An implicit high order discontinuous Galerkin level set method for two-phase flow problems

Mai Ta[1,2], Franck Pigeonneau and Pierre Saramito[2]

[1] *Surface du verre et interfaces, UMR 125 CNRS/Saint-Gobain*
39, quai Lucien Lefranc - BP 135, 93303 Aubervilliers cedex, France
Franck.Pigeonneau@saint-gobain.com,

[2] *Lab. J. Kuntzmann, CNRS and Grenoble university*
51, rue des mathématiques BP 53 - Domaine Universitaire, 38041 Grenoble Cedex 9, France
tathithanhmai@gmail.com, Pierre.Saramito@imag.fr

Abstract

An implicit high order time (BDF) and polynomial degree discontinuous Galerkin (DG) level set method is presented in this talk. The major advantage of this new approach is an accurate mass conservation during the convection of the level set function, thanks to the implicit method. Numerical experiments are presented for the Zalesak and the Leveque test cases. The convergence rates versus time and space are investigated for both BDF and DG high orders. The capture of the zero level set interface is then improved by using an auto-adaptive mesh procedure. The problem is approximated by using the discontinuous Galerkin method for both the level set function, the velocity and the pressure fields.

Keywords: Level set method, discontinuous Galerkin FEM, high order methods

1. Introduction

Let us denote, at any time $t \geqslant 0$ by $\Omega(t) \subset \mathbb{R}^d$, $d = 2, 3$, the bounded moving domain and $\Gamma(t) = \partial\Omega(t)$ its boundary. A level set function ϕ is defined for any time in a bounded computational domain denoted by $\Lambda \subset \mathbb{R}^d$, and containing $\Omega(t)$ at any time, such that $\Gamma(t) = \{\boldsymbol{x} \in \Lambda;\ \phi(t,\boldsymbol{x}) = 0\}$. Since every point belonging to the boundary $\Gamma(t)$ will still continue to belong to it for any time, we have:

$$\frac{\mathrm{d}}{\mathrm{d}t}(\phi(t,\boldsymbol{x})) = 0 \iff \frac{\partial\phi}{\partial t} + \boldsymbol{u}.\nabla\phi = 0 \ \text{ in }]0,+\infty[\times\Lambda \qquad (1)$$

where $\boldsymbol{u} = \dot{\boldsymbol{x}}$ denotes the velocity field. The normal to the boundary $\Gamma(t)$ writes $\boldsymbol{\nu} = \nabla\phi/|\nabla\phi|$ and then deformations of $\Gamma(t)$ are only due to the normal component $\boldsymbol{u}\cdot\boldsymbol{\nu}$ of the velocity on $\Gamma(t)$. For a given velocity field $\boldsymbol{u}$, the problem is to find ϕ defined in $]0,+\infty[\times\Lambda$ and satisfying (1) together with an initial condition $\phi(t\!=\!0) = \phi_0$ where ϕ_0 is given. This is a linear *hyperbolic problem*. The level set method was introduced in 1988 by Osher and Sethian [8] (see also [12]). Notice that several choices are possible for the function ϕ: the only requirement being that a fixed isocontour of ϕ coincides with the front at each time t. A common choice is the signed distance from the front: e.g. $\Omega(t)$ is the part where $\phi(t,.)$ is negative and $\phi(t,\boldsymbol{x}) = -\mathrm{dist}(\Gamma(t),\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega(t)$.

Observe also that, assuming a *divergence free* velocity field $\boldsymbol{u}$, the volume of $\Omega(t)$ remains constant at any time. This property is not strictly maintained after discretization by most numerical methods suitable for hyperbolic problems and this problem is often referred as a *mass loss*, as the mass conservation of the fluid is violated. Several cures to this problem have already proposed. In 1999, Sussman and Fatemi [14] proposed, in the context of a finite difference method, to add a constraint in order to improve the mass conservation. This idea was extended in 2010 by Laadhari *et al* [4] to the finite element context (see also [5]). In 2006, Di Pietro *et al.* [2] and, independently Marchandise *et al.* [7] proposed to discretize the level set equation (1) by using a discontinuous Galerkin finite element method. Note that the discontinuous Galerkin method applies also to a closely related fluid interface problem, the Cahn-Hilliard equations (see e.g. [9]).

In this paper, we revisit this second approach and propose some improvements that dramatically decrease the mass error.

2. Numerical method

2.1. Discontinuous space approximation

Let $\mathscr{T}_h$ be a triangulation of the computational domain Λ. We introduce the following finite dimensional space:

$$X_h = \{\varphi_h \in L^2(\Lambda);\ \varphi_{h|K} \in P_k,\ \forall K \in \mathscr{T}_h\}$$

where $k \geqslant 0$ is the polynomial degree. The variational formulation of the semi-discretized problem writes (see e.g. [11, p. 8] or [1, p. 57]):

$(FV)_h$: *find* ϕ_h *such that* $\phi_h(t) \in X_h$, $\forall t > 0$ *and*

$$\int_\Lambda \frac{\partial\phi_h}{\partial t}\varphi_h\,\mathrm{d}x + \int_{\partial\Lambda}\max(0,-\boldsymbol{u}.\boldsymbol{n})\,\phi_h\,\varphi_h\,\mathrm{d}s + \sum_{K\in\mathscr{T}_h}\int_K \boldsymbol{u}.\nabla\phi_h\,\varphi_h\,\mathrm{d}x + \sum_{S\in\mathscr{S}_h^{(i)}}\int_S [\![\phi_h]\!]\left(\frac{|\boldsymbol{u}.\boldsymbol{n}|}{2}[\![\varphi_h]\!] - \boldsymbol{u}.\boldsymbol{n}\,\{\!\{\varphi_h\}\!\}\right)\mathrm{d}s = 0,$$
$$\forall\varphi_h \in X_h \qquad (2)$$

together with the initial condition $\phi_h(0) = \pi_h(\phi_0)$ where π_h denotes the Lagrange interpolation operator on X_h.

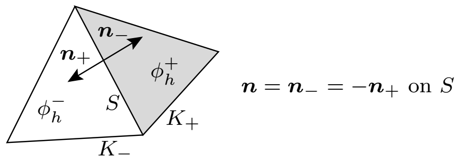


Figure 1: Notations for the discontinuous Galerkin method.

This formulation introduces some notations that are usual in the context of the discontinuous Galerkin method. The second term involves the outer unit normal $\boldsymbol{n}$ on the boundary $\partial\Lambda$ of the computational domain. The last term involves a sum over $\mathscr{S}_h^{(i)}$, the set of *internal sides* of the mesh $\mathcal{T}_h$. Each internal side $S \in \mathscr{S}_h^{(i)}$ has two possible orientations: one is chosen definitively. On this internal side, let $\mathbf{n}$ denotes the normal to the oriented side S: as S is an internal side, there exists two elements K_- and K_+ such that $S = \partial K_- \cap \partial K_+$ and $\mathbf{n}$ is the outward unit normal of K_- on $\partial K_- \cap S$ and the inward unit normal of K_+ on $\partial K_+ \cap S$, as shown on Fig. 1. For all $\phi_h \in X_h$, recall that ϕ_h is in general discontinuous across the internal side S. We define on S the *inner value* $\phi_h^- = \phi_{h|K_-}$ of ϕ_h as the restriction $\phi_{h|K_-}$ of ϕ_h in K_- along $\partial K_- \cap S$. Conversely, we define the *outer value* $\phi_h^+ = \phi_{h|K_+}$. We also denote on S the *jump* $[\![\phi_h]\!] = \phi_h^- - \phi_h^+$ and the *average* $\{\!\{\phi_h\}\!\} = (\phi_h^- + \phi_h^+)/2$. The case $k = 0$, i.e. a piecewise constant approximation, coincides with the popular upwinding finite volume scheme.

2.2. *Time discretization by an implicit scheme*

The final discrete problem is obtained from the semi-discrete one (2) by using a time discretization scheme. For instance, Di Pietro *et al.* [2] and Marchandise *et al.* [7] used an *explicit* Runge-Kutta (RK) method [3]. Here, we propose to use a high order *implicit* scheme based on the following backward differentiation formula (BDF):

$$\frac{\partial \phi}{\partial t} = \frac{1}{\Delta t} \sum_{k=0}^{p} \alpha_k \phi(t - k\Delta t) + \mathcal{O}\left(\Delta t^p\right)$$

where $\Delta t > 0$ is the time step, $p \geqslant 1$ is the order of the scheme and $(\alpha_k)_{0 \leqslant k \leqslant p}$ are the $p+1$ coefficients of the formula. When $p = 1$, the scheme coincides with the usual backward Euler method. For $1 \leqslant p \leqslant 2$, the scheme is inconditionally stable, for $3 \leqslant p \leqslant 6$, the scheme is almost inconditionally stable while when $p > 6$, the scheme is unstable and cannot be used (see e.g. [13, p. 349]). The coefficients when $p \leqslant 6$ are given in Table 1.

p	α_0	α_1	α_2	α_3	α_4	α_5	α_6
1	1	-1					
2	$3/2$	-2	$1/2$				
3	$11/6$	-3	$3/2$	$-1/3$			
4	$25/12$	-4	3	$-4/3$	$1/4$		
5	$137/60$	-5	5	$-10/3$	$5/4$	$-1/5$	
6	$147/60$	-6	$15/2$	$-20/3$	$15/4$	$-6/5$	$1/6$

Table 1: Coefficients of the BDF(p) schemes, $1 \leqslant p \leqslant 6$.

Let us introduce the following bilinear form, defined for all $\phi, \varphi \in X_h$ by

$$\begin{aligned} a(\phi, \varphi) \quad = \quad & \frac{\alpha_0}{\Delta t} \int_\Lambda \phi\, \varphi \,\mathrm{d}x + \int_{\partial\Lambda} \max(0, -\boldsymbol{u}.\boldsymbol{n})\, \phi\, \varphi \,\mathrm{d}s \\ & + \sum_{K \in \mathscr{T}_h} \int_K \boldsymbol{u}.\nabla\phi\, \varphi \,\mathrm{d}x \\ & + \sum_{S \in \mathscr{S}_h^{(i)}} \int_S [\![\phi]\!] \left(\frac{|\boldsymbol{u}.\boldsymbol{n}|}{2} [\![\varphi]\!] - \boldsymbol{u}.\boldsymbol{n}\, \{\!\{\varphi\}\!\} \right) \mathrm{d}s \end{aligned}$$

At a time step $t_n = n\Delta t$, when $n \geqslant p$, the fully discrete problem writes

$(P)_h$: *find* $\phi_h^{(n)}$ *such that*

$$a\left(\phi_h^{(n)}, \varphi_h\right) = -\sum_{k=1}^{p} \frac{\alpha_k}{\Delta t} \int_\Lambda \phi_h^{(n-k)}\, \varphi_h \,\mathrm{d}x$$

When the time step $0 < n < p$, the scheme is started by using the BDF(n) formula while when $n = 0$ we set $\phi_h^{(0)} = \pi_h(\phi_0)$. At each time step, the problem reduces to a linear system in finite dimension, which is solved by a direct method. The simulations are performed with the Rheolef C++ finite element library [10] developed by one of the authors.

3. Tests and discussion

3.1. *The Zalesak rotating disk test*

The Zalesak slotted disk in rotation [15] is a widely used test for comparing the performances of interface capturing methods. In this example, the slotted disk is rotated around the center of the computational domain $\Lambda = [0, H] \times [0, H]$ with $H = 4$ and a constant angular velocity 0.5, such that $\boldsymbol{u}(x, y) = (0.5(y-2), -0.5(x-2))$. The disk of radius 0.5 is initially centered at $(2, 2.75)$. The width of the slot is 0.12 and the maximum width of the upper bridge, that connects two parts of the disk, is 0.4. Notices that the slotted disk returns to its initial position after a period $T = 4\pi$.

Numerical computations are performed on the time interval $[0, T]$ with a time step $\Delta t = T/N$ where N is the number of time steps. Fig. 2 shows the effect of increasing the polynomial degree k of the discontinuous Galerkin method while using an implicit BDF($k+1$) scheme. The mesh is the same for all these computations (5878 elements, 3040 vertices). The comparison between the initial disk position and the final one after one period shows a dramatic improvement when using high order polynomials and schemes: for $k \geqslant 3$, the changes are no more perceptible and the disk after one period fits the initial shape. Fig. 3 shows the effect of an uniform mesh refinement when $k = 2$. Observe that the details of the slotted region, with sharp angles, becomes more precise.

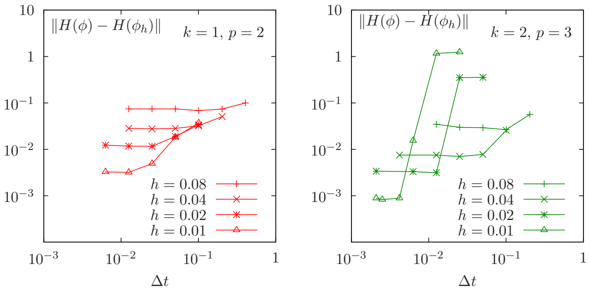
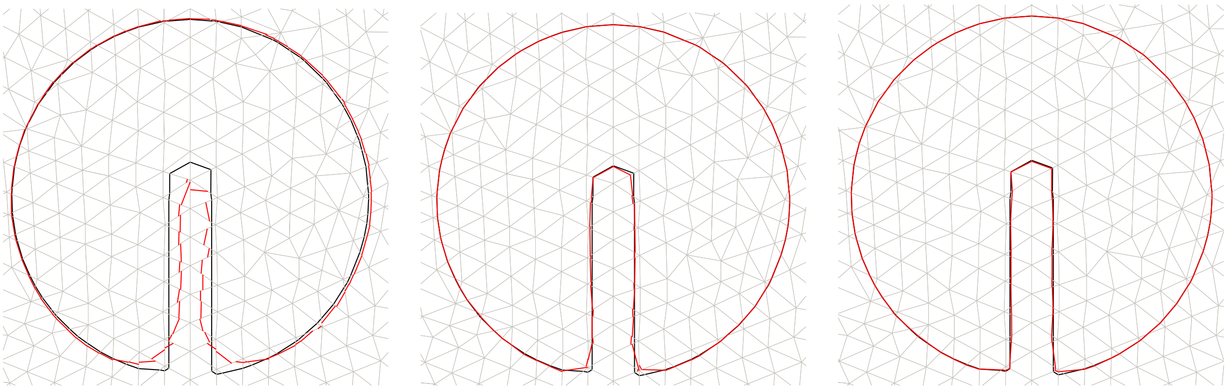


Figure 4: Zalesak test: error vs the time step Δt for fixed mesh refinement h and polynomial degree k with scheme BDF($k+1$).

In order to quantify the difference between the approximate solution and the exact one, let us introduce the following L^1 error between the two shapes:

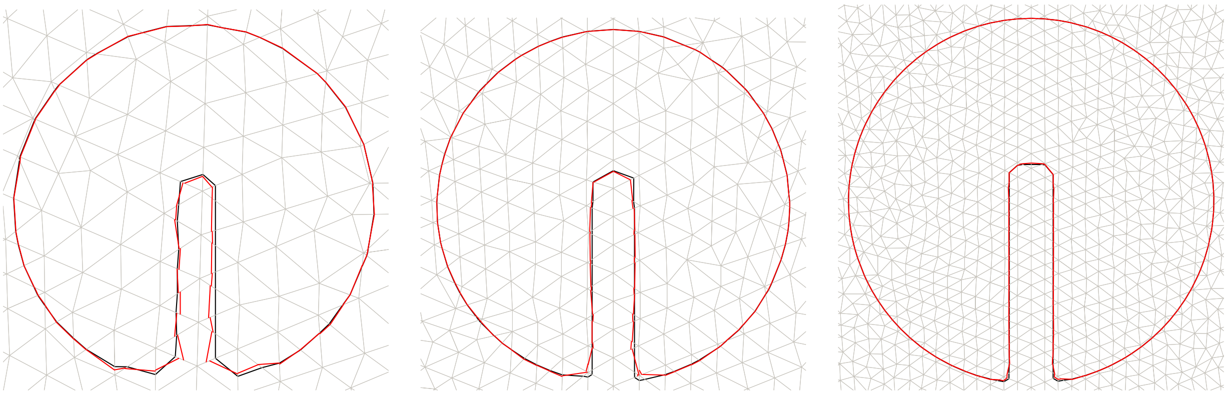
$$\|\phi - \phi_h\| = \max_{0 \leqslant n \leqslant N} \frac{1}{L} \left\| H\left(\phi(t_n)\right) - H\left(\phi_h^{(n)}\right) \right\|_{L^1(\Lambda)}$$

where H denotes the usual Heaviside function and L is the perimeter size of the initial interface. Fig. 4 shows, for this error measurement, the convergence of the approximate solution to the exact one when the time step tends to zero. Remark that, for each h, the error tends to a constant that is independent of the time step. We are looking for a representation of the error as the sum of two terms, one that depends only upon h and the other only upon Δt as $\|\phi - \phi_h\| = \mathcal{O}\left(h^\alpha + \Delta t^p\right)$. Since the solution is regular versus t, the error is expected to depend optimally upon Δt, asymptotically as Δt^p. Conversely, the shape is poorly regular in space, due to sharp corners, and the convergence properties



$k = 1, \Delta t = 4\pi/500$ $k = 2, \Delta t = 4\pi/1000$ $k = 3, \Delta t = 4\pi/2000$

Figure 2: Varying the polynomial degree k with a fixed mesh (5878 elements, 3040 vertices) and using an implicit BDF(k+1) scheme for the Zalesak test. Superposition of the initial disk (black) and after a period (red).



2622 elements, 1378 vertices $h/H = 0.03$ 5878 elements, 3040 vertices $h/H = 0.02$ 23418 elements,11910 vertices $h/H = 0.01$

Figure 3: Varying the mesh refinement with a fixed polynomial degree $k = 2$ and using an implicit BDF(3) scheme for the Zalesak test. Superposition of the initial disk (black) and after a period (red).

will be investigated numerically by looking for the $\alpha \leqslant k+1$ power index. Fig. 5 shows the error versus h for various mesh h and polynomial degrees k. The time step Δt has been chosen sufficiently small for the error to depend only upon h and k and not upon Δt (see also Fig. 4). Observe that the approximate solution converges to the exact one with mesh refinement with a power index $\alpha \approx 2$ that appears to be independent upon $k \geqslant 1$. Table 2 provides the error data for the purpose of comparison: observe that the L^1 error is of about one order of magnitude lower than those in [7]. Let us choose for Ω a full disk, which is regular, instead of a slotted one. Fig. 6 shows the error versus h after one complete revolution of the full disk. It is estimated that $\alpha \approx k/2 + 1$. Thus, the power index α is strongly dependent upon k when the shape Ω is regular and increasing k decreases dramatically the error.

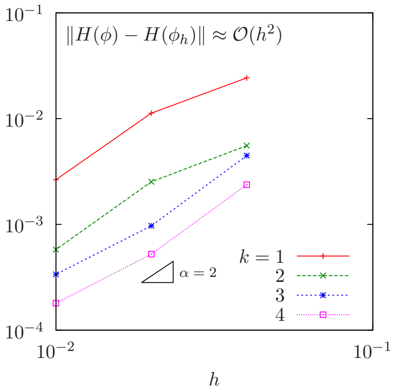


Figure 5: Zalesak test: error vs mesh refinement h and polynomial degree k.

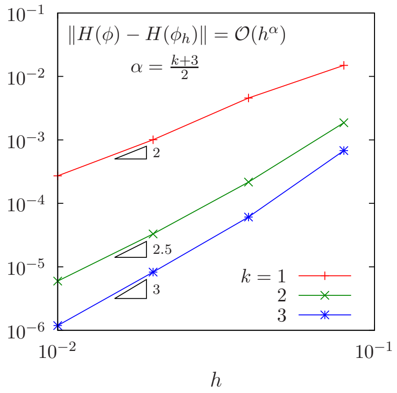


Figure 6: Test of circle: error vs mesh refinement h and polynomial degree k.

order k	mesh h/H	L^1 error from [7]	present
1	0.04		0.02429
2	0.04		0.00555
3	0.04	0.12	0.00446
4	0.04	0.05	0.00237
5	0.04	0.04	0.00148
1	0.02		0.01125
2	0.02		0.00253
3	0.02	0.019	0.00096
4	0.02	0.011	0.00052
5	0.02	0.008	0.00049
1	0.01		0.00264
2	0.01		0.00057
3	0.01		0.00033
4	0.01		0.00017
5	0.01		9.72696e-05

Table 2: Zalesak test: data and comparison with [7] for the error vs mesh refinement h and polynomial degree k.

Notice that the mass of the slotted disk at time t_n expresses:

$$m_h^{(n)} = \int_\Lambda \left(1 - H\left(\phi_h^{(n)}\right)\right) \mathrm{d}x$$

Remark that the mass error

$$m_e = \max_{0 \leqslant n \leqslant N} |m(\phi(t_n)) - m_h^{(n)}|$$

is related to the previous error in L^1 norm $m_e \leqslant L\|\phi - \phi_h\|$. The evolution of the relative mass error is reported on Fig. 7. Observe that when $k = 1$, the mass error on the coarsest mesh is of about 5% of the total mass during the whole period $t = 4\pi$. Conversely, on the finest mesh, it is dramatically decreased and remains bounded. As in the previous computation, the time step Δt has been chosen sufficiently small for the error to depend only upon h and k. While increasing the polynomial degree k, these fluctuations strongly decreases in amplitude and for $k = 4$ the relative error is bounded on all meshes.

3.2. *The vortex-in-box Leveque test*

The second test, proposed in [6], considers a disk of radius 0.15 at $(0.5, 0.75)$ in a unit square domain and the velocity field:

$$\boldsymbol{u}(t,x,y) = \begin{pmatrix} -\sin^2(\pi x)\sin(2\pi y)\cos(\pi t/T) \\ \sin^2(\pi y)\sin(2\pi x)\cos(\pi t/T) \end{pmatrix}$$

for $0 \leqslant t \leqslant T = 8$. The initial disk is compressed and become very stretched: its shape tends to become very thin. The shape reaches its maximum deformation at time $t = T/2$, and has returned to its initial state at time T. This test is also investigated with different polynomial degrees k and with the BDF(k+1) scheme.

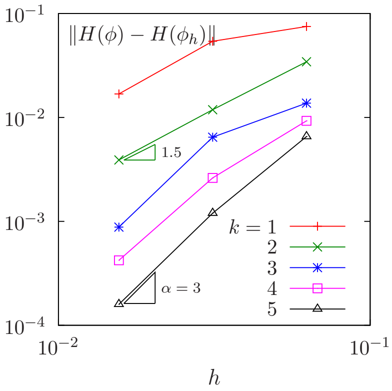


Figure 8: Leveque test: error vs uniform mesh refinement h and polynomial degree k.

Fig. 8 shows the L^1 error versus the mesh refinement h and polynomial degree k. Observe that the convergence rate α increases with k until $k = 4$ and then appears to be bounded by 3. This is due to the poor regularity of the shape, especially at the half period. From the previous paragraph, we can induce that $\alpha = \min((k+1)/2, 3)$. The behavior of the error on the mass conservation is similar to those of the previous test case and is not showed here.

An auto-adaptive mesh procedure is performed at each time step of the evolution of the interface. In the time interval $[t^n, t^{n+1}]$, starting from ϕ^n, the mesh adaptation loop iterates five times, generating an adaptive mesh, solving the transport problem and generating successive predictions $\phi^{*,i}$, $1 \leqslant i \leqslant 5$, to finally set $\phi^{n+1} = \phi^{*,5}$. The anisotropic adaptive mesh is generated accordingly to the metric associated to the Hessian of the criterion $\chi = \delta_\epsilon(\phi^n) + \delta_\epsilon(\phi^{*,i})$, where δ_ϵ is the Dirac-delta function. See the Rheolef library documentation [10] for more about mesh adaptive techniques. Fig. 9 shows some steps of the revolution. The minimum edge length in the mesh was $h_{\min} = 1/200$.

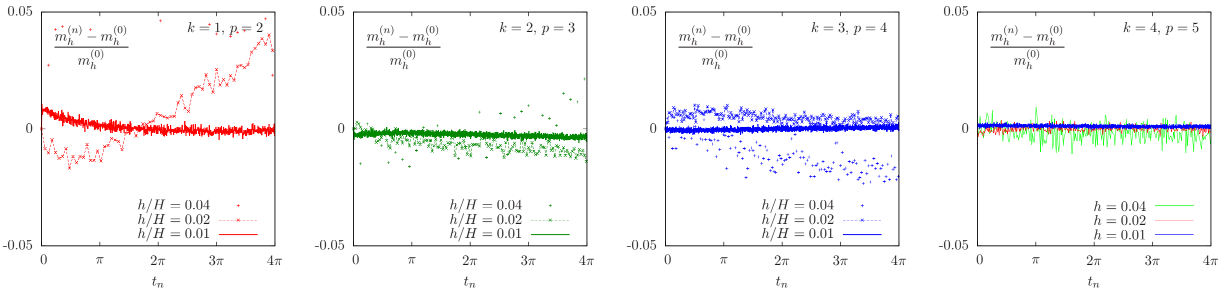
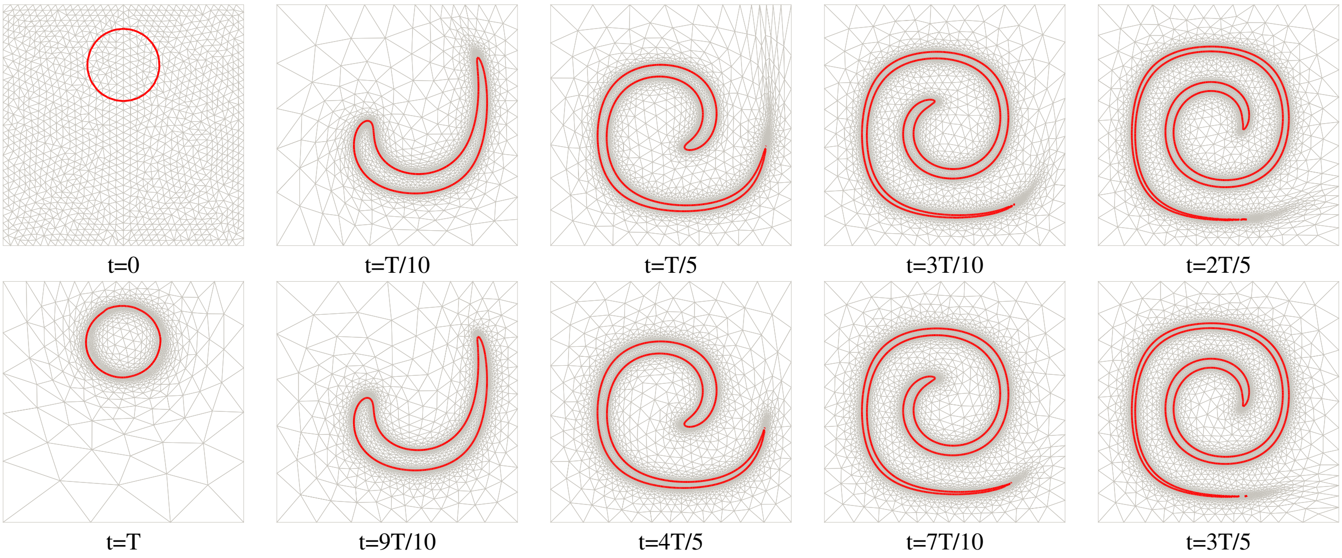


Figure 7: Zalesak test: relative mass error vs time for various polynomial degree k and mesh refinement h.


t=0 t=T/10 t=T/5 t=3T/10 t=2T/5

t=T t=9T/10 t=4T/5 t=7T/10 t=3T/5

Figure 9: Leveque test: auto-adaptive mesh during one period ($\Delta t = T/1000$).

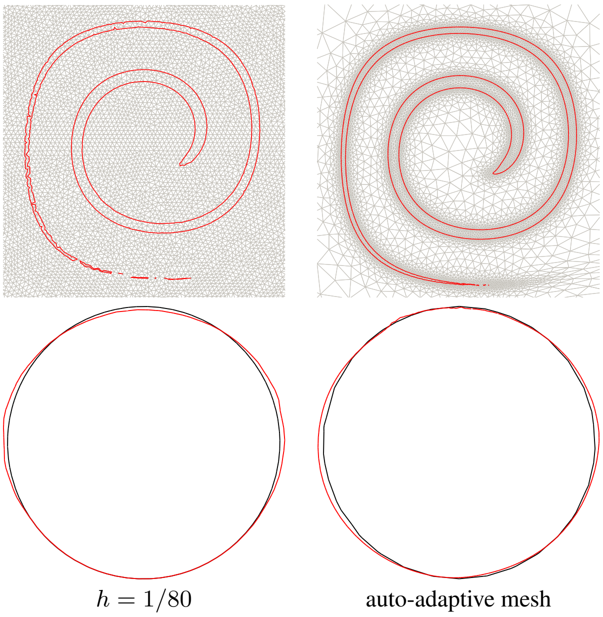

$h = 1/80$ auto-adaptive mesh

Figure 10: Leveque test: evolution of interface at time $t = T/2$ and superposition of the initial circle (black line) and final state (red line) with auto-adaptive mesh.

The final solution is reported in Fig. 10 and compared with the corresponding solution obtained with a uniform mesh with $h = 1/80$.

4. Conclusion

In this contribution, a level set transport is investigated with an implicit high order time (BDF) and polynomial degree discontinuous Galerkin finite element method. Using three well-known cases, i.e. a rotation circle, and the Zalesak and Leveque tests, we establish that our method present a nice mesh convergence. For the Zalesak case, the numerical solution converges toward the exact solution in h^2 whatever the polynomial degree mainly due to the non-regularity of the exact solution. For the Leveque test, the mesh convergence is better but stays limited at high polynomial degree due to the sharp interface observed during the advection process. For the rotating circle, up too oour tests, there is no a priori limit for the convergence rate versus h and we can conclude that the limitation of the convergence rate appears when the region Ω has poor regularity.

It is noteworthy that the mass loss becomes very small when the polynomial degree increases. This means that it is not necessary to introduce artificially the mass conservation. This is an important feature for the future work when the transport scheme will be coupled with the Navier-Stokes equations.

References

[1] D. A. di Pietro and A. Ern. *Mathematical aspects of discontinuous Galerkin methods*. Springer, 2012.

[2] D. A. Di Pietro, S. Lo Forte, and N. Parolini. Mass preserving finite element implementations of the level set method. *Appl. Numer. Math.*, 56(9):1179–1195, 2006.

[3] S. Gottlieb, Chi-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43(1):89–112, 2001.

[4] A. Laadhari, P. Saramito, and C. Misbah. Improving the mass conservation of the level set method. *C. R. Acad. Sci. Paris, ser. I*, 348:535–540, 2010.

[5] A. Laadhari, P. Saramito, and C. Misbah. Computing the dynamics of biomembranes by combining conservative level set and adaptive finite element methods. *J. Comput. Phys.*, 263:328–352, 2014.

[6] R. J. LeVeque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.*, 33(2):627–665, 1996.

[7] E. Marchandise and J.-F. Remacle. A quadrature-free discontinuous Galerkin method for the level set equations. *J. Comput. Phys.*, 212:338–357, 2006.

[8] S. Osher and J. A. Sethian. Front propaging with curvature-dependent speed: agorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(12), 1988.

[9] F. Pigeonneau and P. Saramito. Discontinuous Galerkin finite element method applied to the coupled Stokes/Cahn-Hilliard equations. In *9th international conference on multiphase flow*, 2016.

[10] P. Saramito. *Efficient C++ finite element computing with Rheolef*. CNRS and LJK, 2013. http://cel.archives-ouvertes.fr/cel-00573970.

[11] P. Saramito. *Efficient C++ finite element computing with Rheolef: volume 2: discontinuous Galerkin methods*. CNRS and LJK, 2013. http://cel.archives-ouvertes.fr/cel-00863021.

[12] J. Sethian. *Level set methods and fast marching methods*. Cambridge University Press, UK, 1999.

[13] E. Süli and D. F. Mayers. *An introduction to numerical analysis*. Cambridge University Press, UK, 2003.

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

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