Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 9 (2012) 679 – 688

Procedia Computer Science

International Conference on Computational Science, ICCS 2012

A mathematical model for intracellular HIV-1 gag protein transport and its parallel numerical simulations

Jiangguo Liu[a], Roberto Munoz-Alicea[b], Tingwen Huang[c], Simon Tavener[d], Chaoping Chen[e]

[a] *Department of Mathematics, Colorado State University, Fort Collins, CO 80523-1874, USA,* liu@math.colostate.edu
[b] *Department of Mathematics, Colorado State University, Fort Collins, CO 80523-1874, USA,* munoz@math.colostate.edu
[c] *Science Program, Texas A&M University at Qatar, P. O. Box 23874, Doha, Qatar,* tingwen.huang@qatar.tamu.edu
[d] *Department of Mathematics, Colorado State University, Fort Collins, CO 80523-1874, USA,* tavener@math.colostate.edu
[e] *Department of Biochemistry & Molecular Biology, Colorado State University, Fort Collins, CO 80523-1870, USA,* chaoping.chen@colostate.edu

Abstract

In this paper, we develop a mathematical model for intracellular HIV-1 gag protein trafficking based on the hypotheses that gag proteins employ kinesins for active transport on microtubules and they can also diffuse in cytoplasm. This results in a time-dependent convection-diffusion equation in polar coordinates along with appropriate boundary and initial conditions. A finite element method based on tracking characteristics is established for accurately solving this type of transport problems. The numerical method has been implemented in C++. To validate the mathematical model, we perform numerical simulations on the virion timing, i.e., the time needed for HIV-1 virions (puncta) to first appear on the cell plasma membrane. Numerical simulation results and biological experimental data agree principally. For *in silico* analysis of gag protein trafficking, the numerical simulation code needs to be executed repeatedly on a large collection of sets of model parameters. We further investigate code parallelization strategies using MPI and OpenMP.

Keywords: Active transport, cytoplasm, diffusion, finite element methods, gag proteins, HIV-1, microtubules, MPI, OpenMP

1. Introduction

Incoming viral particles travel from the cell periphery to sites of viral transcription and replication. During egress, subviral particles and/or virions travel back to the cell plasma membrane. In these processes, the host cell cytoskeleton has been utilized for transport of subviral particles and/or virions [25, 27, 33]. It has been revealed [7, 9, 15, 21, 34] that organelles and subviral particles employ dyneins and kinesins for active transport along microtubules.

Human immunodeficiency virus type 1 (HIV-1) is a retrovirus that causes acquired immunodeficiency syndrome (AIDS), a condition in humans in which the immune system fails progressively. In an infected cell, as a critical component of HIV-1, the **g**roup-specific **a**nti**g**en (gag) protein is initially synthesized as a 55 kD polyprotein [11, 12]. The gag polyprotein contains domains that mediate gag-gag interactions and gag association with cellular membranes [2, 25] (and references therein), which are essential for the assembly of progeny virions. About 1500 copies of gag protein are required to assemble one progeny virion [1].

doi:10.1016/j.procs.2012.04.073

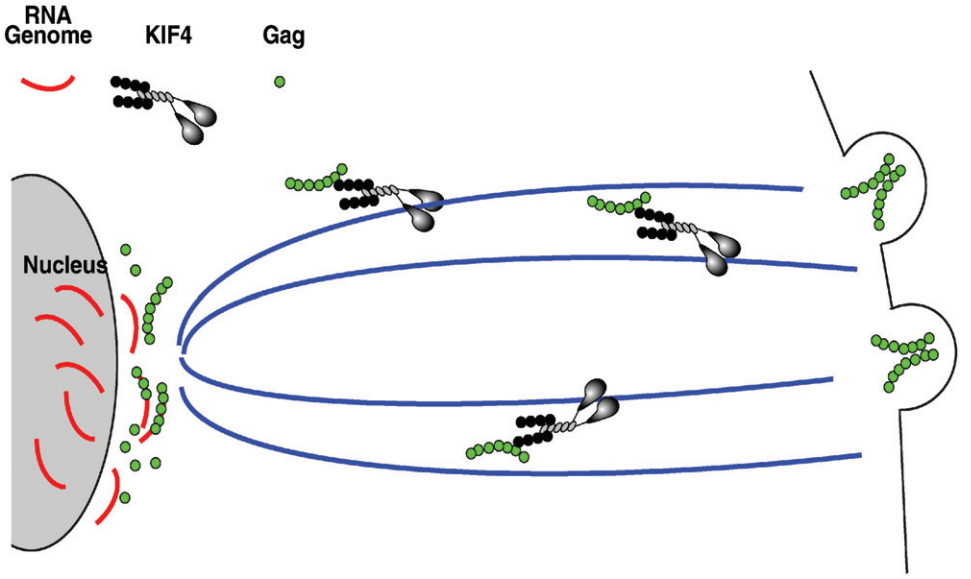

Figure 1: From Ref.[33]: During HIV-1 egress, gag proteins utilize KIF4, a kinesin family member, for transport on microtubules.

It is known [22, 33] that gag proteins engage KIF4, a kinesin family member, for microtubule transport to the plasma membrane where gag proteins and HIV-1 viral genome are budding into virions and released from the infected host cell. Knockdown of KIF4 slows temporal progression of gag through its trafficking intermediates and inhibits virus-like particle production [22].

Although the mechanism of transport along microtubules is still in debate [16, 24] (and references therin), there have been efforts on developing quantitative models for intracellular transport [5, 6, 10, 14, 17, 18, 29]. However, these studies more or less focus on single virus trafficking. In certain applications, it is more important to know how collectively subviral particles are being transported to the place where they assemble and how new virions are formed, for example, the HIV-1 gag protein trafficking and assembly.

In this paper, we develop a mathematical model for intracellular HIV-1 gag protein trafficking based on the hypotheses that gag proteins employ kinesins for active transport on microtubules and they can also diffuse in cytoplasm. This results in a time-dependent convection-diffusion equation in polar coordinates along with appropriate boundary and initial conditions. A finite element method based on tracking characteristics is established for accurately solving this type of transport problems. The numerical method has been implemented in C++. To validate the mathematical model, we perform numerical simulations on the virion timing, i.e., the time needed for HIV-1 virions (puncta) to first appear on the cell plasma membrane and have obtained principal agreement between numerical results and biological experimental data. Moreover, for *in silico* analysis of gag protein trafficking, the numerical simulation code needs to be executed repeatedly on a large collection of sets of model parameters. We also investigate code parallelization strategies using MPI and OpenMP.

2. Modeling HIV-1 Gag Protein Transport Inside Cytoplasm

Existing studies and increasing evidence [7, 9, 25, 26, 27, 31, 33, 34] suggest that viral particles including HIV-1 gag proteins use the host cell cytoskeleton for their intracellular journey: **entry** (from the cell membrane to the nucleus), replication inside the nucleus, and **egress** (from the nucleus to the cell membrane). In these processes, microtubule-associated dynein and kinesin motors are employed, even though the regulation mechanisms for these motors are still in debate [16, 24]. Most viral particles could diffuse in cytoplasm [15, 21].

For HIV-1 gag proteins, it is known [1, 2, 12, 23] that monomers are produced inside cytoplasm. They could associate into dimers, trimers, and even higher order multimers. These gag particles do not enter back into the cell

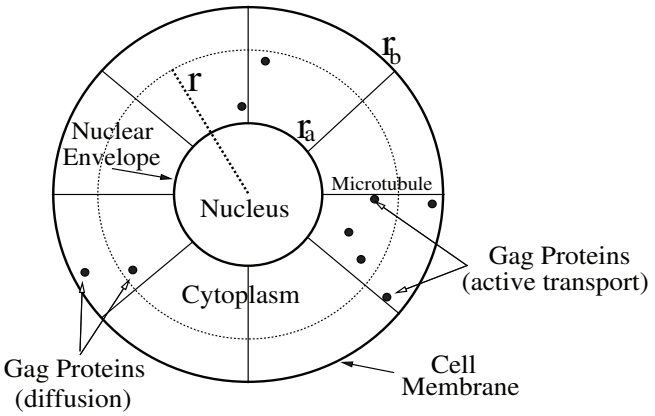

Figure 2: A two-dimensional model for gag protein trafficking: active transport on microtubules and diffusion in cytoplasm. The cytoplasm is geometrically simplified as an annulus. Here r_a is the radius for the nucleus and r_b is the radius for the cell plasma membrane.

nucleus. They travel towards the cell plasma membrane. Once enough gag particles accumulate there, new HIV-1 virions form and escape the cell. Cells take a variety of shapes, but annulus is a reasonably simple geometric model for cytoplasm [6]. Summarized below are our assumptions for a mathemtical model for gag transport inside the cytoplasm of an infected cell.

(A1) Cytoplasm is an annulus;

(A2) All gag species, e.g., monomers, dimers, trimers, and oligomers are treated as quasi-monomers with converted concentration, that is, one dimer is treated as two monomers in terms of mass/concentration, a trimer treated as three monomers, etc;

(A3) No association and dissociation among different species (monomers, dimers, etc.) of gag protein;

(A4) Active transport happens on microtubules and gag can also diffuse in the cytoplasm;

(A5) gag monomers are produced uniformly and constantly in the cytoplasm from the messenger RNA;

(A6) gag particles do not penetrate back into the nucleus;

(A7) gag particles cannot escape through the plasma membrane before virions form (puncta appears), i.e., a threshold concentration is reached;

(A8) The initial monomeric gag concentration is zero.

Let r_a, r_b be the radii of the nucleus and the plasma membrane, $P(r,t)$ the quasimonomeric gag concentration at the radial position r and time moment t, and g_1 the rate of gag production inside the cytoplasm. It is clear that the transport velocity is $\mathbf{v} = (s\cos(\theta), s\sin(\theta))$. The total flux of the transport consists of a convective part and a diffusive part: $\nabla\cdot(\mathbf{v}P - D\nabla P)$. The divergence of the total flux in the polar coordinate system is

$$\nabla\cdot(\mathbf{v}P - D\nabla P) = \frac{1}{r}\frac{\partial}{\partial r}\left(srP - Dr\frac{\partial P}{\partial r}\right).$$

The mass conservation law asserts that the temporal change of the concentration $\frac{\partial P}{\partial t}$ is balanced by the divergence of the total flux and the temporal change of the material sink/source, which is known to be g_1 for gag proteins. We set the time moment when gag monomers are first produced in cytoplasm as 0 and the time moment when new HIV-1 virions first appear on the cell membrane as T_v. Now we have a time-dependent convection-diffusion transport equation along

with the no-penetration no-escape boundary conditions and a zero initial condition for gag concentration, as shown below

$$\begin{cases} \dfrac{\partial P}{\partial t} = \dfrac{1}{r}\dfrac{\partial}{\partial r}\left(Dr\dfrac{\partial P}{\partial r} - srP\right) + g_1, \quad r \in [r_a, r_b], \quad t \in [0, T_v]. \\ \left[Dr\dfrac{\partial P}{\partial r} - srP\right]_{r_a} = \left[Dr\dfrac{\partial P}{\partial r} - srP\right]_{r_b} = 0 \\ P(r, 0) = 0. \end{cases} \tag{1}$$

It is assumed that the averaged speed of gag proteins on microtubule s (μm/sec), the diffusion coefficient D ((μm)^2/sec), and the averaged production rate of monomeric gag proteins inside the cytoplasm g_1 (μM/sec, micro-Molar per second) are constants.

It should be pointed out that the above model is for the time period from the moment gag proteins are being produced inside the cytoplasm until the moment new HIV-1 virions first appear on cell plasma membrane. After that, newly formed HIV-1 virions start the budding process, i.e., they escape the host cell, meanwhile, gag proteins still travel towards the plasma membrane and assemble there, we have a leaking boundary condition.

3. Numerical Methods for Convection-Diffusion Equations

For time-dependent convection-diffusion problems like (1), traditional finite difference methods or finite element methods usually either smear the steep fronts exhibited in the solutions or produce nonphysical oscillations near these fronts [32] (and references therein). In this regard, the finite element methods based on characteristic-tracking have demonstrated their advantages in solution accuracy and algorithm efficiency, due to the use of information along streamlines [8, 20, 32]. Among the characteristic finite element methods, the Eulerian-Langrangian localized adjoint methods (ELLAM) use both Eulerian grids and Lagrangian grids (characteristics or streamlines) and adopt a natural operator splitting. An ELLAM-type method [20, 32] can be developed to numerically solve the boundary initial value problem of the transport equation in polar geometry (1). Here we briefly highlight the main ideas.

Let $0 = t_0 < t_1 < \ldots < t_{n-1} < t_n < \cdots < t_N = T_v$ be a temporal partition of $[0, T_v]$ that is not necessarily uniform and $\Delta t_n = t_n - t_{n-1} (n = 1, \ldots, N)$. Multiplying both sides of the first equation in the boundary initial value problem (1) by a typical test function $\psi(r, t)$ defined on the space-time slab $[r_a, r_b] \times (t_{n-1}, t_n]$, we obtain

$$\int_{t_{n-1}}^{t_n}\int_{r_a}^{r_b}\left(r\frac{\partial P}{\partial t} + \frac{\partial}{\partial r}\Big(srP - Dr\frac{\partial P}{\partial r}\Big)\right)\psi dr dt = \int_{t_{n-1}}^{t_n}\int_{r_a}^{r_b} g_1 r dr dt, \tag{2}$$

where we have applied the no-flow boundary conditions in (1) and the test function is required to satisfy the adjoint equation

$$\frac{\partial \psi}{\partial t} + sr\frac{\partial \psi}{\partial r} = 0. \tag{3}$$

This implies that any test function is actually a constant along each characteristic. Then we end up with a numerical scheme for the transport problem (1) as follows

$$\begin{aligned} &\int_{r_a}^{r_b} P(r, t_n)\psi(r, t_n) r dr + \Delta t_n \int_{r_a}^{r_b} D\frac{\partial P}{\partial r}\frac{\partial \psi}{\partial r} r dr \\ &\quad = \int_{r_a}^{r_b} P(r, t_{n-1})\psi(r, t_{n-1}^+) r dr + \Delta t_n \int_{r_a}^{r_b} g_1 r dr. \end{aligned} \tag{4}$$

The formulation (4) clearly indicates that we have an elliptic problem at the new time step t_n, which opens the door for a variety of spatial finite elements. The formulation also reveals that the ELLAM methodology serves as a very natural operator splitting: one sees only convection along characteristics (Lagrangian grids) (3), whereas the diffusion problem (4) is solved on Eulerian grids.

For simplicity, we use continuous piecewise linear spatial finite elements in the radial direction. A suite of C++ and Matlab code has been developed and used in our preliminary work on numerical simulations for gag protein transport inisde cytoplasm.

It is known that 1500 copies of gag protein are required for an HIV-1 new virion [1] and the spherical virion has a radius in the range of 0.05μm to 0.075μm. This implies that a threshold concentration 1409μM on the cell membrane is needed for new virions (puncta) to appear. Our numerical results for $r_a = 5\mu$m, $r_b = 10\mu$m, $D = 0.04(\mu\text{m})^2/\text{sec}$, $g_1 = 1500\mu\text{M/sec}$, $\Delta t = 1$sec, and the above threshold concentration are shown in Table 1. The time T_v (for puncta to first show up near cell membrane) is in the format of hours, minutes, seconds.

Table 1: Numerical simulation results on the time T_v needed for HIV-1 virions (puncta) to appear on cell plasma membrane. T_v is inversely proportional to the average speed s on microtubules.

Speed $s(\mu m/\text{sec})$	Time T_v
2.00	09h46m41s
1.00	10h14m35s
0.40	10h32m57s
0.20	13h31m29s
0.10	21h09m08s
0.05	34h20m07s

As demonstrated by the numerical simulation results in Table 1, the time needed for HIV-1 virions (puncta) to first appear on the cell plasma membrane is inversely proportional to the average speed on the microtubules. Here we consider only the egress (traveling from the cell center to the cell periphery). If HIV-1 viruses' entry takes roughly the same time as that for egress, then the time post infection for HIV-1 virions to appear near cell membrane is about twice of that for egress. In this regard, our numerical results agree principally with the experimental data in Figure 3.

4. Code Parallelization Based on MPI and OpenMP

To perform *in silico* numerical simulations of HIV-1 gag transport inside cytoplasm, our simulation code will be executed repeatedly for a very large collection of different parameter values s, D. This is essentially a case of Single Instruction Multiple Data (SIMD). Code parallelization is therefore a natural choice for simulation efficiency. We use both Message Passing Interface (MPI) and OpenMP for code parallelization [3, 4, 19].

4.1 Master-Worker Parallelization Using MPI. We adopt a *master-worker* approach. We have an input that is a parameter file containing a collection of sets of parameter values. Itemized below are the major steps in the parallelization, see also Figure 4.

(1) The main process or rank, called the *master*, reads in the parameter data from a parameter file.

(2) The master rank distributes the parameter values to other ranks, called *workers*.

(3) Each worker runs a copy of the code with a particular set of parameter values.

(4) Each worker sends its output to the master.

(5) After collecting the outputs from all workers, the master writes the outputs to a file for later analysis.

Remark. Note that if the number of workers is large enough, compared to the number of parameter combinations, some workers will not perform any work. The actual number of parameter combinations that a worker receives may vary, depending on the total number of ranks.

4.2 OpenMP Parallelization for Each Copy of Code. OpenMP can be used to parallelize the code within ranks. OpenMP parallelizes code through *parallel sections*, e.g., a "for loop" within a subroutine. The loop is split among different *threads*.

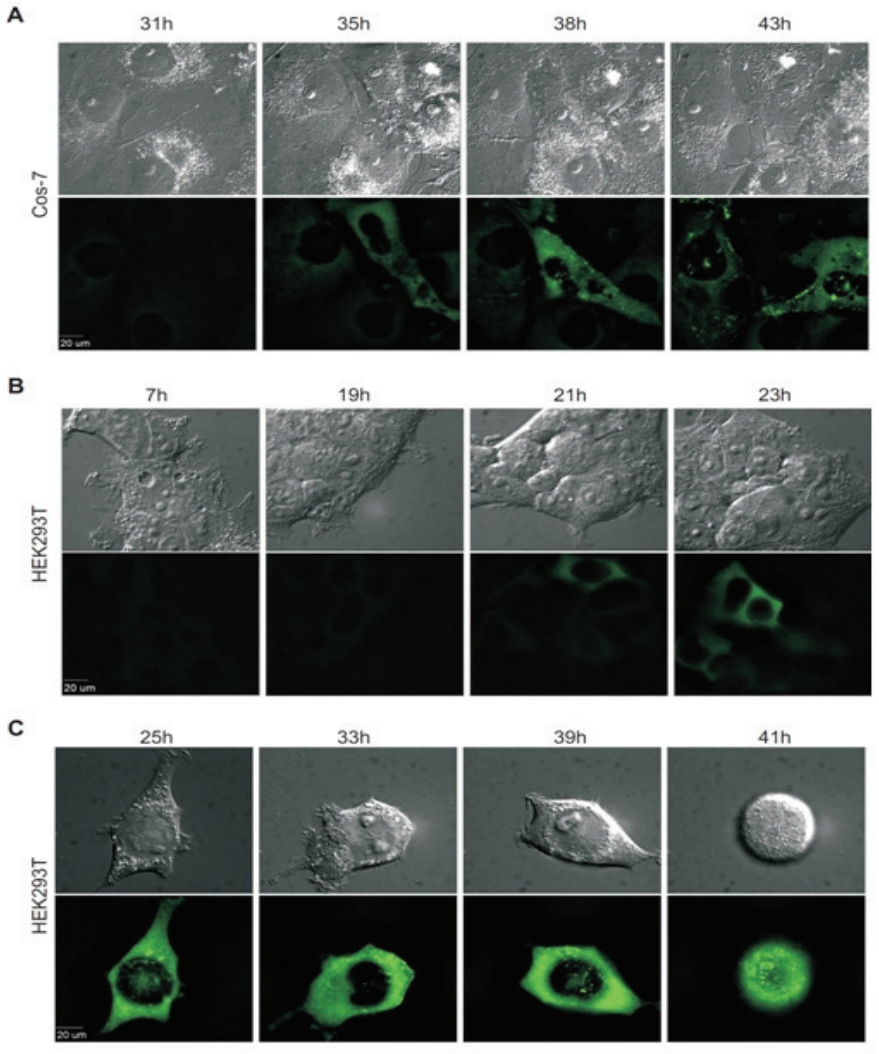

Figure 3: Experimental results: gag distribution near cell plasm membrane at various times post infection: (A) Cos-7 cell, (B&C) HEK293T cell.

`gprof` can be used to create a code profile [13]. This profile helps us find the sections of code that take the longest time to run, and set priority for parallelization within ranks.

4.3 Cray Architecture. Figure 5 is an illustration of the architecture of Cray XT6m.

4.4 `aprun`. On Cray XT6m, one runs a parallel code using the `aprun` command with various values of

- n, the number of processing elements (PEs), which corresponds to the number of ranks used;
- d, depth, the number of cores per PE, which can be set equal to the number of OpenMP threads;

For instance, the command `aprun -n3 -d5 a.out` executes `a.out` with one master, two workers, five threads per rank, one core per thread, and one PE per node, see [4].

4.5 Parallelization Speedup and Efficiency. As is well known, *speedup* is defined as

$$S_i = T_1/T_i,$$

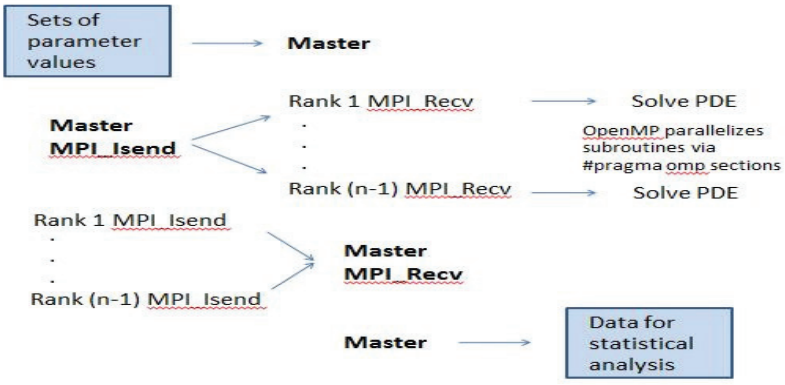

Figure 4: Code parallelization using a master-worker approach based on MPI. OpenMP is used for parallelization within ranks.

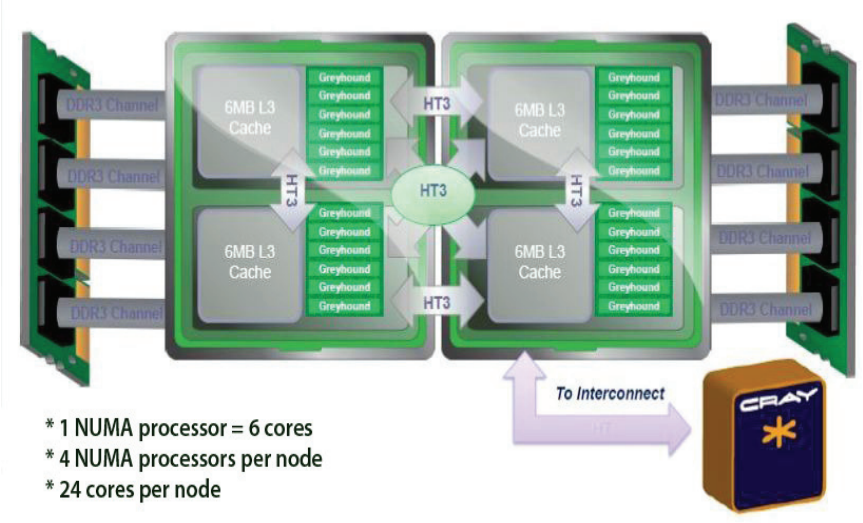

Figure 5: The architecture for Cray XT6m (Adopted from [3])

where i is the number of allocated cores and T_i is the time it takes i cores to run a program. *Efficiency* is defined as

$$E_i = T_1/(iT_i),$$

which is also interpreted as the speedup per core.

5. Parallel Simulation Results

Recall that MPI is used for parallelization among ranks in the master-worker approach. OpenMP is used for parallelization within a rank (via parallel sections).

We have tried a case with 132 sets of parameters: s, D, Δt, where $s \in \{1, 1.2, 1.4, ..., 3.0\}$ with increment 0.2; $D \in \{0.02, 0.04, 0.06\}$; and $\Delta t \in \{0.125, 0.25, 0.5, 1.0\}$. It is clear that we have $11 \times 3 \times 4$ sets of parameters.

When the simulation code is executed sequentially with just one set of parameters, it is observed that for small s, it takes longer for the job to finish, and for larger s, the job is completed faster. Similarly, the job takes longer to be completed when Δt is small, and is completed faster when Δt is larger.

If there are enough workers and each worker does one job, depending on the parameter values of $s, D, \Delta t$ for the jobs assigned by the master, some workers may take longer to complete their jobs than others. Thus, there is clearly **unevenness** among jobs, even though there is no communication among the workers.

Parallelization using `aprun` with various combinations of n and d will produce different effects. Recall that n (the number of ranks) is used for MPI and d (the number of threads) is for OpenMP.

Table 2: Effects of the number of threads (OpenMP)

n (# of ranks)	d (# of threads)	Runtime (seconds)	Speedup
2	1	929.67	
2	5	521.33	1.78
133	1	21.00	
133	2	17.33	1.23

From Table 2 we can see that OpenMP parallelization within ranks had the most impact when we used fewer ranks. In particular, runtime was reduced by almost 50% when only one worker and five threads were used. On the other hand, when 132 workers were used, due to limited resources, we were only able to use two threads. Runtime is reduced by only about 17% in this case.

Table 3: Effects of the number of ranks (MPI)

n (# of ranks)	d (# of threads)	Runtime (seconds)	Efficiency
2	5	521.33	
20	5	31.33	0.81
30	5	23.33	0.71
40	5	19.00	0.65
50	4	22.00	0.45
133	2	17.33	0.21

Notice that we do not really need to run our code with 133 ranks. The code executes with high efficiency when we use 5 threads and between 20 and 40 ranks, see Table 4. If we increase the number of ranks above 40, we are not able to use 5 threads, because of limited resources. For high number of ranks, the code runs only a couple of seconds faster and the efficiency is low, at least partly due to overhead costs.

Table 4: Unevenness of jobs

n (# of ranks)	d (# of threads)	Runtime (seconds)	Efficiency
100	2	21.33	0.23
110	2	24.00	0.18
120	2	22.33	0.18
130	2	17.33	0.21
140	2	17.67	0.20

Table 4 demonstrates an interesting issue. Note that running the parallel code with 110 and 120 worker ranks actually took longer, on average, than running the code with 100 ranks. Two factors might be at play here. First, there could be additional overhead costs for the additionally allocated cores. Second, there exits unevenness of jobs. It is conceivable that, in some situations, some cores are assigned the jobs that take the longest to complete, while other cores are assigned jobs that can be completed quickly. This would cause the program to take longer to run, as the cores would not be used as efficiently as possible.

One approach for addressing job unevenness is to have the master assign jobs to the workers as they become available. Each worker receives a job and sends a message to the master when the job has been completed. At this

point, the master saves the worker's output and assigns the available worker a new job. The process continues until all jobs have been completed.

6. Remarks on Future Work

The mathematical model and simulation code we evaluate in this paper is for a deterministic transport equation in cytoplasm. Since the average speed s on microtubules and the diffusion parameter D vary with cells, we could treat them as random variables on the sample space of cells. In our future work, we will perform stochastic analysis of some quantities of interest, e.g., the virion timing T_v (the time when virions/puncta first appear near the cell plasma membrane). The simulation code for deterministic transport equation in cytoplasm will be used in conjunction with the stochastic collocation methods [35].

In the model proposed in this paper, all species of gag proteins such as monomers, dimers, trimers are treated as quasimonomers with equivalent concentrations. Existing studies [12] suggest that gag proteins dimerize inside cytoplasm and gag dimers further associate with HIV-1 viral RNA. A mathematical model considering gag dimerization will be a system of coupled transport equations with nonlinear reactions that require development of efficient numerical methods for such equations. This is currently under our investigation and will be reported in the future.

Existing studies [28, 30] also reveal that gag trimers play an important role in binding with the infected host cell plasma membrane. Near the plasma membrane, a mixture of the myristoyl-exposed gag trimers and the myristoyl-sequestered gag monomers will evolve into an equilibrium that helps the formation of new HIV-1 virions. A mathematical model for a dynamical system of gag trimers and monomers near cell plasma membrane is currently being developed and will be reported in our future work.

Acknowledgements. J. Liu and R. Munoz-Alicea were partially supported by the US National Science Foundation under Grant No. DMS-0915253. R. Munoz-Alicea was also supported in part by an internal grant from the Center for Interdisciplinary Mathematics and Statistics at Colorado State University. The authors thank the great help from and stimulating discussion with Drs. Lilian Huang and Qing Nie and Ms Judy Mufti.

References

[1] J.A. Briggs et al., *The stoichiometry of gag protein in HIV-1*, Nat. Struct. Mol. Biol., **11**(2004), pp. 672–675.
[2] C. Chen et al., *Association of gag multimers with filamentous actin during equine infectious anemia virus assembly*, Curr. HIV Res., **5**(2007), pp. 315–323.
[3] `http://istec.colostate.edu/istec_cray/tutorial_3_30_11.pdf`
[4] `http://docs.cray.com/cgi-bin/craydoc.cgi?mode=SiteMap;f=xt3_stemap`
[5] A.T. Dinh, C. Pangarkar, T. Theofanous, and S. Mitragotri, *Understanding intracellular transport processes pertinent to synthetic gene delivery via stochastic simulations and sensitivity analyses*, Biophys J., **92**(2007), pp. 831–846.
[6] A.T. Dinh, T. Theofanous, and S. Mitragotri. *A model for intracellular trafficking of adenoviral vectors*, Biophys J., **89**(2005), pp. 1574–1588.
[7] M.P. Dodding and M. Way, *Coupling viruses to dynein and kinesin-1*, EMBO Journal, **30**(2011), pp. 3527–3539.
[8] J. Douglas Jr. and T.F. Russell, *Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures*, SIAM J. Numer. Ana., **19**(1982), pp. 871–885.
[9] K. Döhner, C.-H. Nagel, and B. Sodeik, *Viral stop-and-go along microtubules: taking a ride with dynein and kinesins*, Trends Microbiol., **13**(2005), pp.320–327.
[10] M.R. D'Orsogna and T. Chou, *Optimal cytoplasmic transport in viral infections*, PLoS ONE, **4**(2009), No. 12, e8165.
[11] A.D. Frankel and J.A. Young, *HIV-1: fifteen proteins and an RNA*, Annu. Rev. Biochem., **67**(1998), pp. 1–25.
[12] E.O. Freed, *HIV-1 gag proteins: diverse functions in the virus life cycle*, Virol., **251**(1998), pp. 1–15.
[13] `http://people.sc.fsu.edu/~jburkard/cpp_src/gprof/gprof.html`
[14] M. Gazzola, C.J. Burckhardt, B. Bayati, M. Engelke, U.F. Greber, and P. Koumoutsakos, *A stochastic model for microtubule motors describes the in vivo cytoplasmic transport of human adenovirus*, PLoS Comput. Biol., **5**(2009), e1000623.
[15] J. Han and J. Herzfeld, *Macromolecular diffusion in crowded solutions*, Biophys. J., **65**(1993), pp. 1155–1161.
[16] A. Kunwar et al., *Mechanical stochastic tug-of-war models cannot explain bidirectional lipid-droplet transport*, Proc. Natl. Acad. Sci. USA, **108**(2011), pp. 18960–18965.
[17] A. Kuznetsov and A.A. Avramenko, *Analytical investigation of transient molecular-motor-assisted transport in elongated cells*, Cent. Eur. J. Phys., **6**(2008), pp. 45–51.
[18] T. Lagache and D. Holcman, *Effective motion of a virus trafficking inside a biological cell*, SIAM J. Appl. Math., **68**(2008), pp. 1146–1167.
[19] J. Liu, Z. Chen, R.E. Ewing, G. Huan, B. Li, and Z. Wang, *Parallel computing in the black oil model*, Contemp. Math., Vol. 329 (2003), pp. 253–262.

[20] J. Liu and C. Chen, *A characteristic finite element method with local mesh refinements for the Lamm equation in analytical ultracentrifugation*, Numer. Meth. PDEs, **25**(2009), pp. 292–310.
[21] K. Luby-Phelps, *Cytoarchitecture and physical properties of cytoplasm: volume, viscosity, diffusion, intracellular surface area*, Int. Rev. Cytol., **192**(2000), pp. 189–224.
[22] N.W. Martinez, X. Xue, R.G. Berro, G. Kreitzer, and M.D. Resh, *Kinesin KIF4 regulates intracellular trafficking and stability of the human immunodeficiency virus type 1 gag polyprotein*, J. Virol., **82**(2008), pp. 9937–9950.
[23] D. McDonald et al., *Visualization of the intracellular behavior of HIV in living cells*, J. Cell Biol., **159**(2002), pp. 441–452.
[24] M.L. Müller, S. Klumpp, and R. Lipowsky, *Tug-of-war as a cooperative mechanism for bidirectional cargo transport by molecular motors*, Proc. Natl. Acad. Sci. USA, **105**(2008), pp. 4609–4614.
[25] M.H. Naghavi and S.P.Goff, *Retroviral proteins that interact with the host cell cytoskeleton*, Curr. Opin. Immunol., **19**(2007), pp. 402–407.
[26] A. Ploubidou et al., *Vaccinia virus infection disrupts microtubule organization and centrosome function*, EMBO Journal, **19**(2000), pp. 3932–3944.
[27] K. Radtke, K. Dohner, and B. Sodeik, *Viral interactions with the cytoskeleton: a hitchhiker's guide to the cell*, Cell Microbiol., **8**(2006), pp. 387–400.
[28] M.D. Resh, *A myristoyl switch regulates membrane binding of HIV-1 gag*, Proc. Natl. Acad. Sci. USA, **101**(2004), pp. 417–418.
[29] D.A. Smith and R.M. Simmons, *Models of motor-assisted transport of intracellular particles*, Biophys. J., **80**(2001), pp. 45–68.
[30] C. Tang, E. Loeliger, P. Luncsford, I. Kinde, D. Beckett, and M.F. Summers, *Entropic switch regulates myristate exposure in the HIV-1 matrix protein*, Proc. Natl. Acad. Sci. USA, **101**(2004), pp. 517–522.
[31] R.D. Vale, *The molecular motor toolbox for intracellular transport*, Cell, **112**(2003), pp. 467–480.
[32] H. Wang, H.K. Dahle, R.E. Ewing, M.S. Espedal, R.C. Sharpley, and S. Man, *An ELLAM scheem for advection diffusion equations in two dimensions*, SIAM J. Comput. Sci., **20**(1999), pp. 2160–2194.
[33] B.M. Ward, *The taking of the cytoskeleton one two three: How viruses utilize the cytoskeleton during egress*, Virol., **411**(2011), pp. 244?-250.
[34] M.A. Welte, *Bidirectional transport along microtubules*, Curr. Biol., **14**(2004), pp. 525–537.
[35] D. Xiu, *Numerical methods for stochastic computations: A spectral method approach*, Princeton University Press, 2010.