LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-643813

A New Level-Set Model for the Representation of Non-Smooth Geometries

D. P. Starinshak, S. Karni, P. L. Roe

September 12, 2013

Journal of Scientific Computing

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

A New Level-Set Model for the Representation of Non-Smooth Geometries *

David P. Starinshak[a], Smadar Karni[b], and Philip L. Roe[c]

[a] *Lawrence Livermore National Laboratory*
[b] *University of Michigan, Department of Mathematics*
[c] *University of Michigan, Department of Aerospace Engineering*

Abstract

In [2], we proposed a new level-set model for representing multimaterial flows in multiple space dimensions. Rather than associating each level-set function with the boundary of a material, the new model associates each level-set function with a pair of materials and the interface that separates them. In this paper, we extend the model to represent geometries with non-smooth boundaries. The model uses multiple level-set functions to describe the shape boundary, typically with one level-set function per smooth boundary segment. Sign information is collected from all level-set functions and a voting algorithm is used to determine the interior/exterior of the geometric shape. The model is well suited for representing boundaries with singularities; it offers significant improvement over standard level-set approaches, both in shape preservation and area conservation; and it eliminates the need for costly redistancing of the level-set function. Numerical examples illustrate the superior performance of the proposed model.

*This work was supported in part by DOE NNSA-ASC grant DE-FC52-08NA28616 and NSF Award # DMS 0609766

1 Introduction

Level-set functions are a convenient tool for modeling geometric regions, for example identifying regions occupied by different materials in multimaterial flows, representing the motion of solid geometrical objects in a prescribed flow field, or modeling deformable bodies subject to exterior forces. The traditional approach associates each level-set function with a region. The boundary of the region is embedded as a zero level-curve of a level-set function. The function takes on positive values in the interior of the region, and negative values in the exterior. In this approach, the interior of each region is characterized by exactly one of the level-set functions being positive and all others negative. An evolution equation governs the dynamics of the region, and the zero level-curve defines the boundary location at all times.

This approach works reasonably well for smooth geometries. However, for geometries with corners or other singularities, approximate numerical procedures introduce numerical diffusion that tends to round off sharp corners in the level-set functions. In multimaterial flows, this often creates ambiguities in material identification near material triple junctions. In flows around non-smooth geometries, rounding off sharp corners often introduces large errors in volume as well as inaccuracies in shape representation.

In order to counteract the effects of numerical diffusion, redistancing strategies have been proposed [4]. Their implementation is largely driven by empirical experience. If used too aggressively, redistancing has the tendency to spuriously displace the zero level curve; if use too sparingly, it permits excessive amounts of numerical diffusion. Optimizing the redistancing implementation is problem-dependent and oftentimes requires parameter tuning. In the new level-set model, the functions are initially smooth and do not require redistancing to restore this property.

In [2], we proposed a new level-set model for representing multimaterial flows, which associates each level-set function with a pair of materials and the interface that separates them. A voting algorithm collects sign information from all level-set functions and determines material designation. This new approach offers a flexible tool for representing complex configurations of materials. It is less prone to introducing ambiguities in material identification and renders corrective measures such as reinitialization unnecessary. Numerical results give ample evidence for the superior performance of this approach over the standard approach.

In this paper, we extend the new approach to the representation of non-smooth geometries. Instead of attempting to describe the geometrical object by a single level-set function whose zero level-curve coincides with its boundary, we use multiple level-set functions to describe the smooth boundary segments. As with multimaterial flows, a voting algorithm collects sign information from all level-set functions and determines the interior/exterior of the geometrical shape. A simple projection algorithm may also be used to construct a corresponding single level-set function from the multiple functions (see Appendix).

This approach is vastly superior to the standard approach in representing sharp corners and other boundary singularities, and leads to greatly improved shape and volume preservation. It also renders costly corrective measures, such as redistancing, unnecessary. In Section 2, we briefly discuss the new level-set model for multimaterial flows and describe the extension of this algorithm to representation of non-smooth geometries. We proceed by examples and illustrate, step by step, how the voting algorithm is implemented and how the exterior/interior of the geometry is identified. Section 3 outlines our

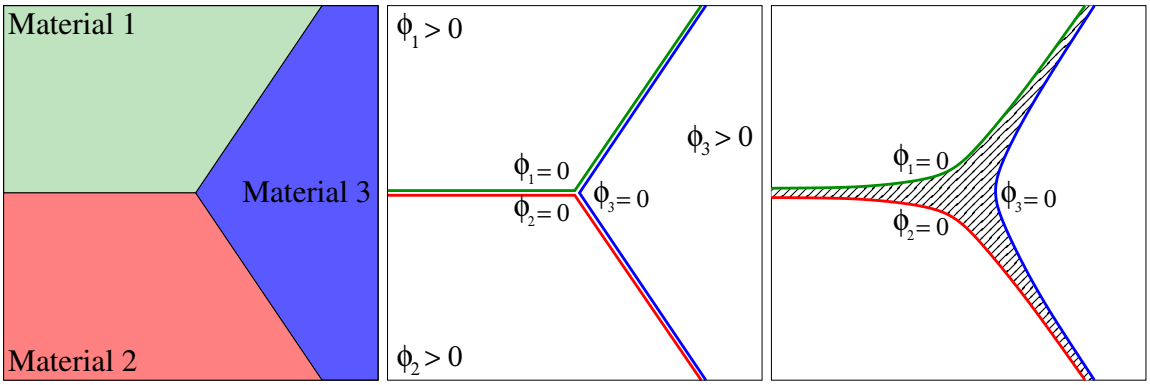

Figure 1: Y-junction schematic (left), zero curves of MLS functions (center), numerical diffusion creates an indeterminate vacuum state (right).

testing strategy and numerical discretization. Results are presented in Section 4.

2 New Level-Set Model

2.1 Representing Multiple Materials

The traditional way for modeling multimaterial flows by level-set functions is to associate material k with a level-set function, $\phi_k(x,t)$ so that

$$\phi_k(\mathbf{x},t) \begin{cases} > 0 & \text{in material } k \\ < 0 & \text{outside material } k \\ = 0 & \text{on boundary of material } k \end{cases} \tag{1}$$

In this model, material l is identified uniquely by the state

$$\begin{cases} \phi_k > 0 & , \quad l = k \\ \phi_k < 0 & , \quad l \neq k \end{cases} \tag{2}$$

While analytically this model represents a consistent picture, ambiguities may arise due to approximate numerical procedures, where regions may not be claimed by any material or are claimed by more than one material. A typical example involves material triple junction (Figure 1).

In [2], we proposed a new level-set model, where level-set functions are defined through material pairs. The level-set function $\phi_{m,n}$ corresponds to materials m and n and satisfies

$$(\mathbf{x},t) \quad \begin{cases} \in \text{material m} & \Longrightarrow \quad \phi_{m,n}(\mathbf{x},t) > 0 \\ \in \text{material n} & \Longrightarrow \quad \phi_{m,n}(\mathbf{x},t) < 0 \\ \text{along } (m,n) \text{ interface} & \Longrightarrow \quad \phi_{m,n}(\mathbf{x},t) = 0 \end{cases} \tag{3}$$

If $(\mathbf{x},t)$ is in a region not occupied by either material, the sign of $\phi_{m,n}(\mathbf{x},t)$ is not constrained and may take on an arbitrary value.

As noted in [2], $\phi_{m,n}(\mathbf{x},t) > 0$ does not imply that $(\mathbf{x},t)$ is occupied by material m, only that it *may be* occupied by material m. The sign of $\phi_{m,n}$ can be viewed as a "negative

vote": $\phi_{m,n}(\mathbf{x},t) > 0$ rules out the possibility that $(\mathbf{x},t)$ is occupied by material n. For the latter to happen, $\phi_{m,n}(\mathbf{x},t)$ would have to be negative. The entire (m,n)-interface at time t is contained in the zero curve $\{\mathbf{x}(t) : \phi_{m,n}(\mathbf{x},t) = 0\}$. The zero curve also contains points not associated with the interface.

To determine which material *actually* occupies $(\mathbf{x},t)$, sign information is collected from all level set function, and a pairwise voting system is used:

$$\phi_{m,n}(\mathbf{x},t) \begin{cases} > 0 & \Longrightarrow \text{ Material } m \text{ receives one vote, } \quad \text{Material } n \text{ is ruled out} \\ < 0 & \Longrightarrow \text{ Material } n \text{ receives one vote, } \quad \text{Material } m \text{ is ruled out} \end{cases} . \tag{4}$$

Materials are identified by observing that, for M materials and $M(M-1)/2$ functions initialized using (3), no material can receive more than $M-1$ votes and no two materials can simultaneously receive $M-1$ votes. Therefore, the material which occupies $(\mathbf{x},t)$ receives exactly $M-1$ votes.

The new level set model is naturally suited to handle interface singularities, such as sharp corners. It exhibits improved accuracy at material triple junctions without the need for expensive reinitialization algorithms. While in theory, the pairwise identification of interfaces requires $\mathcal{O}(M^2)$ level-set functions to represent M materials, in practice far fewer are actually needed. Savings are made by eliminating material pairs that do not share an interface and by combining multiple level-set functions that coincide with each other into a single level-set function.

2.2 Representing Geometrical Figures

In this section, we extend the new level-set model to track the location of moving objects. The new model can accommodate geometries with boundary singularities, for example corners, cusps, or changes in convexity. Such singularities are known to contribute to poor shape preservation and poor volume conservation in standard level-set models. The new model uses multiple smooth level-set functions as described below. Each level-set function is assumed to obey an advection equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u}(\mathbf{x},t) \cdot \nabla \phi = 0. \tag{5}$$

where $\mathbf{u}(\mathbf{x},t)$ is a prescribed velocity field. In examples, we consider $\mathbf{u}(\mathbf{x},t)$ corresponding to solid body rotation as well as deformations that are arbitrarily large in size.

We proceed by examples of increasing complexity. For each example, we describe how to set up multiple (smooth) level set functions, collect sign information from all functions to construct a voting table, and use the table to identify the interior and exterior regions of the given geometry.

Alternatively, the new model may be formulated in terms of geometric intersections and unions. The standard single-function model may be recovered from the set of multiple smooth functions using a simple projection algorithm. A detailed explanation of the projection algorithm, including examples, is provided in the Appendix.

Triangle

We first consider the simple example of a triangle. We begin by partitioning the domain into four regions, as illustrated in Figure 2. This partition is not unique, and many options are available. We observe that region 4 coincides with the interior of the triangle and regions 1, 2, and 3 with the exterior. Corresponding to each pair of regions, we define six level-set functions, $\phi_{m,n}$, where for systematic counting we always use $m < n$.

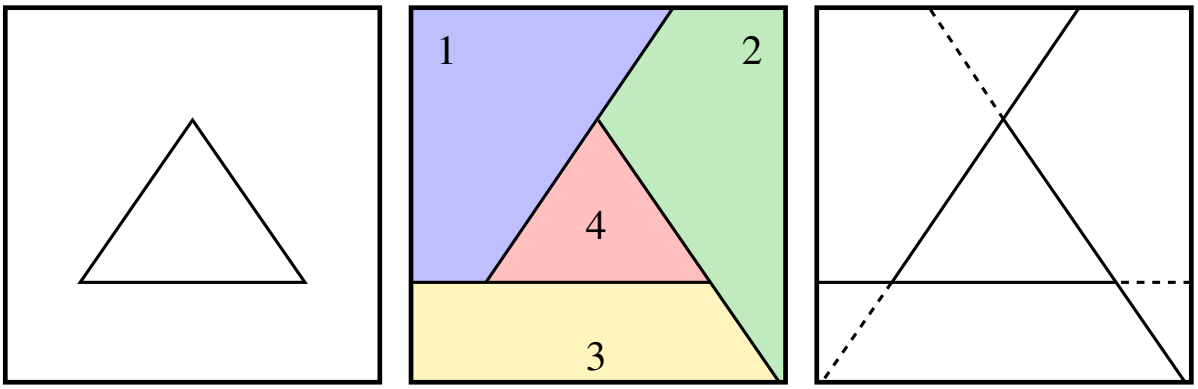

Figure 2: Domain partitioning for a triangle

We initialize $\phi_{m,n}$ in agreement with (3): it is positive in Region m, negative in Region n, and is zero on the interface between them. In regions where the sign of $\phi_{m,n}$ is not constrained, we extend its zero curve so that the function is as smooth as possible. We illustrate one possible choice for the six functions in Figure 3. Here, and in all subsequent schematics, the zero level-curve is plotted as a solid line where it represents an actual interface between regions, and as a dashed line where it does not correspond to an actual interface.

The sign information of each $\phi_{m,n}$, together with (4), enables us to establish whether a given point is interior or exterior to the geometrical shape. The counting argument is similar to the multimaterial setup: for a domain partitioned into M regions, the "winning" region is the only region to receive exactly $M-1$ votes. In the present example, $M = 4$; each region receives at most 3 votes, and only one region receives exactly 3 votes.

The algorithm is complete at this point: all regions are identified by the voting system, and each region corresponds to either the interior or exterior of the geometry. However, the algorithm is redundant; six level-set functions are used to describe the geometry, but three functions are sufficient (see Figure 3). We next describe how functions can be grouped together to eliminate redundancies.

We first observe that $\phi_{1,2}$ and $\phi_{1,4}$ may be initialized so they are identical. Since they are evolved by the same rule, they will remain identical for all time. We may combine them into a single function, ϕ_1, and amend the voting system to reflect the fact that ϕ_1 has the "voting power" of the two functions $\phi_{1,2}$ and $\phi_{1,4}$:

$$\phi_1(\mathbf{x}, t) \begin{cases} > 0 & \Longrightarrow \quad \text{Region 1 gets 2 votes} \\ < 0 & \Longrightarrow \quad \text{Regions 2 and 4 get one vote each} \end{cases} . \tag{6}$$

Similarly, $\phi_{2,3}$ and $\phi_{2,4}$ may be combined into a single function, ϕ_2, and $\phi_{3,4}$ and -$\phi_{1,3}$ combined into ϕ_3. The amended voting system for all functions is summarized in Table 1.

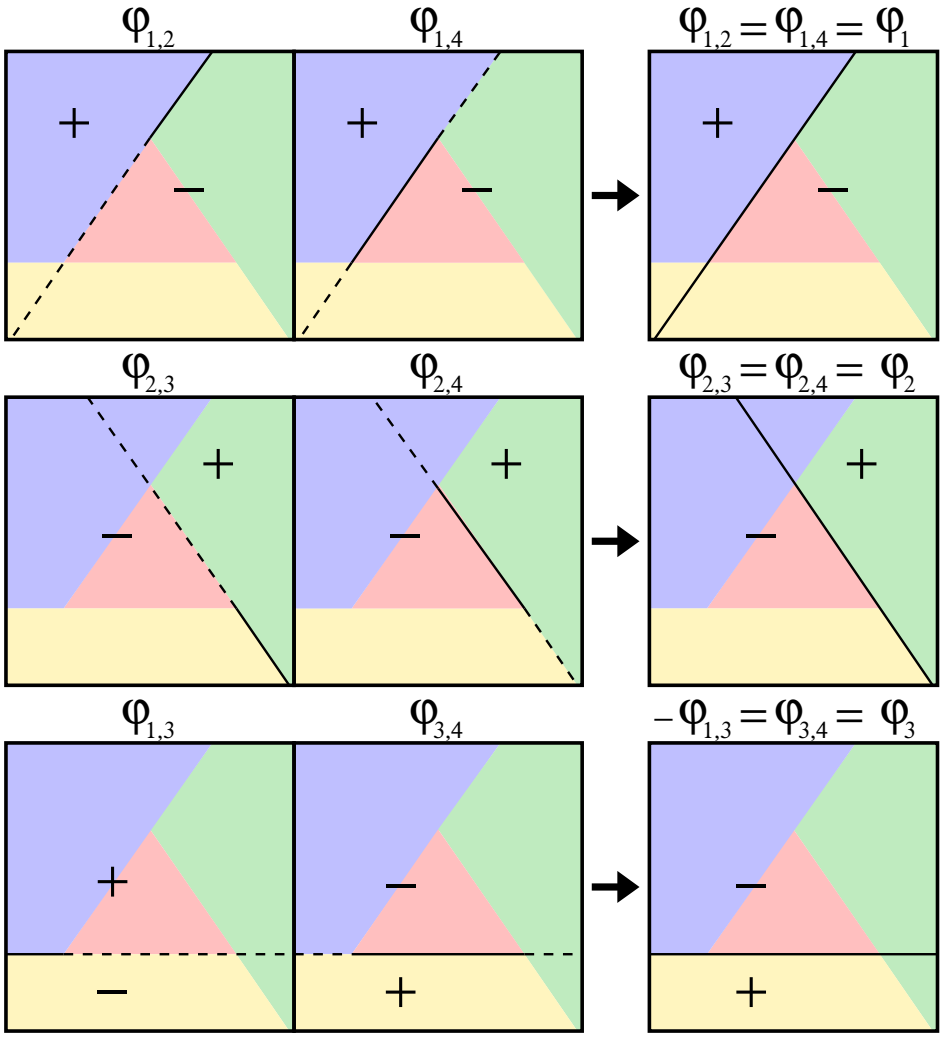

Figure 3: Level-set initialization for a triangle

Function	Sign	Votes for Region			
		1	2	3	4
ϕ_1	$+$	2	-	-	-
	$-$	-	1	-	1
ϕ_2	$+$	-	2	-	-
	$-$	-	-	1	1
ϕ_3	$+$	-	-	2	-
	$-$	1	-	-	1

Table 1: Voting system for 3 ILS functions to represent a triangle.

To determine if $(\mathbf{x}, t)$ is an interior or exterior point to the shape, we appeal to Table 1 and tally the total votes that each region receives. Denote by S_i the sum of all the votes that region i receives and recall that, for this example, the "winning" region is the one and only region that receives 3 votes. It then follows that $S_i - 3$ is zero for some

region i and negative for all other i. In particular, at point $(\mathbf{x}, t)$

(i) If $S_4 - 3 = 0$, $(\mathbf{x}, t)$ is in region 4, and is an interior point.

(ii) If $(S_1 - 3)(S_2 - 3)(S_3 - 3) = 0$, $(\mathbf{x}, t)$ is in one of regions 1,2, or 3, and is an exterior point.

More generally, for a domain partitioned into M regions, denote by I_{in} the set of region indices corresponding to interior regions, and by I_{out} the set of indices corresponding to exterior regions. For this example, $I_{in} = \{4\}$ and $I_{out} = \{1, 2, 3\}$. Then one of the following mutually exclusive conditions holds:

$$\begin{cases} \prod_{i \in I_{in}} \Big(S_i - (M-1)\Big) = 0 \quad \Longrightarrow \quad (\mathbf{x}, t) \text{ is an interior point.} \\ \\ \prod_{i \in I_{out}} \Big(S_i - (M-1)\Big) = 0 \quad \Longrightarrow \quad (\mathbf{x}, t) \text{ is an exterior point.} \end{cases} \tag{7}$$

Zalesak Disk

This has been a popular example in the literature [1, 3, 6]. We construct the voting table similar to the triangle example. Table 2 gives the final tally in concise form and is the only table that is needed in order to implement the algorithm.

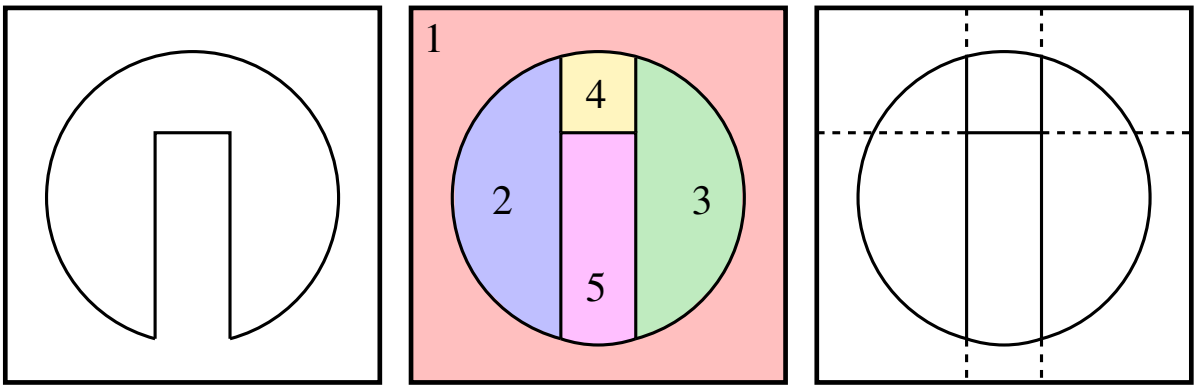

Figure 4: Domain partitioning for Zalesak disk

We begin by partitioning the domain into five regions, as illustrated in Figure 4. Regions 2, 3, and 4 are interior, and Regions 1 and 5 are exterior. Other ways to form a partition are possible.

As before, we define level-set functions corresponding to all possible regions pairs, in this case ten functions. The functions are initialized according to (3), with voting system given by (4) to identify regions. Figure 5 illustrates all level set functions $\phi_{m,n}$, $1 \leq m < n \leq 5$.

We next collapse the 10 functions to only 4 by identifying equivalencies (see Figure 5). For instance, the signs and zero curves of $\phi_{1,2}$, $\phi_{1,3}$, $\phi_{1,4}$, and $\phi_{1,5}$ coincide, and we may initialize them to be identical. We combine them into a single function ϕ_1, and combine their individual votes in (3) into

$$\phi_1(\mathbf{x}, t) \begin{cases} > 0 \quad \Longrightarrow \quad \text{Region 1 gets 4 votes} \\ < 0 \quad \Longrightarrow \quad \text{Regions 2, 3, 4 and 5 get one vote each} \end{cases} \tag{8}$$

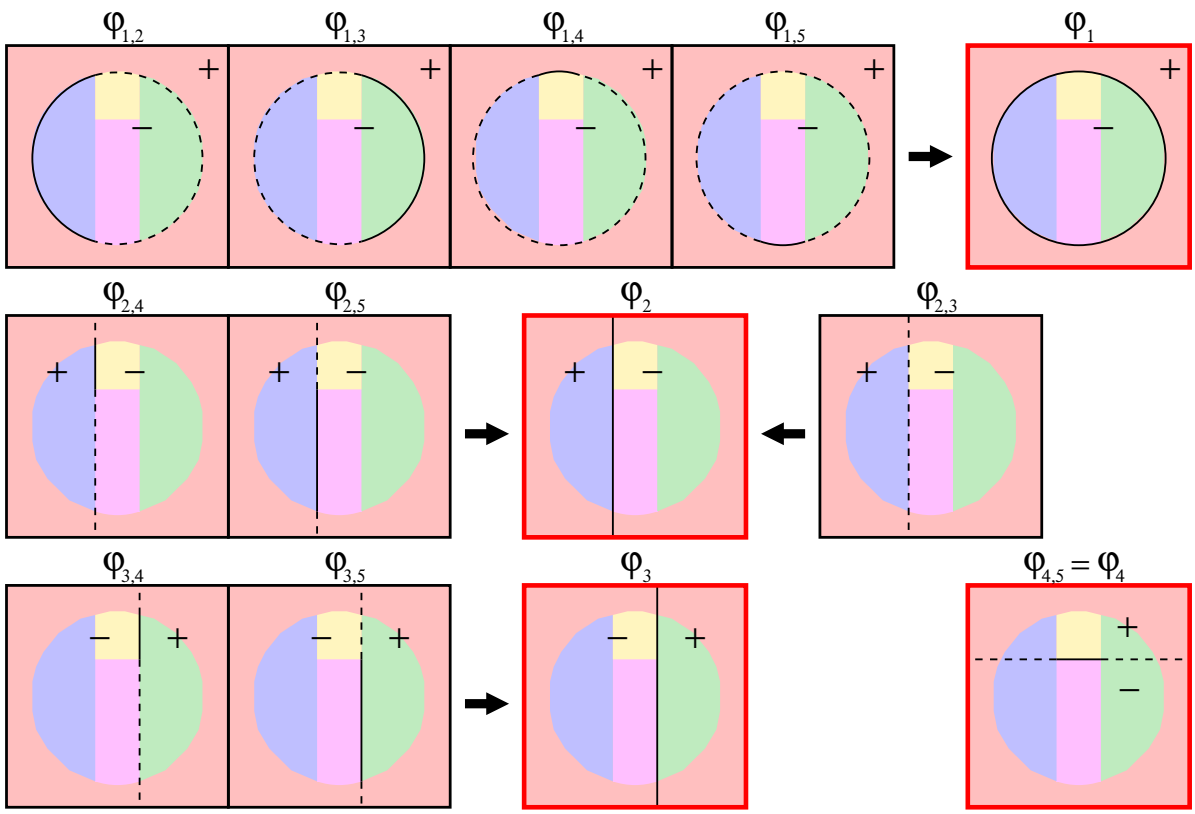

Figure 5: Level set initialization for Zalesak disk and grouping of redundant functions.

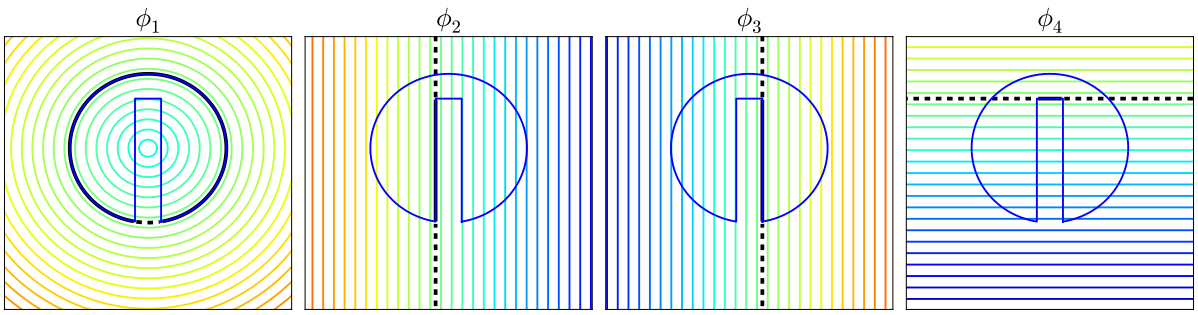

Figure 6: Initial smooth level set functions for Zalesak disk.

Similarly, $\phi_{2,4}$ and $\phi_{2,5}$ are combined into a single function ϕ_2, and $\phi_{3,4}$ and $\phi_{3,5}$ are combined into ϕ_3. For uniformity of notation, we rename $\phi_{4,5}$ as ϕ_4 and retain its original voting rule.

One function, $\phi_{2,3}$, still needs to be accounted for. Since Regions 2 and 3 do not share a boundary in this example, the zero curve of $\phi_{2,3}$ is unconstrained; the function need only be positive in Region 2 and negative in Region 3. We have some freedom here, and choose to initialize $\phi_{2,3}$ so that it coincides with $\phi_{2,4}$ and $\phi_{2,5}$. We include it in the group represented by ϕ_2 and amend the voting system of ϕ_2 to reflect that. (We could have incorporated $\phi_{2,3}$ with ϕ_3 and use $-\phi_{2,3}$ for sign consistency). The votes for each of the combined functions is given in Table 2.

Finally, to determine if a point is interior or exterior to the shape, we use (7), with

Function	Sign	Votes for Region				
		1	2	3	4	5
ϕ_1	$+$	4	-	-	-	-
	$-$	-	1	1	1	1
ϕ_2	$+$	-	3	-	-	-
	$-$	-	-	1	1	1
ϕ_3	$+$	-	-	2	-	-
	$-$	-	-	-	1	1
ϕ_3	$+$	-	-	-	1	-
	$-$	-	-	-	-	1

Table 2: Voting system for 4 smooth functions to represent the Zalesak disk.

$M = 5$, $I_{in} = \{2, 3, 4\}$, and $I_{out} = \{1, 5\}$.

5-Point Star

We walk through setting up the voting table one final time for the example of a 5-point star. We partition the domain into 11 regions, as illustrated in Figure 7. This partition produces 55 region pairs and 55 total level-set functions $\phi_{m,n}$, $1 \leq m < n \leq 11$. Five level-set functions are sufficient to represent the boundary of the star. We describe below how to collapse the 55 functions down to five functions, and amend the voting procedure to reflect that. The process of eliminating redundancies is not unique and can be carried out in any number of ways.

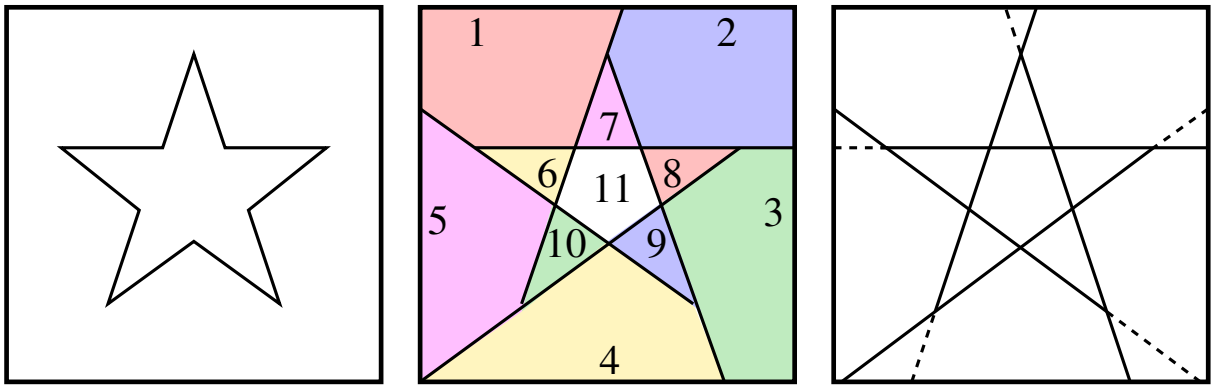

Figure 7: Domain partitioning for a 5-pointed star

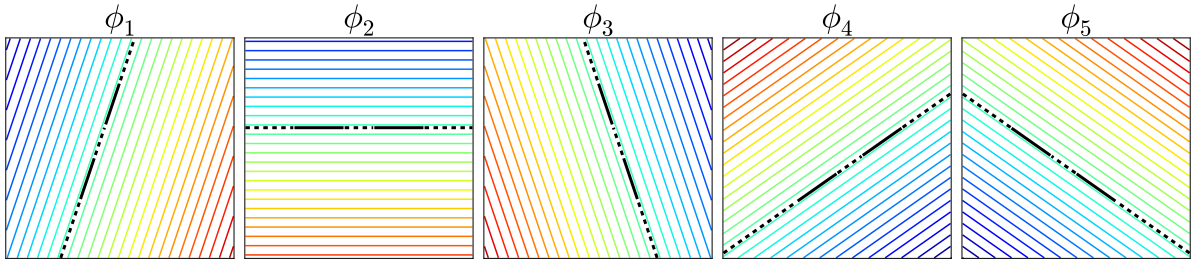

Figure 8: Initial smooth level set functions for 5-point star.

Consider Region 1 and all Region pairs $(1, n)$, $2 < n < 11$. Region 1 has three smooth boundary segments corresponding to three types of boundaries (see Figure 9): (i) shared with Region 2 and Region 7; (ii) shared with Region 6; and (iii) shared with Region 5. The functions $\phi_{1,2}$, $\phi_{1,7}$, $\phi_{1,6}$, and $\phi_{1,5}$ represent actual region interfaces. Six regions do not share an interface with Region 1: Regions 3, 4, 8, 9, 10, and 11. To maintain the integrity of the counting algorithm, it is necessary to account for those too. We have additional freedom in initializing these functions, as their zero curves may be placed based on convenience.

We group level-set functions $\phi_{1,n}$ as follows:

(i) The same zero curve that identifies the Region-(1,2) interface also identifies the Region-(1,7) interface. We use this to initialize $\phi_{1,2} = \phi_{1,7}$.

The same zero curve also separates Region 1 from Regions 3, 8, 9, 10, and 11, all of which do not share an interface with Region 1. We choose to initialize $\phi_{1,3}$, $\phi_{1,8}$, $\phi_{1,9}$, $\phi_{1,10}$, and $\phi_{1,11}$ to be equal. Other choices are possible.

(*ii*) $\phi_{1,6}$ is initialized so that its zero curve identifies the region-(1,6) interface.

The zero curve also separates Region 1 and Region 4. We choose to initialize $\phi_{1,4} = \phi_{1,6}$. (Again, the zero curve is unconstrained by (3) and we have some freedom in this decision.)

(*iii*) Finally, $\phi_{1,5}$ is initialized so that its zero curve identifies the (1,5)-interface.

All pairs involving Region 1 are accounted for, and all corresponding level-set functions are smooth.

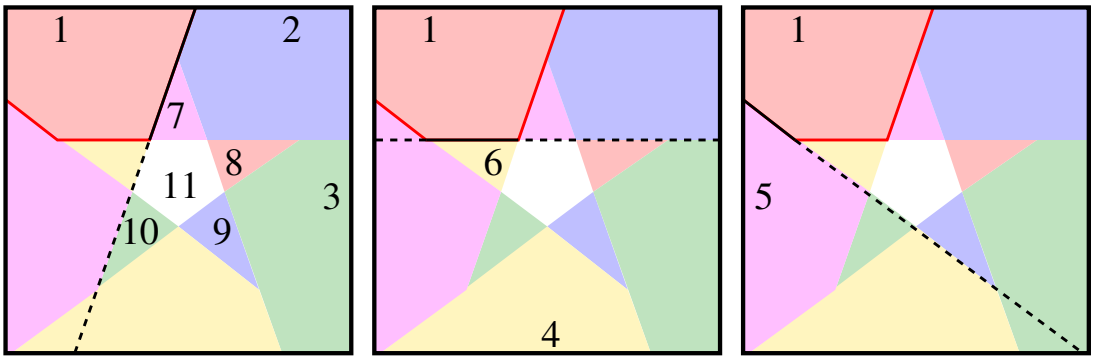

Figure 9: Smooth zero curves representing the boundary of Region 1.

Now consider Region 2 and all region pairs $(2, n)$, $3 < n < 11$. Region 2 also has three smooth boundary segments. One has already been accounted for in the Region 1 bookkeeping. The two that remain are (*i*) a boundary shared with Regions 3 and 8, and (*ii*) a boundary shared with Region 7 (see Figure 10).

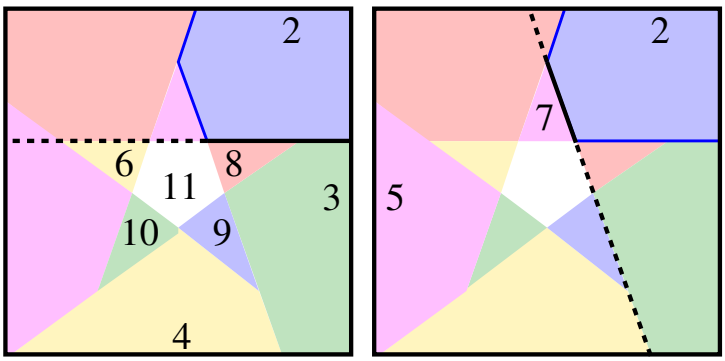

Figure 10: Smooth zero curves representing the boundary of Region 2.

We group the level-set functions associated with Region 2:

(*i*) The same smooth zero curve identifies the region-(2,3) and region-(2,8) interface. We initialize $\phi_{2,3} = \phi_{2,8}$.

The zero curve also separates Region 2 from regions 4, 6, 8, 9, 10, and 11. We choose to initialize the corresponding level set functions to be identical.

(*ii*) The smooth zero curve identifies the region-(2,7) interface and initializes $\phi_{2,7}$.

The curve also separates Region 2 from Region 5. We choose to initialize $\phi_{2,5} = \phi_{2,7}$.

All region pairs involving Region 2 are now accounted for.

We repeat the grouping process region-by-region and summarize results in Table 3. The left column corresponds to region pairs which share an interface. The right column contains region pairs which do not share an interface and have unconstrained zero curves. We refer to such functions as *Tag-Alongs*. All functions listed on the same line are identical.

Region	Region Interfaces	Non-Interface Tag-Alongs
1	$\phi_{1,2},\ \phi_{1,7}$	$\phi_{1,3},\ \phi_{1,8},\ \phi_{1,9},\ \phi_{1,10},\ \phi_{1,11}$
	$\phi_{1,6}$	$\phi_{1,4}$
	$\phi_{1,5}$	-
2	$\phi_{2,3},\ \phi_{2,8}$	$\phi_{2,4},\ \phi_{2,6},\ \phi_{2,9},\ \phi_{2,10},\ \phi_{2,11}$
	$\phi_{2,7}$	$\phi_{2,5}$
3	$\phi_{3,4},\ \phi_{3,9}$	$\phi_{3,5},\ \phi_{3,6},\ \phi_{3,7},\ \phi_{3,10},\ \phi_{3,11}$
	$\phi_{3,8}$	-
4	$\phi_{4,5},\ \phi_{4,10}$	$\phi_{4,6},\ \phi_{4,7},\ \phi_{4,8},\ \phi_{4,11}$
	$\phi_{4,9}$	-
5	$\phi_{5,6}$	$\phi_{5,7},\ \phi_{5,8},\ \phi_{5,9},\ \phi_{5,11}$
	$\phi_{5,10}$	-
6	$\phi_{6,11}$	$\phi_{6,7},\ \phi_{6,8},\ \phi_{6,9},\ \phi_{6,10}$
7	$\phi_{7,11}$	$\phi_{7,8},\ \phi_{7,9},\ \phi_{7,10}$
8	$\phi_{8,11}$	$\phi_{8,9},\ \phi_{8,10}$
9	$\phi_{9,11}$	$\phi_{9,10}$
10	$\phi_{10,11}$	-

Table 3: Region boundaries, organized into real interfaces and non-interface pairs. All functions listed on the same line are set equal.

At this point, we have reduced the number of level-set functions from 55 down to 16. The model can be reduced further. For instance, the level-set functions $\phi_{1,2}$, $\phi_{1,7}$, $\phi_{6,11}$, and $\phi_{5,10}$ and all of their tag-alongs are identical; they may be grouped together into a single function, ϕ_1 (see Table 4 and Figure 8). When ϕ_1 is positive, we consider each function $\phi_{m,n}$ in Table 4 and assign Region m one vote. Similarly, when ϕ_1 is negative, each region n receives one vote. The votes from each of the 13 functions are tallied and the voting power of ϕ_1 is amended (see Table 5).

Similarly, we denote by ϕ_2 the group of identical functions $\phi_{2,3}, \phi_{2,8}, \phi_{7,11}$ and $\phi_{1,6}$ as well as all their tag-alongs. We repeat the process for ϕ_3, ϕ_4 and ϕ_5, with each function

Real Region Interfaces	Non-Interface Tag-Alongs
$\phi_{1,2},\ \phi_{1,7}$	$\phi_{1,3},\ \phi_{1,8},\ \phi_{1,9},\ \phi_{1,10},\ \phi_{1,11}$
$\phi_{6,11}$	$\phi_{6,7},\ \phi_{6,8},\ \phi_{6,9},\ \phi_{6,10}$
$\phi_{5,10}$	-

Table 4: The 13 identical level set functions, organized by type, that collapse to ϕ_1.

Function	Sign	Votes for Region										
		1	2	3	4	5	6	7	8	9	10	11
ϕ_1	$+$	7	-	-	-	1	5	-	-	-	-	-
	$-$	-	1	1	-	-	-	2	2	2	3	2

Table 5: Voting system for ϕ_1 in 5-point star example.

corresponding to one of the five edges of the star. The grouping for each function is summarized in Table 6, and the amended voting algorithm is summarized in Table 7.

Function	Real Region Interfaces	Non-Interface Tag-Alongs
ϕ_1	$\phi_{1,2},\ \phi_{1,7}$	$\phi_{1,3},\ \phi_{1,8},\ \phi_{1,9},\ \phi_{1,10},\ \phi_{1,11}$
	$\phi_{6,11}$	$\phi_{6,7},\ \phi_{6,8},\ \phi_{6,9},\ \phi_{6,10}$
	$\phi_{5,10}$	-
ϕ_2	$\phi_{2,3},\ \phi_{2,8}$	$\phi_{2,4},\ \phi_{2,6},\ \phi_{2,9},\ \phi_{2,10},\ \phi_{2,11}$
	$\phi_{7,11}$	$\phi_{7,8},\ \phi_{7,9},\ \phi_{7,10}$
	$\phi_{1,6}$	$\phi_{1,4}$
ϕ_3	$\phi_{3,4},\ \phi_{3,9}$	$\phi_{3,5},\ \phi_{3,6},\ \phi_{3,7},\ \phi_{3,10},\ \phi_{3,11}$
	$\phi_{2,7}$	$\phi_{2,5}$
	$\phi_{8,11}$	$\phi_{8,9},\ \phi_{8,10}$
ϕ_4	$\phi_{4,5},\ \phi_{4,10}$	$\phi_{4,6},\ \phi_{4,7},\ \phi_{4,8},\ \phi_{4,11}$
	$\phi_{3,8}$	-
	$\phi_{9,11}$	$\phi_{9,10}$
ϕ_5	$\phi_{5,6}$	$\phi_{5,7},\ \phi_{5,8},\ \phi_{5,9},\ \phi_{5,11}$
	$\phi_{4,9}$	-
	$\phi_{10,11}$	-
	$-\phi_{1,5}$	-

Table 6: 55 original level set functions, organized by type, and collapsed to 5 functions.

Finally, to determine if a point is interior or exterior to the shape, we use (7) with $M = 11$, $I_{in} = \{1, 2, 3, 4, 5\}$, and $I_{out} = \{6, 7, 8, 9, 10, 11\}$.

We conclude by emphasizing again that the only table that is required for shape identification is Table 7. All other tables are purely illustrative of its construction.

Function	Sign	Votes for Region										
		1	2	3	4	5	6	7	8	9	10	11
ϕ_1	$+$	7	-	-	-	1	5	-	-	-	-	-
	$-$	-	1	1	-	-	-	2	2	2	3	2
ϕ_2	$+$	2	7	-	-	-	-	4	-	-	-	-
	$-$	-	-	1	2	-	2	-	2	2	2	2
ϕ_3	$+$	-	2	7	-	-	-	-	3	-	-	-
	$-$	-	-	-	1	2	1	2	-	2	2	2
ϕ_4	$+$	-	-	1	6	-	-	-	-	2	-	-
	$-$	-	-	-	-	1	1	1	2	-	2	2
ϕ_5	$+$	-	-	-	1	6	-	-	-	-	1	-
	$-$	1	-	-	-	-	1	1	1	2	-	2

Table 7: Voting table for 5-point star example.

3 Numerical Preliminaries

3.1 Quantification of Error

Let ϕ_{exact} be the (single) level set function which identifies the analytic boundary of some geometry. And let $\Phi^n = \{\phi_1^n, \ldots, \phi_N^n\}$ give the set of computed level set functions at some discrete time n for the new model. We denote the analytic area of the geometric object by

$$A = \int_\Omega H(\phi_{exact}) dA,$$

and the computed area at time level n by

$$A^n = \int_\Omega H(\Phi^n) dA$$

with

$$A^0 = \int_\Omega H(\Phi^0) dA.$$

denoting the computed area at the initial time. We measure the error using the corresponding metrics:

(i) L_1 *error*: This measures the discrepancy between the computed and exact geometry boundary. We use the formulation in [1, 3]:

$$E_1 = \int_\Omega \Big| H(\phi_{exact}) - H(\Phi^n) \Big| dA. \tag{9}$$

(ii) *Relative error in computed area compared to exact area*:

$$E_{exact}^n = \frac{A^n \; - \; A}{A}, \tag{10}$$

(iii) *Relative error in computed area compared to initial computed area*:

$$E^n_{discrete} = \frac{A^n \; - \; A^0}{A^0}, \tag{11}$$

Integrals are computed over the full computational domain Ω in a manner similar to [1]. Each grid cell is partitioned into 100×100 sub-grid cells, with Φ^n interpolated onto each sub-grid cell center. The voting algorithm is used to determine if a sub-grid cell is inside/outside the given geometry. The Heaviside function $H(\Phi^n)$ is computed as constant in each sub-grid cell i:

$$H(\Phi^n)_i \; = \; \begin{cases} 1 & , \; \text{sub-grid cell } i \text{ inside geometry} \\ 0 & , \; \text{sub-grid cell } i \text{ outside geometry} \end{cases} .$$

Integration for a single mesh cell amounts to summing over each sub-cell contribution i.

3.2 Flow Field Examples

We present numerical examples of two-dimensional solid and deformable geometries embedded in prescribed velocity fields. All level set functions are evolved by solving (5) with the underlying velocity field $\mathbf{u}(x, y, t) = (u, v)(x, y, t)$ specified for each test. The following velocity fields are considered:

(i) **Solid Rotation**: The velocity field corresponds to a constant-vorticity flow centered at $(1/2, 1/2)$:

$$\begin{aligned} u(x, y) &= (\pi/314)(0.5 - y) \\ v(x, y) &= (\pi/314)(x - 0.5) \end{aligned} \tag{12}$$

A particle completes one full rotation every $t = 628$ time units.

(ii) **Single Vortex**: The velocity field $\mathbf{u}_0(x, y) = (u_0, v_0)(x, y)$ corresponds to a vortex centered at $(1/2, 1/2)$:

$$\begin{aligned} u(x, y) &= \cos(\pi t/T) \sin(\pi x) \cos(\pi y) \\ v(x, y) &= -\cos(\pi t/T) \cos(\pi x) \sin(\pi y) \end{aligned} \tag{13}$$

where T gives the final time. The vortex field is counterclockwise for the first $T/2$ time units and clockwise for the last $T/2$ time units. This reverses the deformation on the geometry in the flow and returns it to its original state in T time units.

3.3 Numerical Solver Details

We use a second-order flux-corrected upwind scheme with double minmod flux limiter [5], and employ dimension by dimension splitting. All the tests are computed on a unit box, using an $N \times N$ square mesh and solid wall boundary conditions. The specified velocity fields have zero normal velocity at the boundaries.

Results from the new model are compared to the standard single-function level set model. Wherever indicated, we use a redistancing scheme to restore smoothness to the

level-set functions. This step is necessary in order to reduce the effects of numerical diffusion on the standard level set model. The new model performs very well without redistancing, and has little to gain from incorporating a redistancing step. We employ a second-order-accurate version of the Sussman-Fatemi scheme [3] at each time step. Two iterations of redistancing are performed using a CFL number of 0.2 to compute pseudo-timesteps.

4 Numerical Results

4.1 Zalesak Disk

Following [1], we consider a disk of radius 0.15 centered at $(0.5, 0.75)$, with a notch of width 0.05 and length of 0.25. The area of the disk is 5.822e-2. We have implemented the new level-set model, using four level-set functions as described in section 2.2, with initial level set contours given in Figure 6 and voting table given in Table 2.

Solid Rotation:
Figure 11 compares the final location of the disk to the analytic solution after one, two, and three rotations on a grid of 100×100 cells. The top row corresponds to the standard level set model with redistancing. The shaded region is a filled contour of $H(\Phi^n)$ evaluated at each cell center. Round-off at the corners of the notch becomes more prominent and causes the solution to deteriorate with each rotation. Redistancing is necessary in order to be able to resolve the notch, but doing so results in displacement and distortion of the zero curve over time. By comparison, results by the new model (bottom row) are in near-perfect agreement with the analytic solution, with little difference is observed from one rotation to the next. Redistancing is not required for the new model, and is not used.

Tables 8 and 9 summarize the error in the boundary location after one, two, and three rotations under grid resolution. We observe that while redistancing leads to improvements in overall disk area computed by the standard scheme, the shape of the disk is not accurate, and is visibly distorted. The new level-set model does an excellent job preserving sharp corners, and gives excellent agreement with the exact solution both in terms of area preservation and shape preservation. Since the level-set functions in the new model are smooth, there is no need, and indeed little to be gained, from using redistancing.

Deformation by a single vortex:
Figure 12 shows the computed location of the disk at intermediate times t =0.5, 1, 2, 3, 3.5 and at final time $T = 4$ using the new level set model on a grid of 100×100 cells.

We compare the results by the level set models using the time-reversed single-vortex flow field (13) with final time $T = 4$, both with and without redistancing. Figure 13 gives the final locations of the disk on a 100×100 grid along with the analytic boundary. We observe that without redistancing, the standard model is unable to represent adequately the notch. Redistancing recovers the notch, but also displaces the zero curve. By contrast, numerical diffusion has minimal effect on the new model. Computed results preserve the boundary shape extremely well, both with and without redistancing.

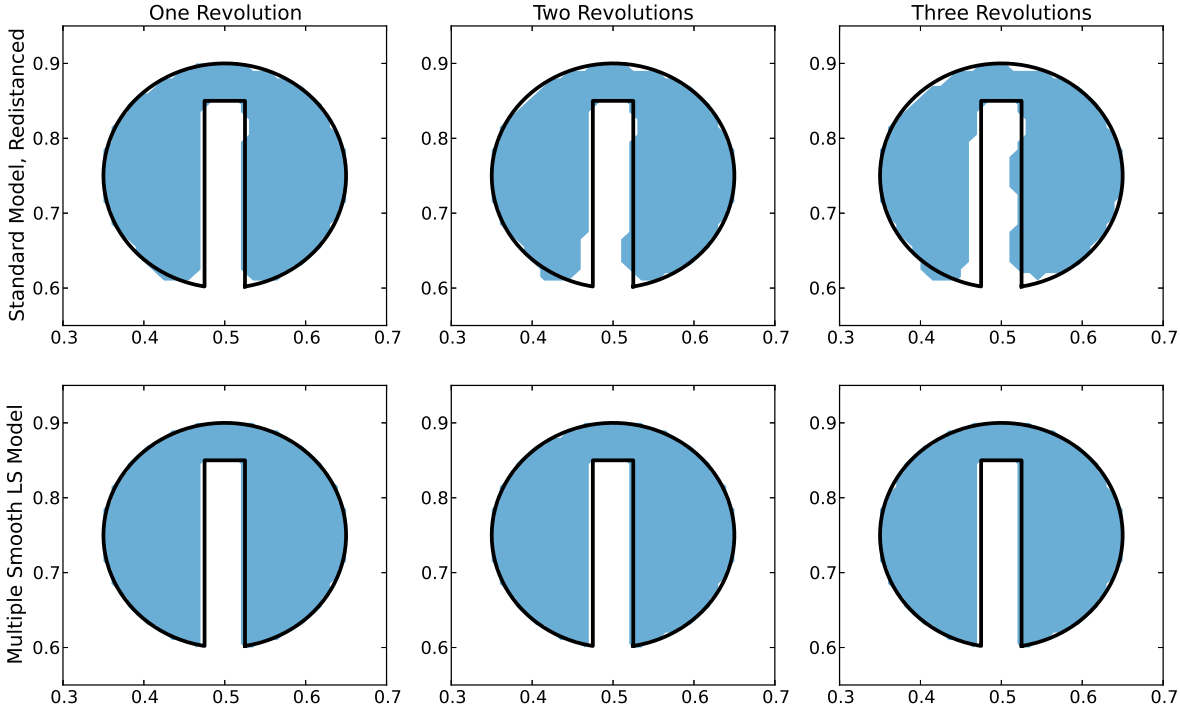

Figure 11: Location of Zalesak disk for solid rotation flow after 1, 2, and 3 revolutions.

	Grid cells	Area	% Area Change	L_1 Error
One revolution	50	5.484e-2	-5.81%	1.764e-1
	100	5.731e-2	-1.56%	3.245e-2
	200	5.821e-2	-0.01%	3.744e-3
Two revolutions	50	5.593e-2	-3.94%	2.363e-1
	100	5.671e-2	-2.59%	4.576e-2
	200	5.821e-2	-0.01%	6.656e-3
Three revolutions	50	5.691e-2	-2.25%	2.729e-1
	100	5.618e-2	-3.50%	6.906e-2
	200	5.820e-2	-0.03%	8.320e-3

Table 8: Solid rotation: standard model with redistancing applied to Zalesak disk

4.2 5-Point Star

We consider a regular 5-pointed star, centered at $(1/2, 1/2)$ with radius (center to point tip) of 0.3. We use five level set functions to represent this shape, initialized as described in Figure 8 and voting system given by Table 7.

Figure 14 shows results for the 5-point star in a time-reversed single-vortex flow (13) with final time $T = 8$ on a both 50×50 and 100×100 grids. The tips of the star round off due to numerical diffusion using the standard model, leading to loss of area. Redistancing reduces area loss but distorts the zero level curve over time, leading to poor shape preservation. By stark contrast, the new model does an excellent job maintaining

	Grid cells	Area	% Area Change	L_1 Error
One revolution	50	5.826e-2	0.07%	2.330e-2
	100	5.809e-2	-0.23%	2.163e-2
	200	5.818e-2	-0.06%	2.080e-4
Two revolutions	50	5.995e-2	2.96%	2.496e-2
	100	5.802e-2	-0.35%	2.330e-2
	200	5.819e-2	-0.06%	4.160e-4
Three revolutions	50	6.120e-2	5.12%	3.495e-2
	100	5.808e-2	-0.25%	2.413e-2
	200	5.820e-2	-0.04%	8.320e-4

Table 9: Solid rotation: new model (without redistancing) applied to Zalesak disk

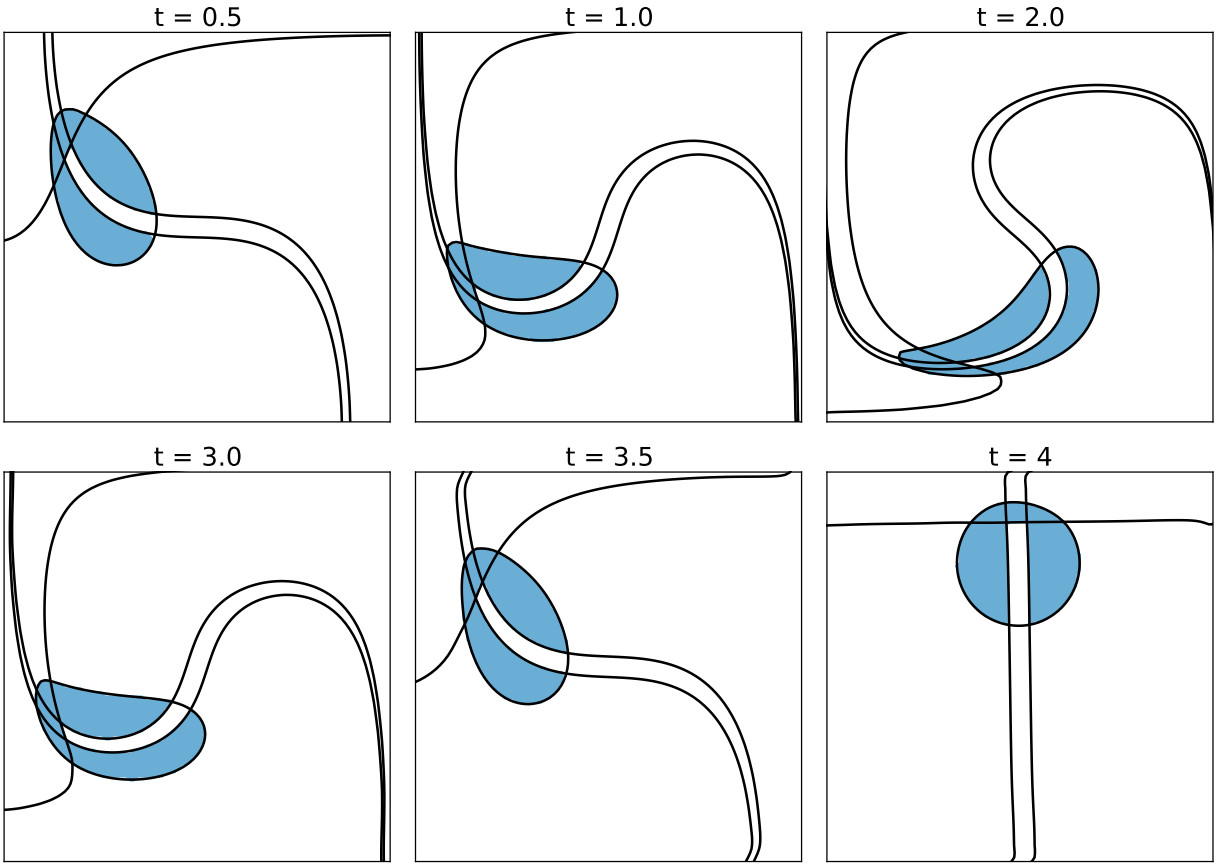

Figure 12: Smooth level set model: location of Zalesak disk and zero curves over time for the single vortex flow.

the sharp corners of the star and its overall shape, even without redistancing. Results from the new model on the coarser mesh even appear qualitatively better than results from the standard model on the finer mesh.

Error histories in the enclosed area are shown in Figure 15 and summarized in Table 12. In each, we compare the new model without redistancing to the standard model

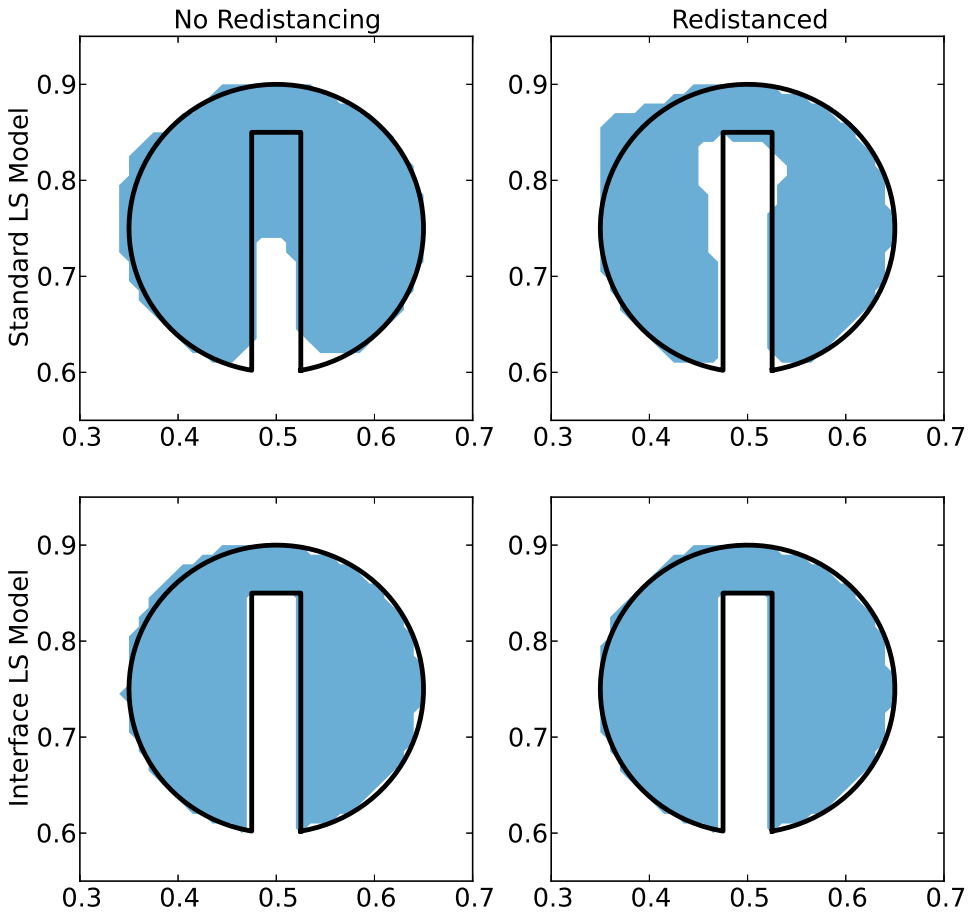

Figure 13: Location of Zalesak disk for single-vortex flow at final time $T = 4$.

	Grid cells	Area	% Area Change	L_1 Error
No redistancing	50	6.782e-2	16.5%	1.398e-1
	100	6.724e-2	15.5%	7.988e-2
	200	5.904e-2	1.40%	3.079e-2
Redistancing	50	6.302e-2	8.25%	2.496e-1
	100	5.762e-2	-1.03%	7.988e-2
	200	5.823e-2	0.01%	1.955e-2

Table 10: Single vortex: final time $T = 4$, standard model applied to Zalesak disk

with redistancing. The new model exhibits significant improvement in all error metrics. We note that the relative error in area computed by the new model increases up to the maximum deformation of the geometry at $t = 4$ and decreases afterwards. In the standard model, errors continue to accumulate, as numerical diffusion prevents the star from recovering its original shape during the reverse deformation.

	Grid cells	Area	% Area Change	L_1 Error
No redistancing	50	6.059e-2	4.07%	9.319e-2
	100	5.904e-2	1.41%	4.576e-2
	200	5.865e-2	0.74%	1.851e-2
Redistancing	50	5.883e-2	1.04%	9.152e-2
	100	5.845e-2	0.39%	4.160e-2
	200	5.839e-2	0.28%	1.518e-2

Table 11: Single vortex: final time $T = 4$, new model applied to Zalesak disk

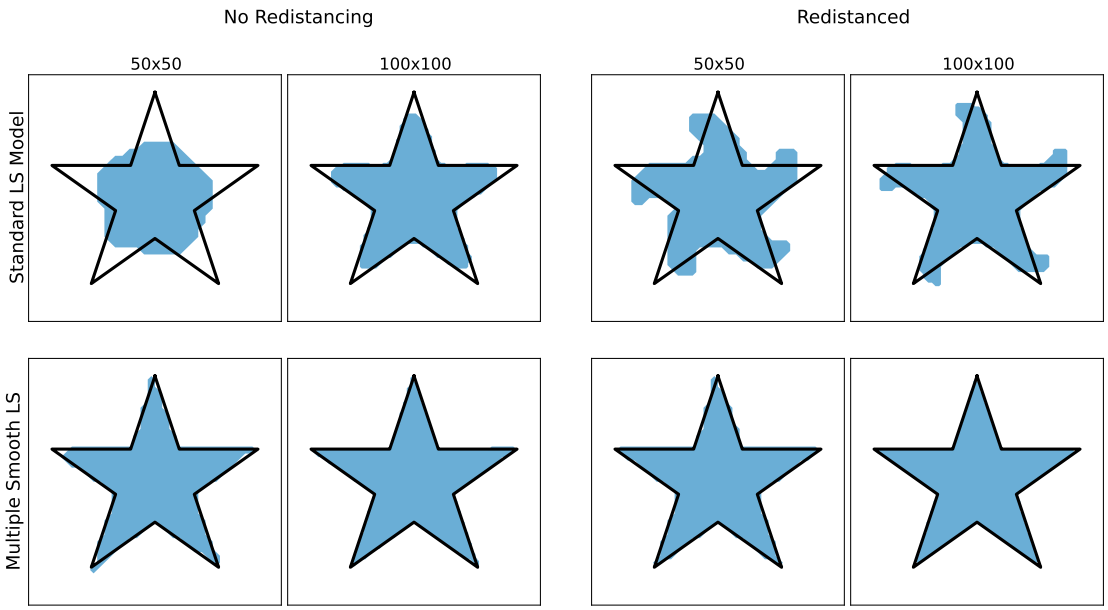

Figure 14: Location of 5-point star for single vortex flow at final time $T = 8$.

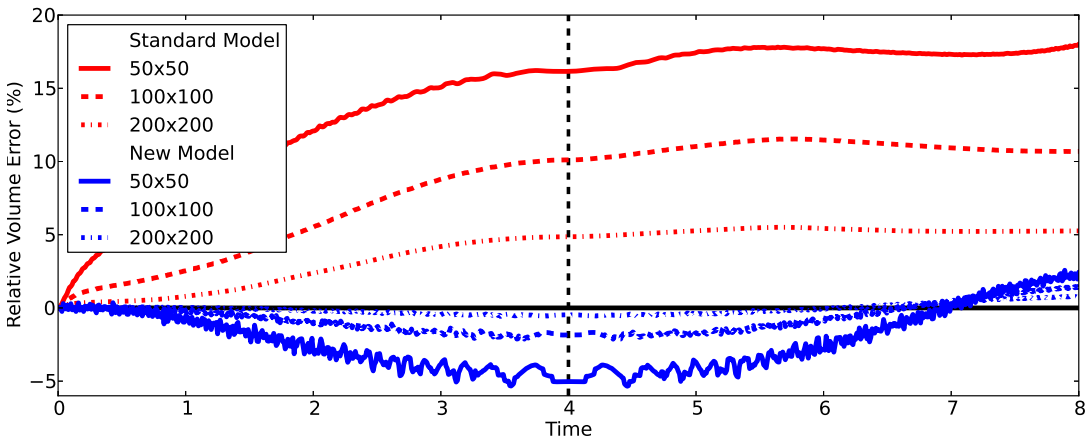

Figure 15: Relative errors in enclosed area over time under grid refinement, 5-point star.

4.3 Cusped Triangle

In this example, we consider a "triangle" with cusp vertices, centered at $(1/2, 1/2)$. We use four level set functions initially set as in Figure 16, and the voting table given in

	Grid cells	Area	% Area Change	L_1 Error
Standard Model	50	1.168e-1	15.6%	6.225e-2
(redistancing)	100	1.113e-1	10.2%	3.722e-2
	200	1.062e-1	5.13%	1.688e-2
New Model	50	1.013e-1	0.27%	1.009e-2
(no redistancing)	100	1.020e-1	0.99%	6.309e-3
	200	1.017e-1	0.69%	3.838e-3

Table 12: Single vortex: final time $T = 8$, 5-point star

	Grid cells	Area	% Area Change	L_1 Error
Standard Model	50	3.495e-2	3.27%	8.640e-2
(redistancing)	100	3.490e-2	3.12%	4.440e-2
	200	3.416e-2	0.96%	2.070e-2
New Model	50	3.311e-2	-2.16%	3.840e-2
(no redistancing)	100	3.352e-2	-0.96%	1.192e-2
	200	3.380e-2	-0.12%	1.320e-2

Table 13: Single vortex: final time $T = 4$, cusped triangle

Table 1.

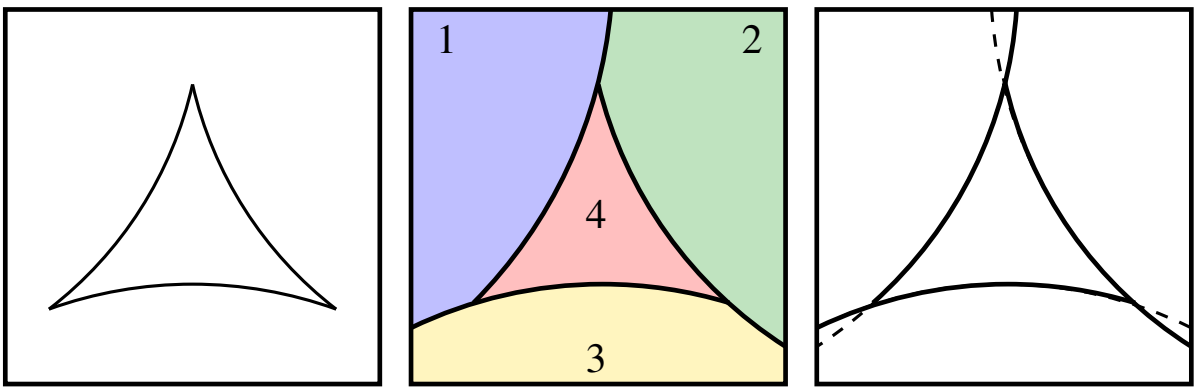

Figure 16: Domain partitioning for a cusped triangle.

Figure 17 gives the final location of the shape for the solid rotation flow (12) after 2, 4, and 8 revolutions on a grid of 100×100 cells. Error histories in the enclosed area are shown in Figure 18. Table 13 compares absolute errors in enclosed area and L_1 errors.

As before, the new model does an excellent job preserving the shape of the cusped triangle. It outperforms the standard model in all error metrics. Volume errors are significantly smaller at every timestep and do not appear to accumulate in time.

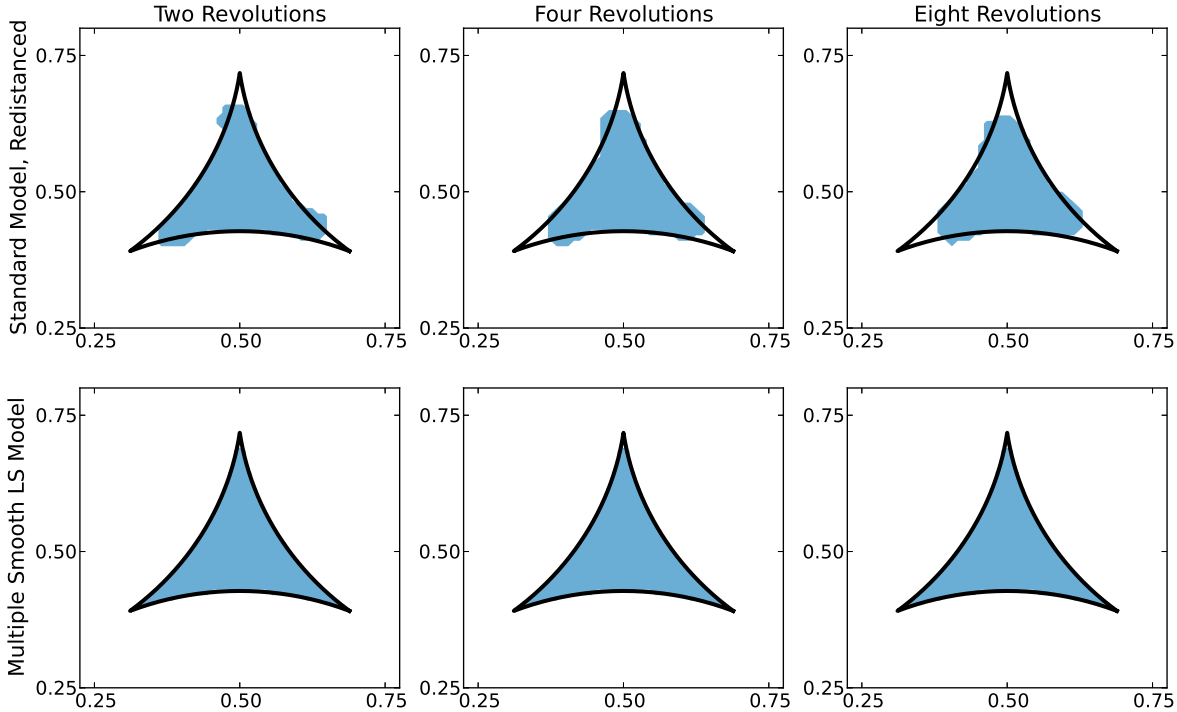

Figure 17: Cusped triangle under solid rotation flow after 2, 4, and 8 revolutions.

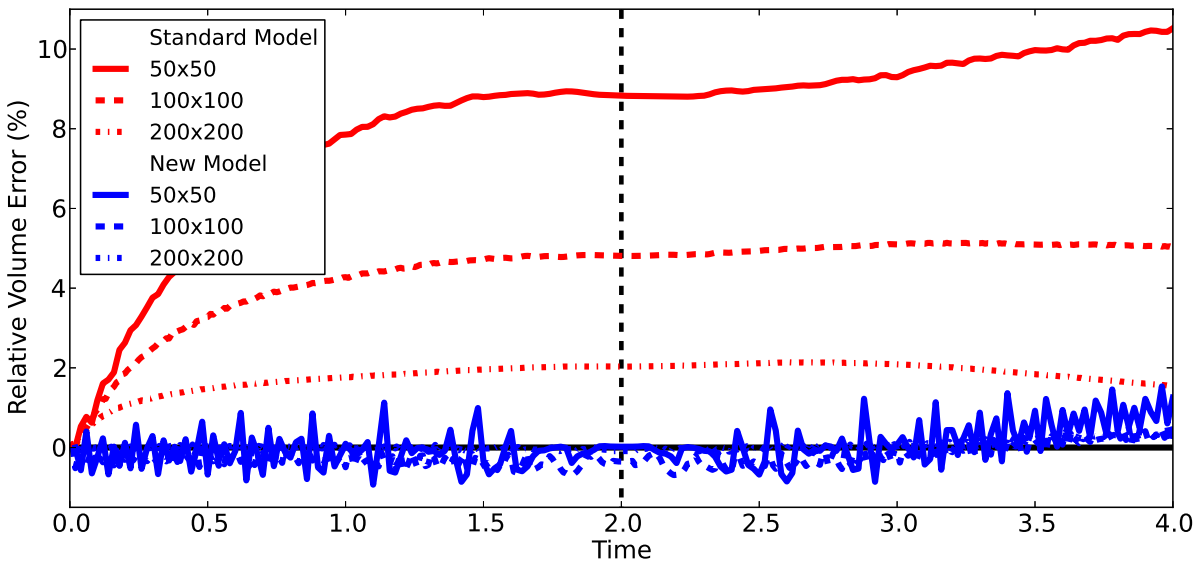

Figure 18: Relative error in area for cusped triangle example.

References

[1] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. J. Comput. Phys., 183(1):83 – 116, 2002.

[2] D. Starinshak, S. Karni, and P. Roe. A new level set model. 2013. In Review.

[3] M. Sussman and E. Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. SIAM J. Sci. Comput., 20(4):1165–1191, February 1999.

[4] M Sussman, P Smereka, and S Osher. A level set approach for computing solutions to incompressible two-phase flow. J. Comput. Phys., 114(1):146 – 159, 1994.

[5] B. van Leer. Towards the ultimate conservative difference scheme iii. upstream-centered finite-difference schemes for ideal compressible flow. J. Comput. Phys., 23(3):263 – 275, 1977.

[6] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. J. Comput. Phys., 31(3):335 – 362, 1979.

Prepared by LLNL under Contract DE-AC52-07NA27344.

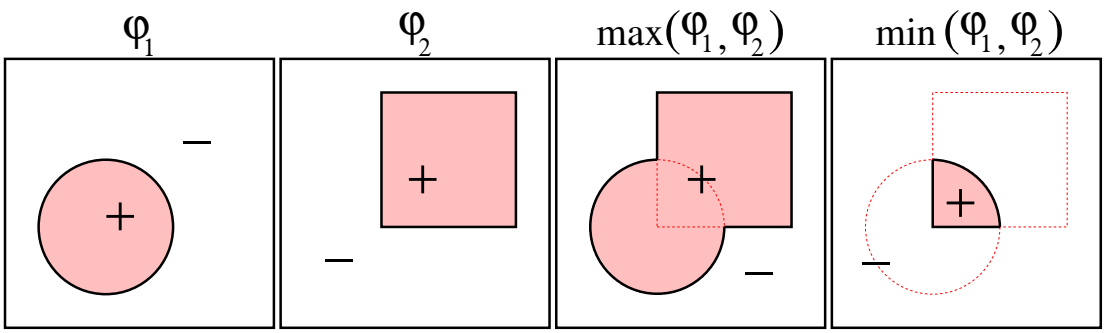

Figure 19: Union and intersection of geometrical shapes via maximum and minimum of their level set functions respectively.

Appendix: Shape Identification by Projection to a Single Function

In this section, we describe a method to project multiple level set functions into a single function, which is positive inside the geometry, negative outside, and preserves the same accurate figure boundaries as the multiple level set functions. It bears some relationship to the voting algorithm and offers an alternative approach to the use of sign information in the multiple level-set model to extract accurate shape information.

The projection is based on computing unions and intersections of convex regions. Consider two level set functions ϕ_1 and ϕ_2 whose positive parts each define a geometric region. We may generate new level set functions by taking the minimal and maximal values of the two functions at every point in space. In particular, the positive parts of $\min(\phi_1, \phi_2)$ and $\max(\phi_1, \phi_2)$ define the intersection and union respectively of the original regions (see Figure 19).

We define the projection method for convex figures first, then generalize to non-convex ones.

Convex Geometries

We represent each smooth boundary segment by a level set function, ϕ_i, whose zero curve contains the corresponding boundary segment and is positive in the interior of the geometry. Since each boundary segment is smooth, the level set functions may be initialized as smooth distance functions (see Figure 20). Convexity of the figure ensures that the intersection of all the positive regions of the set of level set functions determines the interior of the shape. Thus, the projected function

$$\phi_{proj} = \min_i \phi_i.$$

is positive inside the shape and negative outside. In addition, if each smooth function is also a distance function with respect to its respective boundary segment, then so is ϕ_{proj} with respect to the full boundary.

Non-Convex Geometries

The projection algorithm is generalized to non-convex shapes by representing them as a collection of convex sub-geometries. We describe the algorithm in stages and use the

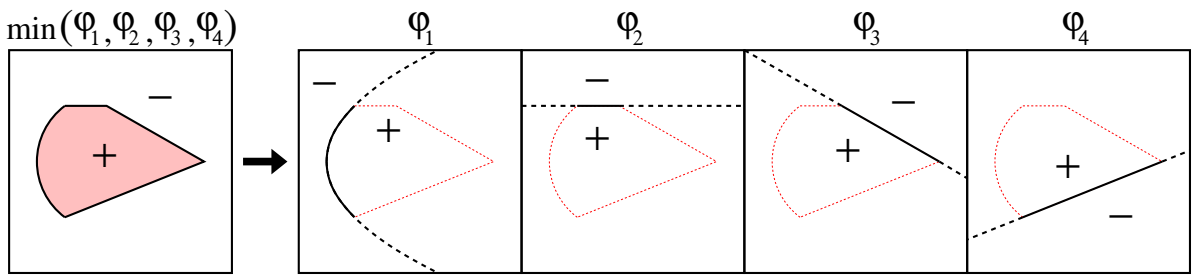

Figure 20: Shape identification as the intersection of positive parts of smooth level set functions associated with boundary segments.

Zalesak disk example for illustration:

1. Divide the full boundary into smooth segments and define a corresponding level set function for each segment (see Figure 21). The level set function is positive on the side of the boundary segment that is interior to the shape and negative on the other side.

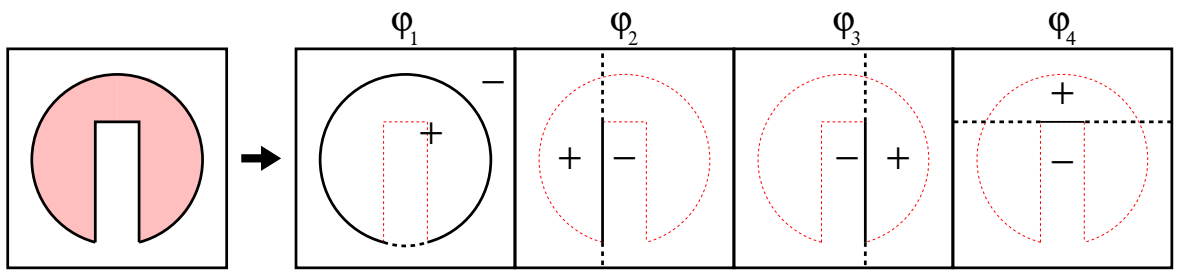

Figure 21: Stage 1: Define a smooth level set function for each boundary segment of the full figure.

2. Divide the geometry into *overlapping* convex sub-geometries. This decomposition is not unique, and many options are available (see Figure 22). Let Φ_k be the set of level set functions whose intersection forms the k-th convex sub-geometry.

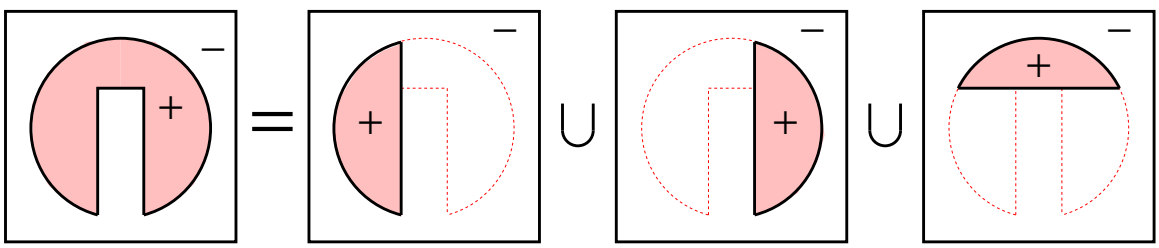

Figure 22: Stage 2: Division of non-convex geometry into overlapping convex regions

3. Define the projected function $\phi_{proj}^{(k)}$ for the k-th convex sub-geometry (See Figure

23:)

$$\phi_{proj}^{(k)} = \min_{\phi \in \Phi_k} \phi.$$

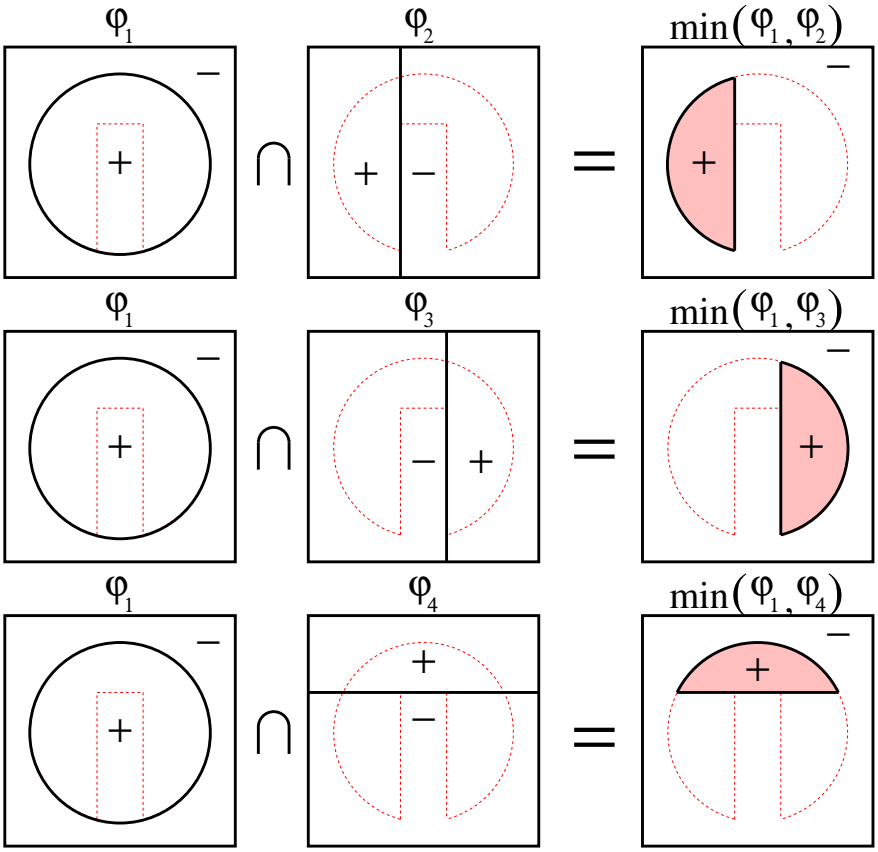

Figure 23: Stage 3: Represent convex regions as the intersection of the corresponding smooth level set functions.

4. Recover the non-convex geometry as the union of its overlapping convex sub-geometries:

$$\phi_{proj} = \max_k \phi_{proj}^{(k)}.$$

Remark: The requirement that the convex sub-geometries overlap is in order to prevent the formation of interior boundaries. If two non-overlapping sub-geometries meet along a boundary curve, their individual projected functions will be zero there. Taking the maximal value of the two produces a function that is zero at the common boundary and positive on either side.