Supplementary Materials

1 Step-By-Step Overview of the Model

As an initial step to our method, we first initialize all relevant variables. The bilayer concentration field $c(\mathbf{x},t)$ is initialized to 0 (to denote bare glass), the list of vesicles V is initialized to the empty set $\emptyset$ (to denote no adsorbed vesicles), the next vesicle arrival time t_{arrival} is generated from the exponential distribution $\text{Exp}(\lambda)$, and finally the current time is initialized to 0. From this we proceed to update these variables at each timestep with timestep size Δt. Our method can be viewed as having four basic steps per timestep:

(1) update the bilayer concentration field $c(\mathbf{x},t)$,

(2) handle vesicle arrival events,

(3) update the positions of the vesicles, and

(4) handle vesicle movement events.

1.1 Step 1: Update Concentration Field

To accomplish this step, we use a simple forward Euler approach:

$$c(\cdot,t+\Delta t) = c(\cdot,t) + \Delta t\left[\nabla^2\left(f'[c] - \varepsilon^2\nabla^2 c\right)\right].$$

In our implementation, we replace the differential operators with their appropriate discretized versions. See section 3.2 for more details on the Cahn-Hilliard equation.

1.2 Step 2: Vesicle Arrival Events

If a vesicle arrival happens within the current timestep, we generate an arrival site $\mathbf{x}^*$ uniformly from the entire substrate domain. First, we check if $\mathbf{x}^*$ is "crowded," as defined in section 3.5. If it is, then we say that the arriving vesicle immediately ruptures and deposits a bilayer patch on the substrate. Implementation-wise, this corresponds to adding to the bilayer concentration field $c(\cdot,t)$. We also decorate the new patch of bilayer, to enforce our hypothesis that bilayer fronts recruit vesicles from the solution very quickly.

If $\mathbf{x}^*$ isn't "crowded," we then check if there is enough bare substrate underneath the arriving vesicle to facilitate adsorption. What this means in our implementation is finding the average value of $c(\cdot,t)$ in a neighborhood of $\mathbf{x}^*$. If the average value is low enough, the vesicle adsorbs.

Once the type of arrival event has been handled, the next arrival time t_{arrival} is generated according to the exponential distribution $\text{Exp}(\lambda)$, and if t_{arrival} is still within the current timestep, step (2) is repeated.

1.3 Step 3: Update Vesicle Positions

To ensure vesicles "stick" to bilayer fronts, we force the vesicles to move with them. This is accomplished by applying a c-dependent force on each vesicle. Assuming the overdamped limit for the motion of the vesicles, the velocity of each is proportional to the force applied to it. Thus, the positions of the vesicles are updated via

$$\mathbf{x}_k = \mathbf{x}_k - m\Delta t\left(c(\mathbf{x}_k,t) - \frac{1}{2}\right)\nabla c(\mathbf{x}_k,t).$$

The constant m denotes the mobility of a vesicle. This is chosen large enough so that the vesicles do not lag behind the SLB edge, but small enough so that vesicle motion is not the limiting timestep. In our simulations, we found that a value of $m = \frac{1}{2}$ satisfied these constraints.

1.4 Step 4: Vesicle Movement Events

Because we allow the vesicles to move on the substrate, it may happen that a vesicle may move to a crowded region of the substrate. To account for this, we check a neighborhood of each adsorbed vesicle k. If the number of vesicles in the neighborhood of vesicle k is high enough, then vesicle k is removed from the list V, and a patch of bilayer is added to the bilayer concentration field c. As in step (1), vesicles are created to decorate the bilayer front on the substrate.

Finally, we remove any vesicle from V whose center lies in a region whose average c value is high (as defined in section 3.6), corresponding to when the vesicle is no longer touching bare substrate.