Level Set Implementation Using FEniCS

Paulo Pimenta[a,b,1,*], Author Two[c], Author Three[a,c]

[a] *Address One*
[b] *Address Two*
[c] *Some University*

Abstract

Text of abstract. Text of abstract. Text of abstract. Text of abstract. Text of abstract.

Keywords: example, LaTeX, template

1. Level set method for two-phase flows

In numerical problems that involve dynamics of interfaces or discontinuities, surface-tracking techniques provide an ambient in which we are able to define different regions and analyze their evolution with time. This procedure also must ensure accuracy and efficiency during the numerical process. The front-tracking methods available nowadays can be simplified into two groups. In the first one, the interfaces are represented by marked particles, which can be seen as an explicit representation of the interface, see for instance [1]. In the second group, we have the techniques in which the interface are represented implicitly by functions with higher dimensional order than the interface [2]. One of the numerical procedures that assume this methodology was proposed by Osher and Sethian (1988) [3], and it is well known as level set method (LSM).

The track mechanism inherent to level set method is activated when it is transported by the velocity field $\mathbf{v}$ via advection equation [4]

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{v}\phi) = 0 \tag{1}$$

In this chapter, we discuss over two approaches within the level set assumption (higher dimensional order than the interface). The first method define the level set $\phi(\mathbf{x}; t)$ as the signed distance function and, in the second one, the level set is defined as the Heaviside-like function.

[☆] This is only an example

[*] I am corresponding author

Email addresses: author.one@mail.com (Paulo Pimenta), author.two@mail.com (Author Two), author.three@mail.com (Author Three)

URL: author-one-homepage.com (Paulo Pimenta)

[1] I also want to inform about...

[2] Small city

1.1. Level set method (LSM) as the signed distance function

The standard LSM introduced by Osher and Sethian define the level set function as the signed distance function [5]. In this context, taking the domain $\Omega := \{\Omega \subset \mathbb{R} \mid \Omega = \Omega_1 \cup \Omega_2\}$ and the interface that separates the two sub-domain as $\Gamma := \{\Omega_1 \cap \Omega_2\}$, we are able to describe $\phi(\mathbf{x}) \in \mathbb{R}$ as the function that determines the minimum distance from any point $\mathbf{x} \in \Omega$ to the point $\mathbf{x}_i \in \Gamma$, or in a compact form as

$$\phi\left(\mathbf{x}\right) = \begin{cases} \min\limits_{\mathbf{x}_i \in \Gamma} |\mathbf{x} - \mathbf{x}_i|, \ \mathbf{x} \in \Omega_1 \\ -\min\limits_{\mathbf{x}_i \in \Gamma} |\mathbf{x} - \mathbf{x}_i|, \ \mathbf{x} \in \Omega_2 \end{cases} \tag{2}$$

The interface between the two sub-domain Ω_1 and Ω_2 can be defined with the use of the level set function as an implicit hypersurface as [6, 7]

$$\Gamma := \{\mathbf{x} \mid \phi(\mathbf{x}) = 0\} \tag{3}$$

One intrinsic property of the signed distance function is that it satisfies the Eikonal equation condition [8], which is

$$|\nabla \phi(\mathbf{x})| = 1 \tag{4}$$

therefor, the normal vector at Γ can be represented as

$$\mathbf{n} = \frac{\nabla \phi(\mathbf{x})}{|\nabla \phi(\mathbf{x})|} \Rightarrow \mathbf{n} = \nabla \phi(\mathbf{x}) \tag{5}$$

During the transport of the interface, the level set lose its singed distance function characteristic. Consequently, this issue leads to numerical instability and loss of accuracy. In order to address this problem, Sussman at al (1998) [9] proposed the following differential equation,

$$\frac{\partial \phi}{\partial \tau} + \text{sign}(\phi_0)(1 - |\nabla \phi|) = 0 \tag{6}$$

which must be solved until the steady state is achieved [7, 2].

In Eq. (6), τ is an intermediate time-step (or pseudo-time at the re-initialization level), which has to be appropriately chosen to avoid divergence and $\text{sign}(\phi_0)$ is the sign function, which is dependent of the value of the level set function evaluated after the transport of the interface.

In practice, $\text{sign}(\phi_0)$ has to ensure that the transition between the two species at the interface must be smooth. Therefore, $\text{sign}(\phi_0)$ is explicitly expressed as a function of the current level set value as

$$\text{sign}(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \epsilon^2 |\nabla \phi|^2}} \tag{7}$$

The introduction of ϵ determines how much the transition will be smooth and it is usually associated with the mesh size.

It is important to mention that the achievement of the steady state of Eq. (6) implies in restore the level set function as the singed distance function without move the interface.

However, this expression is unstable [4] and the addition of an artificial term, such as $\alpha\nabla^2\phi$, can help address this drawback. Then, Eq. (6) can be rewritten as

$$\frac{\partial\phi}{\partial\tau} - \text{sign}(\phi_0)(1 - |\nabla\phi|) - \alpha\nabla^2\phi = 0 \tag{8}$$

The parameter α is responsible for the amount of diffusion and its numerical value is chosen by experimentation. The numerical development of this approach can found in Section 3.

1.2. Conservative level set method

The standard level set method has some drawbacks, as lack of conservation of mass. The approach proposed by Olsson and Kreiss (2005) [10], which is so-called conservative level set method (CLSM), suggest to replace the signed distance function, Eq. (2), with a regularized Heaviside function in order to overcome this conservation issue [7, 10]. Therefore, the level set function assume

$$\phi(\mathbf{x}) = \frac{1}{1 + e^{d(\mathbf{x})/\epsilon}} \tag{9}$$

where $d(\mathbf{x})$ is represented by the signed distance function (2) and ϵ is the perimeter that helps prevent spurious by smoothing the transition of the physical properties between the two sub-domain; it is also called interface thickness.

Within this approach, the interface is represented by the following hypersurface instead of (3),

$$\Gamma := \{\mathbf{x} \mid \phi(\mathbf{x}) = 0.5\} \tag{10}$$

The conservative level set function also needs to be reinitialized in order to maintain the regularized Heaviside function property [11]. Among the approaches proposed in the literature, we use here the one introduced by Olosson and Kreiss (2007) [12], which can be expressed as

$$\frac{\partial\phi}{\partial\tau} + \nabla\cdot\left[\phi\left(1-\phi\right)\mathbf{n}_\Gamma\right] = \epsilon\nabla\cdot\left[\mathbf{n}_\Gamma\left(\nabla\phi\cdot\mathbf{n}_\Gamma\right)\right] \tag{11}$$

where $\mathbf{n}_\Gamma$ is the unit normal vector at the interface Γ, which points into the area surrounded by the interface [11]. It is important to mention that this vector does not change during the re-initialization procedure and it is evaluated assuming as initial value the of the level set information after the previous transport. In other words,

$$\mathbf{n}_\Gamma = \frac{\nabla\phi\left(\mathbf{x},\tau_0\right)}{|\nabla\phi\left(\mathbf{x},\tau_0\right)|} \tag{12}$$

The numerical discretization and the FEniCS algorithm for the re-initialization of the conservative level set is presented in Section 3.2.

2. Integration of the advection equation for transport of the level set function

The FEM integration of the advection equation is given by integrating Eq. (1) and also taking the test function $l \in W^2$ (W^2 denotes the degree of the Lagrangian function used to interpolate the unknowns inside of the element),

$$\int_\Omega \frac{\partial \phi}{\partial t} l \, d\Omega + \int_\Omega \nabla \cdot (\mathbf{v}\phi) l \, d\Omega = 0 \tag{13}$$

From the Gauss Theorem, Eq. (13) yields

$$\int_\Omega \frac{\partial \phi}{\partial t} l \, d\Omega + \int_\Omega (\phi l) \, \mathbf{v} \cdot \mathbf{n} \, ds - \int_\Omega \nabla l \cdot (\phi \mathbf{v}) \, d\Omega = 0 \tag{14}$$

Assuming free-slip condition at the walls ($\mathbf{v} \cdot \mathbf{n} = 0$), the term $\int_\Omega (\phi l) \, \mathbf{v} \cdot \mathbf{n} \, ds$ vanishes and

$$\int_\Omega \frac{\partial \phi}{\partial t} l \, d\Omega - \int_\Omega \nabla l \cdot (\phi \mathbf{v}) \, d\Omega = 0 \tag{15}$$

Assuming the general time interpolation procedure, Eq. (15) became

$$\frac{1}{\Delta t} \left\langle (\phi^{n+1} - \phi^n), l \right\rangle - \theta \left\langle \nabla l \cdot \mathbf{v}^{n+1}, \phi^{n+1} \right\rangle - (1-\theta) \left\langle \nabla l \cdot \mathbf{v}^n, \phi^n \right\rangle = 0 \tag{16}$$

To unsure stability in our numerical procedure, we must add a stabilization term in the advection equation. Here, we apply two approaches, which are SUPG and the IP (interior penalty) stabilization.

The SUPG is shown in the following representation

$$r = \left\langle \frac{\phi - \phi_0}{\Delta t} + \theta \left(\mathbf{v}^{n+1} \cdot \nabla \phi^{n+1} \right) + (1-\theta) \left(\mathbf{v}^n \cdot \nabla \phi^n \right), \tau \mathbf{v} \cdot \nabla l \right\rangle \tag{17}$$

where τ is given by

$$\tau = \frac{h}{2||\mathbf{v}||} \tag{18}$$

The FEniCS algorithm for SUPG is shown below

```
nb = sqrt(inner(v,v))
tau = h*pow(2.0*nb,-1.0)
r = ( ((l-l0)/k) + theta*(inner(v,grad(l)))+ \\
(1.0-theta)*(inner(v0,grad(l0))) )*tau*inner(v,grad(l_))*dx
```

The IP scheme is expressed as

$$\int_{\text{all facets}} \bar{\alpha} h^2 \text{jump} \left(\nabla \phi \cdot \mathbf{n} \right) \text{jump} \left(\nabla \bar{\phi} \cdot \mathbf{n} \right) ds \tag{19}$$

The FEniCS code for IP stabilization is implemented below

```
n = FacetNormal(mesh)
h = CellSize(mesh)
h_avg = (h('+') + h('-'))/2.0
alpha=Constant(0.0)                    # Penalty parameter

# IP (interior penalty) approximation
def IP(l,l_):
    r=alpha('+')*h_avg*h_avg*inner(jump(grad(l),n), jump(grad(l_),n))*dS
    return (r)
```

3. Numerical integration of the re-initialization approach

3.1. Level set method with signed distance function

In the context of the FEM, we rewrite Eq. (8) in the weak form, with w as test function, as

$$\int_\Omega \frac{\partial \phi}{\partial \tau} w \, d\Omega - \int_\Omega \text{sign}(\phi_0)(1 - |\nabla \phi|) w \, d\Omega - \int_\Omega \alpha \nabla^2 \phi w \, d\Omega = 0 \tag{20}$$

since, we can assume

$$\nabla \cdot (\nabla \phi w) = w \nabla^2 \phi + \nabla \phi \cdot \nabla w \tag{21}$$

Therefor, the artificial term presented in (20) can be expressed as

$$\int_\Omega \nabla^2 \phi w \, d\Omega = \int_\Omega \nabla \cdot (\nabla \phi w) \, d\Omega - \int_\Omega \nabla \phi \cdot \nabla w \, d\Omega \tag{22}$$

and using the Gauss Theorem,

$$\int_\Omega \nabla^2 \phi w \, d\Omega = \int_{\partial \Omega} (\nabla \phi w) \cdot \, d\mathbf{s} - \int_\Omega \nabla \phi \cdot \nabla w \, d\Omega \tag{23}$$

in cases when the surface integral vanishes, we have

$$\int_\Omega \frac{\partial \phi}{\partial \tau} w \, d\Omega - \int_\Omega \text{sign}(\phi_0)(1 - |\nabla \phi|) w \, d\Omega + \alpha \int_\Omega \nabla \phi \cdot \nabla w \, d\Omega = 0 \tag{24}$$

Using the forward Euler time discretization, we are able to write

$$\frac{1}{\Delta \tau} \left\langle \phi^{n+1} - \phi^n, w \right\rangle - \left\langle \text{sign}(\phi_0)(1 - |\nabla \phi^n|), w \right\rangle + \left\langle \alpha \nabla \phi^n \cdot \nabla, w \right\rangle = 0 \tag{25}$$

The FEniCS algorithm of Eq. (25) is presented below

```
def reinit(l,mesh,Dx):

    # l    - is the level set field which comes from of the main algorithm
    # mesh - mesh description
    # Dx   - mesh size in the x-direction
```

```
# Space definition
FE = FunctionSpace(mesh, "CG", 2)
V = VectorFunctionSpace(mesh, "CG", 1, dim=2)

# Set the initial value
phi = TrialFunction(FE); phi0 = TrialFunction(FE)
w = TestFunction(FE)

# Gradient norm
def mgrad(b):
    return(sqrt(b.dx(0)**2 + b.dx(1)**2))

# Setup the parameters
dt = 0.0001                    # Time step
k = Constant(dt)               # Time step Python/FEniCS syntax
phi0 = l                       # Initial value
eps = Constant(1.0/Dx)         # Interface thickness
alpha = Constant(0.0625/Dx)    # Numerical diffusion parameter

signp = l/sqrt(l*l + eps*eps*mgrad(l)*mgrad(l))

# FEM linearization
a = (phi/k)*w*dx
L = (phi0/k)*w*dx+signp*(1.0-sqrt(dot(grad(phi0),grad(phi0))))*w*dx-\
    alpha*inner(grad(phi0),grad(w))*dx

# Boundary condition
bc = []

# Flag setup
Ei = 1e10; E_old = 1e10
cont = 0;  num_steps = 10

phi = Function(FE)
for n in range(num_steps):
    solve(a == L, phi, bc)

    # Euclidean norm
    error = (((phi - phi0)/k)**2)*dx
    E = sqrt(abs(assemble(error)))
    print "error:", E

    phi0.assign(phi)

    # Divergence flag
    if (E_old < E ):
        fail = 1
        print "*Diverges_at_the_re-initialization_level*", cont
        break
```

```
        cont +=1
        E_old = E
    return phi
```

It is worth noting that, during the numerical procedure, the standard level set method does not need to be re-initialized at each increment of time. Another important fact is that the achievement of steady state of Eq. (8) is not necessary, since recover the singed distance function propriety is important only nearly the interface instead of the entire domain. Usually, the number of re-initialization increments are chosen by experimentation.

After running some tests, we verify that the lack of the mass conservation and the presence of spurious are important issues in this approach.

3.2. Numerical integration of the re-initialization equation for CLSM

This topic is dedicated to the numerical integration of the re-initialization equation for Heaviside level set function.

Assuming the test function $w \in W^2$, as well as $\phi \in W^2$, the weak form of Eq. (11) is presented by

$$\int_\Omega \frac{\partial \phi}{\partial \tau} w \, d\Omega + \int_\Omega \nabla \cdot \left[\phi \left(1-\phi\right) \mathbf{n}_\Gamma\right] w \, d\Omega = \int_\Omega \epsilon \nabla \cdot \left[\mathbf{n}_\Gamma \left(\nabla \phi \cdot \mathbf{n}_\Gamma\right)\right] w \, d\Omega \tag{26}$$

If we consider the identity shown in (21), the Gauss Theorem and the assumption that the surface integral vanishes, we have

$$\int_\Omega \frac{\partial \phi}{\partial \tau} w \, d\Omega - \int_\Omega \left[\phi \left(1-\phi\right) \mathbf{n}_\Gamma\right] \cdot \nabla w \, d\Omega + \epsilon \int_\Omega \nabla w \cdot \left[\mathbf{n}_\Gamma \left(\nabla \phi \cdot \mathbf{n}_\Gamma\right)\right] \, d\Omega = 0 \tag{27}$$

Using the general time linearization form, we can rewrite Eq. (27) as

$$\begin{aligned} \frac{1}{\Delta \tau} \left\langle (\phi^{n+1} - \phi^n), w \right\rangle - \frac{1}{2} \left\langle \nabla w, (\phi^{n+1} + \phi^n) \mathbf{n}_\Gamma \right\rangle + \left\langle \phi^{n+1} \phi^n \mathbf{n}_\Gamma, \nabla w \right\rangle \\ + \frac{\varepsilon}{2} \left\langle (\nabla \phi^{n+1} + \nabla \phi^n) \cdot \mathbf{n}_\Gamma, \nabla w \cdot \mathbf{n}_\Gamma \right\rangle = 0 \end{aligned} \tag{28}$$

The time step assumes the rule [12]

$$\Delta \tau = \frac{\Delta x^{1+d}}{2} \tag{29}$$

and the interface thickness,

$$\epsilon = \frac{\Delta x^{1-d}}{2} \tag{30}$$

We propose in this report to rewrite Eqs. (29) and (30) adding a parameter β. Once this parameter allow us introduce a thinner skin to the bubble by adopting a number smaller than 0.5.

$$\Delta \tau = \beta \Delta x^{1+d} \tag{31}$$

$$\epsilon = \beta \Delta x^{1-d} \tag{32}$$

With $d = 0.1$ and Δx as the spacial resolution in relation to the smallest dimension, the FEniCS algorithm is represented below

```
def reinit(l,epsilon,beta,Dx,mesh):

    # time-step
    dtau = Constant(1.0/(beta*((1.0/Dx)**(1.1))))

    # space definition
    V = VectorFunctionSpace(mesh, "CG", 1, dim=2)
    FE = FunctionSpace(mesh, "CG", 2)

    # functions setup
    phi = Function(FE); phi0 = Function(FE); w = TestFunction(FE)

    # intial value
    phi0.assign(interpolate(l,FE))

    # Unit normal vector (does not change during this process)
    grad_n = project(grad(l),V)
    n = grad_n/(sqrt(dot(grad_n,grad_n)))

    # FEM linearization (The same form found in ..)
    F = dtau*(phi-phi0)*w*dx-(0.5*(phi+phi0)*(1.0-0.5*(phi+phi0))*\
        dot(n,grad(w)))*dx+epsilon*dot(n,grad((0.5*(phi+phi0))))*\
        dot(grad(w),n)*dx

    bc=[]

    E=1e10; E_old=1e10
    cont=0; num_steps=10

    for n in range(num_steps):

        begin("Reinitialization")
        solve(F == 0, phi, bc)
        end()

        error = (((phi - phi0)*dtau)**2)*dx
        E = sqrt(abs(assemble(error)))
        fail = 0
        if (E_old < E ):
            fail = 1
            print "------------------------------"
            print "Fail!"
            print "at:", cont
            print "------------------------------"

            break
        if (fail != 1):
            phi0.assign(phi)
        cont +=1; E_old = E
```

```
    print "Error:", E, "nincre", cont
    return phi
```

References

[1] J. M. Hyman, Numerical methods for tracking interfaces, Physica D: Nonlinear Phenomena 12 (1-3) (1984) 396–407. doi:https://doi.org/10.1016/0167-2789(84)90544-X.

[2] S. Zahedi, Numerical methods for fluid interface problems, Ph.D. thesis, KTH Royal Institute of Technology (2011).

[3] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations, Journal of computational physics 79 (1) (1988) 12–49. doi:https://doi.org/10.1016/0021-9991(88)90002-2.

[4] N. Johansson, Implementation of a standard level set method for incompressible two-phase flow simulations (2011).

[5] V. Dobrev, F. Grogan, T. Kolev, R. Rieben, V. Tomov, Level set methods for detonation shock dynamics using high-order finite elements, Tech. rep., Lawrence Livermore National Laboratory (LLNL), Livermore, CA (2017).

[6] L. Zhao, J. Mao, X. Bai, X. Liu, T. Li, J. Williams, Finite element implementation of an improved conservative level set method for two-phase flow, Computers & Fluids 100 (2014) 138–154. doi:https://doi.org/10.1016/j.compfluid.2014.04.027.

[7] D. Lindbo, Finite Element Computations for a Conservative Level Set Method Applied to Two-Phase Stokes Flow, Numerisk analys och datalogi, Kungliga Tekniska högskolan, 2006.

[8] C. Basting, D. Kuzmin, J. N. Shadid, Optimal control for reinitialization in finite element level set methods, International Journal for Numerical Methods in Fluids 84 (5) (2017) 292–305. doi:10.1002/fld.4348.

[9] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-phase flows, Computers & Fluids 27 (5) (1998) 663–680. doi:https://doi.org/10.1016/S0045-7930(97)00053-4.

[10] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, Journal of computational physics 210 (1) (2005) 225–246. doi:https://doi.org/10.1016/j.jcp.2005.04.007.

[11] M. Habera, Numerical simulation of ferrofluids, Ph.D. thesis, Charles University in Prague (2015). doi:10.13140/RG.2.1.1718.1928.

[12] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow ii, Journal of Computational Physics 225 (1) (2007) 785–807.