Diffusion as a Rate Limiting Effect On Microtubule Dynamic Instability

Adam Siegel

2010 NSF/REU Program
Physics Department, University of Notre Dame

Mark Alber
Holly Goodson
DIFFUSION AS A RATE LIMITING EFFECT ON MICROTUBULE DYNAMIC INSTABILITY

ADAM D. SIEGEL1,3, MARK ALBER2,3, HOLLY GOODSON1,4, ZILIANG XU2,3
Submitted on August 7, 2010

ABSTRACT

Microtubules play an integral part in cell organization and subcellular structure, including the transportation of organelles and separation of chromosomes during mitosis. The key aspect of microtubules behavior that makes this possible is dynamic instability, stochastic growth and shortening of these proteinaceous structures which allows for exploration of the cell. Based on a previous study (Odde 1997), it is clear that diffusion does not have a rate limiting effect on microtubule assembly. However this paper does not address the effect of diffusion on dynamic instability. We use the diffusion equation as the mathematical model for the dynamics of the molecular interactions of tubulins, coupled with a mesoscopic scale microtubule model to study whether diffusion has any effect on the behavior of the microtubules during dynamic instability. Namely, we mesh a second order accurate Crank-Nicolson finite difference scheme for solving diffusion equation with the mesoscopic microtubule Monte Carlo model to understand the conditions resulting in diffusion having a restrictive effect. Parallelization and code optimization techniques are implemented due to deal with long time steps, such that the effects of dynamic instability become prominent (namely, during steady state interactions of the microtubules with its environment). This research may have profound implications in many respects, as microtubules are the target for numerous anticancer drugs, and understanding their interactions are imminent for anticancer therapy.

Subject headings: Computational Biophysics — Computational Cellular Dynamics, Microtubules, Cellular Dynamic Instability, Microtubule Dynamic Instability

1. INTRODUCTION

Microtubules (MTs) are responsible for routinely carrying out many cellular tasks including the transportation of organelles from different parts of the cell by consecutively bonding down the microtubule, support the cell by forming the rigid cores in sub-cellular bodies (such as those in the mitotic spindle), and participate in mitosis by bonding to and pulling apart the chromosomes before the cell divides (Howard & Hyman 2006, Howard and Hyman 2009). Considerable interest is taken in the need understand how MTs behave and regulate the cell. The motivation to understand this comes from the necessity to understand how the cytoskeleton is regulated, because this will help facilitate the development of microtubule directed drugs for things like cancer therapy. Since MTs are responsible for the separation of the chromosomes during mitosis, cancer directed drugs target microtubular networks. It is necessary and imminent to understand the exact mechanism behind microtubule dynamic instability to determine exactly which factors effect their resultant behavior, and to what extent.

This outwards bonding of protofilaments is a result of the agitation experienced by the MT when the longitudinal bonds are lost. GTP tubulin acts as a more stable environment for the MT than its GDP counterpart. Spontaneous hydrolysis can give rise to the loss of the GTP cap, causing catastrophe. The main goal of this research is to understand exactly how the process of dynamic instability takes place; the precise mechanism of the loss of the GTP cap, and under which conditions this will occur.

1.1. Microtubules

MTs are long chains of proteins composed of $\alpha$ and $\beta$ tubulin dimers, and are found in the cytoskeleton of all living cells (see Figure 1 (a)). These tubulin dimers are proteins that assemble back to back by hydroylizing Guanosine triphosphate (GTP) to Guanosine diphosphate (GDP) via a chemical reaction that occurs spontaneously in order to minimize their free energy. In doing so, these dimers form long polymer chains that can reach tens of microns in length, called protofilaments. In addition to bonding laterally, these dimers also possess the capability to bond horizontally- 13 (sometimes more, sometimes less) protofilaments bond together horizontally to comprise one microtubule. Whether or not these bonded protofilaments first form as a sheet which folds in upon itself via some chemical interruption or if it forms initially as a cylindrical structure is still an open question. At the seam of a closed MT, there is a three monomer shift between the initial and last of the protofilaments, which results in an A lattice, such that the $\alpha$ monomers bond directly to the $\beta$ monomers. Structural analysis of the MT configuration advocates longitudinal bonding between chains of dimers (heterodimers) is much stronger than lateral bonds (Margolin et. al. In Prep. 2010). This observation leads to belief that loss of a GTP cap promotes catastrophe (see below).

After the initial formation of these MTs, stochastic processes take over- they irregularly (but repeatedly), undergo catastrophe and rescue (Figure 1 (b)). Collectively, this unstable “dance” is referred to as microtubule dynamic instability (henceforth, DI). Diffusion in cells is NOT a fast enough process to be able to distribute all nutrients to the proper location within a cell, nor is the cell small enough that nutrients within the cell will be able to get to their desired location without be-
ing brought there. For instance, single nerve cells extending from the spine to fingers in humans would otherwise be unable to transport nutrients throughout the entire cell, since the diffusivity of tubulin through cytoplasm is so low in comparison to the length of cell. Therefore, the purpose of DI is to commission the cytoskeleton to seek out and explore the area inside the cell to find the nutrients for transportation, as well as do things like bond to the kinetochores of chromosomes during mitosis (Howard and Hyman 2009).

In vitro (in living cells) biologists witness the stochastic growth and shrinkage of these microtubules. These MTs are anchored at centrosomes (microtubule “growing stations” from which they nucleate). These centrosomes are composed of centrioles, which are in turn surrounded by pericentriolar proteins (Howard and Hyman 2003). They remain fixed to the centrosomes throughout the cells life growing outwards into the cytoplasm, therefore differing from microtubules observed in vitro (during experiments). In vitro, microtubules are usually grown separated from the centrosome and beginning with some concentration of tubulin dimers, they irregularly assemble and disassemble, where in the process they can lose or gain thousands of dimers in one event, which is similar to the characteristics of instability witnessed in vivo). The exact cause of the sudden change from growth to shrinkage or vice versa has still not been experimentally verified, however there is theoretical evidence that this process is caused by the gain or loss of a GTP-Tubulin cap. As stated above, it has been theorized that the horizontal bond of a GTP cap is more stable than one composed of GDP. Therefore, when this cap disappears, the protofilaments bend outwards as dimers are lost (see Figure 1 (c)).

1.2. Probable Effects on Microtubule Dynamics

Previous work by Odde (Odde 1997, amongst others) has led to the conclusion that diffusion does not have a rate limiting effect on MT assembly. This is due to the fact that tubulin can diffuse rapidly relative to the assembly rate of MTs. Using the diffusivity of sea urchin eggs as the average diffusivity of tubulin ($5.9 \times 10^{-12}$, Salmon et. al. 1984), the average elongation of a MT will be about an order of magnitude less than the average displacement of tubulin in the same time (Odde 1997). In addition, it has been shown that the rate limiting effect in association is not likely to be translational diffusion, but other factors (Northrup and Erickson 1992). Researchers in the field have thus far only specifically pointed this out with respect to MT assembly.

Many in the field remain convinced that diffusion will plan a prominent role in MT dynamic instability. For one, the rates of growth and shrinkage may not be the same during dynamic instability as they are during phases of initial growth. Therefore, although the diffusivities measured by Salmon et. al. may be general, the growth rate of MTs may not be the same. As a result, if tubulin is not present, it won’t be able to attach. This would result in the inability of MTs to grow. The probability of the loss of a GTP cap may be such that while the MT is not growing (or loosing tubulin), the GTP cap is lost due to hydrolysis, giving the MT no choice but to shrink. Our plan will be to run the simulation with and without diffusion, and see whether the MTs interact differently with tubulin during dynamic instability.

1.3. Monte Carlo Methods

The Monte Carlo method for numerical simulation is a powerful method that relies on consecutive stochastic (random) sampling in order to emulate an event that occurs seemingly adventitiously. This method can be utilized for a variety of different things, from computing basic integrals to our case of using it for MT simulation. For instance, consider the integral

$$\int_0^1 x^2 \, dx. \quad (1)$$

This is easily evaluated analytically ($\frac{1}{3}$). However, There are many integrals (even some trivial looking) that cannot be evaluated analytically, because the antiderivatives do not exist. For simplicity, we utilize the Monte Carlo method to solve the integral in equation 1. We chose a square zone that is
1 unit length squared: $0 \leq x, y \leq 1$ (see Figure 2 (a)). We pick some number of points to randomly put in the box (Figure 2 (b) shows the case of 10, (c) the case of 100, and (d) the case of 1000). Various random points are chosen by utilizing a random number generator on a computer. Then, the amount of points that lie below the curve is summed, as is the amount of points that lie above the curve. Taking the ratio of the amounts of points that fall under the curve to those that lie above the curve will give an approximation to the integral.

The rest of the figures (more data points) are not shown, however this can be repeated for any number of points. Figure 3 shows a study when the amount of points are increased (up to 10k). A decrease in accuracy was not observed. Although very useful for evaluating integrals with no analytic solution, we do not utilize Monte Carlo techniques for this reason. For our purposes, we are interested in Monte Carlo simulations because we can utilize random numbers to see if specific events will occur. For instance, at any given time step in the simulation, tubulin can attach to any of the protofilaments. Since we assume diffusion to be instantaneous, if the tubulin concentration is above a specific level, we randomly choose if it will attach or not. The same can be done for hydrolysis of GTP tubulin to GDP tubulin. This is what gives rise to catastrophe: the GTP tubulin cap is randomly lost and the MT falls apart. Ultimately, mathematicians would like to come up with some mathematical model to decide if events such as these will occur, and under which given conditions.

2. MESOSCOPIC MODEL

The current model being used is the result of work done mainly by Dr. Ivan Gregoretti for his PhD thesis in December 2007 (Gregoretti 2007). It is a mesoscopic model, meaning it does not model individual monomers or dimers, but “bricks” of previously bonded dimers. A more advanced dimer-scale model developed at the University of Notre Dame does exist, however it does not yet have the capability of simulating multiple interacting microtubules as the mesoscopic scale model does, in addition to running much slower due to the fact that it models events on much smaller scales (Margolin et. al. in Prep 2010). Therefore, the mesoscopic scale model was chosen for diffusion to be incorporated.

An important feature of this model is that it actually shows dynamic instability. Starting with first principles and a random number generator (for Monte Carlo purposes), information about MTs interacting with tubulin and the rest of the cellular environment was incorporated into the model. Multiple MTs were simulated to interact with each other to see what would happen (Gregoretti et. al. 2006). DI was readily achieved in steady state, as shown in Figure 4.

The simulation starts out with MTs of essentially zero length. Initially, every MT has just one GTP brick “cap” that will cause initial growth in a semi-stable manner (until the GTP cap is lost). At each time step, the MT can either grow, shrink, or pause at its length. The simulation picks MTs randomly (Monte Carlo) and decides what will happen. The behavior is determined by the current tubulin concentration. For each randomly chosen MT, the tubulin concentration is used modify the MT’s activities. Based on the tubulin concentration that is returned, the simulation decides if the MT can grow, shrink, or stay at its current length. Note that the amount of tubulin available locally is the same at each node, since diffusion is not taken into account. Physically, this means that tubulin diffuses instantaneously, which may not always be a physically realizable situation (see Section 3.2).

1.2). The simulation also currently utilizes physical boundary conditions (for example, the MT can now grow outside of the cell, etc.). After the length is checked, the type of tubulin is modified. MTs are picked randomly and their GTP caps have
Fig. 3.—Increasing the amount of random points brings the approximation to the integral closer and closer. The green line represents the actual value and the random data points are in red. This study was done up to 50k random points (data point not shown), and a decrease in accuracy was not observed. Each point represents the average of three trials.

Fig. 4.—Mesoscopic Monte Carlo simulation results in arbitrary units of time. This simulation was run with 128 MTs, each with a maximum length of 512 arbitrary units of length. (a) $T = 70$, this is prior to the MTs achieving steady state. (b) $T = 1548$, Effects of DI are just starting to occur as steady state interactions are beginning. (c) $T = 3329$, Effects of DI are prominent and steady state interactions of MTs are fully developed.

...the cell under ordinary conditions is very high relative to other properties within the cell, such as distance tubulin must travel, speed at which it travels, etc. Our most current attempts to do this are given in section 4.

3. DIFFUSION

Diffusion is the random expansion of particles from an area of higher concentration to an area of lower concentration. It is a general problem that can be applied to many physical situations such as the expansion of a color dye in a glass of water or in this case, the random dispersion of tubulin within a cell. In order to create a model, diffusion is described mathematically by a parabolic partial differential equation (PDE) called the diffusion equation. The particular solution depends on the values the solution takes on the boundaries, and therefore the dependence of diffusion on dynamic instability breaks down to the solution of a boundary value problem meshed with the current mesoscopic simulation.

Diffusion of a substance in a fluid is described mathematically by the diffusion equation

$$\frac{\partial}{\partial t} C = \nabla \cdot (D \nabla C)$$

(2)

where $C$ is the concentration of the substance at any point in the fluid and $D$ is the diffusivity of the substance in the fluid. The diffusivity can be determined from Stokes-Einstein equation

$$D = \frac{k_B T}{6\pi \eta R}$$

(3)
where \( k_B T \) is thermodynamic temperature, \( B \) is a constant, \( R \) is the radius of the particles, and \( \eta \) is the viscosity of the fluid in which the particles are diffusing. This equation gives the relationship between molecular movement and diffusion, since random molecular motion is responsible for diffusion through a fluid. This relationship has been experimentally verified to hold in various scenarios (Miller).

Based on a previous study, the diffusivity of tubulin was measured in sea urchin eggs to be \( 5.9 \times 10^{-12} \text{m}^2/\text{s} \), and is considered to be the diffusivity of tubulin in the average cell (Salmon et al. 1984). This diffusivity was not observed to vary over a wide range of values during experimentation, and therefore the diffusivity is taken to be a constant. In addition, the inclusion or loss of a single phosphate is not enough to change the diffusion of the whole protein. Therefore, we can simplify the diffusion model by using a constant diffusivity in equation 2, which then becomes

\[
\frac{\partial}{\partial t} C = D \nabla^2 C.
\]  

(4)

Here, we are interested in implementing the diffusion equation on a two dimensional Cartesian plane, since the current model (see section 2) employs MT dynamic instability on on a 2-D plane. Therefore, the final version of the diffusion equation will become

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

(5)

Since equation 5 is second order in two space variables and first order in time, we will need four boundary conditions and one initial condition to fully solve the problem. The most physically realizable boundary conditions in this case are Neumann boundary conditions, specifically, the zero flux boundary condition at every boundary. Therefore, we physically describe this situation of no tubulin leaking out of the cell with zero flux boundary conditions at every boundary of the mesh (see Figure 6). We mathematically describe these boundary conditions with the equations

\[
\frac{\partial}{\partial x} C(x = 0, y, t) = 0
\]  

(6)

\[
\frac{\partial}{\partial x} C(x = L, y, t) = 0
\]  

(7)

\[
\frac{\partial}{\partial y} C(x, y = 0, t) = 0
\]  

(8)

\[
\frac{\partial}{\partial y} C(x, y = L, t) = 0
\]  

(9)

The initial condition is describes a constant amount of tubulin equally spaced through the cell, but can be changed to make the distribution whatever one pleases

\[
C(x, y, t = 0) = C_0,
\]  

(10)

where \( C_0 \) is the initial concentration of tubulin specified at compile time.

Equations 4 and 6 through 10 completely describes the problem. The only thing left is implementing this on a grid, thereby iteratively solving the equation using numerical techniques and meshing this with the current simulation.

### 3.1. Numerical Techniques for Diffusion

This code employs a finite difference Crank-Nicolson scheme to solve the diffusion equation. The particular numerical model is utilized because it is second order accurate in time, but mainly because it is numerically stable in time steps (or at least more so than a regular explicit method). An ordinary explicit method is more easily implemented, however is extremely limited in time steps due to the numerical stability of the scheme via the Courant-Friedrichs-Lowy condition (CFL condition, or just CFL). Mathematically, this conditions states that the numerical domain of dependence of the solution must be larger than that of the physical one. Physically, this means that the propagation speed of any physical disturbance (spreading of the particles in this case) must be smaller than that of its numerical counterpart. Therefore, the physical signal is constrained to only move a certain distance in a single time step (Numerical Methods for the Solution of Partial Differential Equations).

The Crank-Nicolson scheme is an approach that weights half towards an explicit method and half towards an implicit method. The Forward Time Central Space (FTCS) method is explicit and is limited in what it can do in a single time step, however is a easy and fast to use. Therefore, it is averaged with a Backwards Time Central Space (BTCS) method in order to utilize the numerical accuracy. The reason that this method is not fully utilized is due to the fact that it requires solving a system of linear equations at each time step. The Crank-Nicolson method also requires solving a system of equations at each time step, however it is second order accurate as opposed to the first order accuracy of the BTCS method. In addition, the system of equations in matrix form will be sparse, meeting the design capabilities of modern day mathematical software packages designed for the solution of systems of equations.

In order to solve the equation numerically, we first set up the standard two dimensional coordinate grid as in Figure 6. Here, we discretize the equation in time and space using the finite difference Crank-Nicolson method. After discretizing Equation 5, it becomes

\[
C_{i,j}^{n+1} = C_{i,j}^n + \frac{D(\Delta t)}{2(\Delta x)^2}(C_{i+1,j}^{n+1} - 2C_{i,j}^{n+1} + C_{i-1,j}^{n+1})
\]  

(11)

\[
+ \frac{D(\Delta t)}{2(\Delta y)^2}(C_{i,j+1}^{n+1} - 2C_{i,j}^{n+1} + C_{i,j-1}^{n+1}) - 2C_{i,j}^{n} + C_{i,j}^{n-1} + C_{i,j+1}^{n} + C_{i,j-1}^{n} - 2C_{i,j}^{n} + C_{i,j}^{n-1},
\]

where \( \Delta x \) and \( \Delta y \) are the grid spacings in the x- and y-directions respectively, and \( C_{i,j}^n \) is the concentration at grid point \( i, j \) at time \( n \). It is now necessary to utilize the boundary conditions. From the zero flux boundary conditions and a Taylor expansion, we can see that we must have the ghost points equal to the values of the adjacent points. For example,
\[
\frac{C_L - C_{L-1}}{\Delta x} = 0 \implies C_L = C_{L-1}
\]

where \(C_L\) is a boundary node in the x-direction (at \(x = L\)). We make this type of substitution at every boundary node to come up with a series of simultaneous equations to solve at every time step.

3.2. **PETSc**

The Portable, Extensible Toolkit for Scientific Computation (PETSc) is specifically designed to solve equations that result from partial differential equations. It contains thousands of functions and routines that can be utilized. Here, we are specifically interested in those functions that can be used to solve coupled linear equations. It exploits advanced mathematical methods as well as the Message Passing Interface (MPI) standard for parallelization. For this system of equations, we solve using the PCLU solver (direct solver), which utilizes LU decomposition to solve the system of equations. PETSc takes the system of equations in matrix form and splits it up into lower and upper tridiagonal matrices. Using an LU algorithm to solve, PETSc then finds the unknowns. This is done at every step, and offers a great advantage to other conventional methods of solution, such as row reduction (see Figure 7). PETSc is so useful in this situation because the matrix that must be solved is sparse. Therefore, most elements are zero, and row reduction is overkill in this case. It is more computationally efficient to use a different method offered by PETSc for such cases.

Therefore, solving the diffusion equation over a large grid does not pose a problem (it is much more computationally efficient than row reduction) Figure 8 shows snapshots taken at time step 0 (initial) and time step 100 for a grid 90 x 90 nodes. Therefore, there are 8100 linear equations to be solved simultaneously, and took PETSc an average of just over 100 seconds to solve this system. See presentation for full diffusion animation.

**FIG. 7.** The solution times for the PETSc routines look constant, but are not. Each solution time is just extremely small (compare for 676 equations—row reduction took 109.975 seconds and PETSc LU decomposition took 0.813 seconds). These times represent the method of solution for an average of three runs at 50 time steps on a single CPU of a 3.8 GHz Intel I5 processor.

**FIG. 8.** Snapshots taken from the diffusion simulation. (a) Time step 0, the concentration at a 30 x 30 section of interior nodes has a high concentration (\(C_0 = 150\), while the rest of the mesh has a concentration of \(C_0 = 5\). (b) By time step=100 the initial concentration at the center has decreased dramatically. In this simulation, the diffusivity is \(D = 0.3\) The entire grid is of unit length, so here \(\Delta x = \Delta y = 0.011\).

4. **CONCLUSIONS AND FUTURE WORK**

The diffusion routine is running with PETSc as its method of solution of linear equations. It runs now on a a grid of about 95 x 95 nodes. With a grid this size, \(95^2 = 9025\) equations must be solved at each time step. Since they are linear equations, there must be an equal amount of unknowns per equation. Therefore, besides the three solution vectors that each contain 9025 elements, there are three matrices that must hold 9025\(^2\) = 81,450,625 elements. Since these are floating point operations, precise variables are used (double precision variables, which are 8 bytes each). A quick analysis will determine that storage of doubles in the matrix and two solution vectors requires the program to have \(\approx 2\) GB of available memory to run. This is conceivable on a modern day computer with this much memory (although some clusters have many processors with only 1 GB each), however future testing might prove this to be too coarse a grid. Therefore, this program might need to be parallelized before we can proceed. PETSc is designed for quick and optimum parallelization using the MPI standard, so this is not a big deal. Splitting up the large matrices onto different computational zones and then few minor changes to the code should suffice to run this program in parallel across multiple processors. This should give an increase in computational efficiency as well as a much finer grid on which diffusion can occur.
Therefore, future work will be to mesh this diffusion routine into the mesoscopic scale simulation to see whether diffusion has a rate limiting effect on MT dynamic instability. Testing of this routine once it is integrated will determine if the grid is too coarse. If so, the routine can be run on single processors with more memory, or parallelized and run on multiple processors. Once this is done, we will be able to tell what type of effect (if any) that diffusion has on MT dynamic instability.

This work was supported by the National Science Foundation (NSF) through grant number NSF MCB 0951264. I would like to acknowledge and thank Dr. Mark Alber and Dr. Holly Goodson for their help and guidance this summer. I would also like to acknowledge the help given to me by Dr. Ziliang Xu on computer related affairs including specific library implementations and optimization. I would like to thank Dr. Umesh Garg and Ms. Shari Herman for their hard work and dedication to the Notre Dame REU program.

REFERENCES

Margolin, Gennady, Goodson, Holly V., Alber, Mark S. In prep. (2010)