Modeling Diffusion Equations... A simple tutorial

Carolina Tropini*
Biophysics Program, Stanford University
(Dated: November 24, 2008)

I. THE DIFFUSION EQUATION IN ONE DIMENSION

In our context the diffusion equation is a partial differential equation describing how the concentration of a protein undergoing diffusion changes over time and space. For a given protein species the behavior in 1D could be described as follows:

$$\frac{\delta P}{\delta t} = D\frac{\delta^2 P}{\delta x^2} + f(P) \tag{1}$$

where P is the concentration of protein, D is the diffusion constant and f(P) is a function that determines how P behaves in relation to itself. For instance f(P)=P would indicate a linear positive feedback etc.

A. How to model this computationally

If you go back to first year calculus, you will remember that from the definition of a derivative we can write equation 1 as follows (ignoring the spatial component):

$$\frac{\delta P}{\delta t} = \frac{P^{t+1} - P^t}{\Delta t} \tag{2}$$

where the derivative is evaluated in the limit $\Delta t \to 0$. We can do the same for the spatial derivative although to simplify things we will use a symmetric derivative so that the second derivative is centered at $x + \frac{1}{2}$ and $x - \frac{1}{2}$ as follows (ignoring the temporal component):

$$\frac{\delta^2 P}{\delta x^2} = \frac{\delta}{\Delta x}\frac{(P_{x+\frac{1}{2}} - P_{x-\frac{1}{2}})}{\Delta x} = \frac{1}{\Delta x}\left(\frac{P_{x+1} - P_x}{\Delta x} - \frac{P_x - P_{x-1}}{\Delta x}\right) \tag{3}$$

this reduces to:

$$\frac{\delta^2 P}{\delta x^2} = \frac{P_{x+1} + P_{x-1} - 2P_x}{\Delta x^2} \tag{4}$$

therefore, equation 1 can be described as:

$$\frac{P_x^{t+1} - P_x^t}{\Delta t} = D\frac{P_{x+1}^{t+1} + P_{x-1}^{t+1} - 2P_x^{t+1}}{\Delta x^2} + f(P_x^t) \tag{5}$$

Let $\alpha = \frac{D\Delta t}{\Delta x^2}$; we can separate the t+1 and t terms as below:

$$P_x^t + \Delta t \cdot f(P_x^t) = -\alpha P_{x-1}^{t+1} + (1+2\alpha)P_x^{t+1} - \alpha P_{x+1}^{t+1} \tag{6}$$

If we imagine P as a vector of positions $x = 0$ to x_n we can visualize the equation as $P^t + \Delta t \cdot f(P^t) = AP^{t+1}$ where A is:

---

*Electronic address: tropini@stanford.edu

$$\begin{pmatrix} 1+\alpha & -\alpha & 0 & 0 & 0 & ... \\ -\alpha & 1+2\alpha & -\alpha & 0 & 0 & ... \\ 0 & -\alpha & 1+2\alpha & -\alpha & 0 & ... \\ ... & ... & ... & ... & ... & ... \\ ... & 0 & -\alpha & 1+2\alpha & -\alpha & 0 \\ ... & ... & 0 & 0 & -\alpha & 1+\alpha \end{pmatrix}$$

Then we can easily solve for P^{t+1} as a function of P^t by performing matrix division:

$$P^{t+1} = A^{-1}(P^t + \Delta t \cdot f(P_x^t)) \tag{7}$$

B. What this might look like in MatLab

In Program 1 below I am trying to solve an arbitrary number of diffusion equation which look like this: $\frac{\delta C}{\delta t} = D\frac{\delta^2 C}{\delta x^2} + f(C)$ The boundary conditions are no flux at the distal end and R0 at the x=0 end. The simulation occurs over time T and the initial conditions are determined by c0. The user needs to input the diffusion constant D, the total iteration time T, the function 'Interactions' which describes the behavior of f(C),the initial value of the concentration c0 and a the boundary condition R which describes the behavior of the derivative of C at the boundary x=0. The matrix A and α were defined above.

We are now ready to extend our diffusion equation to several species.

II. THE DIFFUSION EQUATION FOR MULTIPLE SPECIES

Converting from one to multiple species only requires an extension of the analysis above to multiple species. For instance we could have two proteins interacting, P1 and P2 and let's say that they negatively affect each other but have a self linear positive feedback.

$$\frac{\delta P_1}{\delta t} = D\frac{\delta^2 P_1}{\delta x^2} + P_1 - P_2 \tag{8}$$

$$\frac{\delta P_2}{\delta t} = D\frac{\delta^2 P_2}{\delta x^2} - P_1 + P_2 \tag{9}$$

You can see that this might be arbitrarily complicated, nevertheless in our model we just need to expand the PDE solver to handle Neq equations and to define differently the Interactions program to handle all the cross terms. An example is described in Program 3. The interactions program may now be defined to perform a certain calculation depending on which equation n the loop is going through.

Program 1 Matlab 1D diffusion equation code - single species -

```
function [C,X,A]=PDE_Solver(D,T,Interactions,c0,R1)

%here we define our timestep
dt=0.1;
%here we define the spacial step
dx=0.1;
%Total distance
X=10;
alpha=D*dt/(dx)^2;
R=zeros(X/dx,1);
R(1)=R1;

A=eye(X/dx).*(1+2*alpha);
A(1,1)=1+alpha;
A(1,2)=-alpha;
A(X/dx,X/dx-1)=-alpha;
A(X/dx,X/dx)=1+alpha;
        for ii=2:X/dx-1
          A(ii,ii-1)=-alpha;
          A(ii,ii+1)=-alpha;
        end
%here we also set up the initial conditions, given by a uniform distribution of c0 and some noise added.
C=zeros(X/dx,T/dt);
C(:,1)=c0*ones(X/dx,1)+rand(X/dx,1)*0.01;

%this below will be used to check convergence
Cmean=zeros(T/(5*dt));
check=1;

%here we now go through our time steps
for t=1:(T/dt-1)


%kn defines the part of the equation that is not determined by diffusion but rather by interactions.
        kn=feval(Interactions,C(:,t));

        C(:,t+1)=A(:,:)\(C(:,t)+dt*kn+R(:,n));

%this part below plots to see if get convergence
         if t/5==check
             Cmean(check,n)=mean(C(:,t+1));

                    check=check+1;
                end

        end

    end

end
%here we plot
plot((1:check)*5*dt,Cmean(1:check,1))
%set labels
xlabel('Time (every 5 time intervals)') ylabel('Mean concentration of species 1')
figure
plot((1:X/dx)*dx,C(:,T/dt),'r')
xlabel('Lenght') ylabel('Concentration')
h = legend('Species 1');
set(h,'Interpreter','none')

end
```

Program 2 Matlab 1D diffusion equation code - single species -

```
%here is what the program Interactions might look like if f(P)=P^2:

function kn = Interactions(C)

   kn = 1*C(:).^2;
end
```

Program 3 Matlab -

```
function [C,X,A,Cmean,R]=PDE_Solver(Neq,D,T,Interactions,c0,R1)

dt=0.1;
dx=0.1;
X=10;
alpha=D*dt/(dx)^2;
R=zeros(X/dx,Neq);
R(1,:)=R1;

A=zeros(X/dx,X/dx,Neq);
C=zeros(X/dx,T/dt,Neq);
for i=1:Neq

    A(:,:,i)=eye(X/dx).*(1+2*alpha(i));
    A(1,1,i)=1+alpha(i);
    A(1,2,i)=-alpha(i);
    A(X/dx,X/dx-1,i)=-alpha(i);
    A(X/dx,X/dx,i)=1+alpha(i);

        for ii=2:X/dx-1
         A(ii,ii-1,i)=-alpha(i);
         A(ii,ii+1,i)=-alpha(i);
        end
%here we also set up the initial conditions.
        C(:,1,i)=c0(i)*ones(X/dx,1)+rand(X/dx,1)*0.01;

end

Cmean=zeros(T/(5*dt),Neq);
check=1;

for t=1:(T/dt-1)

    for n=1:Neq

        kn=feval(Interactions,C(:,t,:),n);

        C(:,t+1,n)=A(:,:,n)\(C(:,t,n)+dt*kn+R(:,n));

        if t/5==check
            Cmean(check,n)=mean(C(:,t+1,n));
                if n==Neq
                    check=check+1;
                end

        end

    end


end

plot((1:check)*5*dt,Cmean(1:check,1))
xlabel('Time (every 5 time intervals)') ylabel('Mean concentration of species 1')
figure
plot((1:X/dx)*dx,C(:,T/dt,1),'r',(1:X/dx)*dx,C(:,T/dt,2),'g')
xlabel('Lenght') ylabel('Concentration')
h = legend('Species 1','Species 2',2);
set(h,'Interpreter','none')

end
```