SUPPORTING INFORMATION
Effect of step anisotropy on crystal growth inhibition by immobile impurity stoppers

James P. Lee-Thorp,*,† Alexander G. Shtukenberg,*,‡ and Robert V. Kohn*,†

†*Courant Institute of Mathematical Sciences, New York University, NY, USA*

‡*Department of Chemistry, New York University, NY, USA*

E-mail: leethorp@cims.nyu.edu; shtukenberg@mail.ru; kohn.nyu.edu

Detailed description of our numerical algorithm

We first describe an explicit time-stepping analog of our algorithm that is sufficient to handle isotropic step motion problems. We then discuss our semi-implicit algorithm for anisotropic problems.

Explicit algorithm for isotropic step motion

Given the periodicity of our model problem, it suffices to consider a single period cell (the dotted square in Figure 1), which has side length d. The front of the evolving step in this cell can be described by a simple 1D curve in the plane. We first discretize this 1D curve in space with N grid points, (x_i, y_i), $i = 1, 2, \dots, N$, which are uniformly spaced with respect to the arc length, s. We initialize our algorithm with a horizontal line from $x_1 = 0$ to $x_N = d$, with $y_i \equiv 0$, $\forall i$. Because the curve is initially horizontal, the arc length is simply the length of the line.

The ends of the curve at $(x_1, y_1) = (0, 0)$ and $(x_N, y_N) = (d, 0)$, are initially "pinned" (fixed) at the impurity sites. In the isotropic case, $\kappa_\gamma = \kappa$ and each interior point along the curve evolves according to the growth law (5):

$$\frac{d}{dt}\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} n_{x_i} \\ n_{y_i} \end{pmatrix}(1 - \xi\kappa_i)\,, \quad i \in \{2, 3, \ldots, N-1\}, \tag{S.1}$$

where the normal at the point (x_i, y_i), $\mathbf{n}_i = (n_{x_i}, n_{y_i})^T$, can be computed from appropriate gradients with respect to arc length:

$$\mathbf{n}_i = \begin{pmatrix} n_{x_i} \\ n_{y_i} \end{pmatrix} = \frac{1}{\sqrt{[(\partial_s y)_i]^2 + [(\partial_s x)_i]^2}}\begin{pmatrix} -(\partial_s y)_i \\ (\partial_s x)_i \end{pmatrix}. \tag{S.2}$$

The curvature at each grid point is given by:

$$\kappa_i = -\frac{(\partial_s x)_i(\partial_s^2 y)_i - (\partial_s y)_i(\partial_s^2 x)_i}{\left([(\partial_s x)_i]^2 + [(\partial_s y)_i]^2\right)^{3/2}}. \tag{S.3}$$

We approximate the derivative operators (with respect to arc length), ∂_s and ∂_s^2, that appear in (S.2) and (S.3), using finite differences.

Equipped with $\mathbf{n}_i$ and κ_i, we can solve the interior ODE system (S.1) using an explicit time stepping scheme with a sufficiently small step size, Δt. For $i = 1, 2, \ldots, N$ and $j \in \mathbb{Z}^+$, we denote by (x_i^j, y_i^j), the i^{th} grid point along the front (curve) at time $t_j = j\Delta t$. Our explicit scheme is then

$$\begin{pmatrix} x_i^{j+1} \\ y_i^{j+1} \end{pmatrix} = \begin{pmatrix} x_i^j \\ y_i^j \end{pmatrix} + \Delta t \begin{pmatrix} n_{x_i^j} \\ n_{y_i^j} \end{pmatrix}\left(1 - \xi\kappa_i^j\right). \tag{S.4}$$

The evolution of the ends of the front is determined by whether the front is pinned or free. While the front is pinned at the impurity sites, we impose Dirichlet boundary conditions: $(x_1^j, y_1^j) = (0, 0)$ and $(x_N^j, y_N^j) = (d, 0)$, for all $j \in \mathbb{Z}^+$. As the front evolves, the pinning angle between the front and the vertical at the impurity sites,

$$\phi = \operatorname{atan}\left(\frac{\Delta x}{\Delta y}\right) \approx \operatorname{atan}\left(\frac{x_i - x_{i-1}}{y_i - y_{i-1}}\right), \quad i \in \{2, N\}, \tag{S.5}$$

decreases. When the pinning angle is less than the critical pinning threshold, $\phi < \phi_c$, the front breaks through the impurity fence. At this time, we impose periodic boundary conditions: $(x_1^j, y_1^j) = (x_N^j, y_N^j)$, and evolve the end points according to the explicit scheme (S.4). Once the front hits the next impurity fence, it becomes pinned and we impose Dirichlet boundary conditions again.

There are two subtleties to evolving the free front end points immediately after breakthrough. Firstly, because of the periodic boundary conditions, the front forms a sharp angle at its end points. This makes numerical computation of the normal vector (S.2) unreliable at the end points just after breakthrough. We therefore impose that the normal vector points in the vertical direction: $\mathbf{n}_i = (0, 1)$, $i \in \{1, N\}$. Secondly, immediately after breakthrough, the curvature at the end points, κ_1 and κ_N is very large. Therefore, the velocity near the end points is very large, so we need to ensure that our time step is sufficiently small.

Finally, because the growth law (S.1) determines the evolution of the front in the normal direction only, the grid points along the front will in general not remain equally distributed (with respect to arc length) along the curve. At each time step, we remedy this by reparameterizing the grid points along the curve using numerical interpolation.

Semi-implicit algorithm for anisotropic step motion

In the anisotropic case, $\kappa_\gamma(\theta) = (\gamma(\theta) + \gamma''(\theta))\kappa(\theta)$. The anisotropic surface energy given by Eq. (6) leads to the formation of facets. In this case, the growth law Eq. (5) becomes too stiff to solve explicitly so we turn to a semi-implicit scheme. To simplify our discussion, we assume the mobility is constant, $M(\theta) \equiv 1$. (Introducing variable mobility involves straightforward adjustments to this algorithm.)

The setup in the semi-implicit scheme is very similar to that of the explicit scheme. We uniformly discretize the front (a 1D curve) with the points (x_i, y_i), where $i = 1, 2, \dots, N$. We initialize our algorithm with a horizontal line. The ends of the curve are initially pinned at the impurity sites with Dirichlet boundary conditions. Each interior point evolves according to the growth law (Eq. (5)):

$$\frac{d}{dt}\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} n_{x_i} \\ n_{y_i} \end{pmatrix} \left(1 - \xi\left(\gamma(\theta_i) + \gamma''(\theta_i)\right)\kappa(x_i, y_i)\right), \quad i \in \{2, 3, \dots, N-1\}. \tag{S.6}$$

The curvature term, $\kappa(x_i, y_i)$, is handled differently in the semi-implicit scheme. As before, for $i = 1, 2, \dots, N$ and $j \in \mathbb{Z}^+$, let (x_i^j, y_i^j) denote i^{th} grid point along the front at time $t_j = j\Delta t$. We evaluate the curvature term semi-implicitly in time:

$$\kappa(x_i, y_i) = -\frac{(\partial_s x)_i^j (\partial_s^2 y)_i^{j+1} - (\partial_s y)_i^j (\partial_s^2 x)_i^{j+1}}{\left([(\partial_s x)_i^j]^2 + [(\partial_s y)_i^j]^2\right)^{3/2}}, \tag{S.7}$$

where $i \in \{1, 2, \dots, N\}$ and $j \in \mathbb{Z}^+$. In Eq. (S.7), we evaluate the first order derivatives terms at the current time (time step j) and second order derivatives terms at the future time (time step $j+1$). This choice is motivated by the fact that the second order derivative terms are the largest source of numerical stiffness.

The normal $\mathbf{n}_i = (n_{x_i}, n_{y_i})^T$ is computed as in Eq. (S.2) at the current time (time step j): $\mathbf{n}_i = (n_{x_i^j}, n_{y_i^j})^T$. From $\mathbf{n}_i$, we also compute the polar angle at each grid point, θ_i^j, and the surface energy terms, $\gamma(\theta_i^j) + \gamma''(\theta_i^j)$.

We solve the interior ODE system Eq. (S.6) using a implicit time stepping scheme. Let $\mathbf{x}^j = (x_1^j, x_2^j, \dots, x_N^j)^T$ and $\mathbf{y}^j = (y_1^j, y_2^j, \dots, y_N^j)^T$. Let I denote the $2N \times 2N$ identity matrix, D the $2N \times 2N$ matrix with diagonal entries $(n_{x_1}, \dots, n_{x_N}, n_{y_1}, \dots, n_{y_N})$ and $\mathbf{c}$ the vector $(1, 1, \dots, 1)^T$ of length $2N$. Then, at the level of matrices, the semi-implicit scheme, with time step Δt, takes the form:

$$(I + \Delta t D H) \begin{pmatrix} \mathbf{x}^{j+1} \\ \mathbf{y}^{j+1} \end{pmatrix} = \left(\begin{pmatrix} \mathbf{x}^{j} \\ \mathbf{y}^{j} \end{pmatrix} + \Delta t D \mathbf{c} \right), \tag{S.8}$$

where the matrix $H = H((\partial_s x)_i^j, (\partial_s y)_i^j; \gamma(\theta_i), \gamma''(\theta_i))$ appropriately encodes the surface energy terms and the first order derivatives terms from the curvature given by Eq. (S.7) that are evaluated at time step j.

As the front evolves, we monitor the pinning angle, ϕ, at the impurity sites. When $\phi < \phi_c$, the front breaks through the impurity fence and we impose periodic boundary conditions at the end points of the front. When the front hits the next impurity fence, we impose Dirichlet boundary conditions again. We handle the normal vectors at the end points just after breakthrough in a similar manner to the explicit scheme. We also take similar precautions with the time step and use the same interpolation approach to ensure that the grid points remain equally distributed (with respect to arc length) along the curve.

Benchmarks

We benchmarked our algorithm as follows. We employ an ODE solver to compute the stationary (pinned) configuration of Eq. (5) for fixed ξ and fixed lattice size. We then run our dynamical front tracking algorithm for the same ξ and lattice size, and confirm that the dynamical solution converges to the stationary configuration. We performed this “pinned configuration benchmark” for several anisotropic surface energies of the form given in Eq. (6).

Effect of numerical regularization

As explained in the main text, for numerical purposes we approximate a surface energy γ given by Eq. (6) by the regularization given by Eq. (8).

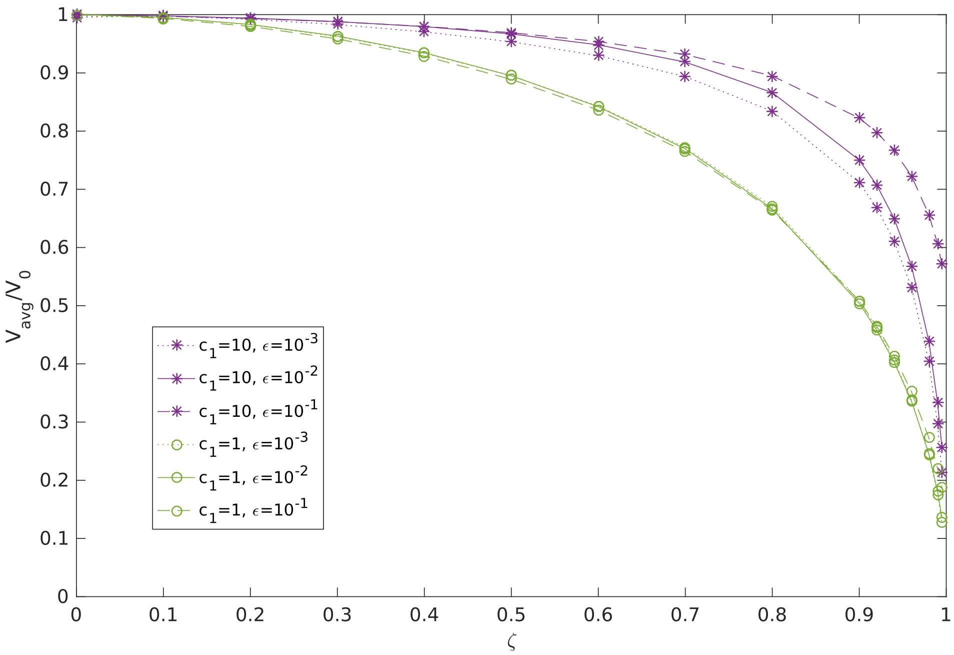

Figure S.1: Comparison of different regularizations for anisotropic, single-faceted step motion. Before regularization, γ is given by Eq. (6) with $c_2 = c_3 = 0$ and $c_1 = 1$ or 10. The growth law is $V = V_0(1 - \xi\kappa_\gamma(\theta))$ and the horizontal axis is $\zeta = \xi/\xi_c$. Data points with the same regularization are connected by dashed lines ($\epsilon = 10^{-1}$), solid lines ($\epsilon = 10^{-2}$), or dotted lines ($\epsilon = 10^{-3}$).

Figure S.1 compares different choices of the regularization parameter ϵ, when $c_2 = c_3 = 0$ and $c_1 = 1$ or 10. For relatively weak anisotropy (such as $c_1 = 1$) there is no noticeable difference between the $\epsilon = 10^{-2}$ and 10^{-3} regularizations. As we increase the anisotropy to $c_1 = 10$, we begin to notice some discrepancies. Since refining the regularization has a computational cost – we require smaller time steps for smaller ϵ – we opted to use $\epsilon = 10^{-2}$ throughout the article.

Derivation of analytical formulas for polygonal cases

For the double-facet case, all angles are $\pi/2$, the percolation parameter is $\zeta = L_{ch}/d$ and $\alpha = L_{cs}/L_{ch} = \gamma_h/\gamma_s$; see Figure S.2. During stage 1, the vertical facets are immobile but the horizontal facet moves with $V_t = V_0(1 - L_{ch}/d)$ (Eq. (11)) until it reaches a vertical position $y_1 = L_{cs}$ at time $\tau_1 = L_{cs}/V_t$. At this moment, the side facets start moving and immediately merge forming a straight infinite horizontal step; in this case, stage 2 is absent and $\tau_2 = 0$. Finally, at stage 3, growth proceeds with $V_t = V_0$ for $y_1 < y \leq d$ and takes time $\tau_3 = (d - L_{cs})/V_0$. The average step velocity can be computed from Eq. (13):

$$V_{avg} = d \left(\frac{L_{cs}}{V_0(1 - L_{ch}/d)} + \frac{d - L_{cs}}{V_0} \right)^{-1}, \tag{S.9}$$

which can then be simplified to the final Eq. (14).

For the triple-faceted case, all angles are $2\pi/3$, the percolation parameter is $\zeta = (L_{ch} + L_{cs})/d$, and $\alpha = L_{cs}/L_{ch} = \gamma_h/(2\gamma_s - \gamma_h)$; see Figure S.3. During stage 1, the side facets are immobile and the top facet moves with step velocity given by Eq. (11):

$$V_t = \frac{dy}{dt} = V_0 \left(1 - \frac{L_{ch}}{L_t} \right). \tag{S.10}$$

The length of horizontal facet is $L_t = d - 2y\cot(\pi/3)$ (see Figure S.3) and stage 1 encompasses the range $0 \leq y < y_1 = \sqrt{3}L_{cs}/2$. One can therefore integrate Eq. (S.10) to obtain the time

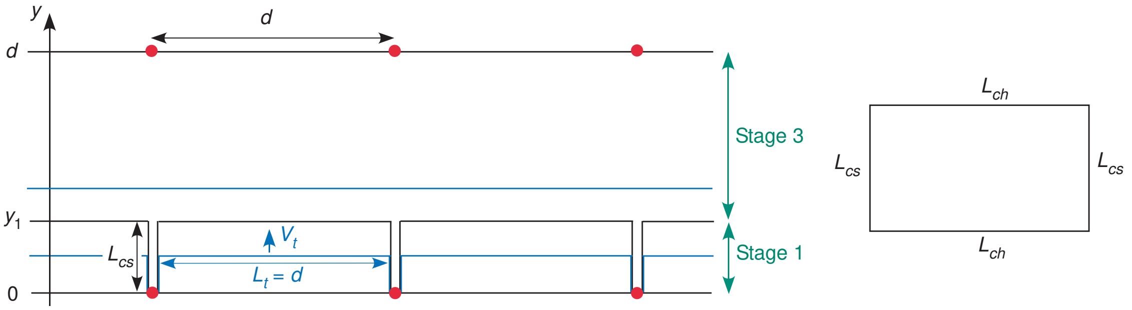

Figure S.2: Schematic illustrating the derivation of the analytical average velocity for double-facet case, Eq. (14). The step is propagating in the positive y direction. Stoppers are shown by red dots. Black lines show start/end step configurations for each stage and the blue lines illustrate typical intermediate configurations for each stage. The equilibrium Wulff shape of the corresponding 2D island is shown in the right inset.

needed to complete stage 1:

$$\tau_1 = \frac{\sqrt{3}}{2V_0}\left(L_{cs} - L_{ch}\ln\left(\frac{d - L_{cs} - L_{ch}}{d - L_{ch}}\right)\right). \tag{S.11}$$

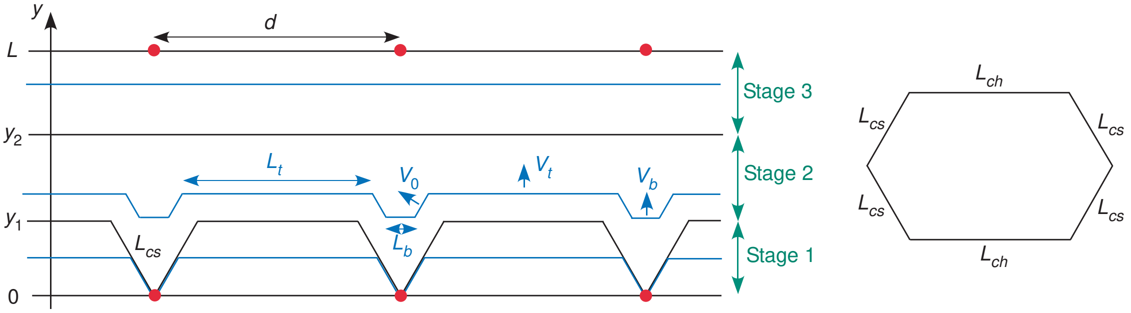

Figure S.3: Schematic illustrating derivation of average velocity for triple-facet case, Eq. (15). The step is propagating in positive y direction. Stoppers are shown by red dots. Black lines show start/end step configurations for each stage and the blue lines illustrate intermediate configurations for each stage. The equilibrium Wulff shape of the corresponding 2D island is shown in the right inset.

Stage 2: The moment the step breaks through the array of stoppers, side facets start moving, and the bottom facet appears (see Figure S.3). In accordance with Eqs. (10) and (11) and Figure 5, velocities of the top horizontal, bottom horizontal, and side facets are given by $V_t = V_0(1 - L_{ch}/L_t)$, $V_b = V_0(1 + L_{ch}/L_b)$, and $V_s = V_0$, respectively. The bottom facet moves faster than the top one. The time required to merge corresponds to the duration of stage 2, τ_2, and will result in step position y_2. Using equations for step velocities one can find the coordinates of facet interception points and write equations determining the top (L_t) and bottom (L_b) facet lengths:

$$\frac{dL_t}{dt} = \frac{2}{\sqrt{3}}(2V_0 - V_t), \quad \text{and} \tag{S.12}$$

$$\frac{dL_b}{dt} = \frac{2}{\sqrt{3}}(V_b - 2V_0). \tag{S.13}$$

Combining these Eqs. (S.12) and (S.13) with the equations for step velocities one can write four equations that completely determine stage 2:

$$\begin{aligned}
\frac{dL_t}{dt} &= \frac{2V_0}{\sqrt{3}}\left(1 + \frac{L_{ch}}{L_t}\right), \\
\frac{dL_t}{dy} &= \frac{dL_t}{V_t dt} = \frac{2}{\sqrt{3}}\left(\frac{L_t + L_{ch}}{L_t - L_{ch}}\right), \\
\frac{dL_b}{dt} &= \frac{2V_0}{\sqrt{3}}\left(\frac{L_{ch}}{L_b} - 1\right), \quad \text{and} \\
\frac{dL_b}{dy} &= \frac{dL_b}{V_b dt} = \frac{2}{\sqrt{3}}\left(\frac{L_{ch} - L_b}{L_b + L_{ch}}\right).
\end{aligned}$$

The limits of integration for these equations are determined by the initial and final times of stage 2 ($t = \tau_1$ and $t = \tau_1 + \tau_2$); the initial and final lengths of the top facet ($L_t(\tau_1) = d - L_{cs}$ and $L_t(\tau_1 + \tau_2)$); the initial and final lengths of the bottom facet (0 and $d - L_t(\tau_1 + \tau_2)$); the initial and final y positions of top facet (y_1 and y_2) and the initial and final positions of the bottom facet (0 and y_2). Integrating each equation, and writing L_t for $L_t(\tau_1 + \tau_2)$ and L_b for

$L_b(\tau_1+\tau_2)$ to simplify the notation, we get

$$\tau_2 = \frac{\sqrt{3}}{2V_0}\left(L_t - d + L_{cs} - L_{ch}\ln\left(\frac{L_t + L_{ch}}{d - L_{cs} + L_{ch}}\right)\right), \tag{S.14}$$

$$y_2 = \frac{\sqrt{3}}{2}\left(L_t - d + 2L_{cs} - 2L_{ch}\ln\left(\frac{L_t + L_{ch}}{d - L_{cs} + L_{ch}}\right)\right), \tag{S.15}$$

$$\tau_2 = -\frac{\sqrt{3}}{2V_0}\left(L_b + L_{ch}\ln\left(\frac{L_{ch} - L_b}{L_{ch}}\right)\right), \quad \text{and} \tag{S.16}$$

$$y_2 = -\frac{\sqrt{3}}{2}\left(L_b + 2L_{ch}\ln\left(\frac{L_{ch} - L_b}{L_{ch}}\right)\right). \tag{S.17}$$

Eliminating τ_2 from (S.14) and (S.16) gives a nonlinear equation relating L_b and L_t. Eliminating y_2 from (S.15) and (S.17) gives a different nonlinear equation involving L_b and L_t. Comparing the two, we reach the conclusion that

$$-L_{cs} + L_{ch}\ln\left(\frac{L_t + L_{ch}}{d - L_{cs} + L_{ch}}\right) = L_{ch}\ln\left(\frac{L_{ch} - L_b}{L_{ch}}\right).$$

Taking the exponential of both sides, we get a linear relation between L_t and L_b:

$$e^{-\alpha}\frac{L_t + L_{ch}}{d - L_{cs} + L_{ch}} = \frac{L_{ch} - L_b}{L_{ch}}$$

where (as usual) $\alpha = L_{cs}/L_{ch}$. But at the end of stage 2 we also have (by definition) $L_t + L_b = d$. These two linear equations determine both L_t and L_b; in particular, they give

$$L_b = L_{ch}\left(1 - \frac{d}{e^{\alpha}(d + L_{ch} - L_{cs}) - L_{ch}}\right). \tag{S.18}$$

Substituting this value for L_b into Eqs. (S.16) and (S.17), one obtains expressions for τ_2 and

y_2, respectively:

$$\tau_2 = -\frac{\sqrt{3}}{2V_0} L_{ch} \left(1 - \frac{d}{e^{\alpha}(d + L_{ch} - L_{cs}) - L_{ch}} + \ln\left(\frac{d}{e^{\alpha}(d + L_{ch} - L_{cs}) - L_{ch}}\right)\right), \quad \text{and} \tag{S.19}$$

$$y_2 = -\frac{\sqrt{3}}{2} L_{ch} \left(1 - \frac{d}{e^{\alpha}(d + L_{ch} - L_{cs}) - L_{ch}} + 2\ln\left(\frac{d}{e^{\alpha}(d + L_{ch} - L_{cs}) - L_{ch}}\right)\right). \tag{S.20}$$

Stage 3 occurs over the range $y_2 \leq y < d$ and proceeds with the unimpeded step velocity $V = V_0$. Therefore, stage 3 takes time

$$\tau_3 = \frac{d - y_2}{V_0}. \tag{S.21}$$

The average step velocity is calculated by combining Eqs. (13), (S.11), (S.20), and (S.21):

$$V_{avg} = V_0 d \left(d + \frac{\sqrt{3}L_{ch}}{2} \ln\left(\frac{e^{\alpha} d(d - L_{ch})}{(e^{\alpha}(d + L_{ch} - L_{cs}) - L_{ch})(d - L_{cs} - L_{ch})}\right)\right)^{-1}, \tag{S.22}$$

which can be simplified to give the final form Eq. (15).

Step velocity as a function of facet length

According to Eq. (9), for the front motion law $V = (1 - \xi\kappa_{\gamma})$, the velocity V_f of the top facet is linearly related to the length L_f of the facet:

$$V_f = \left(1 - \xi\frac{[\gamma'(0)]}{L_f}\right) = \left(1 - \xi_c[\gamma'(0)]\frac{\zeta}{L_f}\right),$$

where $\zeta = \xi/\xi_c$ is the percolation parameter. Figure S.4 assesses the extent to which our numerically-computed solutions satisfy this law, focusing on a single-facet example ($c_1 = 10$, $c_2 = c_3 = 0$) and a symmetric triple-facet example ($c_3 = 0.1$, $c_1 = c_2 = 0$). In both cases the numerically-computed graph of V_f vs ζ/L_f is very nearly a straight line. The numerically-observed slopes are slightly different from those obtained theoretically. Indeed, for both examples $\xi_c = 1/2$ (a result obtained both numerically and analytically), so for the single-facet example the theoretical slope is $-\xi_c[\gamma'(0)] = -c_1/(1+c_1) = -0.91$ while the numerically-obtained slope is -0.94; for the triple-facet example the theoretical slope is $-\xi_c[\gamma'(0)] = -3c_3/(1+c_3) = -0.27$ while the numerically-obtained slope is -0.35. We attribute the discrepancy between the theoretical and numerical slopes to the effect of regularization – specifically, to the fact that in the numerically-computed, regularized setting the "facet" is not exactly flat, so there is some ambiguity in identifying its endpoints and measuring its length. (For Figure S.4 the regularization was $\epsilon = 10^{-2}$ and the facet was defined to be the part of the front where the normal angle $|\theta|$ satisfied $|\theta| < 10\epsilon = 0.1$.)

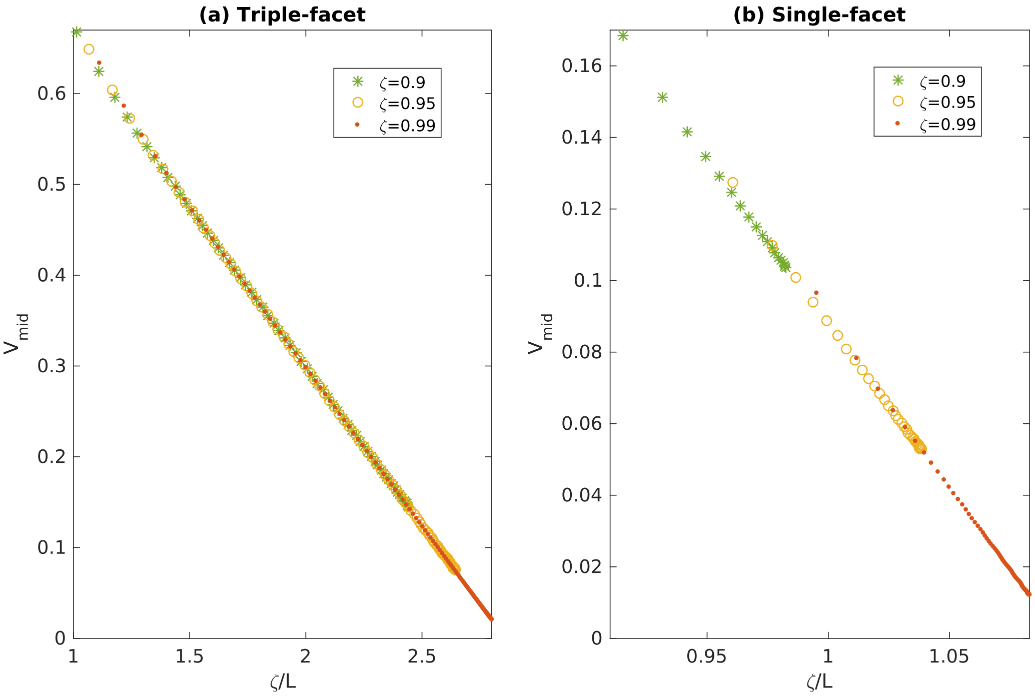

Figure S.4: Velocity of the top facet - characterized by the (unnormalized) midpoint velocity, V_{mid}, plotted as a function of the percolation parameter multiplied by inverse facet length, ζ/L. **(a)**: Triple-facet motion ($c_3 = 0.1$). **(b)**: Single-facet motion ($c_1 = 10$). As the step evolves and and the facet length decreases from $L = 1$, the velocity of the facet decreases linearly. Averaging across the three linear fits in each case, the slope of V_{mid} vs ζ/L is -0.35 in the triple-facet case (a) and -0.94 in the single-facet case (b).