Using and Adapting Our COMSOL Multiphysics® Simulations: How-To Guide

Wildebeest Herds on Rolling Hills: Flocking on Arbitrary Curved Surfaces

Christina L. Hueschen, Alexander R. Dunn, Rob Phillips

We hope that others will find our approach user-friendly and adaptable for solving the Toner-Tu equations on other complex curved surfaces, for solving other continuum equations on curved surfaces, or for additional study of the cases presented here. Our COMSOL Multiphysics® files are available for download, use, and adaptation 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 for download and use:

1. Toner-Tu herding in a 2D channel with an obstacle and periodic boundary conditions. This file is a good starting place for orienting yourself to the COMSOL Multiphysics® General Form PDE interface and to the 2D planar Toner-Tu equations and their syntax. To recapitulate Figure 4, remove the circular obstacle. To recapitulate Figure 7, set the obstacle radius to 15 m. To recapitulate Figure 8, set the obstacle radius to 30 m.

2. Toner-Tu herding on the cylinder. This file introduces the curved-surface implementation of the Toner-Tu equations presented in Section 3. Note that some important components of this formulation, such as the projection operator, can be found in 'Variables 1' under 'Definitions'. Note also that the 'Parametric Sweep' implemented within 'Study 1' explores different choices for the parameters α and β, producing the results shown in Figure 10.

3. Toner-Tu herding dynamics on the sphere. This file solves the curved-surface implementation of the Toner-Tu equations, this time on a spherical geometry, and simulates the damped oscillations shown in Figure 12. Note again that some important components of this formulation are defined in 'Variables 1' under 'Definitions'. To sweep through different parameter choices for D as in Figure 14, enable 'Parametric Sweep' within 'Study 1'. Remember to update plots under 'Results' and 'Exports' to reflect this new solution set ('Study 1/Parametric Solutions X').

4. Toner-Tu herding on the sphere: exploring patterns from an initial wedge confinement. This file builds on the previous one in a simple way: herd dynamics are seeded by initially confining the herd to an angular wedge, as shown in Figure 14. Tuning parameters such as σ leads to different patterns in the long-time limit (after $\sim 10^4$ s). To recapitulate Figure 15 by sweeping through different choices for σ, enable 'Parametric Sweep' within 'Study 1'. Remember to update 'Results' and 'Exports' to show this new solution set ('Study 1/Parametric Solutions X').

5. Herding over a hill: curvature-induced density and velocity changes at a Gaussian hill embedded in a racetrack straightaway. This file imports a racetrack with embedded hill geometry (download 'RacetrackWithHill.mphbin' and import under 'Geometry 1'). In the long-time limit, hill curvature induces the low and high density regions shown in Figure 16.

6. Herds gone wild: dynamics of a Toner-Tu herd on an undulating island landscape. This file imports an island landscape of Gaussian hills (download 'IslandOfHills.mphbin' and import under 'Geometry 1'). Herd dynamics are simulated in the absence or presence of a gravitational force with coefficient ζ, as in Figure 17.

The surface mesh geometries referenced in files 5 and 6 above ('RacetrackWithHill.mphbin' and 'Island-OfHills.mphbin') are available for download in the geometries folder.

A note of caution: This 'How-To Guide' is 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 3 ('Formulating the Surface Toner-Tu Equations for Numerical Implementation') in the main text and Section 8.2 ('Implementation in COMSOL Multiphysics') in the Appendix. Here, in the form of a brief tutorial, we provide a practical guide to using that implementation and the COMSOL Multiphysics® interface.

1. Parameters, Geometry, and Meshing. In COMSOL Multiphysics®, open the file '1_Hueschen Phillips_TonerTu_2DChannel.mph,' which explores Toner-Tu herding in a 2D channel with an obstacle and periodic boundary conditions. Begin to orient yourself to the side panel on the left. Under 'Global Definitions,' you'll find our **parameters** (both Toner-Tu coefficients and geometric parameters, like obstacle radius). Now, under 'Component 1,' expand 'Geometry 1'. Note how we've built our simple channel and obstacle **geometry** by subtracting a circle from a rectangle. Hit 'Build All.' Now, find 'Mesh 1' in the side panel. Choose a **mesh element size** and hit 'Build All.' Note that selecting a user-controlled mesh allows for custom specification of mesh features.

2. Variables and PDEs. Under 'Component 1,' click on the first General Form PDE (velocity). This is our implementation of the Toner-Tu equations for velocity. Note the sections on Units, Discretization, and Dependent variables, which allow us to set the **variables** for which we wish to solve (in this case, v_1 and v_2) and, importantly, choose the element order and type of shape function we'll use to describe our velocity field. Next, expand 'General Form 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 3 in the main text, notice how the contents of $\mathbf{\Gamma}$, f, and d_a combine to **implement the 2D Toner-Tu equations**. Syntax tip: $v1x$ is the derivative of v_1 with respect to x, and $rhox$ is the derivative of ρ with respect to x.

Click on the second General Form 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 **2D continuity equation**, $\partial\rho/\partial t + \partial(\rho v_i)/\partial x_i = 0$. To add your own PDE in this same form, you would right-click on 'Component 1' and choose 'Add Physics / Mathematics / PDE Interfaces / General Form PDE.'

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 nodes'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 PDE for density (continuity) is a uniform density field of magnitude ρ_0, which is similarly defined in 'Parameters 1.'

In this simulation, we want to impose **Dirichlet boundary conditions** of $\mathbf{v} = 0$ at the obstacle and at the sides of the channel, and we want **periodic boundary conditions** at the channel ends. These are imposed within the 'Dirichlet Boundary Condition' and 'Periodic Condition' features within our General Form PDE modules. To see choices for other boundary conditions and constraints, right-click on one of the 'General Form PDE' modules.

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 to 1,000 s (COMSOL syntax: 1e3). 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 '2D Plot Group 1 - velocity.' Note how 'Surface 1' sets up a color table based on velocity magnitude 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 '2D Plot Group.' Next, check out '1D Plot Group - velocity(y)'. Notice how 'Line Graph 1' is set up to **plot $|\mathbf{v}|$ as a function of position** y, along the 'Cut Line' x = -200 m defined under 'Results / Datasets / Cut Line 2D 1.'

Under 'Export,' use 'Animation - file velocity' 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 'Animation - player velocity' is set up to play within COMSOL. To add your own animations, right-click on 'Export' and find 'Animation.' Finally, use 'Plot - velocity(v)' to export a version of the $|\mathbf{v}|$(y) plot we created in the previous paragraph.

Bonus: Remove the circular obstacle (right-click 'Circle 1' under 'Geometry 1' and disable it), then re-run the simulation. Alternatively, re-run the simulation while **sweeping through different parameter choices** (for example, tuning D to change the strength of neighbor coupling or tuning R to explore the effect of obstacle size). Right-click on 'Study 1' and choose 'Parametric Sweep.' As a quick exercise, click on the '+' icon and add a short list of choices for R: 15 m, 30 m. After computing, update your 'Results' plots to visualize this new dataset (Study 1/Parametric Solutions 1).

6. Moving to a Curved Surface in 3D. Now that we've covered some basics of the COMSOL interface and General Form PDE modules, we'll finally take a look at the curved-surface implementation of the Toner-Tu equations developed in this work. Open '3_HueschenPhillips_TonerTu_SphereDamping.mph.' Many things will look familiar (Parameters, the Random function used to set up our velocity initial condition, the Geometry where we've defined a sphere of radius R, several handy visualizations and plots under Results), but we want to highlight key new features. First, click on 'Variables' under 'Component 1 / Definitions.' Here, we've defined important pieces of the curved-surface formulation, such as the projection operator $\mathbf{P}$ and the surface velocity gradient tensor $\mathbf{G}$. Try to familiarize yourself with these quantities, referencing Section 3 ('Formulating the Surface Toner-Tu Equations for Numerical Implementation') in the main text.

Next, explore 'General Form Boundary PDE - velocity.' 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.' Within 'General Form PDE 1,' view the full curved-surface formulation of the Toner-Tu $\partial v_i^{||}/\partial t$ equations (eqn. 26 in the main text). Again, reference Section 3 in the main text and Section 8.2 ('Implementation in COMSOL Multiphysics') in the Appendix to sort through how the contents of $\mathbf{\Gamma}$, f, and d_a combine to **implement the curved-surface Toner-Tu equations**.

Similarly, explore 'General Form PDE - continuity,' which implements the curved-surface continuity equation (eqn. 27 in the main text).

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. 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 wildebeest) is defined in 'Parameters 1.'