

# *Modeling Temperature in Compost*

*Blerta and Marc and others! :-)*

*February 23, 2018*

## *Abstract*

Making compost relies on microbial processes to break organic matter down...

## *Introduction*

### *Problem Statement*

### *Farm Context*

### *Materials and Methods*

### *Session Outcomes*

1. Describe Advection Mathematically
2. Analyze 1-dimensional movement using advection equations
3. Describe Diffusions mathematically
4. Analyze 1-dimensional movement using Fick's law.
5. Two dimensional analysis of advection

## *Theory of Temperature Diffusion*

According to Wikipedia, advection can be described as as "velocity vector" and the del operator.

**Blerta, I sure wish I knew where these terms came from!**

$$\mathbf{u} \cdot \nabla = u_x \frac{\partial}{\partial x} + u_y \frac{\partial}{\partial y} + u_z \frac{\partial}{\partial z}. \quad (1)$$

where  $u = (u_x, u_y, u_z)$  is the velocity field, and  $\nabla$  is the del operator (note that Cartesian coordinates are used here).

The rest of the equation is... ???

Apparently, solving this equation is a nightmare! So, we will come up with an alternative way to predict concentration gradients...

### *Diffusion versus Advection*

Fick's first law relates the diffusive flux to the concentration under the assumption of steady state. It postulates that the flux goes from regions of high concentration to regions of low concentration, with a magnitude that is proportional to the concentration gradient (spatial derivative), or in simplistic terms the concept that a solute will move from a region of high concentration to a region of low concentration across a concentration gradient. In one (spatial) dimension, the law is:

$$J = -D \frac{\partial \varphi}{\partial x} \quad (2)$$

where

J is the "diffusion flux," of which the dimension is amount of substance per unit area per unit time, so it is expressed in such units as  $\text{mol} \cdot \text{m}^2 \cdot \text{s}^{-1}$ . J measures the amount of substance that will flow through a unit area during a unit time interval.

D is the diffusion coefficient or diffusivity. Its dimension is area per unit time, so typical units for expressing it would be  $\text{m}^2/\text{s}$ .  $\varphi$  (for ideal mixtures) is the concentration, of which the dimension is amount of substance per unit volume. It might be expressed in units of  $\text{mol}/\text{m}^3$ . x is position, the dimension of which is length. It might thus be expressed in the unit m.

*What is diffusion?*

*Why is diffusion important?*

*What do we know about diffusion?* In free solution and in the absence of any interactions with other macromolecules, the diffusion process is controlled by the size of the macromolecule as described by the Stokes-Einstein relation:

$$D_0 = \frac{k_B T}{6\pi\eta R_H} \quad (3)$$

where  $k_B$  is Boltzmann's constant, T the temperature in Kelvin,  $\eta$  the solvent viscosity, and  $R_H$  the hydrodynamic radius.

*Steady-state versus ??*

*What are current areas of research concerning diffusion and environmental issues?*

## Modeling with R

### One Dimensional Diffusion

#### Steady-state solution of 2-D PDEs

To develop our models, we will use the **rootSolve** Package. For this portion of our work, we will rely on the `steady.2D()` function that can solve steady-state, 2-dimensional problems.

(4)

a substance C is consumed at a quadratic rate ( $r \cdot C^2$ ), while dispersing in X- and Y-direction. At certain positions (x,y) the substance is produced (rate p).

The model is solved on a square (100\*100) grid. There are zero-flux boundary conditions at the 4 boundaries.

The term  $Dx \dots$ ,

i.e. it is the negative of the  $ux$  gradient, where the  $ux$  is due to diffusion. In the numerical approximation for the  $ux$ , the concentration gradient is approximated as the subtraction of two matrices, with the columns or rows shifted (e.g. `Conc[2:n,]-Conc[1:(n-1),]`).

The flux gradient is then also approximated by subtracting entire matrices (e.g. `Flux[2:(n+1),]-Flux[1:n,]`). This is very fast. The zero-flux at the boundaries is imposed by binding a column or row with 0-s.

```
library(rootSolve)
diffusion2D <- function(t,conc,par){
  Conc <- matrix(nr=n,nc=n,data=conc) # vector to 2-D matrix
  dConc <- -r*Conc*Conc # consumption
  BND <- rep(1,n) # boundary concentration

  # constant production in certain cells
  dConc[ii]<- dConc[ii]+ p

  #diffusion in X-direction; boundaries=imposed concentration

  Flux <- -Dx * rbind(rep(0,n),(Conc[2:n,]-Conc[1:(n-1),]),rep(0,n))/dx
  dConc <- dConc - (Flux[2:(n+1),]-Flux[1:n,])/dx

  #diffusion in Y-direction
  Flux <- -Dy * cbind(rep(0,n),(Conc[,2:n]-Conc[,1:(n-1)]),rep(0,n))/dy
  dConc <- dConc - (Flux[,2:(n+1)]-Flux[,1:n])/dy
```

```
return(list(as.vector(dConc)))
}
```

After specifying the values of the parameters, 10 cells on the 2-D grid where there will be substance produced are randomly selected (ii).

Figure 5: Steady-state solution of the nonlinear 2-Dimensional model

```
# parameters
dy <- dx <- 1 # grid size
Dy <- Dx <- 1.5 # diffusion coeff, X- and Y-direction
r <- 0.01 # 2-nd-order consumption rate (/time)
p <- 20 # 0-th order production rate (CONC/t)
n <- 100
# 10 random cells where substance is produced at rate p
ii <- trunc(cbind(runif(10)*n+1, runif(10)*n+1))
```

The steady-state is found using function `steady.2D`. It takes as arguments a.o. the dimensionality of the problem (`dimens`) and `lrw=1000000`, the length of the work array needed by the solver. If this value is set too small, the solver will return with the size needed. It takes about 0.5 second to solve this 10000 state variable model.

```
Conc0 <- matrix(nr=n, nc=n, 10.)
# print(system.time(
# not working yet...)

ST3 <- steady.2D(Conc0, func=diffusion2D, parms=NULL, pos=TRUE, dimens=c(n,n), lrw=1000000, atol=1e-10, rtol=1e-
```

The S3 image method is used to generate the steady-state plot.

### Diffusion Derivation

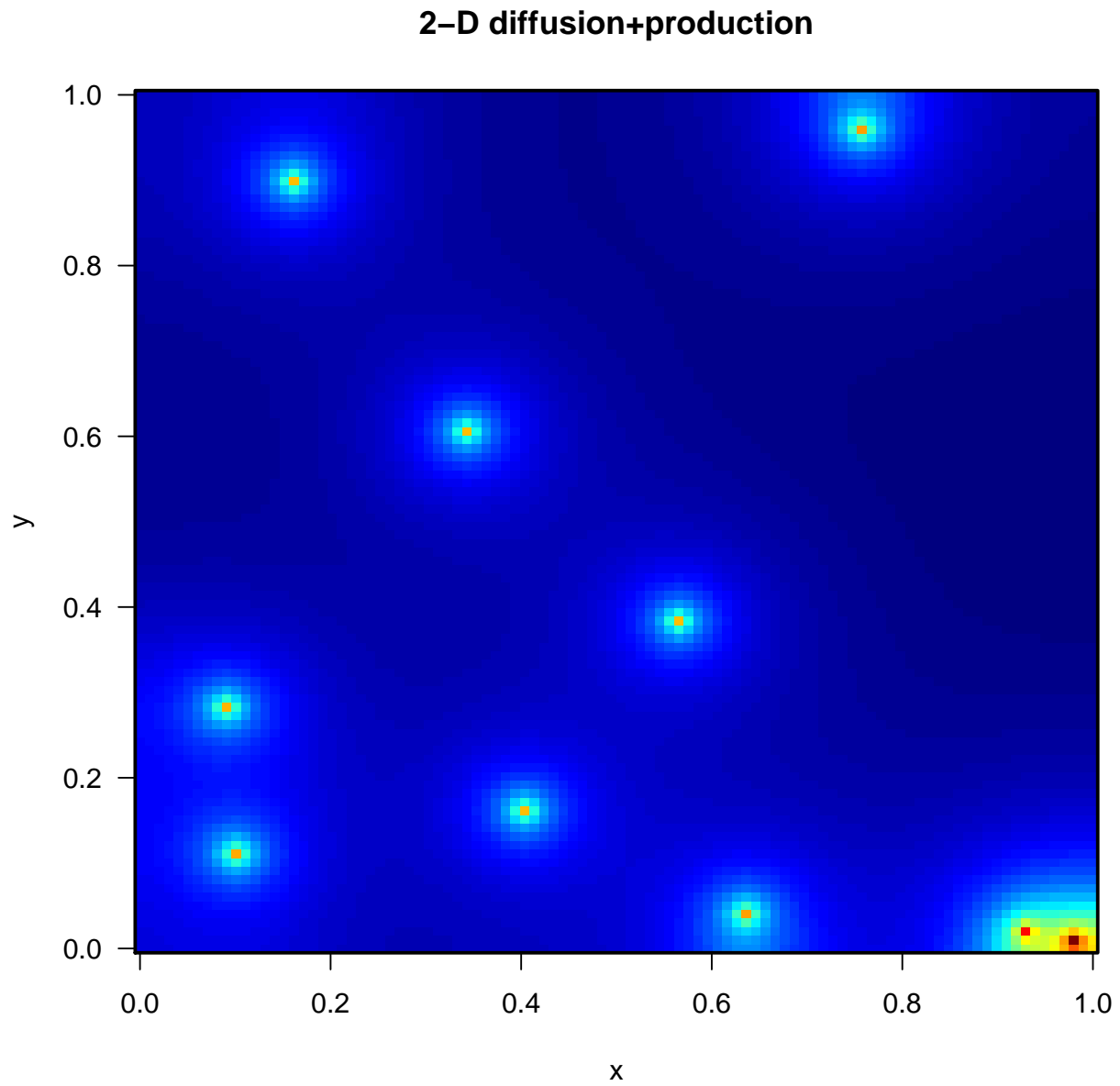
- a.) From 9.3, Special Case 2, we know that if  $A(x, t) = \bar{A}(x) \neq 0$  (if area does not change with time), then the equation can be written as

$$\frac{\partial c(x, t)}{\partial t} = -\frac{1}{\bar{A}(x)} \frac{\partial}{\partial x} [J(x, t) \bar{A}(x)] \pm \sigma(x, t) \quad (5)$$

We now need to find an equation for  $\bar{A}(x)$ . Since arc length equals radius times angle, we get

$$\bar{A}(r) = \theta rh \quad (6)$$

```
image(ST3,main="2-D diffusion+production", xlab="x", ylab="y", las=1)
```



where  $\theta$  is the angle of the arc,  $r$  is radial distance, and  $h$  is height of the section. Therefore we get the equation

$$\frac{\partial c(r, t)}{\partial t} = -\frac{1}{\theta r h} \frac{\partial}{\partial r} [J(r, t) \theta r h] \pm \sigma(r, t) \quad (7)$$

Since  $\theta$  and  $h$  are constants, we can factor them out to get

$$\frac{\partial c(r, t)}{\partial t} = -\frac{1}{r} \frac{\partial}{\partial r} [J(r, t) r] \pm \sigma(r, t) \quad (8)$$

- b) Extending the principle applied in part (a), we first need to find  $\bar{A}(R)$ . Since  $\theta$  is small, we can approximate cross sectional area by taking horizontal arc length times vertical arc length. Therefore we get the equation

$$\bar{A}(R) = \theta_1 \theta_2 R^2 \quad (9)$$

where  $\theta_1$  is the horizontal angle of the arc,  $\theta_2$  is the vertical angle of the arc, and  $R$  is radial distance. Combining this with equation (1) from above, we get

$$\frac{\partial c(R, t)}{\partial t} = -\frac{1}{\theta_1 \theta_2 R^2} \frac{\partial}{\partial R} [J(R, t) \theta_1 \theta_2 R^2] \pm \sigma(R, t) \quad (10)$$

Since  $\theta_1$  and  $\theta_2$  are constants, we can simplify the equation as such:

$$\frac{\partial c(R, t)}{\partial t} = -\frac{1}{R^2} \frac{\partial}{\partial R} [J(R, t) R^2] \pm \sigma(R, t) \quad (11)$$

- c) Part A: In order to obtain the equations in 9.5, we apply Fick's law:

$$J = -\mathcal{D} \nabla c \quad (12)$$

In this case, we use the one-dimensional version:

$$J = -\mathcal{D} \frac{\partial c}{\partial x} \quad (13)$$

Applying this to equation (4), we get

$$\frac{\partial c(r, t)}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} [\mathcal{D} \frac{\partial c}{\partial r} r] \pm \sigma(r, t) \quad (14)$$

Simplifying, and assuming that no particles are created or eliminated at the source, we get:

$$\frac{\partial c(r, t)}{\partial t} = \frac{\mathcal{D}}{r} \frac{\partial}{\partial r} \left( \frac{\partial c}{\partial r} r \right) \quad (15)$$

Part B: Once again, we apply Fick's law in one dimension to get

$$\frac{\partial c(R, t)}{\partial t} = \frac{1}{R^2} \frac{\partial}{\partial R} [\mathcal{D} \frac{\partial c}{\partial R} R^2] \pm \sigma(R, t) \quad (16)$$

We again simplify, assuming no particles are created or destroyed at the source.

$$\frac{\partial c(R, t)}{\partial t} = \frac{\mathcal{D}}{R^2} \frac{\partial}{\partial R} \left( \frac{\partial c}{\partial R} R^2 \right) \quad (17)$$

## Chang 1997 Key Points

### Data

- Size of halos measured with a ruler over constant time intervals
- Experiment repeated three times to calculate average values used in model

### Methodology

- Mathematical model formed using Fick's law
- $c(r,t)$  = lipase concentration as a function of time and radial distance
- Used regression analysis to fit hindered diffusion coefficients and threshold values of lipase concentration as parameters. The finite difference method outlined in Constantinides, 1987 was used specifically.
- Amount of lipase in plate at each time calculated by numerical integration. The total amount of lipase varied by 2.5%, confirming the appropriateness of the mathematical model used.

### Discretization

Fick's second law of molecular diffusion, expressed in cylindrical coordinates.

$$\begin{aligned}\frac{\partial C}{\partial t} &= D\Delta^2 C \\ &= D\left(\frac{1}{r}\frac{\partial C}{\partial r} + \frac{\partial^2 C}{\partial r^2}\right)\end{aligned}$$

Initial conditions:

$$\begin{cases} C = C_0 & \text{for } R_w < r < R_r \\ C = 0 & \text{for } r < R_w \text{ \& } R_r < r \end{cases} \quad (18)$$

where  $D$  is the effective hindered diffusion coefficient,  $C_0$  is the feed enzyme concentration,  $R_w$  is the radius of the well, and  $R_r$  is the outer radius of the absorption ring. The lipase concentration as a function of time and radial distance can be solved using the finite difference method.

$$\frac{1}{\Delta t}(C_{i,j+1} - C_{i,j}) = \left[ \frac{1}{i\Delta r} \frac{D}{2\Delta r}(C_{i+1,j} - C_{i-1,j}) + \frac{D}{\Delta r^2}(C_{i+1,j} - 2C_{i,j} + C_{i-1,j}) \right] \quad (19)$$

$$\frac{1}{\Delta t}C_{i,j+1} = \frac{1}{\Delta t}C_{i,j} - \frac{2D}{\Delta r^2}C_{i,j} + \frac{D}{2i\Delta r^2}(C_{i+1,j} - C_{i-1,j}) + \frac{D}{\Delta r^2}(C_{i+1,j} + C_{i-1,j}) \quad (20)$$

$$C_{i,j+1} = (1 - \frac{2D\Delta t}{\Delta r^2})C_{i,j} + \frac{D\Delta t}{\Delta r^2}[(\frac{1}{2i} + 1)C_{i+1,j} - (\frac{1}{2i} - 1)C_{i-1,j}] \quad (21)$$

For convergence, to hold:

$$\frac{D\Delta t}{\Delta r^2} \leq \frac{1}{2} \quad (22)$$

Lastly, the amount of lipase can be calculated using

$$A = \int_{R_w}^{R_h} 2\pi r l C(r) dr \quad (23)$$

where  $A$  is the total amount of lipase in terms of enzyme activity,  $R_w$  and  $R_h$  are the radius of well and halo, respectively,  $C(r)$  is the concentration at the radial distance of  $r$ , and  $l$  is the thickness of agar plate.