Commun. Comput. Phys.
doi: 10.4208/cicp.191114.140715a

Vol. **19**, No. 2, pp. 329-353
February 2016

Patch-Recovery Filters for Curvature in Discontinuous Galerkin-Based Level-Set Methods

Florian Kummer[1,*] and Tim Warburton[2]

[1] *Department of Mechanical Engineering, Technische Universität Darmstadt, Otto-Berndt-Straße 2, 64287 Darmstadt, Germany.*
[2] *Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005-1892, USA.*

Received 19 November 2014; Accepted (in revised version) 14 July 2015

Abstract. In two-phase flow simulations, a difficult issue is usually the treatment of surface tension effects. These cause a pressure jump that is proportional to the curvature of the interface separating the two fluids. Since the evaluation of the curvature incorporates second derivatives, it is prone to numerical instabilities. Within this work, the interface is described by a level-set method based on a discontinuous Galerkin discretization. In order to stabilize the evaluation of the curvature, a patch-recovery operation is employed. There are numerous ways in which this filtering operation can be applied in the whole process of curvature computation. Therefore, an extensive numerical study is performed to identify optimal settings for the patch-recovery operations with respect to computational cost and accuracy.

AMS subject classifications: 76T99, 65Z05, 65D10

Key words: Discontinuous Galerkin (DG), multiphase flows, curvature, patch recovery, filtering.

1 Introduction and motivating example

In order to simulate immiscible two-phase flows, one usually has to consider surface tension effects. These induce a pressure jump which is proportional to the total curvature κ of the fluid interface $\mathfrak{I}$. Precisely, the momentum equation for the fluid domains $\mathfrak{A}$ and $\mathfrak{B}$ is given as

$$\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \operatorname{div}(\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla \psi = \mu \Delta \boldsymbol{u}, \tag{1.1}$$

*Corresponding author. *Email addresses:* `kummer@fdy.tu-darmstadt.de` (F. Kummer), `timwar@rice.edu` (T. Warburton)

while at the fluid interface $\mathfrak{I}=\overline{\mathfrak{A}}\cap\overline{\mathfrak{B}}$ the velocity $\boldsymbol{u}$ and pressure ψ are coupled via the jump condition

$$\left[\!\!\left[\left(\psi\boldsymbol{I}-\mu\left(\nabla\boldsymbol{u}+\nabla\boldsymbol{u}^T\right)\right)\boldsymbol{n}_{\mathfrak{I}}\right]\!\!\right]=\sigma\kappa\boldsymbol{n}_{\mathfrak{I}}, \tag{1.2}$$

see e.g. [11] or [17]. We briefly introduce the notation required for this formula:

- The computational domain $\Omega\subset\mathbb{R}^D$ is decomposed into fluid $\mathfrak{A}$, fluid $\mathfrak{B}$ and the $(D-1)$-dimensional interface $\mathfrak{I}$, i.e. $\Omega=\mathfrak{A}\cup\mathfrak{I}\cup\mathfrak{B}$. Regarding this work, we restrict ourselves to the case $D=2$.

- $\boldsymbol{n}_{\mathfrak{I}}$ denotes the normal vector on $\mathfrak{I}$, oriented so that "it points from $\mathfrak{A}$ to $\mathfrak{B}$" and κ denotes the total curvature of $\mathfrak{I}$. We assume $\mathfrak{I}$ to be smooth enough, so that both, $\boldsymbol{n}_{\mathfrak{I}}$ and κ are at least in $\mathcal{C}^0(\mathfrak{I})$.

- the jump operator for $u\in\mathcal{C}^0(\Omega\setminus\mathfrak{I})$ is defined as

$$[\![u]\!](\boldsymbol{x})=\lim_{\xi\searrow 0}\left(u(\boldsymbol{x}+\xi\boldsymbol{n}_{\mathfrak{I}})-u(\boldsymbol{x}-\xi\boldsymbol{n}_{\mathfrak{I}})\right). \tag{1.3}$$

- μ and ρ denote the dynamic viscosity and the density of the fluid, which are usually constant within either $\mathfrak{A}$ and $\mathfrak{B}$, but have a jump at the interface.

This setting may be described by a level-set function $\varphi\in\mathcal{C}^2(\Omega)$, which fulfills

$$\begin{cases}\varphi<0 & \text{in } \mathfrak{A},\\ \varphi>0 & \text{in } \mathfrak{B},\\ \varphi=0 & \text{on } \mathfrak{I}.\end{cases} \tag{1.4}$$

Then, obviously,

$$\boldsymbol{n}_{\mathfrak{I}}=\frac{\nabla\varphi}{|\nabla\varphi|_2} \qquad \text{and} \qquad \kappa=\operatorname{div}(\boldsymbol{n}_{\mathfrak{I}}). \tag{1.5}$$

The latter expression is also called Bonnet's formula. In dependence of the level-set gradient $\nabla\varphi$ and the level-set Hessian $\partial^2\varphi$, it can be expressed as

$$\operatorname{div}\left(\frac{\nabla\varphi}{|\nabla\varphi|_2}\right)=:\operatorname{curv}(\nabla\varphi,\partial^2\varphi) \tag{1.6}$$

with

$$\operatorname{curv}(\boldsymbol{g},\boldsymbol{H})=\frac{\operatorname{tr}(\boldsymbol{H})}{|\boldsymbol{g}|_2}-\frac{\boldsymbol{g}^T\boldsymbol{H}\boldsymbol{g}}{|\boldsymbol{g}|_2^3}. \tag{1.7}$$

Note that by the introduction of φ, the properties $\boldsymbol{n}_{\mathfrak{I}}$ and κ, which were initially only defined on $\mathfrak{I}$, were smoothly extended to the whole domain Ω.

The purpose of this paper is to benchmark various filtering strategies for Bonnet's formula, based on patch recovery. In order to assess the quality of a filtering strategy, the