COMSOL Multiphysics® Simulations Guide

Emergent Actin Flows Explain Diverse Parasite Gliding Modes

Christina L. Hueschen, Li-av Segev Zarko, Jian-Hua Chen, Mark A. LeGros, Carolyn A. Larabell, John C. Boothroyd, Rob Phillips, Alexander R. Dunn

We hope that others will find the numerical approach used here and in Hueschen, Dunn, Phillips 2023 [8] to be user-friendly and adaptable for solving the Toner-Tu equations on other complex curved surfaces or for solving other continuum equations on curved surfaces. Our COMSOL Multiphysics® files are available at `https://github.com/chueschen/Toxoplasma_actin`. Related files and a more thorough tutorial introduction to these simulations are found at `https://github.com/RPGroup-PBoC/wildebeest_herds`.

While we chose COMSOL Multiphysics® for its accessibility and learner-friendly interface, we note with regret that the use of these files requires access to a paid COMSOL Multiphysics® license. If you do not already have a license and are affiliated with an institution, we recommend looking into access options through a shared software library.

Files available in this repository:

1. Hueschen_ToxoActinOrganization_StableActin. Actin flocking model predicts self-organized recirculation ("cyclosis") of actin patches in the absence of filament turnover. The simulation begins with a disordered network, and then filament density ρ and velocity $\mathbf{v}$ evolve over time on the *Toxoplasma gondii* tachyzoite cell surface according the actin self-organization Toner-Tu equations presented in Supplementary Info Sections 6 - 9. Filaments are stable and conserved, as in Supplementary Figure S3 A-B. This simulation produced the results shown in Figure 3B and in Video 4.

2. Hueschen_ToxoActinOrganization_ActinTurnover. Actin flocking model predicts the emergence of self-organized unidirectional flow in the presence of filament turnover. The simulation begins with a disordered network, and then filament density ρ and velocity $\mathbf{v}$ evolve over time on the *Toxoplasma gondii* tachyzoite cell surface according the actin self-organization Toner-Tu equations presented in Supplementary Info Sections 6-9. Filaments are polymerized in the conoid (anterior end) at a rate per c $\mu m^{-2}s^{-1}$ and depolymerized throughout the cell surface at rate $\gamma\rho$ $\mu m^{-2}s^{-1}$, as in Supplementary Figure S3C-D and Section 7.2. This simulation produced the results shown in Figure 4 and Video 6.

A note of caution: This document, together with the How-To Guide found at `https://github.com/RPGroup-PBoC/wildebeest_herds`, are intended to provide a quick, practical orientation to our working files and to the use of the COMSOL Multiphysics® interface for solving PDEs. For those new to the world of finite elements, we recommend seeking out general training from local experts or written resources on both principles of the finite element method and practical tips: avoiding common pitfalls; carefully choosing a solver, mesh size, time step size; verifying against analytical results, etc. COMSOL's own thorough introduction to the finite element method is available at `https://www.comsol.com/multiphysics/finite-element-method`.

Quick-start tutorial:

To solve our custom surface partial differential equations, we used the COMSOL Multiphysics® General Form Boundary PDE interface and took advantage of COMSOL's built-in tangential differentiation operator, dtang(f,x). The heart of our implementation of a general curved-surface formulation of the Toner-Tu equations is described in Section 8 ('Deriving a Tangential Formulation of the Filament Self-Organization Equations') in the supplementary information. Here, in the form of an abridged tutorial,

we provide a practical guide to using that implementation and the COMSOL Multiphysics® interface. A more complete version is available at `https://github.com/RPGroup-PBoC/wildebeest_herds`.

1. Parameters, Geometry, and Meshing. In COMSOL Multiphysics®, open the file 'Hueschen_ToxoActinOrganization_StableActin.mph,' which explores F-actin self-organization in the absence of filament turnover. Begin to orient yourself to the side panel on the left. Under 'Global Definitions,' you'll find our **parameters** (e.g., Toner-Tu coefficients and initial conditions). Now, under 'Component 1,' expand 'Geometry 1'. Here, we could use simple geometric features to build a geometry or import one, as was done in our case for the *Toxoplasma* cell shape extracted from soft X-ray tomograms.

2. Variables and PDEs.

First, click on Variables 1 under Component 1 / Definitions. Here, weve defined important pieces of the curved-surface formulation, such as the projection operator **P** and the surface velocity gradient tensor **G**. Try to familiarize yourself with these quantities, referencing Supplementary Information Section 8.

Under 'Component 1,' click on the first General Form Boundary PDE (velocity). This is our implementation of the Toner-Tu-based self-organization equations for velocity. Note the sections on Units, Discretization, and Dependent variables, which allow us to set the **field variables** for which we wish to solve (in this case, v_1, v_2, and v_3) and, importantly, choose the element order and type of shape function we'll use to describe our velocity field. Next, expand 'General Form Boundary PDE (velocity)' and click on 'General Form PDE 1.' Now we're really getting to the heart of things! Expand the Equation section to see the PDE form used here. Referencing Section 8 in the SI, notice how the contents of $\mathbf{\Gamma}$, f, and d_a combine to **implement our curved-space self-organization velocity equations**, Eqn. 84. Syntax tip: $v1x$ is the derivative of v_1 with respect to x, and $rhox$ is the derivative of ρ with respect to x. Note that we are now working in 3D and have defined three components of our velocity vector, but we're using a 'boundary' PDE module to solve our equations on the surface of our sphere. To practice adding your own Boundary PDE module, right-click on 'Component 1' and choose 'Add Physics / Mathematics / PDE Interfaces / Lower Dimensions / General Form Boundary PDE.'

Click on the second General Form Boundary PDE (continuity). Here we have defined a field of the scalar dependent variable density, ρ, and described it with a linear shape function. Within 'General Form PDE 1,' you'll find our implementation of the curved-space **continuity equation** presented in Eqn. 86. In this example of stabilized actin, the source and sink terms (c and γ) are not present.

Next, click on 'Boundary ODEs and DAEs.' Here, we are using an ODE simply to create a field of normal vectors (n_1, n_2, n_3) that is defined between mesh nodes using a specific shape function. (Our thanks to Yue Huang of COMSOL technical support for suggesting this trick.) Note that the normal vector components nx, ny, and nz are built-in COMSOL geometric variables.

Finally, notice a couple of constraints specific to the curved-surface implementation. These constraints are called for by the imperfect world of numerics, not by the mathematics, and prevent small numerical errors from expanding over time. Within the velocity PDE module, 'Weak Constraint 1' helps to enforce the condition that the velocity vector is nonzero only in the surface tangent plane ($\mathbf{n} \cdot \mathbf{v} = 0$), using terminology defined in 'Variables 1.' Within 'General Form Boundary PDE - continuity,' notice the 'Global Constraint 1' that enforces global conservation of mass. The global integration function 'intop1' is defined within 'Component 1 / Definitions / Integration 1,' and the global integrated density (total number of filaments) calculated using that function is defined in 'Parameters 1.'

3. Initial and Boundary Conditions. Look back at the General Form PDE for velocity and click on 'Initial Values 1'. To **initialize with a disordered velocity field**, we are drawing every node's velocity orientation randomly from a uniform distribution of angles between 0 and 2π, and scaling that by magnitude v_0. Our random angle is generated by the function $rn1(x, y)$, which is defined under 'Global Definitions / Random 1.' The magnitude v_0 was defined under 'Global Definitions / Parameters 1.' Our **initial condition** for our General Form Boundary PDE for density (continuity) is a uniform density field of magnitude ρ_0, which is similarly defined in 'Parameters 1.'

4. Solving. Under 'Study 1', you'll find a **Time Dependent solver** module. Click on 'Step 1: Time Dependent' to customize the time range of the simulation. We used default COMSOL solvers and settings: implicit backward differentiation formula (BDF) for time stepping and multifrontal massively parallel sparse direct solver (MUMPS) for the linear direct spatial solver. Note the Time Stepping settings under 'Solver Configurations / Solution 1 / Time-Dependent Solver 1.'

Now, another big moment! Hit 'Compute' to run the simulation. If you're in a rush, shorten the time range of the simulation. Note: at the end of step 5, we'll discuss **parametric sweeps**.

5. Visualizing and Exporting Results. Move to the Results section on the side panel to **visualize our simulation**. As an example, expand 'Density + velocity arrows.' Note how 'Surface 1' sets up a color table based on density and how 'Arrow Surface 1' displays velocity with a field of customizable arrows. To create your own surface plot, try right-clicking on 'Results' and choose '3D Plot Group.'

Under 'Export,' use 'density velocity file export' to practice **exporting a .gif movie** of the velocity plot discussed above. Undersampling by selecting a limited number of frames or a limited time window may be useful to reduce file size. Then, notice how 'density velocity player' is set up to play within COMSOL. To add your own animations, right-click on 'Export' and find 'Animation.'

Bonus: To re-run the simulation while **sweeping through different parameter choices**, right-click on 'Study 1' and choose 'Parametric Sweep.' Click on the '+' icon and add a short list of choices for a parameter of interest. After computing, update your 'Results' plots to visualize this new dataset (Study 1/Parametric Solutions 1).