

Aberystwyth University

Remarks on the accuracy of algorithms for motion by mean curvature in bounded domains

Cox, Simon John; Mishuris, Gennady

Published in:

Journal of Mechanics of Materials and Structures

DOI:

[10.2140/jomms.2009.4.1555](https://doi.org/10.2140/jomms.2009.4.1555)

Publication date:

2008

Citation for published version (APA):

Cox, S. J., & Mishuris, G. (2008). Remarks on the accuracy of algorithms for motion by mean curvature in bounded domains. *Journal of Mechanics of Materials and Structures*, 4(9), 1555–1572.
<https://doi.org/10.2140/jomms.2009.4.1555>

General rights

Copyright and moral rights for the publications made accessible in the Aberystwyth Research Portal (the Institutional Repository) are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Aberystwyth Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Aberystwyth Research Portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

tel: +44 1970 62 2400

email: is@aber.ac.uk

Remarks on the accuracy of algorithms for motion by mean curvature in bounded domains

S.J. Cox^{†‡}, G. Mishuris^{†‡}

[†] Institute of Mathematics and Physics, Aberystwyth University, UK

[‡] Wales Institute of Mathematical and Computational Sciences

Abstract Simulations of motion by mean curvature in bounded domains, with applications to bubble motion and grain growth, rely upon boundary conditions that are only *approximately* compatible with the equation of motion. Three closed form solutions for the problem exist, governing translation, rotation and expansion of a single interface [1], providing the only benchmarks for algorithm verification. We derive new identities for the translation solution. Then we estimate the accuracy of a straightforward algorithm to recover the analytical solution for different values of the velocity V given along the boundary. As expected, for large V the error can reach unacceptable levels especially near the boundary. We discuss factors influencing the accuracy and propose a simple modification of the algorithm which improves the computational accuracy.

1 Introduction

Foams are ubiquitous in everyday life [2]. They are used daily in the home, in both food and cleaning products. Moreover, their industrial uses are varied and contribute significantly to the UK economy. For example, foams are used to separate ores (e.g. zinc, lead) from the rock in which they are found, and to push oil out of porous rock. They are also used in the decontamination cleaning of vessels, and in firefighting. In all these applications, it is the flow of foam that is driving the process. Therefore, if it were possible to predict how a foam would behave in this range of scenarios, when subjected to a complex collection of deformations, each process could be made more efficient, either in terms of yield/output or cost-effectiveness.

Aqueous foams and concentrated emulsions, which are very similar to foams, have peculiar and remarkable properties: they are complex fluids whose properties lie between the familiar extremes of liquid and solid. For small strains, a foam behaves as an elastic solid while at large strains (or strain-rate) a foam moves like a liquid. They therefore generate a rich range of behaviours, but with a well-defined local structure that obeys Plateau's geometric laws and the Laplace-Young law that balances interface curvatures with bubble pressure differences. Our idea is to use the precise, known, structure of a foam to predict its rheological response.

One of the leading tools for the analysis of foam structure is the Surface Evolver developed by Ken Brakke [3]. This consists of software expressly designed for the modelling of soap bubbles, foams, and other liquid surfaces shaped by minimizing energy (such as surface tension and gravity), and subject to various constraints (such as bubble volumes). Originally designed to model grain growth in metals, it is easily extended to the case of foams, and has been used in a number of engineering disciplines to look at, for example, capillary surfaces, solder joints and fluid behaviour in microgravity. A surface is represented as a collection of triangles, so that the complicated topologies found in foams are routinely handled. In particular, the Evolver can deal with the topological changes encountered during foam flow.

Here, we concentrate on a two-dimensional model of a foam, for ease of analysis. We use the following "viscous froth model" (VFM) [4] to examine the evolution of foam films under shear:

$$p_i - p_j = \gamma \kappa_{ij} + \lambda v_{ij} \quad (1)$$

where γ is the surface tension in the films (assumed constant), κ_{ij} is the curvature of the film separating bubble i from bubble j and v_{ij} its normal velocity. Each bubble has a well-defined pressure p_i . In essence, the model augments the (equilibrium) Laplace-Young law with a term proportional to the velocity; the constant of proportionality λ is a drag coefficient, representing the external dissipation due to friction with the walls of

the container, allowing the investigation of strain-rate effects. If the external dissipation is negligible, then we recover the Laplace-Young law for a foam in equilibrium and a model for quasi-static evolution.

A Surface Evolver simulation with the VFM proceeds in the following way. Each film is discretised into short straight segments and curvatures κ_{ij} are calculated pointwise at the intersections of these segments. Bubble pressures are obtained by deriving a matrix equation based upon an integral of (1) around each cell. Then the endpoints of the segments are moved according to the motion equation (1) using a small time step Δt . Boundary conditions are applied according to the problem under consideration.

In the ideal model of the evolution of crystalline grains in a polycrystalline metal, known as normal grain growth, the size of each grain evolves due to the normal motion of each of its boundaries [5]. Each boundary has a certain ‘‘mobility’’ λ , and moves in such a way as to reduce the total perimeter of the pattern, as in foams. However, without area (volume in 3D) constraints, this is motion by *mean curvature*. As noted above, this is a well-posed limit of the VFM (1) when bubble pressure differences are negligible, such as in freely translating films (grain boundaries) and ordered (hexagonal) foams. The grain growth model requires that we have 120° at vertices, justifying *a posteriori* that assumption in the VFM.

Here, we ask how such a solution can be commensurate with the boundary of the domain and how well this solution can be calculated numerically. To the best of our knowledge, answers to this question are not available in the literature.

2 Curvature-driven motion of a bounded interface

2.1 Problem formulation

In vector form, the motion of an interface in the model of ideal grain growth (1) can be described by

$$\mathbf{v} = \kappa \mathbf{n}, \quad (2)$$

where \mathbf{n} and \mathbf{s} are the normal and tangential unit vectors to the interface (see Fig. 1):

$$\mathbf{s} = [n_1, n_2], \quad \mathbf{n} = [-n_2, n_1]. \quad (3)$$

If the representation of the interface is taken in the form:

$$x = x(y, t), \quad y \in [\underline{y}(t), \bar{y}(t)], \quad (4)$$

then the vector components n_1, n_2 are calculated as follows

$$n_1 = \frac{dx}{ds} = \cos \theta = \frac{x'_y}{\sqrt{1 + (x'_y)^2}}, \quad n_2 = \frac{dy}{ds} = \sin \theta = \frac{1}{\sqrt{1 + (x'_y)^2}}, \quad (5)$$

where $ds = \sqrt{(dx)^2 + (dy)^2}$ and θ is the tangential angle to the interface (see Fig. 1).

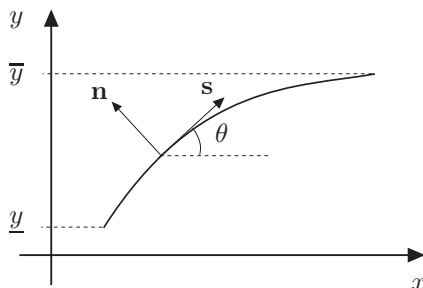


Figure 1: The bounded interface considered here.

Finally, the vector $\mathbf{v} = [v_1, v_2]$ is the instantaneous velocity of the point (x, y) lying on the interface at time t and κ is the curvature of the interface at that point:

$$\kappa = \frac{d\theta}{ds} = \frac{-x''_{yy}}{\sqrt{(1 + (x'_y)^2)^3}}. \quad (6)$$

Equation (2) can be also written in component form:

$$v_n = \mathbf{v}\mathbf{n} = -v_1n_2 + v_2n_1 = \kappa, \quad (7)$$

$$v_s = \mathbf{v}\mathbf{s} = v_1n_1 + v_2n_2 = 0. \quad (8)$$

In this paper we will consider only Mullins' solution [1], also known as the grim reaper (see below), which is symmetrical with respect to the x -axis. Taking into account the direction of the interface motion, we can assume in what follows that:

$$n_1 > 0, \quad n_2 > 0, \quad y'_x > 0, \quad 0 < \theta < \pi/2, \quad y''_{xx} < 0, \quad \kappa < 0, \quad (9)$$

$$v_n < 0, \quad v_s > 0, \quad (v_2 < 0, \quad v_1 > 0). \quad (10)$$

Equation (7) is widely discussed in the literature [1, 6], while the second equation (8) is somehow usually forgotten in this context. If one is only interested in reconstruction of the interface position at any time step an approach based only on equation (7) is sufficient. However, if it required to control the position of each tessellation point along the interface, as in the case of numerical computation, then both equations are equally important. In particular, equation (8) allows us to find relation between the two unknown components of the velocity vector \mathbf{v} and the normal vector \mathbf{n} in the form:

$$n_1 = -\frac{v_2}{v_1}n_2. \quad (11)$$

This allows us to eliminate components of the normal vector \mathbf{n} from equation (7) to give:

$$-\left(v_1 + \frac{v_2^2}{v_1}\right) = \frac{d\theta}{dy}. \quad (12)$$

2.2 Mullins' solution for translation revisited

Let us assume that the interface conserves its shape but moves in the x -direction with a constant speed V . We consider two points A and C having the same y -coordinate $y = y_0$ at two consecutive time steps t_0 and $t_0 + dt$ (see Fig. 2). It is clear that these two points correspond to two different material points. Namely, there exists a point B on the interface at time t_0 which moves according to the curvature law (2) to the point C for an infinitesimally small time step dt . If the coordinates of the point A are (x_0, y_0) then the coordinates of B and C can be written $(x_0 + dx, y_0 + dy)$ and $(x_0 + Vdt, y_0)$.

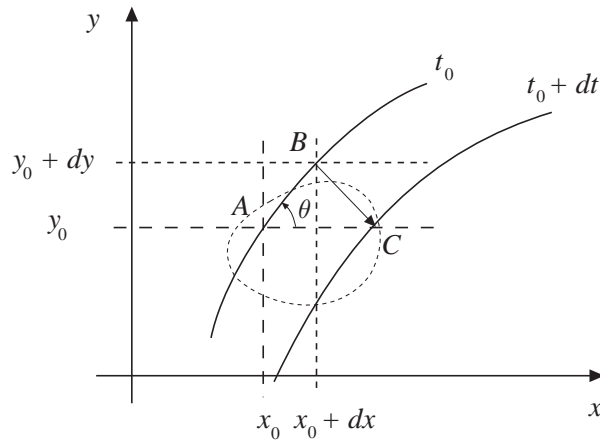


Figure 2: Interface under translational motion at two consecutive instants in time t_0 and $t_0 + dt$.

Note that

$$\overline{BC} = \mathbf{v}(x_0 + dx, y_0 + dy)dt = \mathbf{v}(x_0, y_0)dt + O(dt ds).$$

On the other hand, $|BC| = V \sin \theta dt$, $\tan \theta = v_1/|v_2|$. As a result one can conclude:

$$V = \frac{1}{v_1(y)} (v_1^2(y) + v_2^2(y)), \quad y \in [\underline{y}(t), \overline{y}(t)]. \quad (13)$$

This relation immediately follows in case of the translation movement of the interface in x -direction from (12).

Now, to reconstruct the solution obtained by Mullins [1] it is sufficient to substitute (2) into (12) to have:

$$\pi/2 - Vy = \theta. \quad (14)$$

Here we have taken into account the second of the two symmetry conditions at the point $y = 0$:

$$v_2(0) = 0, \quad \theta(0) = \pi/2. \quad (15)$$

Equation (14) can be written in the form

$$\cot Vy = y'_x, \quad (16)$$

which after direct integration leads to Mullins' solution [1]:

$$x(y) = x(0) + Vy - \frac{1}{V} \log \cos(Vy), \quad y \in [0, \bar{y}(t)]. \quad (17)$$

This solution exists only under the condition $\bar{y}(t) < h_{max}$, where

$$h_{max} = \frac{\pi}{2V}. \quad (18)$$

Note also that the angle θ defined by such a solution monotonically decreases in the interval $y \in (0, h)$ ($h < h_{max}$), taking values

$$\theta(y) \in (\theta_{min}, \pi/2), \quad \theta_{min} = \theta_{min}(h) = \pi/2 - Vh. \quad (19)$$

It is now possible to write analytical representations of all problem variables in the interval $y \in (0, h)$:

$$v_1 = V \cos^2 Vy, \quad v_2 = -\frac{1}{2}V \sin 2Vy, \quad n_1 = \sin Vy, \quad n_2 = \cos Vy, \quad \kappa = V \cos Vy. \quad (20)$$

Note that the first symmetry condition (15) has not been used but the reconstructed Mullins solution (17) satisfies it automatically. It exhibits the following asymptotics near the symmetry axis:

$$x(y) = x(0) + Vy + \frac{1}{2}Vy^2 + O(y^4), \quad y \rightarrow 0, \quad (21)$$

and has an asymptote near the maximal thickness:

$$x(y) \sim -\frac{1}{V} \log(h_{max} - y) + O(1), \quad y \rightarrow h_{max}. \quad (22)$$

Remark: Note that relation (13) also represents the boundary condition for any moving interface whose upper point lies on the line $y = \bar{y} = h$ and which moves in the x -direction with velocity V . Moreover, the velocity can be in that case a function of time $V = V(t)$:

$$V(t)v_1(h, t) = v_1^2(h, t) + v_2^2(h, t). \quad (23)$$

Substituting (2) into (23) such a boundary condition can be equivalently rewritten in other forms:

$$\kappa(h, t) = -V(t)n_2(h, t), \quad \text{or} \quad x''_{yy}(h, t) = V(t) (1 + (x'_y(h, t))^2). \quad (24)$$

2.3 Arbitrary instantaneous solution of equations (7) - (8) in bounded domain.

Let us consider any instantaneous solution of the equations (7) - (8) with the prescribed boundary condition (23). This effectively means that, for a particular time t , the velocity components $v_1(y, t)$ and $v_2(y, t)$ as well as the end points of the interface $y = \bar{y}(t)$ and $y = h$ are known as functions of the variable y and the problem is to determine using this information the position of the interface in space variables (y, x)

Let us introduce a function which in what follows is considered known:

$$F(y) = v_1(y) + \frac{v_2^2(y)}{v_1(y)} > 0. \quad (25)$$

Note that the condition (13) may be not valid at all inside the interval $y \in (\underline{y}, h)$ as it was for Mullins' solution; as a result, $F(y)$ is not a constant, in general. Equation (12) can be integrated to give:

$$-\int_{\underline{y}}^y F(\xi) d\xi = \theta(y) - \underline{\theta}, \quad (26)$$

and should be considered together with equation (11) which, in this case, takes the form:

$$\frac{dx}{dy} = -\frac{v_2}{v_1} = \cot \theta, \quad (27)$$

or

$$x(y) = \underline{x} - \int_{\underline{y}}^y \frac{v_2(\xi)}{v_1(\xi)} d\xi. \quad (28)$$

Equations (26) and (27) together indicate that the functions v_1 and v_2 cannot be chosen arbitrarily to satisfy the vectorial equation (2) as one might expect. Instead, the following identity has to be satisfied:

$$\arctan \frac{v_1}{v_2} = \int_{\underline{y}}^y F(\xi) d\xi + \underline{\theta}, \quad (29)$$

or writing $w = v_1/v_2$, equation (29) becomes:

$$\frac{w'}{1+w^2} = \frac{(1+w^2)v_2}{w}, \quad (30)$$

which leads to the following identity valid within the entire interval $y \in (\underline{y}, h)$:

$$v_1^2(y) = -v_2^2(y) \left(\frac{1}{2 \int_{\underline{y}}^y v_2(\xi) d\xi + C} + 1 \right). \quad (31)$$

Note that any solution of equations (7) - (8) satisfies this additional relation, which makes sense only under the additional constraint:

$$-1 \leq 2 \int_{\underline{y}}^y v_2(\xi) d\xi + C \leq 0. \quad (32)$$

As $0 < \theta < \underline{\theta}$, one can also deduce another condition which has to be true for any admissible velocities:

$$0 \leq \int_{\underline{y}}^y F(\xi) d\xi \leq \underline{\theta}. \quad (33)$$

Note that the constant \underline{x} in (28) is arbitrary (it changes only the position of the interface in the x -direction and does not influence any other variables). To determine the other constants $\underline{\theta}$ and C , we need to use boundary conditions at the end of the interface. Thus, condition (23) together with (31) leads to

$$v_1(h) = V(1 + C + 2I_2), \quad v_2(h) = -V\sqrt{-(1 + C + 2I_2)(C + 2I_2)}. \quad (34)$$

where we denote

$$I_2 \equiv I_2(h) = \int_{\underline{y}}^h v_2(\xi) d\xi.$$

If the boundary condition on the other end is given in the form

$$\theta(\underline{y}) = \underline{\theta}, \quad (35)$$

then all the constants have been defined. Such an instantaneous solution, assuming that the functions $v_1(y)$ and $v_2(y)$ satisfy equation (29), conditions (34) and restrictions (32) and (33), can be realised during interface evolution at some step.

In the case of the symmetrical solution, when the both symmetry conditions (15) have to be satisfied and the additional condition $v_1(0) = W > 0$, one can show that

$$C = 0, \quad v_2(y) \sim -W^2 y, \quad y \rightarrow 0. \quad (36)$$

Note here that the value $W = W(V, h)$ should be found from the constructed solution and is not an additional (arbitrary or given) constant.

Finally, both restrictions (32) and (33) should be valid for the symmetrical interface:

$$\int_0^h v_2(\xi) d\xi \geq -1/2, \quad \int_0^h F(\xi) d\xi \leq \pi/2. \quad (37)$$

Note that the tangential angle θ for this solution is a monotonically decreasing function in the interval $y \in (0, h)$ so that

$$\theta(y) \in (\theta_{min}, \pi/2), \quad \theta_{min} = \pi/2 - \int_0^h F(\xi) d\xi. \quad (38)$$

It is straightforward to see that Mullins' solution (20) satisfies all these relationships with $C = 0$, $\underline{\theta} = \pi/2$ and $W = V$, as expected. In the Appendix, we construct analytical examples of the symmetrical instantaneous solutions which are different from the Mullins one. Those solutions are not generally speaking steady-state ones. This means that they can be reached for some time step t , given the interface boundary velocity $V(t)$ and the position of the ends $h(t)$, but all these parameters may later change with time. What is extremely interesting about these solutions is that some of them are well-defined for any velocity $V > 0$ and arbitrary position of the boundary $y = h$. This shows a "rich behaviour" of possible instantaneous solutions. It is also clear that there is an infinite number of admissible instantaneous solutions. Some of them can be realised during some specific non steady-state interface motion. For example, *any* instantaneous solution obtained during a numerical computation, for any particular time step, boundary velocity and topology, has to satisfy all the relations (25) - (35). This will allow us to use the relations as indicators of the accuracy of computations. Moreover, they could provide a means to improve the accuracy of the algorithms.

Note also that a family of arbitrary (asymmetric) instantaneous solutions constitutes an even larger set in comparison to the symmetric case. In fact, the family of symmetrical solutions has one degree of freedom (since the constant C is equal to zero in this case) and correspondingly one less boundary condition (compare (35) and (15)). Moreover, in the context of further applications of this result to a given algorithm, where the angle-type boundary condition has to be preserved at the interface intersection point, it is worth mentioning that the boundary condition (35) is therefore more important for application than the symmetry condition. On the other hand, symmetrical instantaneous solutions can also be considered a subset of the asymmetric solutions if one considers the interval (h_0, h) instead of $(0, h)$; ($0 < h_0 < h$).

3 Numerical Simulations

To indicate the computational inaccuracy, we discuss Mullins' solution for a symmetrical reaper, for which all quantities are known in closed form (see (14) and remarks thereafter), and compare it with the result of a numerical computation using a simple algorithm (implemented in the Surface Evolver). This takes the form of a single film separating two bubbles of equal pressure being pulled at a velocity V at each boundary (fig. 3).

The numerical procedure can be briefly described as follows. We start from a straight (vertical) line joining the two walls a distance $2h = 2$ apart. This is subdivided into 2^5 short elements ("edges" which meet at points). A time step $\Delta t = 1 \times 10^{-5}$ is chosen for the computations and bounds on the possible length L of each edge ($0.01 \leq L \leq 0.05$). The algorithm proceeds as follows: (i) each boundary point is moved in the x direction a distance $V\Delta t$; (ii) the curvature at each point that is not on the boundary: $\kappa = \mathbf{F} \cdot \mathbf{n} / \bar{L}$ where \bar{L} is the average length of the neighbouring edges and \mathbf{F} is the negative energy (perimeter) gradient; (iii) each point that is not on the boundary is then moved according to $\Delta \mathbf{r} = \Delta t \kappa \mathbf{n}$ (fig. 3 b)). This procedure (one "step") is repeated

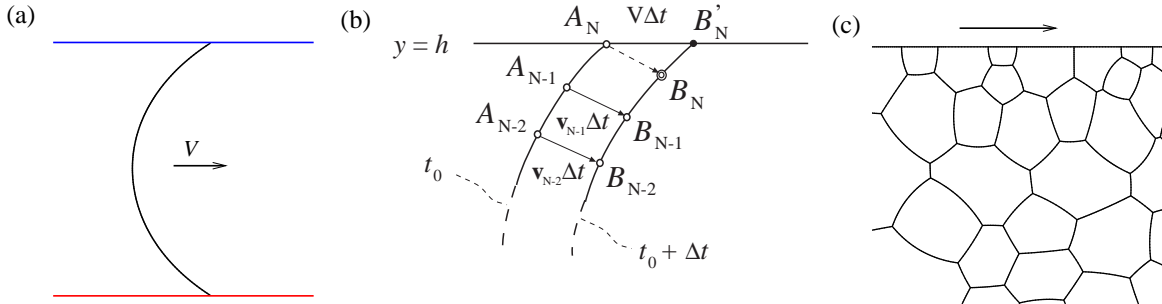


Figure 3: (a) The test problem considered here consists of a single film that is sheared symmetrically by translating the boundaries. The shape should correspond to Mullins' solution. (b) Standard algorithm for implementation of the boundary condition at each time step Δt . A_j and B_j are the respective tessellation points on the film at times t_0 and $t_0 + \Delta t$. (c) Example of a multi-bubble foam simulation, for which the numerical procedure developed here will be of use.

until the largest change in x position is less than a critical value (1×10^{-8}). Every 20 steps we check the edge length bounds and add or remove edges as necessary. Note that this choice of the parameters for numerical simulation is standard and allows us to obtain acceptable accuracy in reasonable computational time [7]. On the other hand, when one computes the dynamics of foams with many bubbles, the total computational error accumulates. Therefore information about the error is crucial, since it gives us a lower bound for the total computational error.

Two important observations illustrating the weakness of the algorithm should be noted here:

- The density of the tessellation points near the symmetry axis ($y = 0$) increases with each time step (fig. 3 b). Since the time step is constant, this may lead to failure of the stability condition for the linearized finite difference (FD) scheme applied to the nonlinear parabolic equation (2) due to this algorithm.
- The opposite effect occurs near the external boundary $y = h$. However, the situation here is even much worse. In fact, there is not enough information to reconstruct the curvature and the unit vector at a point A_N lying on the boundary and the algorithm, in fact, simply eliminates it. It creates instead the point B'_N along the boundary which should be the next point B_{N+1} (fig. 3 b)).

First we should underline here that the computations were stable (the stable steady-state regime has been reached) for every value of the external velocity V under consideration. Note that in our computations at high V , the number of tessellation points has increased from 2^5 to about 220 at steady-state. As expected, the worst situation (in the sense of computational time) occurs for the largest value of the velocity, $V = 1.560796$, which is slightly less than the critical value $V_{cr} = \pi/(2h)$ predicted by the analytical solution [1]. In numbers, it takes about 1 hour to complete the computation in the case $V \leq 0.5$, rising to over 48 hours for $V = 1.560796$ (SGI Altix Itanium 2, 1.5GHz). The algorithm could not reach the steady state regime at all for $V > 1.560796$, and lost physical meaning since the interface had a branch lying outside the external boundary $y = h$. All this illustrates that the existing algorithm is well organised and works according to expectations but it is naturally sensitive to the value of the boundary velocity V . Thus it makes sense to ask about algorithm accuracy for a specific velocity versus space and time steps.

As the exact analytical solution to the Mullins problem is known, we can estimate errors in the computations for all the physical and geometrical quantities: position of the interface $x(y)$, curvature $\kappa(y)$ and the velocity components $v_1(y), v_2(y)$. Corresponding relative errors for all solution parameters are presented in fig. 4 for different applied velocities.

As expected, the least accurate solutions are those for the largest velocity $V = 1.560796$. The error can be as large as 90% near the boundary ($y = h$) in the x component of velocity v_1 and 60% near the symmetry axis for the y component. These errors are a consequence of the phenomena discussed in the two observations above. As V decreases, the accuracy increases for given bounds on the edge lengths L .

At first glance, it would appear that the position of the interface, $x(y)$, should be computed with better accuracy than all other solution parameters, which are, in fact, results of some derivative procedure. However, our computations show that this is not in the case and the relative error for $x(y)$ varies from 6% to 28% for the

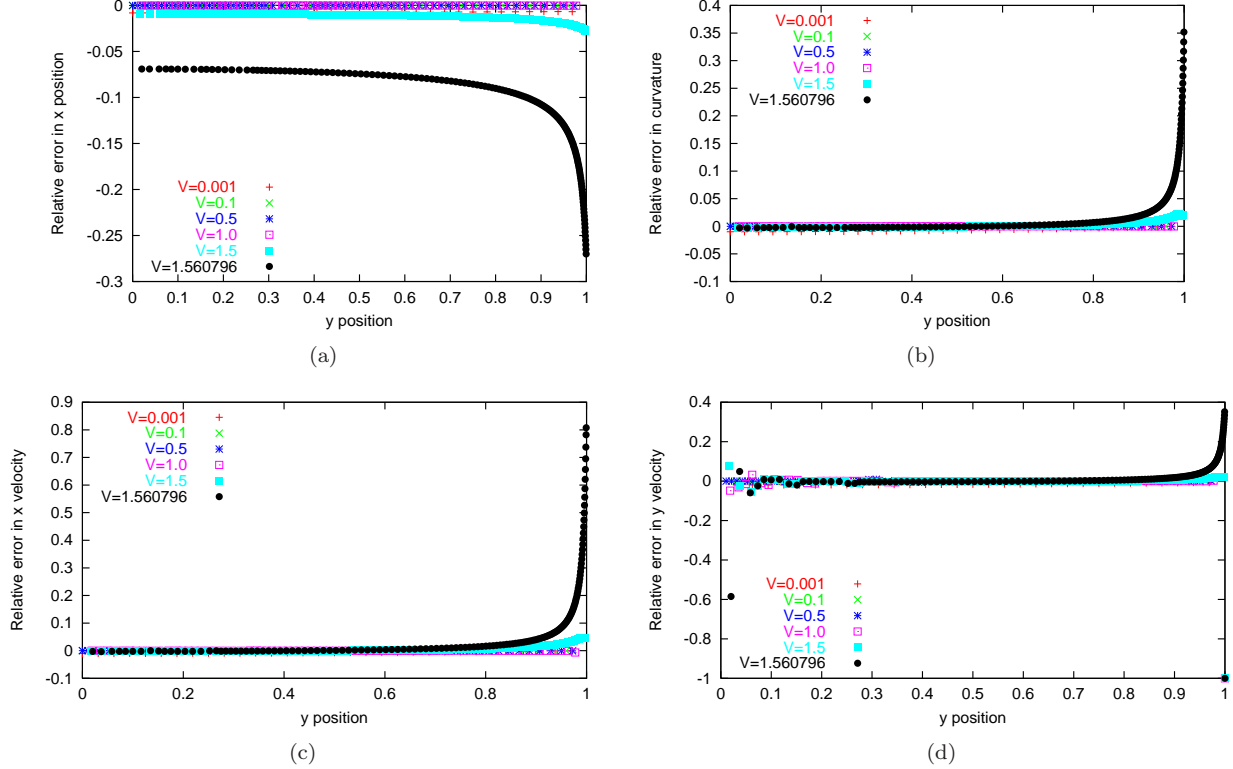


Figure 4: Relative error in (a) the position of the interface $x(y)$, (b) its curvature $\kappa(y)$, and the velocity components (c) $v_1(y)$ and (d) $v_2(y)$, for different velocities V of the external boundary.

velocity $V = 1.560796$ while the curvature error is lower. Note also that the error for smaller velocities reaches a few percent near the boundary or symmetry axis.

The maximal absolute errors for all solution parameters mostly appear near the external boundary $y = h(= 1)$. This highlights that the implementation of the boundary condition in the existing algorithm cannot be considered as sufficient and should be improved.

Moreover, in the case of Mullins' solution an additional simple local indicator defined by identity (13) (independent of the integration of the solution variables) could equally be considered. It is clear from the results presented in fig. 5(a), that the error in this condition is not localized near the ends of the interface, as one might expect from the above. Moreover, this “internal error” is present for all values of V and is comparable with that near the interface ends.

To investigate accurately this “internal” error we repeated the computations for a specific velocity $V = 1$ and decreased both the time increment, Δt , and the minimum edge length, L_{\min} (fig. 5(b)). This has improved the quality of the computations, but there is still an error at some internal points of the interval that is comparable with the error near the ends. Unfortunately, it considerably increases the computational time by several hours. The obvious route of decreasing Δt but fixing L_{\min} , to ameliorate this effect, leads to greater error in the solution (fig. 5(b)). The accuracy of the solution can be improved internal by making the line segments of equal length (fig. 5(b)), but this doesn't prevent an error at the boundary.

Possible sources of the “internal” error include: a) non-optimal distribution of the tessellation points along the interface after some time; b) imperfections in the correction procedure (which adds and eliminates points from the interface at some prescribed time); c) point-to-point error variation related to the fact that the “diffusion”-type coefficient changes from point to point along the interface (recall that equation (2) is a nonlinear parabolic equation which is solved by a direct FD scheme with a fixed time step).

In the last computation in figure 5(b), for the line segments of equal length, we have redistributed points to make the segments equal at every time step. Apart from the fact that the number of tessellation points is smaller than for the standard algorithm, such a comparison is not absolutely fair as the redistribution in the

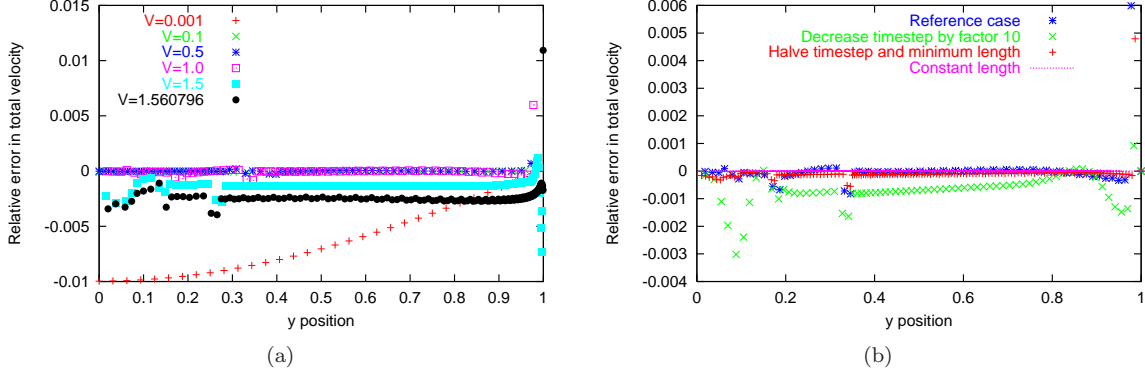


Figure 5: Relative error for the characteristic relation (13) for (a) varying velocity V , and (b) varying numerical parameters ($\Delta t, L$). The best accuracy is obtained by keeping the line segments of equal length.

standard algorithm was done every 20 time steps. To discover if there is an effect of redistribution frequency on the accuracy, we have tested these two algorithms under the same strategy by redistributing the points (in a different way) every time step and every 20th time step. The error in the function F defined in (25), which is a constant in the case of Mullins solution, are presented in 6(a). It is evident that the standard algorithm is quite sensitive to the chosen strategy. For a given position of the points on the interface, the relative error may differ by as much as two orders of magnitude, whereas this is not the case for the equal segment strategy. In this case, only near the external boundary is there some small fluctuation in the accuracy. Comparing the two redistribution algorithms for the same frequency of redistribution, the largest error always appears in the case of the standard algorithm – by up to two orders of magnitude – despite the fact that the number of tessellation points was greater. Moreover, in the standard algorithm this error is irregularly distributed along the interface. Recall that the number of tessellation points in the standard algorithm changes during the computations from 2^5 initially to around 150 (for $V = 1$) in the steady-state regime, while the number of the points in the second (equal length) algorithm remains constant. Therefore the computational time for the second algorithm was less by a factor of approximately two. On the other hand, the difference in the computational time between the different frequencies for redistribution for the equal length algorithm was only a few percent. This indicates the further possibility to optimise this algorithm by redistributing points every M time steps. It is clear that $M = M(V)$ and this needs further investigation.

Note that the function F from (25) can be used as an indicator of the accuracy of the computation only for the Mullins solution. However, there are three universal indicators which can be helpful to estimate the accuracy for any computations. Namely, the relative errors of the numerical representations of the identities (26), (28) and (31). The respective results are shown in 6(b)-(d).

As in the case of the specific indicator discussed above (Fig. 6(a)), all general indicators presented in Fig. 6 (b) – (d) indicate that the equi-distant distribution of the tessellation points is much better than the standard algorithm, regardless of the chosen strategy. Moreover, even near the symmetry point, $y = 0$, where the value of the indicators all tend to zero and have a large influence on the relative errors, the accuracy of the computations for the second algorithm is extremely high. This is not the case for the standard algorithm.

In Fig. 7, the relative errors of the solution variables are presented for both algorithms: the standard one and the equi-distant distribution. The result shown in Fig. 7(a) looks surprisingly at first glance: although the accuracy of the computations performed with these two algorithms is of the same order and the error related to the new algorithm is distributed more uniformly, it appears that the accuracy of the standard algorithm is better than the equal-segment-length algorithm, at least with respect to the accuracy of the position of the interface. However, this not in the case. In fact, as was shown above, the computational error for the standard algorithm is redistributed along the interface irregularly whereas that for the equal-segment algorithm is practically uniform. As a result, the criterion to stop the iteration process to find the steady state solution works differently for the two algorithms. The prescribed maximal step-to-step incremental growth 10^{-8} is reached more quickly for the new algorithm. This is the second reason (together with number of tessellation points) why this algorithm is faster. If one were to run both algorithms for the same time, or for the same number of iterations, and compare

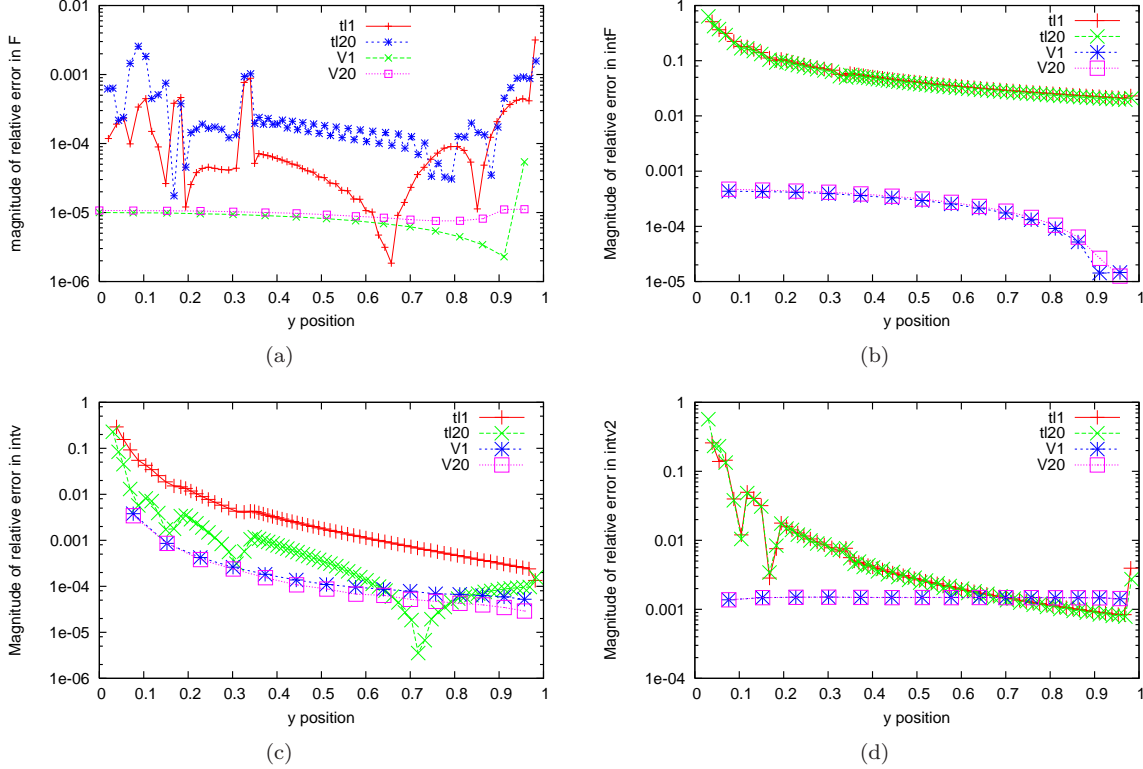


Figure 6: Relative errors in the computations shown by the integral measures for unit velocity of the external boundary, $V = 1.0$, obtained with four different computational strategies: the standard redistribution of the tessellation points, $tl1$ and $tl20$; uniform length segment redistribution, $V1$ and $V20$, at every time step and every 20th step, respectively. (a) the function F defined in (25) (as in Fig. 5); (b,c,d) the three general internal measures; for the notation used in the graphs see Appendix A.4. All integrals were computed with the trapeze rule.

the corresponding results, the discussed paradox should not appear and the new algorithm always provides better accuracy by as much as two orders of magnitude.

However, for the accuracy of other problem variables, the film curvature, κ , and the interface velocities, v_1 and v_2 , cf. in Fig. 7 (b)-(d), the new algorithm is more accurate, notwithstanding the above argument.

Finally, let us stress again that the proposed three indicators are better measures than a comparison of numerical steady state solution with the analytical one, since the latter comparison includes an additional error related to the determination of the steady state regime, while the indicators show us accuracy of the solution at any time step.

4 Discussion and Conclusions

All these results clearly indicate that the existing algorithm should be used with caution, especially when investigating foam behaviour near the critical velocity. Moreover, when there are many bubbles in a simulation (fig. 3(c)), the user is restricted to some critical number of tessellation points M , which gives a limitation on accuracy even for low velocity. In fact, the foam structure is highly non-uniform, in the sense that the bubbles may have different sizes. Effectively this means that every interface has its own critical velocity and the bigger bubbles thus introduce larger errors. This creates the following duality: to accurately describe the process of bubble motion (for example a neighbour switching event [2]) it is necessary to have the computational error as small as possible, while the error generated near the critical velocity takes its greatest value. This is true for boundary or internal bubbles equally. Problems requiring high accuracy of the solution near the external boundary are related to the investigation of the boundary effects describing the total phenomenological

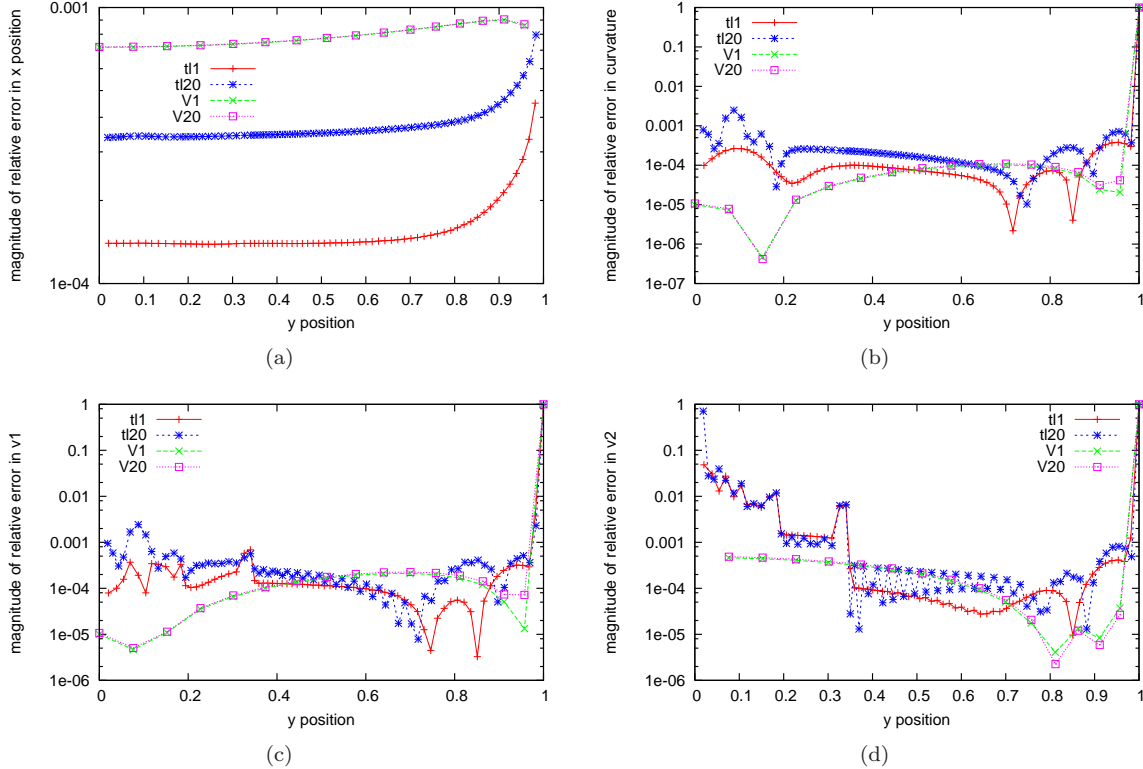


Figure 7: Relative error in (a) the position of the interface $x(y)$, (b) the curvature $\kappa(y)$, and (c) and (d) the velocity components $v_1(y)$ and $v_2(y)$, for velocity of the external boundary $V = 1$ and different numerical algorithms.

behaviour of the foam structures.

To revise and improve the existing numerical algorithm, we propose to use another strategy for the redistribution of the tessellation points: an equal-segment-length distribution of the tessellation points is much more favourable.

Additionally, taking advantage of the auxiliary identities (26), (28) and (31), we may correct the instantaneous solution obtained within any algorithm at any or even every time step without time-consuming computations, as the identities are valid for any instantaneous solution. They also make possible further investigation of the asymptotic behaviour of the bounded interface solution near the ends. For example, any possible solution behaves at the symmetry axis according to (36), which allows us to tackle the error in the solution near the symmetry axis. On the other hand, the results obtained in section 2.3 may allow us to construct and implement a new numerical procedure/elements tackling the boundary condition in a more accurate way (without losing any near-boundary points).

Finally, as we have shown, the identities (26), (28) and (31) may be used to probe the accuracy of computations. These indicators are extremely helpful as they are not based on information about the exact solution and can therefore illuminate inaccuracy of the numerical solution without any preliminary knowledge about the exact solution itself.

Summarizing, we have shown that an improvement of the numerical algorithm is highly desirable and possible. Apart from the fact that some of the improvements have been indicated and proven in this paper, there is still an open question how to deal with accuracy of the computations near the critical velocities and near the external boundaries. We have also detected possible directions for future investigation: improved implementation of the boundary condition, creation of additional near-boundary points. Further, to check new results related to the numerical algorithm we need a larger set of analytical benchmarks.

References

- [1] W.W. Mullins, Two-Dimensional Motion of Idealized Grain Boundaries. *J. Appl. Phys.*, 1956, **27**:900–904.
- [2] D. Weaire and S. Hutzler, *The Physics of Foams*. Clarendon Press, Oxford, 1999.
- [3] K. Brakke, The Surface Evolver. *Exp. Math.*, 1992, **1**:141–165.
- [4] N. Kern, D. Weaire, A. Martin, S. Hutzler, and S.J. Cox, Two-dimensional viscous froth model for foam dynamics, *Phys. Rev. E*, 2004, **70**:041411.
- [5] D. Weaire and S. McMurry, Some Fundamentals of Grain Growth. *Solid State Physics*, 1996, **50**:1–36.
- [6] A. Peleg, B. Meerson, A. Vilenkin Area-preserving dynamics of a long slender finger by curvature:A test case for globally conserved phase ordering, *Phys. Rev. E*, 2001, **63**:066101.
- [7] S.J. Cox, A Viscous Froth Model for Dry Foams in the Surface Evolver, *Coll. Surf. A*, 2005, **263**:81–89.

A Appendix: A family of symmetrical instantaneous solutions

In this section we present analytical representations of some instantaneous symmetrical solutions for the interface satisfying the same boundary (23) and symmetry (15) conditions as Mullins’ solution.

A.1 First example

Let us consider the following simple combination of compatible velocities

$$v_2(y) = -W^2y, \quad v_1(y) = W\sqrt{1 - W^2y^2}, \quad W = \frac{V}{\sqrt{1 + V^2h^2}}, \quad (39)$$

which satisfy (31) with $C = 0$ and, as a result, can be used to construct a symmetrical instantaneous solution. Here W is the same constant as in (36). Natural restrictions (37) for the existence of such solution give the same estimate:

$$W < \frac{1}{h}, \quad \text{or} \quad \frac{V}{\sqrt{1 + V^2h^2}} < \frac{1}{h}, \quad (40)$$

which, remarkably, holds true for any values of V and h . The shape of the interface is described by equation (28)

$$x = x_0(0, t) - \sqrt{1 - W^2y^2}. \quad (41)$$

The tangential angle θ for this solution is a monotonically decreasing function in the interval $y \in (0, h)$ and

$$\theta(y) \in (\theta_{min}, \pi/2), \quad \theta_{min} = \pi/2 - \arcsin(Wh) > 0. \quad (42)$$

A.2 Second example

Let us now consider another specific instantaneous solution assuming that $v_1 = W < V$. Then the second component of the velocity satisfies the equation

$$\frac{W^2}{v_2^2(y)} = -\frac{1}{2 \int_0^y v_2(\xi) d\xi} - 1. \quad (43)$$

To find $v_2(y)$ it is more convenient to return to the differential equation (30) rather than working with the nonlinear integral equation (43). After integration it takes the form

$$\Phi\left(\frac{v_2}{W}\right) = -Wy, \quad (44)$$

where the odd function Φ is defined as

$$\Phi(\xi) = \frac{1}{2} \left(\arctan \xi + \frac{\xi}{1 + \xi^2} \right), \quad \left(\Phi'(\xi) = \frac{1}{(1 + \xi^2)^2} \right). \quad (45)$$

Note that $\Phi : \mathbb{R}_+ \rightarrow [0, \pi/2)$ is a monotonic function. Moreover, one can easily obtain the constraint $Wh < \pi/4$, which is similar to (37) and (40). Then the required velocity component v_2 can be found from:

$$v_2 = -W\Phi^{-1}(Wy), \quad (46)$$

and we can finally find the complete solution using (27):

$$x = x_0(0, t) + \frac{1}{W} \int_0^{Wy} \Phi^{-1}(\xi) d\xi. \quad (47)$$

Finally, note that the tangential angle θ for this solution is a monotonically decreasing function in the interval $y \in (0, h)$

$$\theta(y) \in (\theta_{min}, \pi/2), \quad \theta_{min} = \pi/2 - \int_0^{Wh} \left(1 + \left(\Phi^{-1}(\xi)\right)^2\right) d\xi. \quad (48)$$

It remains only to find possible values of the unknown constant W in order to satisfy the boundary condition (23). The relevant equation takes the form

$$\Phi^{-1}(Wh) = \sqrt{\frac{Vh}{Wh} - 1}. \quad (49)$$

This equation has the unique solution $W = W_*(V, h) < V$. In fact, the left hand side is an increasing function from zero to infinity as $Wh \rightarrow \pi/4$, whereas the right-hand side is a decreasing function taking values between ∞ when $Wh \rightarrow 0$ and 0 when $Wh \rightarrow Vh$. Additionally one can conclude from this that $Wh < \min\{Vh, \pi/4\}$, so the restriction defined after (45) always holds. In other words, this solution, as well as that of the first example, is well-defined for arbitrary velocity V and position of the boundary $y = h$. We can also show that θ_{min} is always positive:

$$\theta_{min} > \pi/2 - \int_0^{\pi/4} \left(1 + \left(\Phi^{-1}(\xi)\right)^2\right) d\xi = \pi/2 - \int_0^{\infty} (1 + \eta^2) \Phi'(\eta) d\eta = \pi/2 - \int_0^{\infty} \frac{d\eta}{1 + \eta^2} = 0. \quad (50)$$

It is interesting to note that in the case $V \ll 1$ both of the instantaneous solutions constructed above coincide with Mullins' solution to within an accuracy of $O(V^2)$ for any fixed value of h .

A.3 General case

The second example above indicates how to build a wider class of symmetrical instantaneous solutions. Let us introduce the following set $\mathfrak{A} \subset C^2([-a, a])$, ($a > 0$) of even functions, $\psi(\xi) = \psi(-\xi)$, satisfying the following four conditions:

$$\psi(\xi) = \frac{1}{2}\xi^2 + O(\xi^4), \quad \xi \rightarrow 0; \quad \psi(a) \leq \frac{1}{2}; \quad \psi' > 0; \quad \left(\frac{\psi'}{\sqrt{\psi(1-2\psi)}}\right)' \geq 0, \quad \xi \in (0, a). \quad (51)$$

Note that a may differ from function to function, but it is necessary that for every function there exists some $a > 0$ for which all conditions (51) hold. For some functions it may happen that $a = \infty$.

For example, the following three functions belong to the set \mathfrak{A} :

$$\psi_1(\xi) = \frac{1}{2}\sin^2 \xi, \quad \psi_2(\xi) = \frac{1}{2}\xi^2, \quad \psi_3(\xi) = \int_0^{\xi} \Phi^{-1}(\zeta) d\zeta = \frac{1}{2} \left(1 - \frac{1}{1 + [\Phi^{-1}(\xi)]^2}\right), \quad (52)$$

with $a_1 = \pi/2$, $a_2 = 1$ and $a_3 = \infty$ respectively. These three functions have been collected from Mullins' solution and two previous examples. Thus, the set \mathfrak{A} is not empty.

Using any function from this set we can construct a symmetrical instantaneous solution with velocity components in the form:

$$v_2(y) = -W\psi'(Wy), \quad v_1(y) = W\psi'(Wy) \sqrt{\frac{1 - 2\psi(Wy)}{2\psi(Wy)}}, \quad (53)$$

that identically satisfies equation (31) with $C = 0$. Then the unknown constant W should be taken to be of the form $W = W_*(Vh)/h$ where $W_*(b) > 0$ is a solution of the implicit equation:

$$\frac{\psi'(W_*)}{\sqrt{2\psi(W_*)(1-2\psi(W_*))}} = \frac{b}{W_*}. \quad (54)$$

Because of the last condition in (51), there may exist only one solution of this equation. If, in addition, the left-hand side of (51) tends to infinity as $W_* \rightarrow a$, then the solution always exists and $W_* < a$. However, if the left-hand side of (51) tends to a finite value $L > 0$ as $W_* \rightarrow a$, then the solution exist only under additional condition

$$Vh < La. \quad (55)$$

One can check that for Mullins' solution $L = 1$ and (55) coincides with (18). For the other two cases previously discussed above, we have $L = \infty$ so the solution of the implicit equation (54) always exists and no solvability condition (55) is needed in these cases.

To reconstruct the complete symmetrical instantaneous solution based on (53) it is enough to substitute it in (25), (26) and (28).

Note that in the case $V \ll 1$, the solution to (54) gives $W_* \sim V$, as one can conclude from the first part of (51). This means that any instantaneous solution differs negligibly from the Mullins' solution for small values of the velocity V .

It remains to investigate two important constraints (37). Taking into account that

$$\int_0^h v_2(\xi)d\xi = -\psi(Wh), \quad \int_0^h F(\xi)d\xi = \frac{1}{2} \left(\arcsin(4\psi(Wh) - 1) + \frac{\pi}{2} \right),$$

then the two constraints (37) are equivalent in this case and correspond to $\psi(Wh) \leq 1/2$, which coincides with the second part of (51).

In fact, the third condition, $\psi' > 0$, from (55) was never used. We added it only to have a convex instantaneous solution; without it, we can construct non-convex interfaces.

A.4 Integral measures

Here we define explicitly the integral measures of problem quantities, used for assessing the accuracy of the computations, and their relative errors.

After (26):

$$\text{intF} = - \int_0^y F(\xi)d\xi; \quad \text{relative error} = - \left(\int_0^y F(\xi)d\xi \right) / \left(\theta(y) - \frac{1}{2}\pi \right) - 1. \quad (56)$$

After (28):

$$\text{intv} = \int_0^y \frac{v_2(\xi)}{v_1(\xi)}d\xi; \quad \text{relative error} = \left(\int_0^y \frac{v_2(\xi)}{v_1(\xi)}d\xi \right) / (x(0) - x(y)) - 1. \quad (57)$$

After (31):

$$\text{intv2} = \int_0^y v_2(\xi)d\xi; \quad \text{relative error} = \left(\int_0^y v_2(\xi)d\xi \right) / \left(\frac{v_2^2(y)}{2(v_2^2(y) - v_1^2(y))} \right) - 1. \quad (58)$$