

A PROJECTED GHOST FLUID METHOD FOR A MIMETIC APPROACH FOR EXTREME CONTRAST INTERFACES IN MULTIPHASE FLOWS

N. Valle^{1*}, F.X. Trias¹, J. Castro¹ and A. Oliva¹

¹ Heat and Mass Transfer Technological Centre (CTTC), Universitat Politècnica de Catalunya - BarcelonaTech (UPC), ESEIAAT, Carrer Colom 11, 08222 Terrassa (Barcelona), nicolas.valle@upc.edu, www.cttc.upc.edu

Key words: Mimetic Methods, Multiphase Flow, Computational Methods

Abstract. In the context of a multimaterial domain, the GFM for the Laplacian operator is reformulated from an algebraic perspective, where operators are represented by matrices. From this perspective, a whole new level of analysis can be attained. Namely, conceiving mass, heat and momentum transfer applications, the discretization of both value and flow discontinuities is discussed, with particular emphasis on extreme contrast interfaces, such as those arising in liquid-vapor mixtures.

Regarding flow discontinuities, the GFM is shown to result in an off-image linear system of equations for the Poisson problem. In order to fix this issue, the Projected Ghost Fluid Method (PGFM) takes a global approach to the discretization strategy in order to correct the standard GFM and thus guarantee a consistent discretization.

Apart from being mathematically neat and resulting in a compatible linear system of equations, the PGFM shows enhanced robustness in dealing with extreme contrast interfaces. A comparison study is presented for several coefficient ratios.

1 INTRODUCTION

The use of Mimetic Finite Difference Methods (MFD) [1] provides a mathematically consistent approach for computational physics. Independently, symmetry-preserving schemes [2, 3] establish a set of conditions that need to be satisfied in order to preserve several physical properties in the simulation of turbulent flows. Both can be seen as different sides of the same coin, whereas MFD takes a mathematical approach, symmetry-preserving schemes take a physical one. Inspired in these two approaches, we try to elucidate how these ideas can be applied to the simulation of multiphase flows, in particular when extreme density ratios are present.

The Ghost Fluid Method (GFM) has been successfully implemented in the framework of finite difference methods [4, 5] and gained popularity for sharply capturing multiphase flows. In particular, the seminal work of Kang et al. [6] extended this method for multiphase incompressible flows. The method has been successfully used to incorporate the pressure jump arising due to surface tension in the Poisson equation of the Fractional Step

Method [7]. This has been used in atomization [8], vaporization [9], oil-water flow [10] and electrohydrodynamics simulations [11], among many others. Including discontinuities in the viscous term, however, remains still a challenge in terms of complexity and suitability for implicit temporal integration schemes [8].

Nonetheless, the use of the GFM can lead to instabilities when both flux discontinuities and extreme contrast interfaces are present. These arise from an incompatible system of equations which may be remedied by projecting the resulting discretization. The results and suitability of such a method under these conditions are discussed.

The rest of the paper is organized as follows. In Section 2 an introduction to the standard method is presented, along with a proposal of improvement. In Section 3 results and comparison between the GFM and the PGFM are presented. Finally, Section 4 highlights the advantages of the method and analyzes its potentiality.

2 METHOD

2.1 Mimetic operators

Assuming a partition of unity given by an arbitrary mesh, as the one in Figure 1, the relation between faces and cells can be cast in an incidence matrix as follows:

$$T_{\mathcal{FC}} = \begin{matrix} & f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 & f_9 \\ \begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{matrix} & \begin{pmatrix} 0 & 0 & -1 & +1 & +1 & 0 & 0 & 0 & 0 \\ -1 & -1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & 0 & 0 & 0 & 0 & -1 & +1 & 0 & 0 \\ 0 & +1 & 0 & 0 & 0 & 0 & 0 & +1 & +1 \end{pmatrix} \end{matrix} \quad (1)$$

Where $T_{\mathcal{FC}}$ represents the oriented connection between faces and cells, while $T_{\mathcal{CF}} = T_{\mathcal{FC}}^T$ is the incidence matrix representing the connection between faces and their adjacent cells. This matrix will serve as the basis for the development of the forthcoming numerical methods. From a classical stencil-based perspective, it corresponds with the connection established between faces and cells in nested *for* loops, e.g., the oriented sum of face fluxes \mathbf{u}_f for every cell may be seen as the matrix-vector product $T_{\mathcal{FC}}\mathbf{u}_f$. The use of the incidence matrix allows us to remain in an algebraic perspective and analyze, easily, forthcoming properties.

By defining the divergence as the primal operator at every mesh cell [1], we get:

$$\int_{\Omega} \nabla \cdot \mathbf{u} = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} \approx -T_{\mathcal{FC}}S_{\mathcal{F}}\mathbf{u}_f = M_{\mathcal{C}}D_{\mathcal{C}}\mathbf{u}_f \quad (2)$$

Where the integral is taken over the control volume Ω (i.e., over surface in 2D or volume in 3D). Then, by means of the Gauss-Ostrogradsky's theorem, moved to the fluxes at the boundaries $\partial\Omega$ (i.e., line or surface integral, correspondingly). $D_{\mathcal{C}}$ is the discrete divergence operator, while $M_{\mathcal{C}}$ and $S_{\mathcal{F}}$ correspond with the size the control volumes and the face surfaces, respectively. Both are arranged as diagonal matrices. Conversely, we can define the control volumes at faces as $M_{\mathcal{F}}$, which in this case corresponds with the classical staggered control volume.

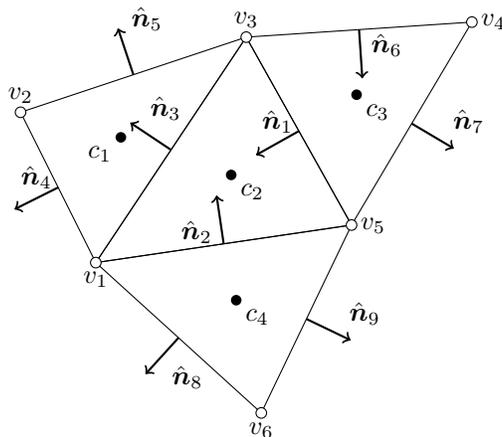


Figure 1: Mesh \mathcal{M} . c_i corresponds with the i th cell, $\hat{\mathbf{n}}_j$ corresponds with the normal vector to the j th face (i.e., f_j) and v_k corresponds with the k th vertex. Note that in this 2D mesh faces and edges collapse in the same entity.

$$M_{\mathcal{F}} = \Delta_X S_{\mathcal{F}} \quad (3)$$

Where Δ_X is a diagonal matrix which corresponds with the distance between cell centers. From a geometric perspective, it can be seen as the extrusion of the face surface between its two adjacent cell centers.

Next, following Verstappen and Veldmann [3], we impose the following duality condition in order to preserve kinetic energy.

$$\langle \mathbf{v}_f, G_{\mathcal{F}} \mathbf{u}_c \rangle_{\mathcal{F}} = - \langle D_C \mathbf{v}_f, \mathbf{u}_c \rangle_C \quad (4)$$

This is equivalent to the duality condition used in Lipnikov et al. [1] to define the derived operator. In our discrete approach, we define the dot product $\langle \cdot, \cdot \rangle_{\mathcal{Q}}$ for any space \mathcal{Q} (i.e., \mathcal{F} , \mathcal{C}) as follows:

$$\langle \mathbf{v}_q, \mathbf{u}_q \rangle = \mathbf{v}_q^T M_{\mathcal{Q}} \mathbf{u}_q \quad (5)$$

Where both M_C and $M_{\mathcal{F}}$ are used again. This provides a consistent definition of the gradient $G_{\mathcal{F}}$ from the definition of the primal divergence operator and the staggered control volume and results in:

$$G_{\mathcal{F}} = \Delta_X^{-1} T_{\mathcal{C}\mathcal{F}} \quad (6)$$

Which is nothing but the classical form of the gradient located at the face in a finite difference scheme.

The concatenation of D_C and $G_{\mathcal{F}}$ results in a proper discretization of the Laplacian operator. However, we will be interested in the integral form of such operator, which preserves symmetry:

$$L_C = M_C D_C G_{\mathcal{F}} = -T_{\mathcal{F}C} S_{\mathcal{F}} \Delta_X^{-1} T_{\mathcal{F}C} \quad (7)$$

2.2 Ghost Fluid Method

The regular GFM [12] will be redefined from a fully algebraic point of view. The method arises in trying to solve the discontinuous Poisson equation for a scalar u and arbitrary jump conditions at the interface Γ in its value, a , and its derivative in the interface normal direction \hat{n}_Γ , b .

$$\nabla \cdot \lambda \nabla u = f \quad (8a)$$

$$[u]_\Gamma = a \quad (8b)$$

$$[\lambda \nabla u]_\Gamma \cdot \hat{n}_\Gamma = b \quad (8c)$$

Being originally a finite difference method, the GFM directly imposes the interfacial jump conditions by modifying the gradient operator. An illustrative example can be seen in Figure 2.

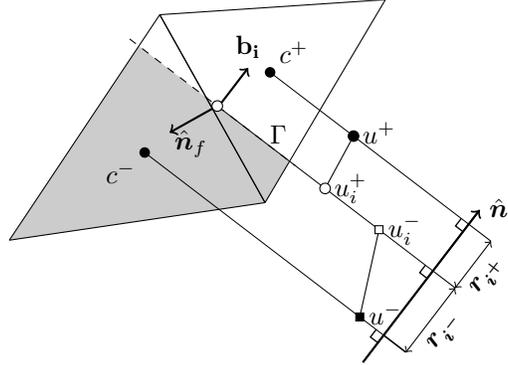


Figure 2: Variable u reconstruction in the interface-normal direction. Filled symbols correspond with actual variables while empty ones correspond with interface values.

With the help of the interfacial values \mathbf{u}_{i+} and \mathbf{u}_{i-} , the value of the gradients in the interface-normal direction at both sides (i.e., $(\nabla \mathbf{u})_{i+} \cdot \hat{\mathbf{n}}_i$ and $(\nabla \mathbf{u})_{i-} \cdot \hat{\mathbf{n}}_i$) are forced to satisfy the following 2nd order Taylor expansion:

$$\mathbf{u}_{i+} + \mathbf{r}_{i+} (\nabla \mathbf{u})_{i+} \cdot \hat{\mathbf{n}}_i \approx \mathbf{u}^+ \quad (9a)$$

$$\mathbf{u}_{i-} + \mathbf{r}_{i-} (\nabla \mathbf{u})_{i-} \cdot \hat{\mathbf{n}}_i \approx \mathbf{u}^- \quad (9b)$$

Where \mathbf{r}_{i+} and \mathbf{r}_{i-} are the signed distances from the cell center to the interface, a readily available value in a Level-Set formulation, whereas \mathbf{u}^+ and \mathbf{u}^- corresponds with the mesh values of \mathbf{u}_c segregated for the positive or negative phase.

In order to re-write this system of equations in matrix terms, the interface oriented incidence matrix, T_{CI} , is required. In turn, T_{CI} is obtained from the face-oriented incidence matrix as follows:

$$T_{CI} = QT_{CF} \quad (10)$$

Where Q corresponds with a $\mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ diagonal matrix (being $|\mathcal{F}|$ the number of faces) that corrects the proper orientation of the interface relative to the face. It contains $+1$ when the interface orientation is already coincident with the face, -1 when it is opposite and 0 when there is no interface related to that face. By splitting $T_{\mathcal{CI}}$ into its positive and negative coefficients as $T_{\mathcal{CI}} = T_{\mathcal{CI}^+} - T_{\mathcal{CI}^-}$, \mathbf{u}^+ and \mathbf{u}^- can be obtained as $\mathbf{u}^+ = T_{\mathcal{CI}^+}\mathbf{u}_c$ and $\mathbf{u}^- = T_{\mathcal{CI}^-}\mathbf{u}_c$.

Finally, equation (9) can be cast in the matrix form:

$$\mathbf{u}_{i\pm} + R_{\mathcal{I}\pm}(\nabla\mathbf{u})_{i\pm} \cdot \hat{\mathbf{n}}_i \approx T_{\mathcal{CI}\pm}\mathbf{u}_c \quad (11)$$

Where $R_{\mathcal{I}\pm}$ is the diagonal matrix arrangement of $\mathbf{r}_{i\pm}$.

Once the gradients have been defined for both sides of the interface, we can finally incorporate the jump conditions from equation (8) to close the linear system of equations:

$$\mathbf{u}_{i+} - \mathbf{u}_{i-} = \mathbf{a}_i \quad (12a)$$

$$((\lambda\nabla\mathbf{u})_{i+} - (\lambda\nabla\mathbf{u})_{i-}) \cdot \hat{\mathbf{n}}_i = \mathbf{b}_i \quad (12b)$$

Where the jump values \mathbf{a}_i and \mathbf{b}_i need to be interpolated from cell centers to the interface by means of the homothetic operator $H_{\mathcal{CI}}$, which is defined as:

$$H_{\mathcal{CI}} = (\Delta_R)^{-1} (R_{\mathcal{I}^+}T_{\mathcal{CI}^-} - R_{\mathcal{I}^-}T_{\mathcal{CI}^+}) \quad (13)$$

Where Δ_R is just the total distance between cell centers across the interface and can be computed as $\Delta_R = R_{\mathcal{I}^+} - R_{\mathcal{I}^-}$. Keep in mind that $R_{\mathcal{I}^-}$ is a negative quantity and thus needs to be subtracted in order to provide the total distance.

By solving the linear system of equations (11) and (12), and after rearranging, we obtain the following form of the flux at the interface:

$$(\lambda\nabla\mathbf{u})_{i\pm} \cdot \hat{\mathbf{n}}_i \approx (\lambda\nabla\mathbf{u})_{\mathcal{I}\pm} = \hat{\Lambda}_{\mathcal{I}}R_{\mathcal{I}\pm}^{-1}(T_{\mathcal{CI}\pm}\mathbf{u}_c - \mathbf{a}_i) - \Lambda_{\mathcal{I}^-}^{-1}\Lambda_{\mathcal{I}^+}(\Delta_R)^{-1}R_{\mathcal{I}^+}\mathbf{b}_i \quad (14)$$

Where $\Lambda_{\mathcal{I}}$ is the homothetic interpolation of λ arranged in an $\mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ diagonal matrix. On the other hand, the harmonic mean of λ , $\hat{\Lambda}_{\mathcal{I}} = \Lambda_{\mathcal{I}^+}\Lambda_{\mathcal{I}^-}\Lambda_{\mathcal{I}}^{-1}$ represents the diagonal matrix arrangement of the harmonic mean at the interface.

At this point is worth noticing that the construction of the gradient operator is not derived from a divergence as a primal operator but from a finite difference approach. Even when we could, eventually, derive a phase centered divergence operator from the gradient, this is out of the scope of this work.

Once the gradient at the interface has been successfully computed, the discretization of the Laplacian operator in equation (7) requires the gradient operator to be defined at the faces of the cell rather than at the interface.

In order to achieve this, and following Liu et al. [12], we will reconstruct the full gradient assuming no interfacial tangent contribution.

$$(\lambda\nabla\mathbf{u})_{\mathcal{F}\pm} = (\Delta_X)(\Delta_R)^{-1}Q(\lambda\nabla\mathbf{u})_{\mathcal{I}\pm} = \cos^{-1}(\Phi)(\lambda\nabla\mathbf{u})_{\mathcal{I}\pm} \quad (15)$$

Where $\cos^{-1}(\Phi)$ corresponds to the diagonal matrix arrangement of the arccosine of Φ , which is the angle between interface and face normals.

This produces the following form:

$$(\lambda \nabla u)_{\mathcal{F}\pm} = \hat{\Lambda}_{\mathcal{I}}(G_{\mathcal{F}}\mathbf{u}_c - Q(\Delta_X)^{-1}\mathbf{a}_i) - \Lambda_{\mathcal{I}}^{-1}\Lambda_{\mathcal{I}\pm}(\Delta_X)^{-1}R_{\mathcal{I}\mp}\mathbf{b}_i \quad (16)$$

Which can, effectively, be used to produce a proper (integrated) Laplacian operator discretization as the final form:

$$\begin{aligned} \nabla \cdot \lambda \nabla u \approx & L_C \mathbf{u}_c - D_C \hat{\Lambda}_{\mathcal{I}} I(\Delta_X)^{-1} \mathbf{a}_i \\ & - V_C^{-1} \Lambda_C H_{\mathcal{C}\mathcal{I}}^T(\Delta_R)(\Delta_X)^{-1} S_{\mathcal{F}} \Lambda_{\mathcal{I}}^{-1} \mathbf{b}_i \end{aligned} \quad (17)$$

We note that this is a slightly different approach to the one taken in [12], where the jump condition is aligned with the Cartesian coordinates before proceeding with the discretization, whereas in this approach the jump is captured in the interfacial direction and then moved to the face in order to proceed with the finite volume discretization. As far as this operation is linear, the result is the same.

2.3 Projected Ghost Fluid Method

Even when the GFM presents a sharp method that locally reflects the jump boundary conditions, the presence of flow discontinuities under extreme contrast interfaces can produce instabilities and, eventually, prevent it from convergence.

Contrary to what is stated in [12], in a typical discontinuous Poisson system of equations convergence may not be guaranteed if Neumann boundary conditions are posed.

Note that in this particular setup, the following condition must hold in order to achieve a solution:

$$\int_{\Omega} \nabla \cdot \lambda \nabla u = \int_{\partial\Omega} (\lambda \nabla u) \cdot \hat{\mathbf{n}}_f + \int_{\Gamma} b = \int_{\Omega} f \quad (18)$$

Which implies that, in the absence of boundary contributions, the surface integral of the flow discontinuity must be compensated by the source term. On the other hand, when Dirichlet boundary conditions are posed, the boundary terms does not vanish anymore, although the global balance needs to be satisfied as well. In that situation, the values next to the boundary will adapt such that they provide the appropriate gradient. However, there is no way to separate the contribution to the boundary flow of the discretization errors from the actual posing of the system.

However, even when condition given in equation (18) may be satisfied at the continuous level, the GFM discretization may not. In particular, the resulting discretization of the Laplace equation may produce an incompatible system of equations.

By moving to the right the terms corresponding to the discontinuities in equation 17, we obtain the non-homogeneous linear system of equations:

$$L_C \mathbf{u}_c = \mathbf{s}_a + \mathbf{s}_b + \mathbf{s}_f \quad (19)$$

Where $\mathbf{s}_a = D_C \hat{\Lambda}_I I(\Delta_X)^{-1} \mathbf{a}_i$, $\mathbf{s}_b = V_C^{-1} \Lambda_C H_C^T (\Delta_R) (\Delta_X)^{-1} S_{\mathcal{F}} \Lambda_{\mathcal{I}}^{-1} \mathbf{b}_i$ and $\mathbf{s}_f = f$. In order to guarantee that the system of equations lies in the image of the operator L_C , we may ensure that $\langle \mathbb{1}_C, \mathbf{s}_a + \mathbf{s}_b + \mathbf{s}_c \rangle_C = 0$. This can be seen as to check whether or not their projection over the kernel space is 0 or not for a symmetric, positive-definite, simple connected Laplacian operator [13].

It can be readily seen that because of the form of \mathbf{s}_a , this term already belong to the image of L_C , as it presents a divergence form D_C . Unfortunately, there is no guarantee that the sum of $\langle \mathbb{1}_C, \mathbf{s}_b + \mathbf{s}_f \rangle_C$ will cancel out as well. Indeed, not even when $f = 0$, as in the discontinuous Laplace problem, it is guaranteed that \mathbf{s}_b belongs to the image of the operator.

The reason behind such an imbalance lies in the step that moved the gradient at the interface to the gradient at the face, which was conceived to preserve local jump conditions but not to preserve the global balance.

The simplest solution may be to correct such an imbalance by projecting the resulting discretization to the image of the Laplacian operator. This correction obviously produces a disturbance in the local jump conditions, although the overall results are shown to improve convergence when extreme contrast interfaces are present. Because this is a global correction method, the way of correcting the equation is not unique. In this work we propose to correct the imbalance as follows:

$$L_C \mathbf{u}_c = \mathbf{s}_a + \mathbf{s}_b + \mathbf{s}_f - \frac{\langle \mathbb{1}_J, \mathbf{s}_b + \mathbf{s}_f \rangle_C}{\langle \mathbb{1}_C, \mathbb{1}_C \rangle_C} \mathbb{1}_C \quad (20)$$

It is worth noticing that this condition provides with an appropriate correction only when Neumann boundary conditions are posed.

3 RESULTS

The simulation of realistic physical situations may involve extreme contrast interfaces, which may result not only in a stiff system but in an incompatible one, as it is shown below.

The following are canonical cases that allow for a comparative of both methods when the solution presents a flow discontinuity. Both obey the general form of equation (8) with Neumann boundary conditions and present the following solution.

$$u^+(x, y) = \exp(-x^2 - y^2) \quad (21a)$$

$$u^-(x, y) = 0 \quad (21b)$$

The value and flow jump discontinuities a and b as well as the source term f stated in equation (8) can be readily obtained from the solution.

The problems are solved in a $[-1, 1] \times [-1, 1]$ domain and use a uniform Cartesian mesh. In both cases the linear system of equations is solved with a Conjugate Gradient method preconditioned with an incomplete Cholesky factorization. The residual of all cases is below 10^{-8} .

3.1 Pure flux discontinuity

This case is inspired by a static phase change. Although a practical case may involve a transient term, such a term may mask the imbalance arising from a GFM, and so static conditions have been imposed in order to test a pure flow discontinuity case. Under these conditions, the integral of the source term in the domain must be equal to the interface integral of the flow discontinuity in order to attain a solution, as can be seen in equation (18). This could be seen as including a source term that compensates the energy released by the phase change such that the solution is steady. This is obviously an artifact without physical meaning but useful to assess the performance of our scheme.

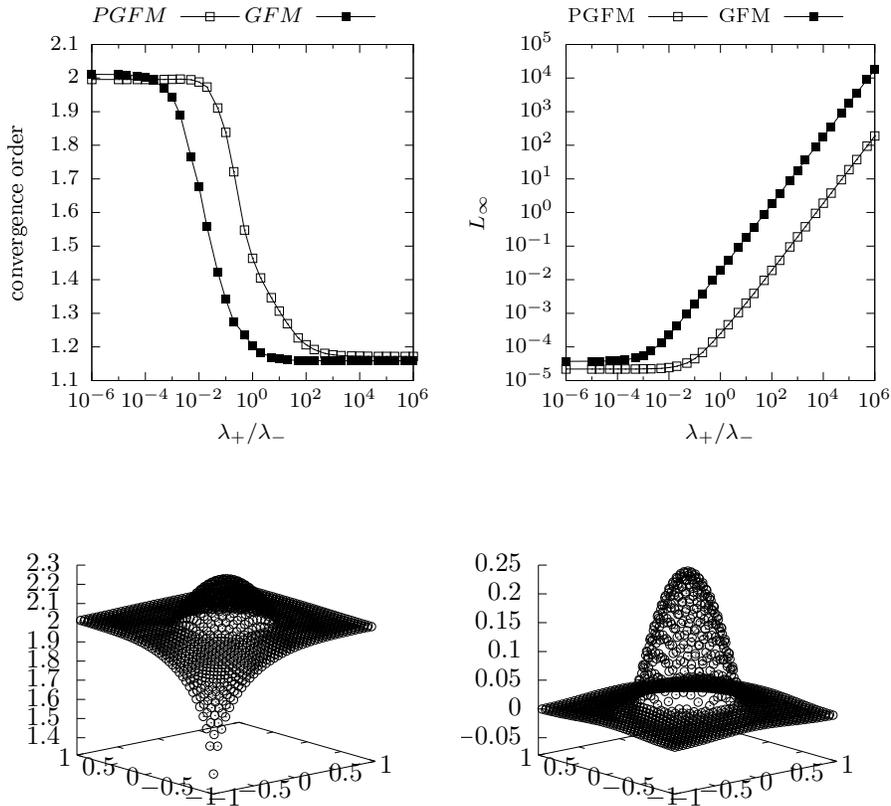


Figure 3: Results for a flux discontinuity case. Top left Figure shows converge in terms of the L_∞ norm of the error, while the top right one shows the error norm. The results in a 40×40 mesh for $\lambda_+/\lambda_- = 50$ obtained with the GFM and the PGFM are shown at the bottom left and right, respectively.

Convergence order collapses to *2nd* order when $\lambda_+/\lambda_- < 1$ for both methods, while it rapidly decreases as λ_-/λ_+ approaches 1. It can be seen as well how the adoption of the correction in the PGFM delays such a degradation.

However, as the top right of Figure 3 suggest, the errors increase dramatically as far as coefficient ratios increase further, providing with uncertain quality solutions.

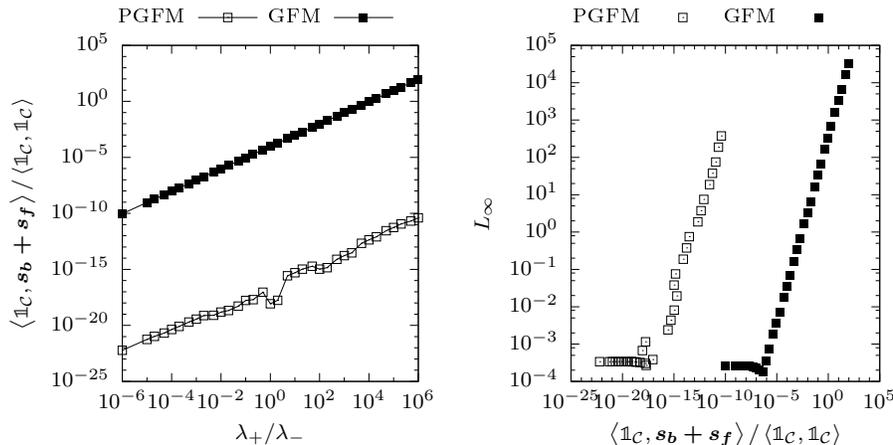


Figure 4: Normalized kernel imbalance vs. coefficient ratio (left) and L_∞ norm of the error vs. normalized kernel imbalance (right) for the pure flux discontinuity case in a 40x40 mesh.

At the bottom of Figure 3 it can be seen how the GFM presents a divergent solution (left) and is obviously not a reliable solution, while the PGFM present a stable one (right). This is an expected result for the inconsistent system of equations resulting from the GFM discretization. On the other hand, the PGFM corrects such an imbalance, providing a consistent discretization of the problem. It is remarkable that, despite the efforts, the solution degenerates rapidly when increasing the density ratio beyond 50, obtaining a stable but imprecise solution.

From Figure 4 it can be seen how the PGFM provides with an almost 0 kernel imbalance even for extreme λ_+ / λ_- values, whereas kernel imbalance seems to delay the increase of the error, it does not guarantee an accurate solution *per se*.

3.2 Face-aligned interface

In this case, the interface is now aligned with faces by switching the previous case to a square interface, which may then fit perfectly with a cartesian mesh, provided that the mesh size is set properly. On the other hand, because the interface will not lie with an iso-value curve of the solution, a value jump must be imposed in order to achieve a solution.

The face-aligned interface presents the advantage that faces and interfaces are perfectly aligned, and so the computation of $\cos(\Phi)$ in equation (15) is trivial. This will provide with an exact discretization at the interface, removing thus any interface-to-face interpolation errors. In addition, because phases will be perfectly contained in all cells of the mesh (i.e., there will be no “mixed” cells), the evaluation of the source term integral f is immediate.

As it can be seen from the top right figure, the alignment of the interface with the faces provides a way more robust method in terms of convergence order, whereas the in top left the error is seen to increase as λ_+ / λ_- increases. Again, the use of the PGFM delays and reduces the order of the error.

Bottom left and bottom right figures show, again, how the GFM will diverge where

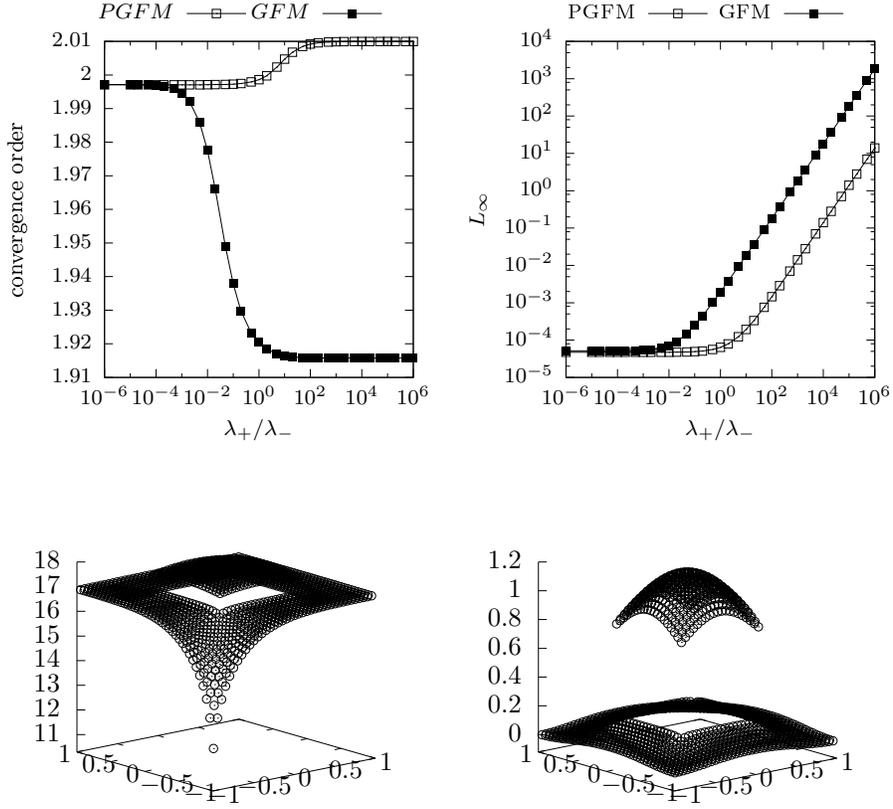


Figure 5: Results for an interface-aligned case. Top left Figure shows the convergence order in terms of the L_∞ norm of the error. Top right show L_∞ norm of the error. Results for a 40x40 mesh at $\lambda_+/\lambda_- = 10^3$ are shown at the bottom left for a conventional GFM (left) and PGFM (right).

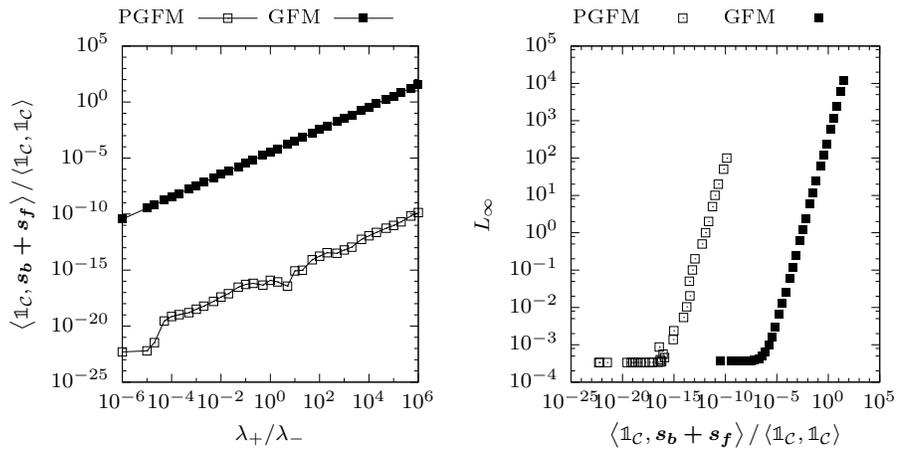


Figure 6: Normalized kernel imbalance vs. coefficient ratio (left) and L_∞ norm of the error vs. normalized kernel imbalance (right) for the face-aligned interface case in a 40x40 mesh.

the PGFM will provide with a converged, yet perturbed, solution.

Even when the perturbation included from the interface-to-face mapping can be removed, it can be seen from Figure 6 that the kernel imbalance is still inaccurate when extreme values of λ_+/λ_- are imposed to the system. This holds for both the GFM and, to a lesser extent, the PGFM as well.

4 CONCLUSIONS AND FUTURE WORK

Despite the ability of the GFM to efficiently handle value discontinuities, even under extreme contrast conditions, the method fails to adequately include flow discontinuities in its discretization. The reasons behind such instability lie in the resulting incompatible linear system of equations. From this perspective, the condition for stability has been highlighted. Imbalances results from both i) the interpolation from interface gradients to face gradients and ii) the approximation of the jump condition from cells to faces.

A global approach has been taken in order to enforce stability of the method in the presence of Neumann boundary conditions. From a purely algebraic reasoning the PGFM has been formulated. The new method has succeed at providing a stable, yet inaccurate, solution.

Results show that, despite the correction does indeed reduce the kernel imbalance, the higher the coefficient ratio, the more sensitive the system becomes. For high coefficient ratios, even a small discretization error of the flux jump condition introduces errors that cannot be removed by projection.

In addition, the inclusion of such a correction produces a distortion of the local discretization, degrading then the global quality of the solution and violating the physical meaning of the problem.

The use of such a global approach has enforced the fulfillment of condition stated in equation (18), however it is valid only when Neumann boundary conditions are posed and still results in a modified equation which may not be accurate. From this perspective, we see this approach more as a measure of merit of the discretization rather than a solution method suitable for all cases.

Future developments of this method are aimed at the development of new correction strategies which may be derived from a local perspective, and physically compliant with the discretization of multiphase flows.

5 ACKNOWLEDGEMENTS

This work has been financially supported by the Ministerio de Economía y Competitividad, Spain (ENE2017-88697-R and ENE2015-70672-P) as well as an FI AGAUR-Generalitat de Catalunya fellowship (2017FI_B_00616) and a Ramón y Cajal postdoctoral contract (RYC-2012-11996). We would like to acknowledge prof. Roel Verstappen for his enriching discussions.

REFERENCES

- [1] K. Lipnikov, G. Manzini, and M. Shashkov, “Mimetic finite difference method,” *J. Comput. Phys.*, vol. 257, no. PB, pp. 1163–1227, 2014.
- [2] F. Trias, O. Lehmkuhl, A. Oliva, C. Pérez-Segarra, and R. Verstappen, “Symmetry-preserving discretization of Navier–Stokes equations on collocated unstructured grids,” *J. Comput. Phys.*, vol. 258, pp. 246–267, feb 2014.
- [3] R. Verstappen and A. Veldman, “Symmetry-preserving discretization of turbulent flow,” *J. Comput. Phys.*, vol. 187, no. 1, pp. 343–368, 2003.
- [4] R. P. Fedkiw and X. D. Liu, “The ghost fluid method for viscous flows,” *Prog. Numer. Solut. Partial Differ. Equations*, pp. 1–33, 1998.
- [5] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, “A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method),” *J. Comput. Phys.*, vol. 152, no. 2, pp. 457–492, 1999.
- [6] M. Kang, R. P. Fedkiw, and X. D. Liu, “A boundary condition capturing method for multiphase incompressible flow,” *J. Sci. Comput.*, vol. 15, no. 3, pp. 323–360, 2000.
- [7] A. J. Chorin, “A Numerical Method for Solving Incompressible Viscous Flow Problems,” *J. Comput. Phys.*, vol. 135, pp. 118–125, 1997.
- [8] O. Desjardins, V. Moureau, and H. Pitsch, “An accurate conservative level set/ghost fluid method for simulating turbulent atomization,” *J. Comput. Phys.*, vol. 227, no. 18, pp. 8395–8416, 2008.
- [9] S. Tanguy, T. Ménard, and A. Berlemont, “A Level Set Method for vaporizing two-phase flows,” *J. Comput. Phys.*, vol. 221, no. 2, pp. 837–853, 2007.
- [10] M. Kang, H. Shim, and S. Osher, “Level set based simulations of two-phase oil-water flows in pipes,” *J. Sci. Comput.*, vol. 31, no. 1-2, pp. 153–184, 2007.
- [11] B. P. Van Poppel, O. Desjardins, and J. W. Daily, “A ghost fluid, level set methodology for simulating multiphase electrohydrodynamic flows with application to liquid fuel injection,” *J. Comput. Phys.*, vol. 229, no. 20, pp. 7977–7996, 2010.
- [12] X.-D. Liu, R. P. Fedkiw, and M. Kang, “A Boundary Condition Capturing Method for Poisson’s Equation on Irregular Domains,” *J. Comput. Phys.*, vol. 160, no. 1, pp. 151–178, 2000.
- [13] N. Valle Marchante, F. X. Trias, J. Castro, and A. Oliva, “Boundary condition capturing method for the discontinuous Laplacian operator under extreme contrast interfaces,” *J. Comput. Phys. (to be submitted)*