GFM treatment of jump conditions in proximity of a membrane

Aurelio Spadotto

February 2, 2023

1 The GFM correction scheme

Consider a simple closed curve $\Sigma \subset \mathbb{R}^2$. At each point $\boldsymbol{X} \in \Sigma$ define the outward facing normal vector $\boldsymbol{n}(X)$. Σ partitions $\mathbb{R}^2$ into an internal region Ω^- and an external one, $\Omega^+ = (\overline{\Omega^-})^c$. Define a region-wise constant function σ such that:

$$\sigma(\boldsymbol{x}) = \begin{cases} \sigma^- & \boldsymbol{x} \in \Omega^- \\ \sigma^+ & \boldsymbol{x} \in \Omega^+ \end{cases} \frac{\sigma^{in}}{\sigma^{out}} = 10 \tag{1}$$

$$\begin{cases} \nabla \cdot (\sigma \nabla \phi(\boldsymbol{x})) = 0 & \boldsymbol{x} \in \Omega^{+/-} \\ [\![\phi]\!]_{\boldsymbol{X}} = V_m(\boldsymbol{X}) & \boldsymbol{X} \in \Sigma \\ [\![\sigma \nabla \phi]\!]_{\boldsymbol{X}} \cdot \boldsymbol{n} = 0 & \boldsymbol{X} \in \Sigma \end{cases} \tag{2}$$

$$\boldsymbol{E} = \nabla \phi, \; = 6\boldsymbol{e} = [\![\epsilon(\boldsymbol{E}\boldsymbol{E} - \frac{1}{2}(\boldsymbol{E} \cdot \boldsymbol{E})\bar{\bar{\boldsymbol{I}}})]\!] \tag{3}$$

where the jump operator $[\![\cdot]\!]$ for a function $f : \Omega \to \mathbb{R}$ is defined as follows:

$$[\![f]\!](\boldsymbol{X}) := \lim_{\boldsymbol{x} \to \boldsymbol{X}} f^+(\boldsymbol{x}) - \lim_{\boldsymbol{x} \to \boldsymbol{X}} f^-(\boldsymbol{x}),$$

$V_m : \Sigma \to \mathbb{R}$ is a given function expressing the jump of ϕ across Σ and $\phi_{,n} := \nabla \phi \cdot \boldsymbol{n}$.

We set up a finite differences scheme to discretize the differential problem (2). For this purpose, consider a cartesian grid over a squared domain centered in the origin. We define the family of nodes $\{\boldsymbol{x}_{i,j}\}$ with $i = -N, ..., N, j = -N, ..., N$ and a step h such that $\boldsymbol{x}_{i,j} = hi\hat{\boldsymbol{x}} + hj\hat{\boldsymbol{y}}$. Over a generic node, we define the discretized function $\phi_{ij} = \phi(\boldsymbol{x}_{i,j})$. In order to rewrite our problem in form of a linear system with the values $\phi_{i,j}$ as the unknowns, we use a 2nd order centered finite difference scheme to discretize the first line of (2) at each node of the grid:

$$\Delta(\phi(\boldsymbol{x}_{i,j})) \approx \frac{1}{h^2}(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{i,j}) = 0 \qquad \forall\ (i,j). \tag{4}$$

This choice of discretization is fine as long as we suppose that ϕ is regular around $\boldsymbol{x}_{i,j}$. In the present example, this condition is lost when a node has a neighbour that falls on the opposite side of Σ, as ϕ is supposed to vary sharply along the edge conneting the two. The GFM scheme consists in a correction of the discretization (4) which explicitly accounts for the jump conditions in (2). To have another applied example, we make reference to [1], where the method is used in the context of compressible multiphase fluidodynamics.
To fix the ideas, let's consider the situation depicted in figure 1. Discretizing the derivative along $\hat{\boldsymbol{x}}$ at node (i,j) we get:

$$\phi_{,xx} \approx \frac{1}{h^2}(\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}). \tag{5}$$

Since the node $(i+1,j)$ falls on the other side of the interface Σ, using this differencing scheme would lead to a sensible error as a strong variation is expected between consecutive nodes. To face this problem, we can consider a field ϕ^- that restricts ϕ onto Ω^-. We can think of prolonging with continuity ϕ^- over Ω^+. This idea can be repeated to define ϕ^+ and extend it over Ω^-. We can also think of extending $[\![\phi]\!]$ from Σ to the surrounding space by setting:

$$[\![\phi]\!](\boldsymbol{x}) := \phi^+(\boldsymbol{x}) - \phi^-(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \ \in \Omega. \tag{6}$$

With this framework we can write:

$$\begin{aligned} \phi_{,xx} &\approx \frac{1}{h^2}(\phi^-_{i+1,j} - 2\phi^-_{i,j} + \phi^-_{i-1,j}) \\ &= \frac{1}{h^2}(\phi^+_{i+1,j} - [\![\phi]\!](\boldsymbol{x}_{i+1,j}) - 2\phi^-_{i,j} + \phi^-_{i-1,j}) \qquad . \\ &= \frac{1}{h^2}(\phi^+_{i+1,j} - 2\phi^-_{i,j} + \phi^-_{i-1,j}) - \frac{1}{h^2}[\![\phi]\!](\boldsymbol{x}_{i+1,j}) \end{aligned} \tag{7}$$

Since we only know $[\![\phi]\!]$ at Σ we can get an estimate of $[\![\phi]\!](\boldsymbol{x}_{i+1,j})$ by Taylor expansion:

$$\begin{aligned} [\![\phi]\!](\boldsymbol{x}_{i+1,j}) &= V_m(\tilde{\boldsymbol{x}}) + (\boldsymbol{x}_{i+1,j} - \tilde{\boldsymbol{x}}) \cdot [\![\nabla\phi]\!](\tilde{\boldsymbol{x}}) \\ &= V_m(\tilde{\boldsymbol{x}}) + (\boldsymbol{x}_{i+1,j} - \tilde{\boldsymbol{x}}) \cdot \boldsymbol{n}(\tilde{\boldsymbol{x}})[\![\phi_{,n}]\!](\tilde{\boldsymbol{x}}). \end{aligned} \tag{8}$$

At this point, we exploit the third line of (2) to write:

$$[\![\phi_{,n}]\!](\tilde{\boldsymbol{x}}) = -\frac{[\![\sigma]\!]}{\sigma^+}\phi^-{}_{,n} = (r^{-1} - 1)\phi^-{}_{,n}\,, \tag{9}$$

where we introduce the ratio $r = \sigma^+/\sigma^-$. This equality allows to describe the jump of the normal derivative of ϕ in terms of the derivative from the interior side only. In order to approximate $\phi^-{}_{,n}$ we choose a neighbour of (i,j) called *pivot* such that it is found on the same region of (i,j). In the example provided, the chosen pivot is $(i, j+1)$. We have:

$$
\begin{aligned}
\phi^{-}{}_{,n} &= \nabla\phi^{-} \cdot \boldsymbol{n} \\
&= (\phi^{-}{}_{,x}\, \hat{\boldsymbol{x}} + \phi^{-}{}_{,y}\, \hat{\boldsymbol{y}}) \cdot \boldsymbol{n} \\
&= \phi^{-}{}_{,x} \cos(\theta) + \phi^{-}{}_{,y} \sin(\theta) \\
&\approx \frac{1}{h}[(\phi^{-}_{i+1,j} - \phi^{-}_{i,j}) \cos(\theta) + (\phi^{-}{}_{,i+1,j} - \phi^{-}{}_{,i,j}) \sin(\theta)] \\
&= \frac{1}{h}[(\phi^{+}_{i+1,j} - [\![\phi]\!]_{i+1,j} - \phi^{-}_{i,j}) \cos(\theta) + (\phi^{-}{}_{,i+1,j} - \phi^{-}{}_{,i,j}) \sin(\theta)]
\end{aligned} \tag{10}
$$

By substituting in (8) the result obtained in (10) we get an equation for $[\![\phi]\!]_{i+1,j}$ involving only the unknowns of the linear system. By defining:

$$
\chi = \frac{(r^{-1} - 1)(\boldsymbol{x}_{i+1,j} - \tilde{\boldsymbol{x}}) \cdot \boldsymbol{n}(\tilde{\boldsymbol{x}})}{h} \tag{11}
$$

we get:

$$
[\![\phi]\!]_{i+1,j} = \frac{V_m(\tilde{\boldsymbol{x}})}{1+\chi} + \frac{\chi}{1+\chi}(\cos(\theta)\phi_{i+1,j} - (\cos\theta + \sin\theta)\phi_{i,j} + \sin\theta\phi_{i,j+1}). \tag{12}
$$

After making all the substitutions we end up with a correction for the discrete equation to be applied in proximity of Σ:

$$
\begin{aligned}
0 = \Delta(\phi(\boldsymbol{x}_{i,j})) &\approx \frac{1}{h^2}(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{i,j}) \\
&- \frac{1}{h^2}\left(\frac{V_m(\tilde{\boldsymbol{x}})}{1+\chi} + \frac{\chi}{1+\chi}(\cos(\theta)\phi_{i+1,j} - (\cos\theta + \sin\theta)\phi_{i,j} + \sin\theta\phi_{i,j+1})\right).
\end{aligned} \tag{13}
$$

2 Implementation Outline

The application of the GFM correction requires a preliminary implementation step. The purpose of it is to identify the edges of the cartesian grid which are intersected by the interface Σ. The tracking of the interface is realized sharply and explicitly. In our application, Σ is discretized as a collection of connected edges. With this in mind, We can consider every pair of the type (surface segments, grid edge) and check for an intersection cryterion. Whenever this cryterion is met, we store the properties which are necessary to perform the correction. We need to register the intersection position and the angle the interface forms with the cut edge. We also mark interface nodes with a signed flag according to the side of the node. In second instance, we loop over the cut edges, and for each of the two ends of of it, we look for a pivot node, such that is found on the same side. A schematic presentation of the algorithm is presented in the pseudocode 1.

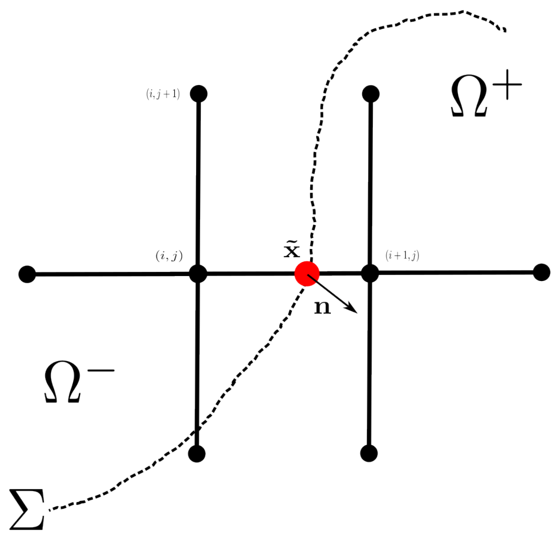

Figure 1: A node (i,j) which is close to the interface Σ. Its neighbour $(i+1,j)$ falls on the opposite side, whereas $(i,j+1)$ is selected as pivot.

Once this is done, we proceed by assembling the linear system. We can start by setting up the well-known discrete Laplacian matrix of the finite difference scheme. Recalling that the i-th line of the system is different from 0 only at the so-called 5-point stencil of node i, a sparse representation of the matrix is allowed. In a second time, we can loop over the nodes which have been flagged as interface points. For each of those, 4 corrections must be operated. The stencil coefficients must be corrected for the central node, the node over the interface, and the pivot. Also, the right hand side of the system must be accomodated. Please, notice that $\chi = 0$ when σ is constant over all Ω, so that the matrix correction is activated only when the ratio $r \neq 1$. On the other hand, the right hand side has to be corrected only when $V_m \neq 0$.

References

[1] Olivier Desjardins, Vincent Moureau, and Heinz Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of computational physics*, 227(18):8395–8416, 2008.

Algorithm 1 Interface tracking algorithm

```
for i ← 1,no_edge_interface do                              ▷ Intersection Recognition
    for j ← 1,no_edge_domain do
        if (check_intersection (i, j) == true) then
            Mark ends of j;
            Register intersection position for edge i;
            Register intersection angle for edge i;
        end if
    end for
end for
for (i ← 1,no_edge_domain) do                                       ▷ Pivot Recognition
    if (is_cut (i) ==true) then
        for j ← 1,no_edge_domain do
            if (share_node (i, j) == true) then
                flag ← shared_node_int_or_ext (i, j)
                if (check_if_other_node_on_same_side (i, j) ==true) then
                    Register_pivot (i, flag, j);
                end if
            end if
        end for
    end if
end for
```