: **Analysis of fluorophore diffusion by continuous distributions of diffusion coefficients: application to photobleaching measurements of multicomponent and anomalous diffusion**. *Biophysical journal* 1998, **75**:557–67.

[9] Saxton MJ: **Anomalous subdiffusion in fluorescence photobleaching recovery: a Monte Carlo study**. *Biophysical journal* 2001, **81**(4):2226–40.

[10] Lubelski A, Klafter J: **Fluorescence recovery after photobleaching: the case of anomalous diffusion**. *Biophysical journal* 2008, **94**(12):4646–53.

[11] Haugh JM: **Analysis of reaction-diffusion systems with anomalous subdiffusion**. *Biophysical journal* 2009, **97**(2):435–42.

[12] Sprague BL, McNally JG: **FRAP analysis of binding: proper and fitting**. *Trends Cell Biol* 2005, **15**(2):84–91.