PhD Diary

Nathan Hughes

March 27, 2019

Contents

1 Tasks [3/9]

1.1 DONE Redo Kitagawa et al. [2] data figures

1.1.1 DONE Write optimising procedure for q

```
from node_diffusion import do_internode_diffusion
from diffusion_functions import D_eff
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import leastsq

cells = 151  # need to take two extra measurements to get sample of 5 without slicing
cell_um = 100
points_per_cell = 10  # Needs to be divisible by 2
Xs = cells*points_per_cell
dx = cell_um//points_per_cell
dx2 = dx**2

b = 0.0
dt = 0.01
# convert to seconds
t = 1/dt
num_seconds = 60*60*14
ts = int(t * num_seconds)

moss_values = [0.08, 0.15, 0.3, 0.2, 0.1]

err = 1
q = 0.0015


def optimize_q(q):
    u = np.zeros((Xs, 1))
    u[(Xs//2)-(points_per_cell//2):(Xs//2)+(points_per_cell//2)] = 1
    D = D_eff(83, q, cell_um)
    nds, u = do_internode_diffusion(
        u, dx2, D, dt, b, cell_um, points_per_cell, ts)
    fig = plt.figure(0, figsize=(10, 10))
    fig.clf()
    fig, ax = plt.subplots(1, 1, figsize=(5, 5), num=0)

    ax.plot(np.arange(-(cells*cell_um)//2, +(cells*cell_um)//2, step=cell_um)+50,
            nds, marker='o', label='Model')

    ax.set_xlim(-250, 250)
    ax.set_ylim(-0.02, 1)

    ax.set_xlabel(r'$\mu m$')
    ax.plot(np.linspace(-200, 200, num=5), moss_values,
            label='Kitawga et al.', marker='o')

    err = np.square((nds[(len(nds)//2)-2:(len(nds)//2)+3] - moss_values)).sum()

    fig.suptitle('Q={0} \t err^2 = {1}'.format(q, err))
    fig.tight_layout()
    fig.savefig('./images/{0}.png'.format(str(q).replace('.', '_')))

```

```
    # Minimise sum of squared errors
    return err

result = leastsq(optimize_q, [q])
print(result)
```

1.1.2 DONE minimise difference of model to data

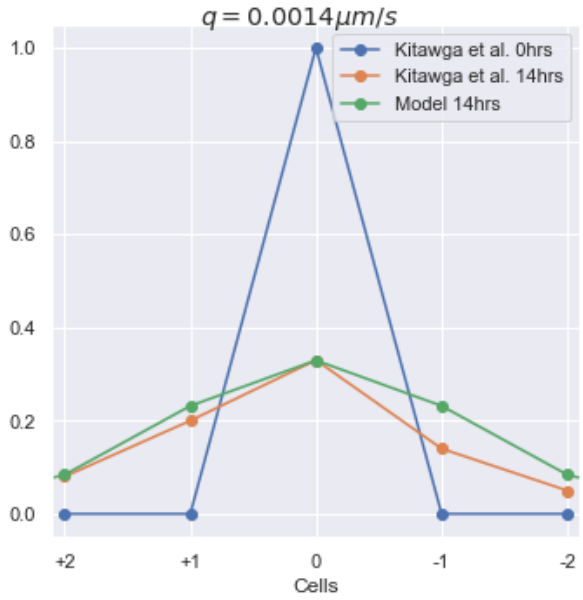

Figure 1: Testing against Kitagawa data

1.1.2.1 Mock treatment

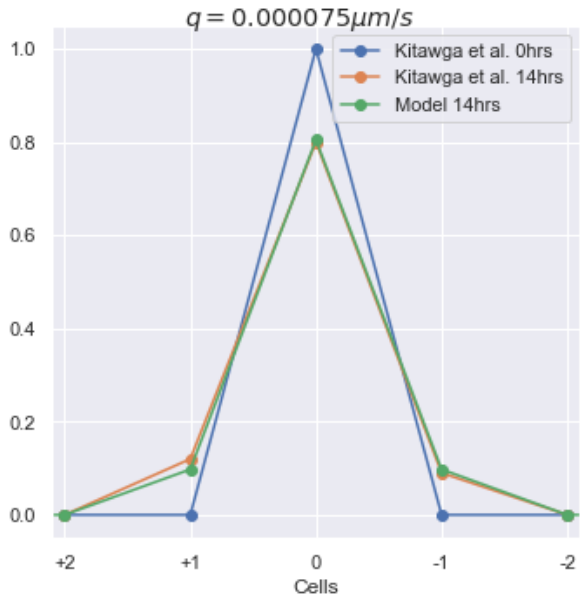

Figure 2: Testing against Kitagawa data with aba treatment

1.1.2.2 ABA treatment

1.2 TODO Does it make sense to use D_{eff} even for intra-cellular diffusion

1.2.1 TODO Test effect of using D vs D_{eff}

1.3 TODO Move towards fully node based graph model with proof of results

Still use a low Δx but just use dist = cell to approx

- Cell length / 10 works consistently

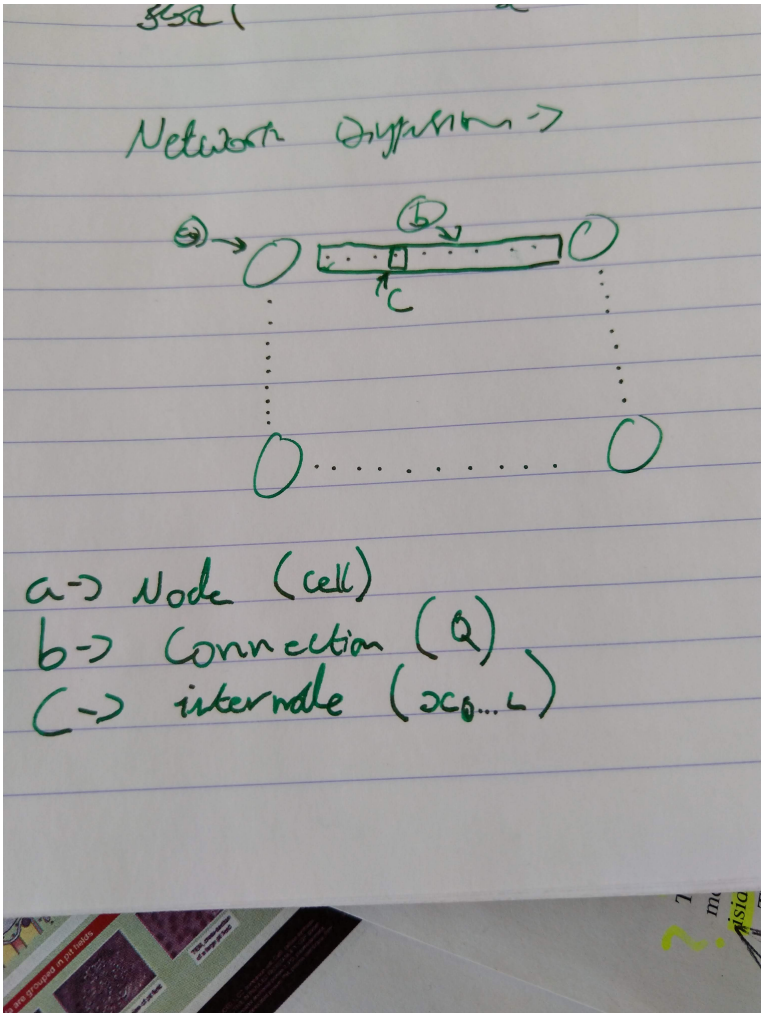

Figure 3: Sketch of idea

1.4 TODO Test D_{eff} vector idea

```
def diffuse_vectorise(un, g, b, dt, dx2, dy2, a):
    """
    Takes a state, rate of decay, production, delta time, delta space and
    flux of molecule. Uses these data to compute next time state
    """
    return (un[1:-1, 1:-1] + a *
            (((un[2:, 1:-1] - 2 * un[1:-1, 1:-1] + un[:-2, 1:-1]))/dx2 +
             ((un[1:-1, 2:] - 2 * un[1:-1, 1:-1] + un[1:-1, :-2]) / dy2))) *\
        g + b
```

1.5 DONE Finish presentation

1.6 TODO Finish chapter 1.5 of report

1.7 TODO Give practice presentation to Morris group

1.8 TODO Follow up on images of bombardment assays with Christine

1.9 TODO Use Deinum [1] A.69 with numerical solution

2 Questions

2.1 How best to make use of D_{eff}

Not sure how best to use, as it will give a linear curve for diffusion.

- Could probably do a D_{eff} with a k, for cell number, term

$$\alpha = \frac{Dql}{D + ql} \tag{1}$$

References

[1] E E Deinum. *Simple Models for Complex Questions on Plant Development*. PhD thesis, s.n., S.l., 2013. 00011.

[2] Munenori Kitagawa, Takumi Tomoi, Tomoki Fukushima, Yoichi Sakata, Mayuko Sato, Kiminori Toyooka, Tomomichi Fujita, and Hitoshi Sakakibara. Abscisic Acid Acts as a Regulator of Molecular Trafficking through Plasmodesmata in the Moss iPhyscomitrella patens/i. *Plant and Cell Physiology*, December 2018. ISSN 0032-0781. doi: 10.1093/pcp/pcy249.