

Alternating Direction Explicit – Barakat et al. 1966

Previously the FTCS method was used to estimate the solution to the diffusion equation at each time step. Based on the explicit nature of this algorithm the method is conditionally stable based on the following criteria (any time step larger will cause oscillations in the solution).

$$\lambda = D\Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \leq \frac{1}{4}$$

Using the largest value of diffusivity in our model (TNF, IL10, and chemokines = $\sim 5 \times 10^{-8} \text{ cm}^2/\text{s}$), as this value limits the time step, the estimated maximum time step is 12 seconds. Previous iterations of the ABM have reduced this time step to 6 seconds to maintain the accuracy of the solution. This small time step leads to a large amount of iterations, and thus computation time, needed in order to simulate a 10 min agent time step. Many different classical algorithms were explored as possible replacements for FTCS (Crank-Nicolson, Peaceman and Rachford etc.) but most are implicit methods that require the assembly of a system of algebraic equations and subsequent iterative solutions to convergence by linear algebra methods. This requires a significant amount of calculations and the incorporation of C++ libraries that have built in LA solvers (i.e. LAPack). Herein, we present a more accurate and computationally faster algorithm for estimating the solution to the diffusion equation called the alternating direction explicit method (ADE) (Barakat et al. 1966), which retains the explicit nature of FTCS but manages to obtain unconditional stability that is normally a characteristic of implicit solvers.

For simplicity the explanation is given in 2-D but the algorithm is easily adapted to 3-D. The 2-D diffusion equation is given as:

$$\frac{\partial C}{\partial t} = D \left[\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right]$$

In our hybrid ABM rows are defined by the subscript i and columns are defined by the subscript j. Let u and v be finite difference representations of the equation above.

$$\begin{aligned} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} &= \frac{(u_{i+1,j}^n - u_{i,j}^n) + (u_{i-1,j}^{n+1} - u_{i,j}^{n+1})}{(\Delta x)^2} + \frac{(u_{i,j+1}^n - u_{i,j}^n) + (u_{i,j-1}^{n+1} - u_{i,j}^{n+1})}{(\Delta y)^2} \\ \frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} &= \frac{(v_{i+1,j}^{n+1} - v_{i,j}^{n+1}) + (v_{i-1,j}^n - v_{i,j}^n)}{(\Delta x)^2} + \frac{(v_{i,j+1}^{n+1} - v_{i,j}^{n+1}) + (v_{i,j-1}^n - v_{i,j}^n)}{(\Delta y)^2} \end{aligned}$$

Where the solution at any time point is given by the arithmetic average of u and v.

$$C_{i,j}^{n+1} = \frac{(u_{i,j}^{n+1} + v_{i,j}^{n+1})}{2}$$

Upon rearrangement and simplification u and v become the following equations with similar constants.

$$u_{i,j}^{n+1} = au_{i,j}^n + b(u_{i,j-1}^{n+1} + u_{i,j+1}^n) + c(u_{i-1,j}^{n+1} + u_{i+1,j}^n)$$

$$v_{i,j}^{n+1} = av_{i,j}^n + b(v_{i,j-1}^n + v_{i,j+1}^{n+1}) + c(v_{i-1,j}^n + v_{i+1,j}^{n+1})$$

$$a = \frac{\left[1 - D\Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)\right]}{\left[1 + D\Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)\right]} = \frac{1 - \lambda}{1 + \lambda}$$

$$b = \frac{\left[D\Delta t \left(\frac{1}{\Delta x^2}\right)\right]}{1 + \lambda} \quad c = \frac{\left[D\Delta t \left(\frac{1}{\Delta y^2}\right)\right]}{1 + \lambda}$$

In the case of u the iterations proceed forward in the i,j direction while in the case of v the iterations occur backward in i,j. This marching sequence is necessary since the ADE method uses time points at n+1, which must fall ‘behind’ the direction of the iterator thus allowing the method to proceed explicitly.

The ADE method is shown to be unconditionally stable in Barakat et al. 1966 by using Von Neumann analysis. This allows the time step to be chosen on an accuracy basis instead of a stability basis. The FTCS and ADE methods were implemented in MATLAB® on a 100x100 grid and 10,000 molecules were initialized at grid space [50,50]. Shown in Figure 1 is the concentration profile, through the center slice, after 12 hrs of simulation. There is no difference in solution accuracy between the FTCS method with a time step of 6 seconds and the ADE method with a time step of 60 seconds and little change in accuracy of the solution with upwards of 120 seconds time steps.

These times steps are applicable only when diffusion is occurring without reactions. When receptor-ligand dynamics of TNF and IL10 are added the diffusion equation becomes:

$$\frac{\partial C}{\partial t} = D \left[\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right] + Rxn_{Deg} + Rxn_{R-L}$$

Each portion of this equation can be solved separately by the principle of operator splitting. Since they are occurring on different relative time scales, where receptor-ligand

processes happen faster than diffusion processes, care must be taken when determining the appropriate time step for diffusion and the appropriate time step for reaction-ligand dynamics. The receptor-ligand dynamic time step is taken as 6 seconds to maintain the accuracy of the forward-euler approximation to the ODEs. The largest diffusion time step to maintain mathematical accuracy is estimated by calculating the diffusion time (the relative time it takes a molecule to move a certain distance) given below:

$$t_D = \frac{L^2}{D}$$

Based on the size of a grid compartment, 20 μm , the diffusion time is estimated to be ~ 80 seconds. This means that a molecule will take around 80 seconds for it to leave its current grid space, thus any diffusion time step higher than 80 seconds will be a less physical representation of the diffusion process (since the receptor-ligand dynamics depend on the concentration in the grid space at shorter times). Therefore, a diffusion time step of 60 seconds was chosen to maintain solution accuracy and to represent the physical time scale of diffusion accordingly.

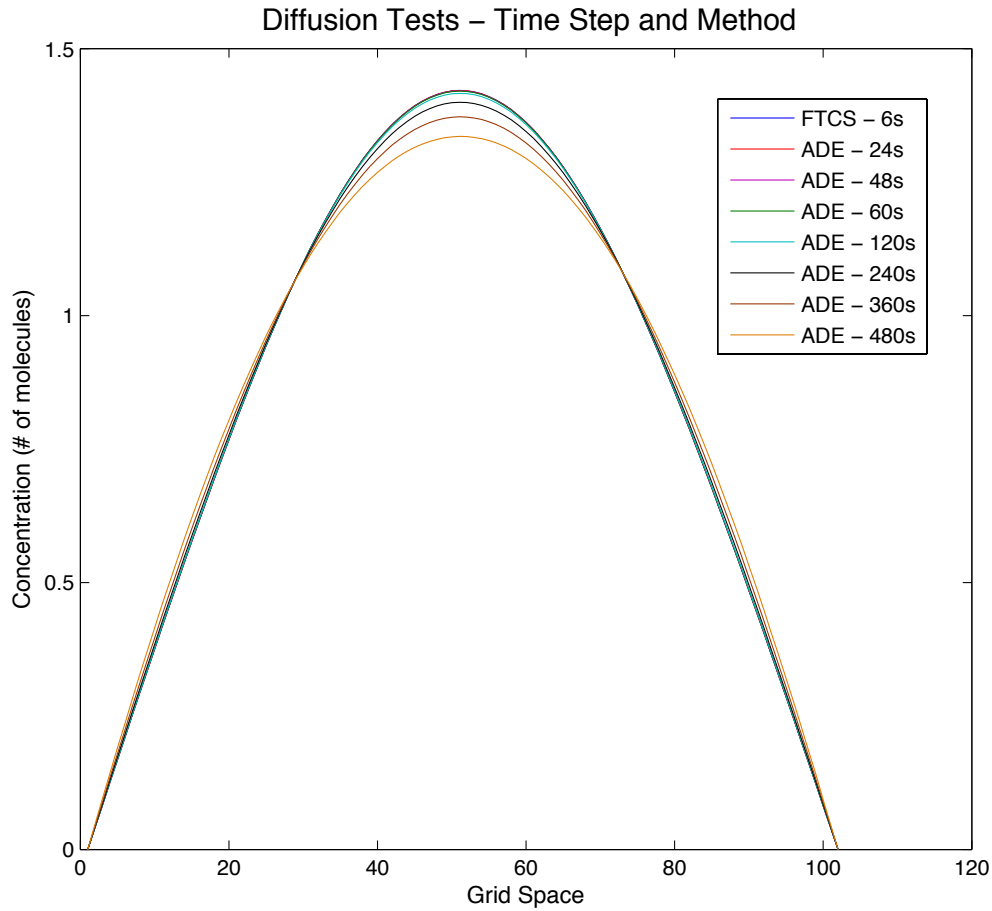


Figure 1. Testing the FTCS and ADE methods in MATLAB for different time steps. 100x 100 grid with 10,000 molecules initialized at [50,50] at time zero.