myc2d.h,myc.h Documentation

Haochen (Langford) Huang

July 24, 2025

Version: draft   Updated: 2025-07-15

Section 1

Introduction

'MYC' stands for the Mixed Young's Centered scheme, introduced by Scardovelli et al. [2]. It is used to compute the interface normal vector **n** at a given cell, based on the volume fraction c in the current cell and its neighboring cells (9 cells in 2D and 27 cells in 3D).

In the following sections, I will first introduce the basic concept of representing an interface with a linear function, followed by the algorithm that enables the mixed scheme to determine the normal direction in 2D cases. This will then be extended to the 3D scenario.

1.1 2D scenario

1.1.1 Linear Representation of the Interface

Consider an interfacial cell on a 2D plane, as shown in figure 1. Let the normal direction of the interface be denoted by $\mathbf{n} = (n_x, n_y)$. It is straightforward to show that the interface can be represented by the linear equation

$$n_x X + n_y Y = \alpha, \tag{1}$$

where α is a constant that determines the position of the interface.
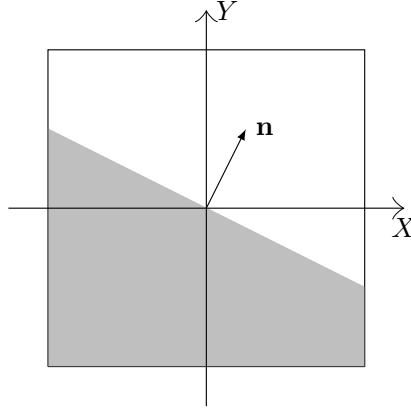
Figure 1: Linear representation of the interface, with normal direction $\mathbf{n} = (n_x, n_y)$.

This equation indicates that the linear interface within a cell is fully characterized once the triplet (n_x, n_y, α) is known.

If we treat Y as a function of X, the equation takes the slope-intercept form

$$sign(n_y)Y = -\frac{n_x}{|n_y|}X + \frac{\alpha}{|n_y|} = m_y X + b_y, \tag{2}$$

where $m_y = -\frac{n_x}{|n_y|}$ and $b_y = \frac{\alpha}{|n_y|}$.

Similarly, X can be expressed as a function of Y, with slope $m_x = -\frac{n_y}{|n_x|}$. The normal direction obtained from equation 2 can thus be written in the form $(-m_x,, \text{sign}(n_y))$ (or equivalently, $(\text{sign}(n_x),, -m_y)$). This form is derived by transforming equation 2 back into the original form of the interface representation, as shown in equation 1.

With the interface representation clarified, we now introduce the centered scheme, Young's scheme, and finally the mixed scheme. As will be shown, the centered scheme estimates the interface normal direction through the form of equation 2, while Young's scheme directly computes the normal vector (n_x, n_y).

1.1.2 Centered columns scheme

For the linear equation 2, the most straightforward way to compute the slope is by selecting an arbitrary pair of points on the line, denoted as (X_1, Y_1) and (X_2, Y_2), and calculating

$$m_x = \frac{Y_2 - Y_1}{X_2 - X_1}. \tag{3}$$

The centered columns scheme adopts the same idea but operates under discrete conditions. This naturally raises several questions:

1. Given a value of X, how can we determine the corresponding output Y in a discrete setting?

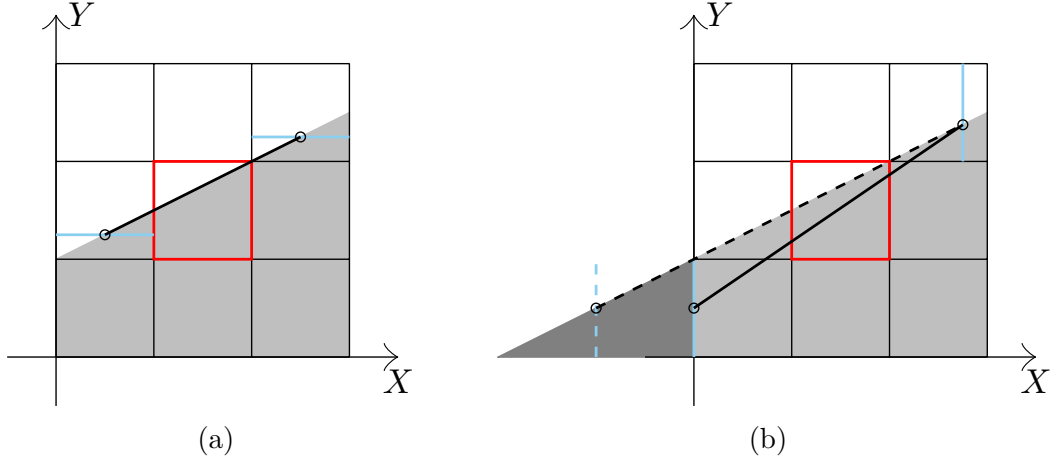2. Since there are two slope estimates, which one should be selected?

(a) (b)

Figure 2: Sketch of the computation process for the centered scheme. The red cell highlights the current cell where the interfacial slope is being computed. The blue lines represent the summation of volume fractions across columns, and the black solid line shows the resulting slope obtained by the centered scheme. In figure (a), the scheme accurately captures the interface when it spans across opposite boundaries. By contrast, figure (b) presents the same scenario but from a complementary perspective. The result reveals a significant discrepancy between the computed and actual slopes. Note that the dashed and solid lines are intentionally reversed to facilitate comparison with the true slope of the interface.

Figure 2a illustrates the first question by showing how the centered scheme computes the interfacial slope. Similar to the height function method described in the `heights.h` documentation, the value of Y at each column is approximated by summing the volume fractions c of three adjacent cells aligned along the same X coordinate. Once two neighboring values (marked as empty circles) are obtained, the slope at the current cell (highlighted in red) is computed using a centered finite difference. The result is indicated by the solid black line.

Let the current cell be denoted by index $(0,0)$. The slope in figure 2a is then given by:

$$m_x = \frac{1}{2}\left(\sum_{j=-1}^{1} c_{1,j} - \sum_{j=-1}^{1} c_{-1,j}\right) \tag{4}$$

Since $m_x = -\frac{n_y}{|n_x|}$, it follows that $\text{sign}(n_x) = -\text{sign}(m_x)$, and similarly, $\text{sign}(n_y) = -\text{sign}(m_y)$. This dual representation for the same interface raises the second question: which of the two directional estimates gives the more accurate result?

Figure 2b presents the same scenario but treats Y as the independent variable and X as the dependent output. In this case, the slope computed by the centered scheme deviates significantly from the true interface, as part of the left column (shown in dark gray) is omitted. For a perfectly linear interface, the centered scheme yields accurate results only when the interface intersects the left and right boundaries of the current cell. However, when the interface crosses adjacent boundaries (e.g., left and top) or spans the top and bottom boundaries, the accuracy deteriorates due to partial volume loss, as

illustrated in figure 2b. In other words, the smaller the absolute value of the slope, the more accurate the estimate [1].

Based on this observation, the normal direction is chosen as $(-m_x, sign(m_y))$ when $|m_y| > |m_x|$, and as $(sign(m_x), -m_y)$ otherwise.
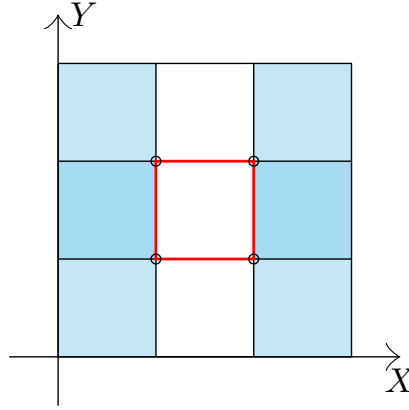
1.1.3 Young's Scheme

Figure 3: Illustration of Young's scheme. The interface normal is obtained by averaging the normals at four cell corners, indicated by empty circles. Blue shading marks the cells involved in the final algebraic expression, with dark blue showing overlaps.

Young's scheme estimates the interface normal vector $\mathbf{n}$ by averaging the normals computed at four cell corners. As shown in figure 3, the normal at each corner is evaluated using central differences of the surrounding cell values. For example, considering the target cell with index $(0, 0)$, the x-component of the normal at corner $(1/2, 1/2)$ is given by

$$n_{x,(1/2,1/2)} = \frac{1}{2\Delta} \left(c_{1,1} + c_{1,0} - c_{0,1} - c_{0,0} \right) \tag{5}$$

The average x-component of the normal over all four corners is then computed as

$$n_x = \frac{1}{4} \left(n_{x,(1/2,1/2)} + n_{x,(-1/2,1/2)} + n_{x,(1/2,-1/2)} + n_{x,(-1/2,-1/2)} \right) \tag{6}$$

Alternatively, a more compact expression for n_x can be obtained by directly summing the contributions from relevant cells:

$$n_x = \frac{1}{8} \left(C_1 - C_{-1} \right) \tag{7}$$

where

$$C_1 = c_{1,1} + 2c_{1,0} + c_{1,-1}, \quad C_{-1} = c_{-1,1} + 2c_{-1,0} + c_{-1,-1}$$

as indicated by the blue regions in figure 3. The same procedure applies for computing n_y. Note that the coefficient $\frac{1}{8}$ may be omitted or replaced, provided both components of $\mathbf{n}$ are scaled consistently.

References

[1] Eugenio A. et al. "Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry". In: *J. Comput. Phys.* 225.2 (2007), pp. 2301–2319.

[2] R. Scardovelli and S. Zaleski. "Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection". In: *Int. J. Numer. Methods Fluids* 41.3 (2003), pp. 251–274.