Levelset methods (and XFEM) in FEniCS

Mischa Jahn, Timo Klock, Andreas Luttmann
The Center for Industrial Mathematics (ZeTeM), University of Bremen

Motivation

- Our project within the Collaborative Research Center 747 investigates i.a. a material accumulation process based on rod-end melting, see Fig. 1.
- This processes can be modeled by PDEs (Stefan problem, Navier-Stokes) and simulated using FEM [3]. The model includes time-dependent discontinuities.

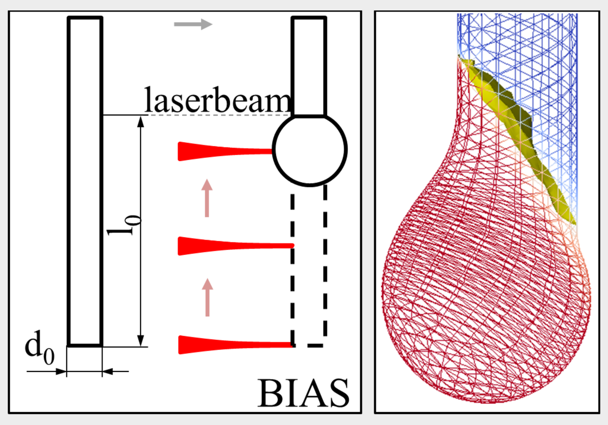

Fig. 1: Material accum. process

The Levelset method: Review

- The zero level set of a signed distance function $\varphi\colon \Omega\times[t_0,t_f]\to\mathbb{R}$ (continuous, scalar) represents a time dependent discontinuity [6], separating Ω into $\Omega(t)=\Omega^+(t)\cup\Omega^-(t)\cup\Gamma(t)$, cf. Fig. 2.
- The level set problem is given by: Find $\varphi(x,t)\in C^1(\Omega,[t_0,t_f])$, s.t.

$$\begin{aligned}\varphi_t+\vec{u}\cdot\nabla\varphi&=0 &&\text{in } \Omega\times[t_0,t_f],\\ \varphi(x,t_0)&=\varphi_0(x) &&\text{in } \Omega,\\ \varphi(x,t)&=\varphi_D(x,t) &&\text{on } \partial\Omega_{\text{in}}(t)\times[t_0,t_f].\end{aligned}$$

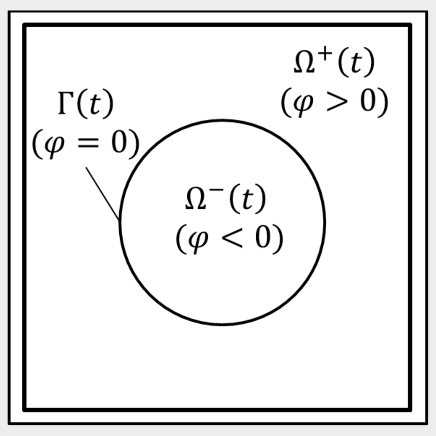

Fig. 2: Subdomains $\Omega^+(t)$ and $\Omega^-(t)$ separated by $\Gamma(t)$.

Weak formulation: With $V_{u,D}=\{v\in L^2(\Omega): u\cdot\nabla v\in L^2(\Omega)\wedge v|_{\partial\Omega_{\text{in}}}=\varphi_D\}$ the weak formulation of the level set problem is given by: For $t\in[t_0,t_f]$ find $\varphi(\cdot,t)\in V_{u,D}$ s.t. $\varphi(\cdot,t_0)=\varphi_0$ and

$$(\varphi_t,v)_{L^2}+(\vec{u}\cdot\nabla\varphi,v)_{L^2}=0,\quad \forall v\in L^2(\Omega).$$

The Levelset method: Numerical aspects

Discretization: Using standard Lagrangian function spaces and the $\theta-$scheme, the fully discretized and stabilized problem is given by

$$\sum_{S\in\mathcal{S}_h}\left(\tfrac{\varphi_h^{n+1}-\varphi_h^n}{\Delta t}+\vec{u}\cdot\left(\theta\varphi_h^{n+1}+(1-\theta)\varphi_h^n\right),v_h+\delta_S\vec{u}\cdot\nabla v_h\right)_{L^2(S)}=0. \qquad (1)$$

Interface representation [2]:

- Γ_h is given by the linear interpolation of φ_h on a regularly refined mesh.
- # refinements depends on the polynomial degree k of the basis functions. An example for $k=2$ is shown in Fig. 3.

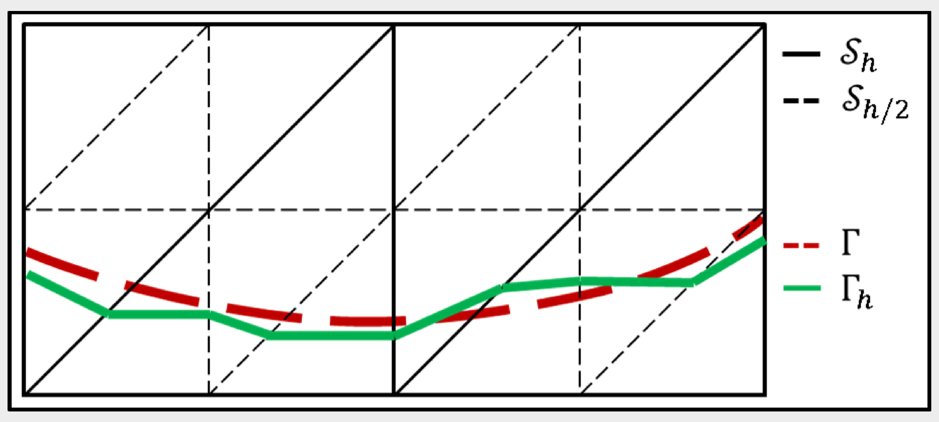

Fig. 3: Construction of Γ_h on $\mathcal{S}_{h/2}$

Reinitialization [2,5]:

- During the evolution of φ_h in time, i.a the signed distance property get lost.
- A reinitialization $\hat{\varphi}_h$ of φ_h is required with $\hat{\Gamma}_h\approx\Gamma_h$ and $||\nabla\hat{\varphi}_h||\approx 1$.
- The Fast Marching Method of [2] is used as reinitialization technique. It consists of an initialization and an iteration phase, cf. Fig. 4.

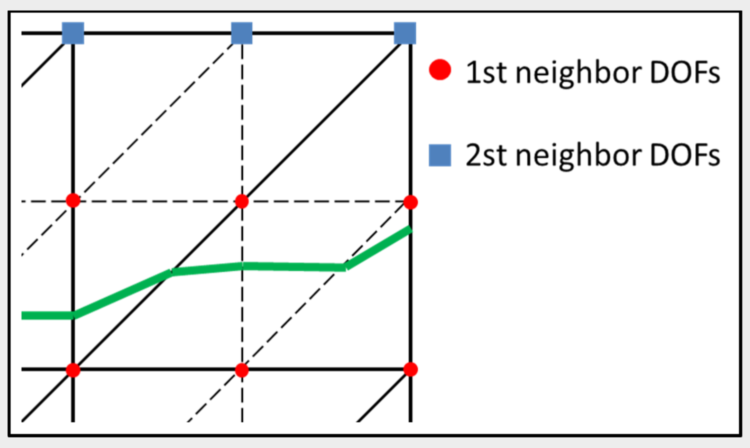

Fig. 4: Construction of Γ_h on $\mathcal{S}_{h/2}$

Volume correction [1]:

- the level set method is (on a discrete level) not volume conserving
- volume correction methods take advantage of the signed distance property, shifting φ_h by $\epsilon_S\in\mathbb{R}$, $S\in\mathcal{S}_{h/2}$, the roots of the non-linear equation

$$Z_S(\epsilon_S):=V_{h,S}^-(I\varphi_h^{\text{old}}(\cdot))-V_{h,S}^-(I\varphi_h^{\text{new}}(\cdot)+\epsilon_S)=0,\ S\in\mathcal{S}_{h/2} \qquad (2)$$

- the corrected function is given by $I\varphi_h^{c}:=I\varphi_h^{\text{new}}+\psi_h$, with ψ_h interpolating the values ϵ_S.

Levelset Toolbox (C++)

LevelSetEquationXD	LevelSetCalculatorXD	LevelSetCalculatorUtils
Form resp. header files containing problem specific data (1).	Object class with dimension-dependent constructors and methods	Utility class with dimension-independent functions

Fig. 5: Implementation structure: Form (header) files, object classes and utility functions.

Example 1 Using the Levelset Toolbox in FEniCS

```
[...] // Initialization etc.
// Creating and extending the parameters structure: Since all methods are
// hidden within the object class, the reinitialization frequency, the volume
// correction method etc. are added and defined within the parameters structure.
dolfin::Parameters parameters; parameters.add("foo", foo);
// Creating an object of type LevelSetCalculatorXD
LevelSetCalculatorXD lc(mesh, u(x,t), φh(t0), parameters);
// Update LevelSetCalculatorXD lc object based on the specified parameters.
lc.updateLevelSetFunction();
[...]
```

Results

- Example 2D [4]: On $\Omega=[0,1]^2$, consider φ_0 for a disk with $r=0.15$ centered at $(0.5,0.75)$ for $t\in[0,2]$. The velocity field $u(t,x,y)$ is given by

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

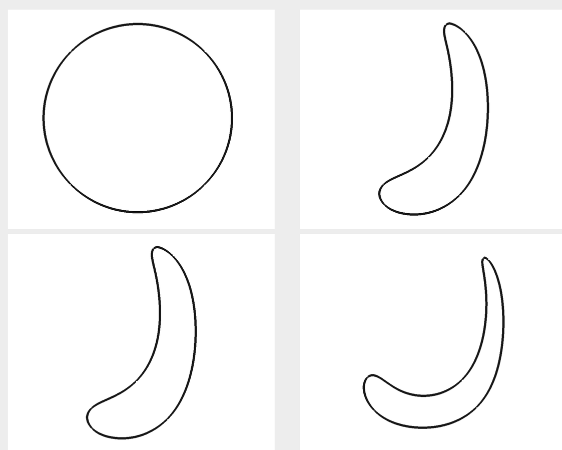
Fig. 6: Sketch 2D Example: Ref. φ_h at $t=\{0,0.5,1,1.5\}$. It is $\varphi_h(t=0)=\varphi_h(t=2)$.

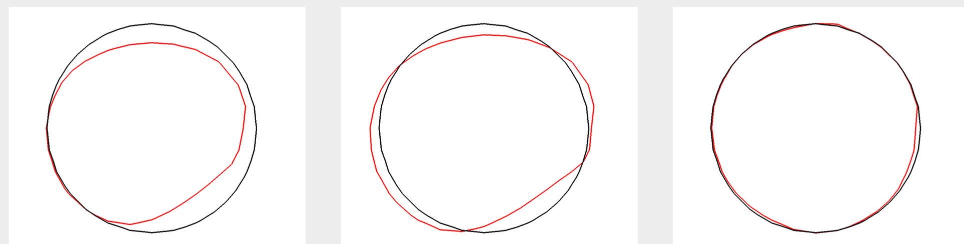
Fig. 7: 2D Results at $(t=2)$ with $2\times32\times32=2048$ elements (form left to right): reinit., reinit. and global vol. corr, reinit. and local vol. corr., (red), reference solution in black.

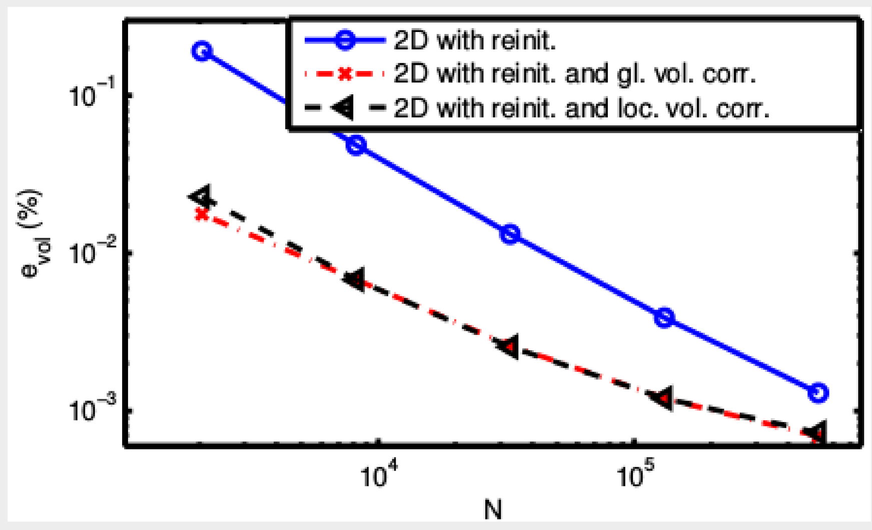

Fig. 8: Rel. volume error in L_2: Comparison of different maintaining methods on different mesh sizes.

Δt	$\theta=1$	$\theta=0.5$
$2^0/10$	$3.25e-2$	$6.10e-3$
$2^{-1}/10$	$1.86e-2$	$1.54e-3$
$2^{-2}/10$	$1.01e-2$	$3.87e-4$
$2^{-3}/10$	$5.36e-3$	$9.68e-5$
$2^{-4}/10$	$2.71e-3$	$2.42e-5$
$2^{-5}/10$	$1.32e-3$	$5.99e-6$

Tab. 1: L_2-error for different time discretization schemes on a mesh consisting of $2\times10\times10=200$ elements.

- Example 3D [4]: On $\Omega=[0,1]^3$, consider φ_0 for a sphere, centered at (0.35, 0.35, 0.35) with $r=0.15$ for $t\in[0,2]$. The velocity field is given by

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

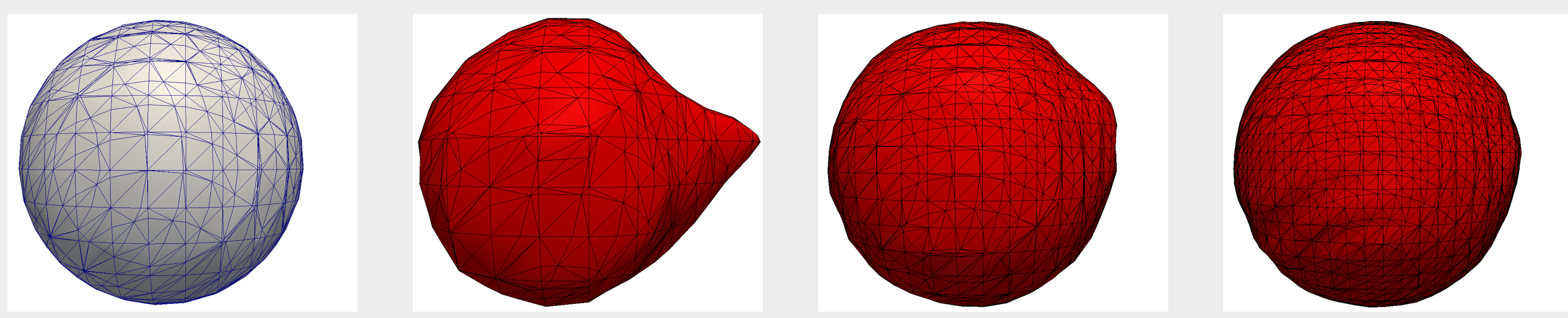
Fig. 9: 3D Results at $(t=2)$: Reference solution and numerical solution for mesh sizes $6\times24\times24$, $6\times44\times44$ and $6\times64\times64$ using reinitialization and local volume correction.

Acknowledgement

The authors gratefully acknowledge the financial support by the DFG (German Research Foundation) for the subproject A3 within the Collaborative Research Center SFB 747 “Mikrokaltumformen - Prozesse, Charakterisierung, Optimierung”.

References

1 R. Ausas, E. Dari, and G. Buscaglia. A mass-preserving geometry-based reinitialization method for the level set function. Mecanica Computacional, 27:25-27, 2008.
2 S. Gross and A. Reusken. Numerical Methods for Two-phase Incompressible Flows. Springer Series in Computational Mathematics. Springer, 2011.
3 M. Jahn, H. Brüning, A. Schmidt, F. Vollertsen. Energy dissipation in laser-based free form heading: a numerical approach. Production Engineering - Research and Development, Springer Verlag, 2013.
4 R. LeVeque. Wave propagation algorithms for multidimensional hyperbolic systems. Journal of Computational Physics, 131:327-353, 1997.
5 S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. Journal of Computational Physics, 79(1):12-49, 1988.
6 J. A. Sethian. A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences, 93(4):1591-1595, 1996.