A high-order elliptic PDE based level set reinitialisation method using a discontinuous Galerkin discretisation

Thomas Adams[a], Stefano Giani[a], William M. Coombs[a]

[a]*Department of Engineering, Durham University, South Road, Durham, DH1 3LE, UK*

Abstract

In this paper, an efficient, high-order accurate, level set reinitialisation method is proposed, based on the elliptic reinitialisation method (Basting and Kuzmin, 2013 [1]), which is discretised spatially using the discontinuous Galerkin (DG) symmetric interior penalty method (SIPG). In order to achieve this a number of improvements have been made to the elliptic reinitialisation method including; reformulation of the underlying minimisation problem driving the solution; adoption of a Lagrange multiplier approach for enforcing a Dirichlet boundary condition on the implicit level set interface; and adoption of a narrow band approach. Numerical examples confirm the high-order accuracy of the resultant method by demonstrating experimental orders of convergence congruent with optimal convergence rates for the SIPG method, that is h^{p+1} and h^p in the L^2 and DG norms respectively. Furthermore, the degree to which the level set function satisfies the Eikonal equation improves proportionally to h^p, and the often ignored homogeneous Dirichlet boundary condition on the interface is shown to be satisfied accurately with a rate of convergence of at least h^2 for all polynomial orders.

Keywords: Level set, Reinitialisation, Discontinuous Galerkin.

1. Introduction

The level set method is a popular technique used for representing and tracking evolving interfaces in computer simulations which has found use across a wide range of areas interesting to computational physicists and engineers. This includes applications in fluid dynamics [2], shape optimisation [3], computer vision [4] and biomechanics [5] to name just a few; an extensive review

Email addresses: thomas.d.adams@durham.ac.uk (Thomas Adams), stefano.giani@durham.ac.uk (Stefano Giani), w.m.coombs@durham.ac.uk (William M. Coombs)

into the methods surrounding and further applications of the level set method can be found in the textbooks written on the subject, in particular those by Sethian [4], and Osher and Fedkiw [6]. The aim of the work to be presented in this article is to extend the level set methodology, through the development of a level set reinitialisation method which employs a discontinuous Galerkin (DG) spatial discretisation. The remainder of the introduction is divided into the three following sections: Section 1.1 provides an introduction to the level set method, Section 1.2 provides an introduction to DG methods, and finally Section 1.3 provides a review of the level set reinitialisation literature.

1.1. Level set method

The level set method was originally developed by Osher and Sethian [7], in 1988. The idea behind the level set method is to use a real scalar valued function, ϕ, called a level set function to divide a problem domain, Ω, into a number of subdomains. In general, there will be two subdomains and an interface between the two which can be defined by the value of the level set function at any point in the domain. This can be stated as

$$\begin{aligned} \phi &> 0 \quad \text{in } \Omega \backslash D, \\ \phi &= 0 \quad \text{on } \Gamma \text{ and} \\ \phi &< 0 \quad \text{in } D, \end{aligned} \tag{1}$$

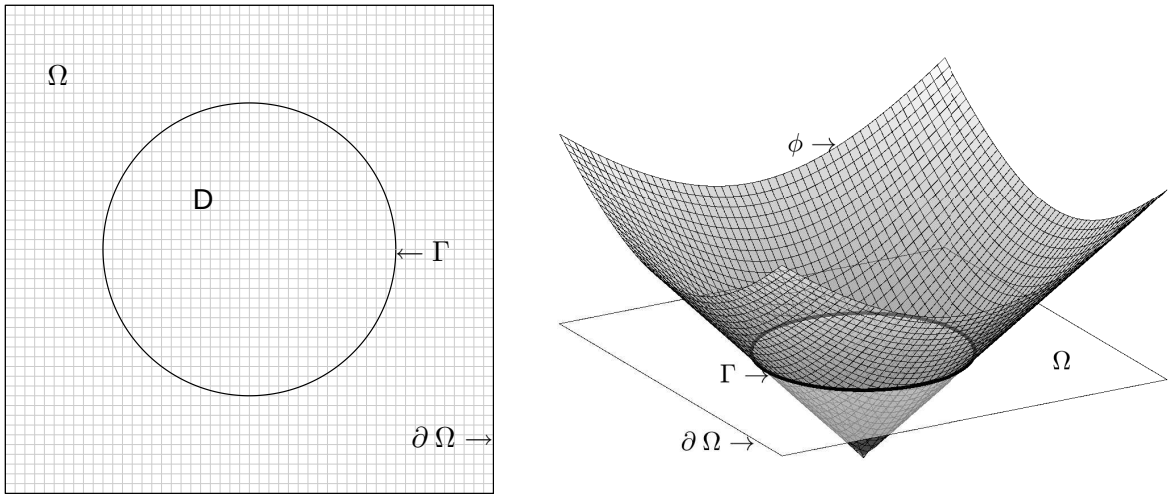
where Ω is the problem domain, Γ denotes the level set function's zero isocontour also known as the interface (i.e. the interface between the subdomains) and D is a subdomain implied by the level set function. An example of a circular interface defined by a level set function in a square domain can be seen in Figure 1.

The level set function can be evolved through the solution of a scalar transport problem, sometimes called the level set equation, which can be stated as

$$\frac{\partial \phi}{\partial t_e} + \boldsymbol{b} \cdot \boldsymbol{\nabla} \phi = 0, \tag{2}$$

where t_e is pseudotime as related to the evolution equation and $\boldsymbol{b}$ is the interface velocity.

When evolving a level set function through the solution of (2), the level set interface can only be transported along its normal, and as such it is a natural choice to initialise the level set function as a signed distance function to the interface. That is $\phi = \pm \text{dist}(\boldsymbol{x}, \Gamma)$, where $\text{dist}(\boldsymbol{x}, \Gamma)$ is the minimum distance from the point $\boldsymbol{x}$ to the interface, Γ, and the sign of the function is defined as



(a) The problem domain with interface, Γ.

(b) The level set function intersecting the problem domain.

Figure 1: Graphical representation of a discrete level set function on a square domain.

positive for $\phi \in \Omega \backslash D$ and negative for $\phi \in D$, using the notation from equation (1). One property of a signed distance function is that it will satisfy the Eikonal equation, which can be stated as

$$|\boldsymbol{\nabla}\phi| - 1 = 0. \tag{3}$$

The example level set function shown in Figure 1(b), is a signed distance function to the circular interface.

For a given velocity field, $\boldsymbol{b}$, it is unlikely that after any given time step in the solution of the evolution equation, that the level set function will maintain the properties of a signed distance function. Whilst it is not required that the level set function must maintain these properties, it is often preferred, as it has been shown that large variations in the gradient of the level set function, can cause numerical instability during the solution of the level set evolution problem, [8]. The desire to maintain the level set function as a signed distance function led Chopp [9], to introduce the idea of reinitialisation, by which, between iterations of the level set evolution problem, the level set function can be reinitialised as a signed distance function to the new interface. Reinitialisation makes it possible to ensure that for all time, t_e, the level set function is, as defined by the user, a 'good' approximation of a signed distance function, and therefore allows one to generate numerically stable results.

In terms of a full level set methodology it can be observed that there is a further advantage to always ensuring that the level set function satisfies the Eikonal equation. Given that the advection velocity can be written, $\boldsymbol{b} = b\boldsymbol{n_\phi}$ where, $\boldsymbol{n_\phi} = \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}$, is the normal of the level set function, and b, is the scalar magnitude of the advection velocity normal to the interface, then the evolution equation (2) can be simplified as follows

$$\frac{\partial \phi}{\partial t_e} = -b|\boldsymbol{\nabla}\phi| = -b. \tag{4}$$

In this way, much of the complexity in solving the evolution equation is now exchanged for the complexity of solving the reinitialisation problem (with the potential added expense of having to reinitialise more often).

1.2. Discontinuous Galerkin methods

DG methods are a class of non-conforming finite element methods where the test and trial functions are not continuous across the faces and edges of the mesh, [10]. This decoupling between adjacent elements allows the methods to be both highly parallelisable and well suited for $hp-$adaptivity, which in turn should allow one to develop methodologies which are both efficient and high-order accurate, [11]. This would be advantageous for use with some of the more expensive applications of the level set method, topology optimisation for example. Furthermore, when it comes to the level set method, DG methods are particularly well suited for solving the PDE's which naturally arise in the level set context, i.e. hyperbolic/advection-dominated problems, due to the built in stabilisation mechanisms DG methods possess, [12]. Details of the DG methods as applied to elliptic problems can be found in [10]. We choose in this work to use the discontinuous Galerkin symmetric interior penalty (SIPG) method. There are a number of reasons why SIPG can be considered preferable to other DG discretisations for elliptic problems, some of which are outlined in the conclusions of [13]. In particular, we highlight the well established optimal rates of convergence which are not possible for the nonsymmetric DG methods (which includes the nonsymmetric interior penalty method (NIPG), [14], and the method of Baumann-Oden (BO), [15]), as well as the increased efficiency in terms of the linear solve (compared with the nonsymmetric methods) and memory requirements (compared with the Local Discontinuous Galerkin method (LDG), [16]).

1.3. Level set reinitialisation

There are many existing methods which are capable of reinitialising a level set function as a signed distance function. In general these fall into two categories: geometric methods, and PDE

based methods. Geometric methods reinitialise the level set function at discrete nodal points by measuring the distance from the nodes to the level set interface, and using this information as well as the sign of the level set function pre-reinitialisation to generate a signed distance function. PDE based methods involve generating a signed distance function, by solving a PDE. PDE based methods can be further categorised into two types. One type of PDE based reinitialisation we call *pure reinitialisation* methods, by which between iterations of the evolution equation, a separate PDE is solved, the solution of which will be a signed distance function to the interface. The other type of PDE based reinitialisation methods are known as *all-in-one*, by which the level set equation (4) itself is modified to include a constraint enforcing that the level set function always satisfies the Eikonal equation (3), such that both the evolution of the interface and reinitialisation of the level set function are computed simultaneously. In this section the literature surrounding level set reinitialisation, which includes all of the types mentioned in this paragraph, will be reviewed, with a focus on where such methods have been applied to DG.

It should be noted that the aim of any of the reinitialisation methods to be discussed below is to ensure that at the beginning of each iteration of the solution of the level set equation, the level set function is a signed distance function to the current position of the level set interface; i.e. the interface returned at the end of the previous iteration of the evolution equation. In this sense, all of the reinitialisation techniques discussed below are equivalent, however, they vary in terms of their computational efficiency, stability and accuracy, especially when applied to a DG level set methods.

The original reinitialisation method introduced by Chopp [9], reinitialises the level set function as a signed distance function using a direct geometric approach. This approach works by first explicitly discretising the interface, $\Gamma_h \sim \Gamma$, and then at each point in the problem domain setting the value of the level set function equal to the minimum distance from that point, to the discretised interface multiplied by the sign of the original level set function at that point, which can be stated as

$$\phi = \text{sign}(\phi^0)\,\text{dist}(\boldsymbol{x}, \Gamma_h(\phi^0)), \tag{5}$$

where ϕ^0 is the pre-reinitialisation level set function, and $\text{sign}(\cdot)$ denotes the signum function.

Whilst conceptually simple, there are a number of issues with the geometric reinitialisation method. One of the key advantages of the level set method in terms of computational efficiency, is the implicit nature of the evolving interface. Not only is this advantage surrendered by discretising the interface, but the expense required to both discretise the interface and compute the minimum

distance at each mesh node, to the discrete interface, increases with mesh density, with the number of points used to discretise the interface and with the length of the interface itself. Chopp notes that the complexity of this reinitialisation method is $\mathcal{O}(n^6)$, [9]. Furthermore, adaptation of such an approach to a discontinuous Galerkin discretisation also poses some additional difficulties. As there is no longer a requirement of continuity across element edges, the zero isocontour can be discontinuous and thus the computation of the distance from a point to the interface can become problematic. Similarly, the sign of the level set function at each degree of freedom for a given node will not necessarily be well-defined, particularly if a node is near to the interface. These problems will either cause strong discontinuities to develop in the level set function, or lead to a smoothing of the level set function which will cause movement in the position of the interface post-reinitialisation. Lastly, when using the geometric reinitialisation method, the approximation of the interface on each element will only be first-order, and any benefits arising from the high-order approximations possible through the use of DG methods will be surrendered each time the reinitialisation routine is called.

One of the most popular methods of reinitialisation is a PDE based, pure reinitialisation method, often referred to as the hyperbolic reinitialisation method, which was introduced by Sussman *et al.* [17]. This method involves solving a hyperbolic PDE which can be stated as

$$\frac{\partial \phi}{\partial t_r} = \text{sign}(\phi^0)\left(1 - |\boldsymbol{\nabla}\phi|\right), \tag{6}$$

where t_r denotes pseudotime as related to the reinitialisation problem. The steady state solution to (6) will be achieved once the level set function provides a sufficient approximate solution to the Eikonal equation, (3). The multiplication by the sign of the pre-reinitialisation level set function, ϕ^0, works as a weak Dirichlet boundary condition on the interface of the level set function, and thus the resulting function will be a signed distance function to the interface, $\Gamma(\phi^0)$.

The main issue with the hyperbolic reinitialisation method is that the (potentially poor) characteristics of the original level set function, ϕ^0, can be propagated during the reinitialisation process, which most often presents as a 'smearing' of the interface [18]. Mousavi [19], presented a solution to (6), using a DG discretisation along with a third-order Runge-Kutta scheme in time. Mousavi outlines quite clearly the difficulties encountered in trying to produce a stable solution using these methods of discretisation; the results presented demonstrate that at some point in pseudotime the solution will always gradually begin to diverge. Independent work done by the author of this paper,

instead using an explicit Euler discretisation in time, found a similar issue when trying to solve the hyperbolic reinitialisation problem using a spatial DG discretisation. Mousavi [19] found that it was possible to create a method which was practically viable by utilising a severe time step restriction, a sufficiently smoothed signum function and including an artificial viscosity term. Such a solution to the reinitialisation problem is less than ideal however, as a large number of iterations are required to return a signed distance function everywhere in the domain, which could be considered prohibitively expensive. Similar issues were found by Karakus *et al.* [20], in which the author takes advantage of the high level of parallelisation possible with DG methods to speed up the computation of the resulting reinitialisation method.

Gomes and Faugeras [21], showed that the resulting level set function when solving a Hamilton-Jacobi equation would not in general satisfy the Eikonal equation. They proposed modifying the evolution equation as follows

$$\frac{\partial \phi}{\partial t_e} = b(\boldsymbol{x} - \phi \boldsymbol{\nabla} \phi), \tag{7}$$

such that it was no longer a Hamilton-Jacobi equation, thus developing the first all-in-one type method. Whilst theoretically such a formulation should force the level set function to maintain it's signed distance properties, it was found that once discretised there could still be a drift in the level sets leading to a loss of the signed distance property over time [22]. This idea however, prompted other all-in-one type methods whereby the evolution equation is modified to include a signed distance constraint, such that at each time step the resulting level set function is a solution to both the evolution problem and the Eikonal equation. For example, Weber *et al.* [22], set up their evolution problem as an optimisation problem driven by an error functional which minimises deviations in the desired interface movement and also deviations from the signed distance property. A similar solution was presented by Li *et al.* [23] whereby the level set evolution problem was reframed as an optimisation problem including an energy driving the evolution and a penalty term restricting deviation from a signed distance function. This lead to a formulation of the evolution equation which could be stated as

$$\frac{\partial \phi}{\partial t_e} = \underbrace{b|\boldsymbol{\nabla}\phi|}_{\text{advection term}} + \underbrace{\alpha \boldsymbol{\nabla} \cdot \left(\boldsymbol{\nabla}\phi - \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|} \right)}_{\text{signed distance constraint}}, \tag{8}$$

where α is a penalty parameter. Later, Li *et al.* [24], named this, distance regularised level set evolution (DRLSE).

Basting and Kuzmin [1], took the distance regularisation part of the DRLSE, and considered it as a pure reinitialisation problem, which is a parabolic PDE and can be stated as

$$\frac{\partial \phi}{\partial t_r} = \boldsymbol{\nabla} \cdot \left(\boldsymbol{\nabla} \phi - \frac{\boldsymbol{\nabla} \phi}{|\boldsymbol{\nabla} \phi|} \right). \tag{9}$$

By removing the time dependent part, so as to avoid pseudotime stepping, and also including an appropriate boundary condition, Basting and Kuzmin reformulated the problem as a quasilinear elliptic PDE to be solved iteratively which can be stated as

$$\boldsymbol{\nabla} \cdot \left(\boldsymbol{\nabla} \phi - \frac{\boldsymbol{\nabla} \phi}{|\boldsymbol{\nabla} \phi|} \right) + \gamma \phi = 0, \tag{10}$$

where γ is a penalty parameter. The work presented in this paper provides a solution to the elliptic reinitialisation problem using a DG method for the spatial discretisation.

Whilst the work presented in this paper was completed independently, Utz *et al.* [25] recently presented a similar DG solution to the elliptic reinitialisation problem. However, a number of issues were found with the work presented in [1] and [25], solutions to which are discussed here. Explicitly, in this paper issues are addressed concerning: boundary conditions on an implicit surface; experimental orders of convergence which align with the theoretically optimal rates of convergence in the relevant norms through the use of a narrow band approach; and the construction of a new potential function which removes the issues when reinitialising level set functions with small gradients, i.e. $|\boldsymbol{\nabla} \phi| \leq 0.5$.

The rest of the paper is organised as follows. Section 2 presents the proposed elliptic reinitialisation method. Section 3 presents three numerical examples as demonstrations of the efficacy of the proposed method. The article is then concluded in Section 4.

2. Discontinuous Galerkin elliptic level set reinitialisation

Section 2 consists of the following subsections. Section 2.1 presents the mathematical preliminaries required for the discussion of DG methods. Section 2.2 presents the elliptic reinitialisation problem and the proposed DG discretisation. Section 2.3 presents a discussion on the use of a narrow band approach, which is required to allow one to demonstrate optimal rates of convergence.

2.1. Symmetric interior penalty discontinuous Galerkin method preliminaries

Let $\mathcal{T}_h$ denote any partition of a domain, Ω, into nonoverlapping quadrilateral elements, τ, with element size, h, such that the computational domain can be defined, $\Omega = \cup_{\tau \in \mathcal{T}_h} \tau$, with boundary vertices on $\partial\Omega$. The skeleton of the mesh, S, is defined as the set of all interior edges, that is $S = \cup_{\tau \in \mathcal{T}_h} \partial\tau \backslash \partial\Omega$. The unit outward normal on the boundary, $\partial\tau$, of a given element, τ, is denoted as $\hat{\boldsymbol{n}}$. For any mesh $\mathcal{T}_h$ of Ω, with elements of maximum polynomial degree, p, the DG finite element space is defined as

$$V_{DG}(\mathcal{T}_h) := \{v \in L^2(\Omega) : \ \forall \tau \in \mathcal{T}_h, \ v|_\tau \in \mathcal{Q}_p(\tau)\}, \tag{11}$$

where $\mathcal{Q}_p(\tau)$ denotes the space of polynomials of degree no more than, p, in each coordinate direction.

It should be noted that the work to be presented here is restricted to regular quadrilateral elements, on Cartesian grids. This is due to the Eulerian framework within which the level set method operates, which allows one to exploit the simplicity of such an approach.

2.2. Elliptic level set reinitialisation

The reinitialisation problem can be stated, for a given level set function, ϕ^0, find a new level set function, ϕ, which is a signed distance function to the original position of the level set interface, $\Gamma(\phi^0)$. This can be stated mathematically as finding a solution to the Eikonal equation, stated in equation (3), relative to the following Dirichlet boundary condition

$$\phi = 0 \quad \text{on } \Gamma(\phi^0). \tag{12}$$

As first presented by Basting and Kuzmin in [1], the elliptic reinitialisation method aims to solve the level set reinitialisation problem by minimising the least squares residual to the Eikonal equation, (3), that is

$$\min \left(\int_\Omega \frac{1}{2} (|\boldsymbol{\nabla}\phi| - 1)^2 \, \mathrm{d}\boldsymbol{x} \right). \tag{13}$$

Taking the derivative of the objective functional (13), leads to a strong formulation of the problem which can be stated as

$$\begin{aligned} \boldsymbol{\nabla} \cdot \left(\boldsymbol{\nabla}\phi - \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|} \right) &= 0 && \text{in } \Omega, \\ \phi &= 0 && \text{on } \Gamma(\phi^0) \text{ and} \\ \boldsymbol{\nabla}\phi \cdot \hat{\boldsymbol{n}} &= \text{sign}(\phi^0) && \text{on } \partial\Omega. \end{aligned} \tag{14}$$

The first equation forming (14) is a diffusion equation which will have positive diffusion where $|\boldsymbol{\nabla}\phi| > 1$ and negative diffusion where $|\boldsymbol{\nabla}\phi| < 1$, with a solution at $|\boldsymbol{\nabla}\phi| = 1$. There is a homogeneous Dirichlet boundary condition which ensures that there is a unique solution defined by the position of the pre-reinitialisation level set interface, as well as a Neumann boundary condition on the natural boundary stating that the gradient of the solution at the domain boundary must also be equal to the sign of the pre-reinitialisation level set function at that point. This Neumann boundary condition actually exists as a homogeneous Neumann boundary condition, as it could be rewritten as

$$\left(\boldsymbol{\nabla}\phi - \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}\right) \cdot \hat{\boldsymbol{n}} = 0 \quad \text{on } \partial\Omega. \tag{15}$$

Applying a Picard linearisation to the terms which are nonlinear with respect to $\boldsymbol{\nabla}\phi$, allows one to rewrite the above diffusion equation as

$$\boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\phi^m = \boldsymbol{\nabla} \cdot \frac{\boldsymbol{\nabla}\phi^{m-1}}{|\boldsymbol{\nabla}\phi^{m-1}|} \quad \text{in } \Omega, \tag{16}$$

where the superscript denotes the m^{th} iteration. Discretising the problem spatially using the SIPG method then leads to a variational formulation which can be stated as; find $\phi_h^m \in V_{DG}$, as $m \to \infty$, such that the following weak form statement of equilibrium is satisfied

$$\int_\Omega \boldsymbol{\nabla}\phi_h^m \cdot \boldsymbol{\nabla} v \, \mathrm{d}\boldsymbol{x} - \int_S \{\{\boldsymbol{\nabla}\phi_h^m\}\} \cdot [\![v]\!] \, \mathrm{d}s - \int_S [\![\phi_h^m]\!] \cdot \{\{\boldsymbol{\nabla} v\}\} \, \mathrm{d}s + \mu \int_S [\![\phi_h^m]\!] \cdot [\![v]\!] \, \mathrm{d}s =$$

$$\int_\Omega \frac{\boldsymbol{\nabla}\phi_h^{m-1}}{|\boldsymbol{\nabla}\phi_h^{m-1}|} \cdot \boldsymbol{\nabla} v \, \mathrm{d}\boldsymbol{x} - \int_S \left\{\left\{ \frac{\boldsymbol{\nabla}\phi_h^{m-1}}{|\boldsymbol{\nabla}\phi_h^{m-1}|} \right\}\right\} \cdot [\![v]\!] \, \mathrm{d}s, \quad \forall v \in V_{DG}(\mathcal{T}_h), \tag{17}$$

were μ is a penalty parameter, henceforth referred to as the *discontinuity penalisation parameter*, which for elliptic problems can be chosen as $\mu = Cp_e^2/h_e$, where C is a constant usually equal to 10, p_e is the maximum polynomial order of the two elements sharing that edge, and h_e is the length of the edge. For further information on the discontinuity penalisation parameter in quasilinear elliptic problems, please refer to [26]. The jump and average operators denoted by $[\![\cdot]\!]$ and $\{\{\cdot\}\}$ respectively, are as defined in [10] and are reproduced here as follows: for an arbitrary scalar valued function,

ψ, and vector valued function, $\boldsymbol{\Psi}$, on adjacent elements, τ^+ and τ^-, which share an edge

$$[\![\psi]\!] = \begin{cases} (\psi^+ - \psi^-)\hat{\boldsymbol{n}}^+ & \text{on } \partial\tau \backslash \partial\Omega, \\ \psi^+ \hat{\boldsymbol{n}}^+ & \text{on } \partial\tau \cap \partial\Omega, \end{cases} \tag{18}$$

$$\{\{\boldsymbol{\Psi}\}\} = \begin{cases} (\boldsymbol{\Psi}^+ + \boldsymbol{\Psi}^-)/2 & \text{on } \partial\tau \backslash \partial\Omega, \\ \boldsymbol{\Psi}^+ & \text{on } \partial\tau \cap \partial\Omega. \end{cases} \tag{19}$$

The resulting linear system can then be solved using a fixed point iterative method as follows; find $\phi_h^m \in V_{\mathrm{DG}}$, as $m \to \infty$, such that

$$K\phi_h^m = F(\phi^{m-1}), \tag{20}$$

where the matrix $K = (k_{ij})$, has elements given by

$$k_{ij} = \int_\Omega \boldsymbol{\nabla} v_j \cdot \boldsymbol{\nabla} v_i \, \mathrm{d}\boldsymbol{x} - \int_S \{\{\boldsymbol{\nabla} v_j\}\} \cdot [\![v_i]\!] \, \mathrm{d}s - \int_S [\![v_j]\!] \cdot \{\{\boldsymbol{\nabla} v_i\}\} \, \mathrm{d}s + \mu \int_S [\![v_j]\!] \cdot [\![v_i]\!] \, \mathrm{d}s, \tag{21}$$

and the column vector $F = (f_i)$, has elements given by

$$f_i = \int_\Omega \frac{\boldsymbol{\nabla}\phi_h^{m-1}}{|\boldsymbol{\nabla}\phi_h^{m-1}|} \cdot \boldsymbol{\nabla} v_i \, \mathrm{d}\boldsymbol{x} - \int_S \left\{\!\!\left\{ \frac{\boldsymbol{\nabla}\phi_h^{m-1}}{|\boldsymbol{\nabla}\phi_h^{m-1}|} \right\}\!\!\right\} \cdot [\![v_i]\!] \, \mathrm{d}s. \tag{22}$$

It should be noted that the above formulation is incomplete as it does not enforce the Dirichlet boundary condition. This will be discussed separately in Section 2.2.2. The homogeneous Neumann boundary condition is, however, naturally satisfied in the above formulation.

Modifications to the above formulation will be discussed in the following subsections. Section 2.2.1 presents a discussion on the reformulation of the elliptic reinitialisation problem by modifying the underlying objective functional such that the proposed reinitialisation method is better suited for dealing with level set functions with small gradients. Section 2.2.2 presents a discussion on methods for imposing Dirichlet boundary conditions on implicit surfaces. Section 2.2.3 presents a discussion on methods for integration on implicit surfaces. Each section concludes with the method adopted in this work.

2.2.1. Objective functionals for the elliptic reinitialisation problem

The reinitialisation method presented in Section 2.2, begins by attempting to minimise the residual to the Eikonal equation by taking the most natural form of a functional, the minimisation

of which would be equivalent to the minimisation of the least squares residual to the Eikonal equation, i.e. one could rewrite the problem in (13) as

$$\min\left(\int_{\Omega} t_1(|\boldsymbol{\nabla}\phi|)\,\mathrm{d}\boldsymbol{x}\right), \tag{23}$$

where

$$t_1(|\boldsymbol{\nabla}\phi|) = \frac{1}{2}(|\boldsymbol{\nabla}\phi| - 1)^2. \tag{24}$$

Choosing the objective functional to be equal to t_1, leads to a diffusion term in the weak formulation which could be stated as

$$\boldsymbol{\nabla}\cdot(d_1(|\boldsymbol{\nabla}\phi|)\boldsymbol{\nabla}\phi) = 0, \tag{25}$$

where

$$d_1(|\boldsymbol{\nabla}\phi|) = 1 - \frac{1}{|\boldsymbol{\nabla}\phi|}. \tag{26}$$

It can be seen that the diffusion functional, d_1, becomes singular as $|\boldsymbol{\nabla}\phi| \to 0$. To combat this problem, authors such as Li [24] and Basting [1], have modified the objective functional such that it minimises the least squares residual to the Eikonal equation everywhere except in the region where $|\boldsymbol{\nabla}\phi|$ is small. For example, [1] presents the following functional

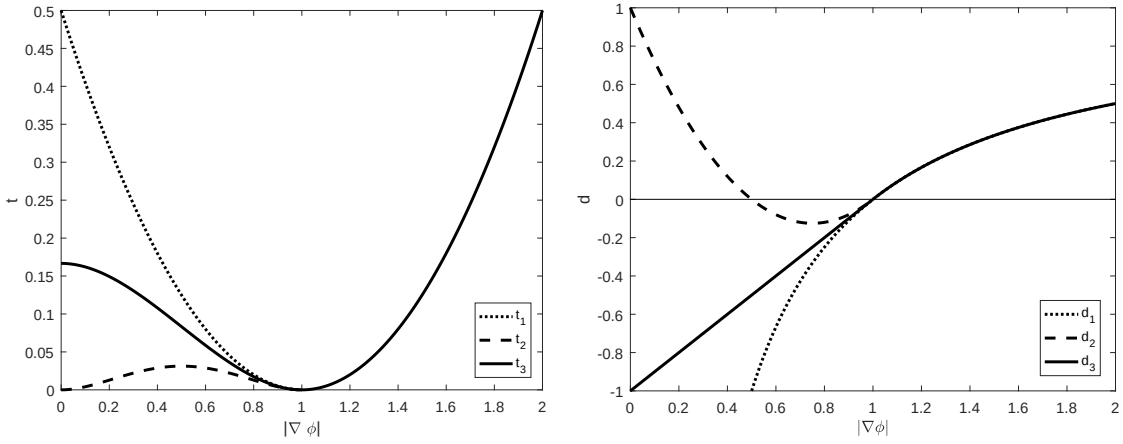
$$t_2(|\boldsymbol{\nabla}\phi|) = \begin{cases} \frac{1}{2}(|\boldsymbol{\nabla}\phi| - 1)^2 & \text{if } |\boldsymbol{\nabla}\phi| > 1, \\ \frac{1}{2}|\boldsymbol{\nabla}\phi|^2(|\boldsymbol{\nabla}\phi| - 1)^2 & \text{if } |\boldsymbol{\nabla}\phi| \leq 1, \end{cases} \tag{27}$$

which leads to a diffusion term

$$d_2(|\boldsymbol{\nabla}\phi|) = \begin{cases} 1 - \frac{1}{|\boldsymbol{\nabla}\phi|} & \text{if } |\boldsymbol{\nabla}\phi| > 1, \\ 1 - (3|\boldsymbol{\nabla}\phi| - 2|\boldsymbol{\nabla}\phi|^2) & \text{if } |\boldsymbol{\nabla}\phi| \leq 1. \end{cases} \tag{28}$$

Figure 2 shows a plot of the objective and diffusion functionals presented in this section. For the objective functional, t_2, it can be observed that there are two solutions to the minimisation problem, one corresponding to the Eikonal equation, and a second at $|\boldsymbol{\nabla}\phi| = 0$. Furthermore, Figure 2(b), shows that for the corresponding diffusion functional, d_2, that where the gradient is small, i.e. $|\boldsymbol{\nabla}\phi| < 0.5$, the diffusion is positive, which corresponds to forcing the level set function towards the solution at $|\boldsymbol{\nabla}\phi| = 0$.

In order to overcome these two issues, here we propose a new objective functional which both avoids the singularity at $|\boldsymbol{\nabla}\phi| = 0$ and always has negative diffusion for $|\boldsymbol{\nabla}\phi| < 1$. One such



(a) Objective functionals.

(b) Diffusion functionals.

Figure 2: Three different objective functionals and their corresponding diffusion rates.

functional could be stated as

$$t_3(|\boldsymbol{\nabla}\phi|) = \begin{cases} \frac{1}{2}(|\boldsymbol{\nabla}\phi| - 1)^2 & \text{if } |\boldsymbol{\nabla}\phi| > 1, \\ \frac{(|\boldsymbol{\nabla}\phi|)^3}{3} - \frac{(|\boldsymbol{\nabla}\phi|)^2}{2} + \frac{1}{6} & \text{if } |\boldsymbol{\nabla}\phi| \leq 1, \end{cases} \tag{29}$$

which leads to a diffusion term

$$d_3(|\boldsymbol{\nabla}\phi|) = \begin{cases} 1 - \frac{1}{|\boldsymbol{\nabla}\phi|} & \text{if } |\boldsymbol{\nabla}\phi| > 1, \\ 1 - (2 - |\boldsymbol{\nabla}\phi|) & \text{if } |\boldsymbol{\nabla}\phi| \leq 1. \end{cases} \tag{30}$$

It should be stated that conceptually any function which satisfies these conditions would suffice. Figure 2 demonstrates that the objective functional, t_3, does indeed satisfy both of these conditions.

To include any of the above defined diffusion functionals using the formulation presented in Section 2.2, the only modification required to the linear system stated in (20) is the entries to the F vector, which can now be written

$$f_i = \int_\Omega (1 - d_k(|\boldsymbol{\nabla}\phi_h^{m-1}|))\boldsymbol{\nabla}\phi_h^{m-1} \cdot \boldsymbol{\nabla} v_i \, \mathrm{d}\boldsymbol{x} - \int_S \left\{\left\{(1 - d_k(|\boldsymbol{\nabla}\phi_h^{m-1}|))\boldsymbol{\nabla}\phi_h^{m-1}\right\}\right\} \cdot [\![v_i]\!] \, \mathrm{d}s, \; k = 1, 2, 3. \tag{31}$$

Figure 3 demonstrates by example the relative performance of these different objective functionals. A level set function, $\phi^0 = -(|x|/2) + 0.5$, is projected onto a mesh of 38 square elements on the domain $\Omega = (-2, 2) \times (0, 8/19)$ with $h = 4/19$, such that a singularity falls at the centre of the

2 central elements. Using a mesh of linear elements, both components of the gradient throughout these elements will therefore be close to zero, and everywhere else in the mesh the gradient can also be considered small, i.e. $|\boldsymbol{\nabla}\phi| \leq 0.5$. The initial projection of the level set function can be seen in Figure 3(a). For these examples, the Dirichlet boundary condition on the level set interface is enforced using the Lagrange multiplier method, described in Section 2.2.2, along with the integration method of Müller *et al.* [27] described in Section 2.2.3, and the solution is considered to have converged when $\sum(\phi^m - \phi^{m-1}) < 10^{-8}$, that is when the relative change between iterations is less than a threshold value.

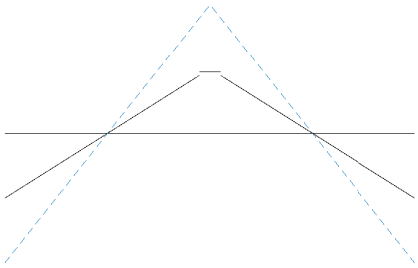
When using the objective functional, t_1, it can be observed that the solution immediately begins to oscillate and does not converge. Figure 3(b) shows a snapshot of the level set function after 50 iterations when using t_1. It can be seen that the attempt to correct the almost zero gradients in the centre element, leads to an overcorrection causing the level set function to twist as it tries to force the gradient back to unity, after which the solution breaks down and continues to get worse over time. Figure 3(c) shows the converged solution when using the objective functional, t_2. It can be seen that there are no longer overshoots as a result of the initial 'zero' gradients, however, some parts of the level set function converge to the additional solution at $|\boldsymbol{\nabla}\phi| = 0$. Figure 3(d) shows the converged solution using the objective functional, t_3. The limited diffusion for small gradients, removes any overshoots or oscillations, and the level set function at steady state is congruent with the analytical solution as far as possible given the coarseness of the mesh. Therefore the objective functional adopted in this work is that defined as t_3.

2.2.2. Boundary conditions on implicit surfaces

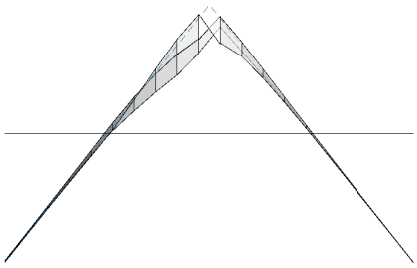
In both [1] and [25], the Dirichlet boundary condition on the level set interface is enforced using a penalty method. As such the weak formulation would be stated as, find $\phi_h^m \in V_{DG}$, as $m \to \infty$ such that

$$\int_\Omega \boldsymbol{\nabla}\phi_h^m \cdot \boldsymbol{\nabla} v \, \mathrm{d}\boldsymbol{x} - \int_S \{\{\boldsymbol{\nabla}\phi_h^m\}\} \cdot [\![v]\!] \, \mathrm{d}s - \int_S [\![\phi_h^m]\!] \cdot \{\{\boldsymbol{\nabla} v\}\} \, \mathrm{d}s + \mu \int_S [\![\phi_h^m]\!] \cdot [\![v]\!] \, \mathrm{d}s$$
$$\underbrace{+ \gamma \int_{\Gamma(\phi^0)} \phi_h^m v \, \mathrm{d}s}_{\text{penalty term}} = \int_\Omega \frac{\boldsymbol{\nabla}\phi_h^{m-1}}{|\boldsymbol{\nabla}\phi_h^{m-1}|} \cdot \boldsymbol{\nabla} v \, \mathrm{d}\boldsymbol{x} - \int_S \left\{\!\!\left\{ \frac{\boldsymbol{\nabla}\phi_h^{m-1}}{|\boldsymbol{\nabla}\phi_h^{m-1}|} \right\}\!\!\right\} \cdot [\![v]\!] \, \mathrm{d}s, \quad \forall v \in V_{DG}(\mathcal{T}_h), \quad (32)$$

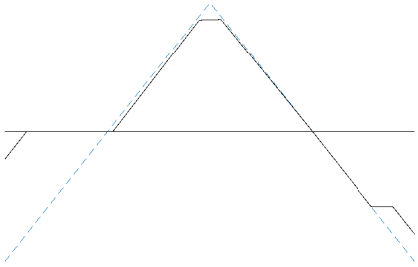
where γ is a penalty parameter, henceforth referred to as the *interface penalisation parameter*.



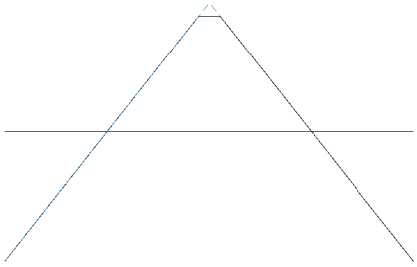
(a) Pre-reinitialisation level set function.



(b) Diverging solution after 50 iterations using objective function, t_1.

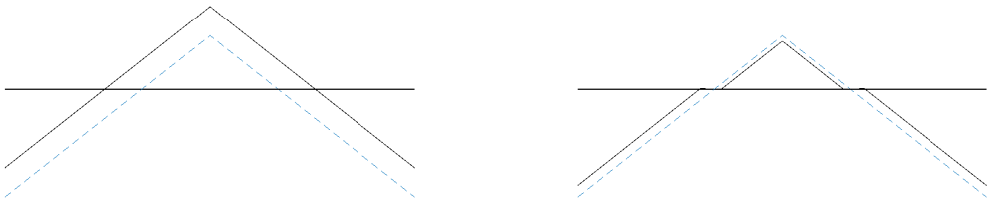


(c) Converged solution using objective functional, t_2.



(d) Converged solution using objective functional, t_3.

Figure 3: Converged solutions to a simple problem using the different objective functionals. The solid line shows the level set function, the dashed line shows the analytical solution, and the horizontal line shows the problem domain.



(a) Converged solution with $\gamma = 0$.

(b) Converged solution with $\gamma = 10^6$.

Figure 4: Effect of the value of the penalty parameter, γ, on the solution at the boundary. The solid line shows the level set function, the dashed line shows the analytical solution, and the horizontal line shows the problem domain.

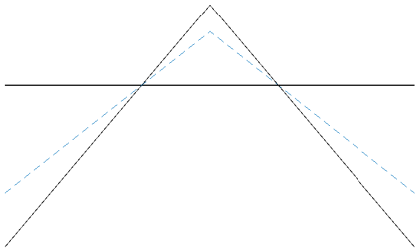
It should be noted here that throughout this section, as the interface of the original level set function, $\Gamma(\phi^0)$, is in general, immersed within an element, and does not correspond with an element edge for example, the integral over the interface, is computed as a volume integral over each element intersected by the interface, multiplied by some weighting function. Any examples presented in this section will thus be computed using the method of Müller *et al.* [27], the details of which will be discussed in Section 2.2.3.

In this work, difficulty was encountered in deciding the best choice for the value of the interface penalisation parameter, γ. Babuška *et al.* [28], note that when using a penalty method, that if the value of the penalty parameter is chosen to be too large or too small, it can significantly decrease the accuracy of the underlying method. This can be demonstrated through a simple numerical example. For all of the examples in this section, the problem is defined by an initial level set function, $\phi^0 = 1.5|x| + 1$, which is discretised with 40 square elements on $\Omega = (-2, 2) \times (0, 0.4)$, such that $h = 0.2$. Once again, the solution is considered to have converged when $\sum(\phi^m - \phi^{m-1}) < 10^{-8}$. Figure 4(a) demonstrates that if the penalty parameter is too small then there is no longer a unique solution and equation (32) holds such that the solution found satisfies the Eikonal equation, but the level set function is no longer sufficiently constrained as a rigid body in space, which appears as a movement of the interface. Figure 4(b), demonstrates that if the value of the interface penalisation parameter is too large, there will be boundary locking, [29], in elements intersected by the interface.

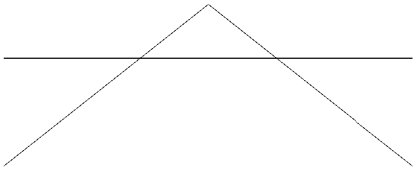
Evidence is provided in [25], which supports the idea that an appropriate choice for the value of the interface penalisation parameter for a given mesh, is equal to the discontinuity penalisation parameter, μ such that, $\gamma = \mu$. Whilst it can be observed that the interface penalisation parameter is problem dependent, it is not necessarily apparent that it is related to the mesh size in the same way as the discontinuity penalisation parameter. Repeating the example problem from the previous paragraph, with a mesh of linear elements, the interface penalisation parameter would therefore be computed, $\gamma = \frac{10p^2}{h} = 50$. As evidenced at a glance by the solution in Figure 5(b) this is an appropriate value for this penalty parameter in this case. Increasing the order of the elements to $p = 5$ causes an increase in this value to $\gamma = 1250$; Figure 5(c) shows that this value is too large and causes locking/spurious oscillations in the elements at the boundary and therefore is not appropriate. However, once again using quintic elements, but choosing $\gamma = 50$ allows one to return a solution which no longer displays locking at the boundary as shown in Figure 5(d). The same is true when changing the number of elements used to discretise the problem. This implies that the problem itself has a significant (and difficult to quantify) influence on the range of admissible values for the interface penalisation parameter. This difficulty in choosing a value of the interface penalisation parameter within the admissible range of values for a given problem led to the exploration of other possible methods for the imposition of a Dirichlet boundary condition on an implicit surface.

The literature highlights four main approaches for the imposition of Dirichlet boundary conditions on implicit surfaces; the aforementioned penalty method, Nitsche's method [30], the method of Lagrange multipliers [31], and methods involving enrichment or modification of shape functions, for example [32]. Nitsche's method is akin to the penalty method in that there is a penalty term which imposes the prescribed value on the boundary. Without re-presenting the evidence, the same arguments against using the penalty method described above were found to also be true of Nitsche's method when applied to the implicit surface. The methods involving the modification of the shape functions require *a priori* knowledge of the position of the interface, whereas the methodology here deals with evolving and implied interfaces only, and therefore methods such as these are not appropriate in this context.

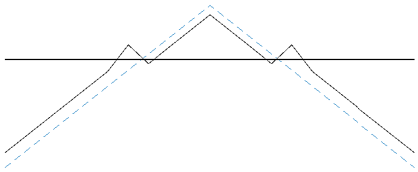
The method of Lagrange multipliers involves the reformulation of the weak form of the problem such that a new unknown, the Lagrange multiplier, λ, is to be solved for in addition to the original unknown, in this case the level set function, ϕ, such that the solution on the Dirichlet boundary is



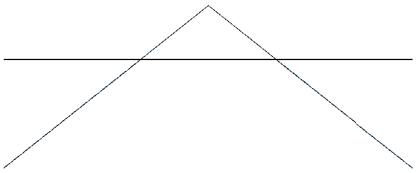
(a) Pre-reinitialisation level set function.



(b) Converged solution using linear elements with $\gamma = 50$.



(c) Converged solution using quintic elements with $\gamma = 1250$.



(d) Converged solution using quintic elements with $\gamma = 50$.

Figure 5: Examples showing problem dependency of the penalty parameter. The solid line shows the level set function, the dashed line shows the analytical solution, and the horizontal line shows the problem domain.

constrained by a prescribed value. The weak form of the elliptic reinitialisation problem can thus be reformulated: find $\phi_h^m \in V_{DG}$ and $\lambda \in L$, as $m \to \infty$ such that

$$\int_\Omega \boldsymbol{\nabla}\phi_h^m \cdot \boldsymbol{\nabla} v \, \mathrm{d}\boldsymbol{x} - \int_S \{\{\boldsymbol{\nabla}\phi_h^m\}\} \cdot [\![v]\!] \, \mathrm{d}s - \int_S [\![\phi_h^m]\!] \cdot \{\{\boldsymbol{\nabla} v\}\} \, \mathrm{d}s + \mu \int_S [\![\phi_h^m]\!] \cdot [\![v]\!] \, \mathrm{d}s$$

$$+ \underbrace{\int_{\Gamma(\phi^0)} \lambda v \, \mathrm{d}s}_{\text{LM term}} = \int_\Omega \frac{\boldsymbol{\nabla}\phi_h^{m-1}}{|\boldsymbol{\nabla}\phi_h^{m-1}|} \cdot \boldsymbol{\nabla} v \, \mathrm{d}\boldsymbol{x} - \int_S \left\{\!\!\left\{ \frac{\boldsymbol{\nabla}\phi_h^{m-1}}{|\boldsymbol{\nabla}\phi_h^{m-1}|} \right\}\!\!\right\} \cdot [\![v]\!] \, \mathrm{d}s, \quad \forall v \in V_{DG}(\mathcal{T}_h), \tag{33}$$

and

$$\int_{\Gamma(\phi^0)} \phi_h^m \zeta \, \mathrm{d}s = 0, \quad \forall \zeta \in L. \tag{34}$$

One of the difficulties of using such a method, is choosing the correct interpolation space for the Lagrange multipliers, L. One natural choice is choosing the space, L, as follows

$$L = \operatorname*{span}_{\tau \in \mathcal{T}_h^\Gamma} \{\mathcal{Q}_p(\mathcal{T}_h)\}, \tag{35}$$

where

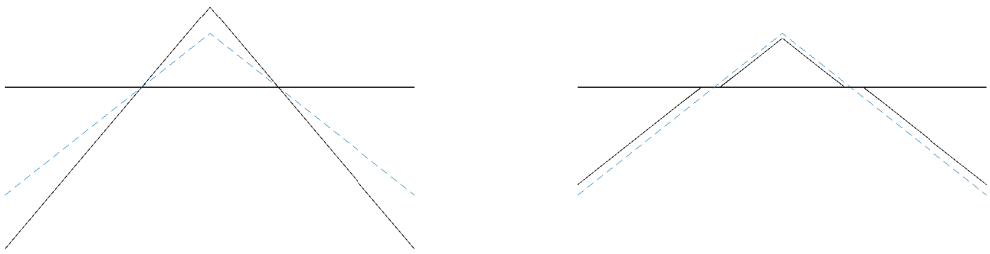
$$\mathcal{T}_h^\Gamma = \{\tau \in \mathcal{T}_h : \tau \cap \Gamma(\phi^0) \neq 0\}, \tag{36}$$

that is, $\mathcal{T}_h^\Gamma$ denotes the subset of elements in $\mathcal{T}_h$ which are intersected by the level set interface, Γ. This means that the Lagrange multiplier space will consist of the same basis functions as the finite element space, and therefore one can solve for one Lagrange multiplier per degree of freedom on any element intersected by the interface.

When choosing the Lagrange multiplier interpolation space, it is necessary that the space is rich enough such that it contains the approximate solution, but not so large as to overconstrain the problem. It is a known phenomena, [33], that boundary locking or spurious oscillations can occur when the approximation spaces V_{DG} and L are chosen to be of equal order. Repeating the previous experiment, using a Lagrange multiplier approach to enforce the boundary condition with the Lagrange multiplier space defined as in (35) gives the results shown in in Figure 6, which confirms that such a choice will in fact lead to boundary locking.

In order to rectify this problem, the order of the Lagrange multiplier space has been reduced to the space of piecewise constant functions with one degree of freedom per element intersected by the interface. This can be stated as

$$L = \operatorname*{span}_{\tau \in \mathcal{T}_h^\Gamma} \{\mathbb{1}^\tau\}, \tag{37}$$



(a) Pre-reinitialisation level set function.

(b) Converged Solution.

Figure 6: Effect of using too large of an interpolation space for the Lagrange multipliers to enforce the Dirichlet boundary condition. The solid line shows the level set function, the dashed line shows the analytical solution, and the horizontal line shows the problem domain.

where $\mathbb{1}^\tau$ is the indicator function defined as follows

$$\mathbb{1}^\tau(\boldsymbol{x}) := \begin{cases} 1 & \text{if } \boldsymbol{x} \in \tau, \\ 0 & \text{if } \boldsymbol{x} \notin \tau. \end{cases} \tag{38}$$

This choice of space means that for each element, $\tau \in \mathcal{T}_h^\Gamma$, the integral of the level set function over the portion of the interface contained within that element, averages to be zero over the element. In other words, this reduction in the order of the constraint space allows some movement to occur at the interface (limited by the size of the element), which is a sufficient relaxation to remove the boundary locking observed above and allows the boundary condition to be satisfied without affecting the signed distance property. It should be noted that even for higher order elements, i.e. $p \geq 2$, choosing the Lagrange multiplier space, as the space of piecewise constants, is required to ensure that there is no boundary locking.

As such, the preferred method of the author therefore for enforcing a homogeneous Dirichlet boundary condition on an implicit surface, is to use a Lagrange multiplier approach, where the Lagrange multiplier space is the space of piecewise constant functions. Using this formulation, to

enforce the Dirichlet boundary condition, (12), the linear system, (20), will be modified as follows

$$\begin{bmatrix} K & A^T \\ A & 0 \end{bmatrix} \begin{Bmatrix} \phi^m \\ \lambda \end{Bmatrix} = \begin{Bmatrix} F(\phi^{m-1}) \\ 0 \end{Bmatrix}, \tag{39}$$

where $A = (a_{ij})$ is a matrix, where the number of rows is equal to the number of elements in $\mathcal{T}_h^\Gamma$ and the number of columns is the total number of degrees of freedom in the problem, with elements given by

$$a_{ij} = \int_{\Gamma(\phi^0)} v_j \mathbb{1}_i^\tau \, \mathrm{d}s, \tag{40}$$

and K and F are defined in equations (21) and (31) respectively.

2.2.3. Integration on immersed implicit surfaces

Regardless of the method chosen to impose a boundary condition on an immersed implicit surface, it will require a method for integrating a function on that surface. There are three general approaches found in the literature: explicit reconstruction of the interface through mesh refinement [34]; implicit reconstruction of the interface using an approximate Dirac delta function such as in the original immersed boundary method, [35]; and methods which generate a new quadrature rule over the volume of an element $\tau \in \mathcal{T}_h^\Gamma$, which is equivalent to integrating an arbitrary function over the implicit surface [36, 27].

As the Eulerian nature of the level set method allows one to take advantage of the use of Cartesian meshes, methods involving r-adaptivity to explicitly reconstruct the interface are not appropriate in the context of this work. Such methods also suffer from extreme computational expense, especially when the desired level of accuracy is high. Methods involving the use of an approximate Dirac delta function, allow one to replace the surface integral over the interface with an equivalent volume integral weighted by the Dirac delta function. Whilst this method is simple to implement, and has found use in other works, even prompting research into high order approximations of the delta function [37], the method depends on the global cancellation of errors over the domain. Thus such an approach has limited accuracy when working with piecewise discontinuous level set functions.

The final group of methods are able to provide arbitrarily high-order elementwise approximations of integrals on implicit interfaces. One such method presented by Müller *et al.* [27], involves the construction of a new quadrature rule based on the solution to the moment-fitting equations [38],

and can be stated as follows

$$\begin{bmatrix} \boldsymbol{g}'_1(\boldsymbol{p}_1)\cdot\boldsymbol{n}_{\boldsymbol{\phi}}(\boldsymbol{p}_1) & \cdots & \boldsymbol{g}'_1(\boldsymbol{p}_N)\cdot\boldsymbol{n}_{\boldsymbol{\phi}}(\boldsymbol{p}_N) \\ \vdots & \ddots & \vdots \\ \boldsymbol{g}'_M(\boldsymbol{p}_1)\cdot\boldsymbol{n}_{\boldsymbol{\phi}}(\boldsymbol{p}_1) & \cdots & \boldsymbol{g}'_M(\boldsymbol{p}_N)\cdot\boldsymbol{n}_{\boldsymbol{\phi}}(\boldsymbol{p}_N) \end{bmatrix} \begin{Bmatrix} w_1 \\ \vdots \\ w_N \end{Bmatrix} = \begin{Bmatrix} -\int_{\partial\tau} H(-\phi)\boldsymbol{g}'_1\cdot\hat{\boldsymbol{n}}\,\mathrm{d}s \\ \vdots \\ -\int_{\partial\tau} H(-\phi)\boldsymbol{g}'_M\cdot\hat{\boldsymbol{n}}\,\mathrm{d}s \end{Bmatrix}. \quad (41)$$

For a given set of divergence free vector valued functions, $\boldsymbol{g}'$, an integral over the unknown interface, $\Gamma(\phi^0)$, can be transformed to an integral over the known surface, $\partial\tau$, using the divergence theorem, which forms the RHS of (41) which can then be approximated using a standard Gauss quadrature rule. Then the weights, $\boldsymbol{w}$, for a new quadrature rule over the element, τ, using the standard 2D Gauss quadrature abscissae, $\boldsymbol{p}$, equivalent to integrating these functions, $\boldsymbol{g}'$, over the interface can be solved for, which can then be used to compute the integral of any function over the interface.

Müller *et al.* [27], chose the functions, g, to be the monomial basis functions, where the derivatives $\boldsymbol{g}'$ are orthonormalised using a Gram-Schmidt procedure. The maximum order of these functions, determines the number of equations, M, to be solved and it is noted that care should be taken to ensure that the number of quadrature points, N, is chosen such that the resulting linear system is underdetermined, i.e. $N > M$. The full details of the method can be found in [27].

The integration method presented in [27] is the preferred method of the author, with two caveats. Firstly, it was found that the accuracy of this integration method depends heavily on the accuracy with which one is able to compute the terms on the RHS of equation (41). The Heaviside function, $\mathrm{H}(-\phi)$, in each of the integrals is present such that the integral is computed only along the part of the edge where, $\phi < 0$. When using standard 1D Gauss quadrature along element edges, the discontinuity present in the Heaviside function is smoothed to such an extent that it becomes difficult to predict whether a given quadrature rule will be sufficient to ensure that the method is sufficiently accurate, without using a (potentially prohibitively) high-order quadrature rule. As such for edges intersected by the interface, a Newton/bisection method is used to find the intersection point(s) and a standard quadrature rule is used to integrate over this newly defined interval. Secondly, as the number of quadrature points is chosen to ensure that the system is underdetermined, the linear system will likely be rank deficient and ill-conditioned. Thus the numerically stable singular value decomposition approach is used to solve for the least squares solution. Any singular values, s, deemed too small, that is $s < \max(s)/10^{12}$, are removed to further improve stability.

These two choices have proven imperative in ensuring the quadrature rule produced is then able to accurately integrate a function on the interface. As a final note, whilst generally robust, this

integration method is problem dependent and small perturbations in the relative position between the mesh and the immersed surface will have an influence on the accuracy for a given problem.

2.3. Narrow band level set methods

When using the level set method for problems involving evolving interfaces it can be noted that the maximum amount of movement of the interface at each time step will be a known value limited by the Courant-Friedrichs-Lewy (CFL) condition, which, if the level set function is always a signed distance function, will be a function of the smallest element size, h_{min}. In other words, the evolution can only occur within a small banded region around the interface, and therefore the information about the level set function outside of this band can effectively be ignored. Narrow band strategies such as that presented in [4], can therefore be a useful tool in reducing the computational expense when using level set methods, as the computation of both the evolution problem and the reinitialisation problem can be restricted to a set of elements, defined by some measure as being close to the interface.

Computational efficiency isn't the only benefit of using a narrow band approach. One of the issues with choosing the level set function to be a signed distance function, is that if the zero isocontour of the level set function, $\Gamma(\phi)$, has at least one loop surrounding a simply-connected subdomain, D, there will always be a singularity which occurs in the level set function, this can be observed in Figure 1 for example. An added benefit of narrow band strategies is that, for a 'sufficiently refined' mesh, almost all of these areas would be far enough away from the level set interface so as to fall outside of the narrow band. When using SIPG, it is known that optimal convergence rates are a function of the smoothness of the problem [39]. Since these singularities will always occur, the use of a narrow band approach is therefore necessary to allow one to demonstrate optimal orders of convergence when using an SIPG discretisation.

'Sufficient refinement' is, as of right now, a poorly defined term. In order to capture a given interface to a prescribed level of accuracy, there is some requirement on the number and order of the elements present along the interface. In the literature, where adaptive meshes are used, the general refinement strategy can be stated as "split any cell whose edge length exceeds its minimum distance to the interface", [40]. Whilst the simplicity of such a strategy is attractive and will result in high levels of h-refinement close to the interface, it is unlikely that such a refinement strategy is optimal. One area of future work therefore could be to develop appropriate error estimators and

refinement strategies for the level set reinitialisation problem. For the purposes of this article it will suffice to demonstrate that the combination of sufficient mesh refinement and a narrow band approach, are required to return optimal convergence; this will be demonstrated in Section 3.

For the purposes of this article, it is noted that when deciding on an appropriate width for the narrow band one needs to consider that in order to satisfy the CFL condition, the furthest that the interface should be able to move to maintain stability is from the element within which it currently resides to one of its neighbours. As such the best case scenario for a narrow band is the union of the set of elements cut by the interface and the set of elements which share a node with any element cut by the interface. If one where to start with, and always maintain, a level set function as a signed distance function, on a uniform mesh, the width of the narrow band could always be linearly related to the minimum absolute value of the level set function in a given element, i.e. to maintain for each timestep a narrow band approximately two elements wide, at each time step one need only consider all of the elements within which the minimum value of the level set function is less than the threshold value equal to twice the width of the smallest element. This constitutes an efficient way to compute which elements belong to the narrow band and which do not. For the problems to be considered in this article we always start with a level set function which is not a signed distance function however, and as such a slightly more conservative value is used, equal to four element widths, to account for the variation in the gradient either side of the interface. Furthermore, it is noted that the boundary conditions stated in Equation (14) naturally extend to the narrow band region. Since the narrow band will contain the set of elements cut by the interface $\mathcal{T}_h^\Gamma$, no change has to be made to the imposition of the Dirichlet condition presented in Equation (39). Finally, whilst the set of element edges constituting the Neumann boundary, $\partial\Omega$, will change, it is the same homogeneous Neumann condition, (15), which is to be applied on this new set of edges.

3. Numerical Examples

3.1. Error measures

Where the analytical solution, ϕ, is known, the error for the example problems in this section is given in the L^2 norm which can be stated as

$$E_{L^2}^2 = \int_\Omega (\phi_h - \phi)^2 \, \mathrm{d}\boldsymbol{x}, \tag{42}$$

the L^∞ norm which can be stated as

$$E_{L^\infty} = \max |\phi_h - \phi|, \tag{43}$$

and the DG norm which can be stated as

$$E^2_{\mathrm{DG}} = \int_\Omega (\boldsymbol{\nabla}(\phi_h - \phi))^2 \, \mathrm{d}\boldsymbol{x} + \mu \int_S [\![\phi_h - \phi]\!]^2 \, \mathrm{d}\boldsymbol{x}. \tag{44}$$

For elliptic problems discretised using SIPG the optimal convergence rates in the L^2 norm are known to be h^{p+1}, and in the DG norm, h^p, [10], assuming the problem is sufficiently smooth. Similarly, it has been shown that optimal convergence when using the L^∞ norm is proportional to $\ln(h^{-1})^{\bar{s}} h^{p+1}$, where $\bar{s} = 1$ for $p = 1$, and $\bar{s} = 0$ otherwise, [41]. It is shown in [39], that for a problem which lacks sufficient smoothness, the convergence rates fall back equal to the linear case for all p.

When the analytical solution is not known, there are two additional error measures which can demonstrate the efficacy of the reinitialisation method. The first is an error measure which measures globally, the degree to which the computed solution satisfies the Eikonal equation, that is

$$E^2_{\mathrm{SD}} = \int_\Omega (|\boldsymbol{\nabla}\phi_h| - 1)^2 \, \mathrm{d}\boldsymbol{x}. \tag{45}$$

This *signed distance error measure* acts similarly to the H^1 seminorm, computing the difference between measures of the gradient of the solution. As such it would be reasonable to expect optimal convergence rates to be equivalent to optimal convergence in the H^1 seminorm, which is known to be h^p, once again assuming sufficient smoothness.

The second of these, is a measure of the movement of the interface in the L^2 norm, which is evaluated by integrating the difference between the computed and desired value of the solution along the original position of the interface, that is

$$E^2_{\mathrm{Int}} = \int_{\Gamma(\phi^0)} \phi_h^2 \, \mathrm{d}\boldsymbol{x}, \tag{46}$$

which will be referred to as the *interface error measure*.

For all of the numerical experiments presented in this section the following statements are true. The objective functional defining the problem to be solved is that defined in equation (29), i.e. t_3. The fixed point iterative method is considered to have converged when $\left|E_{SD}(\phi^m) - E_{SD}(\phi^{m-1})\right| <$

10^{-8}, or the number of iterations required exceeds 1000. The Dirichlet boundary condition is enforced using the Lagrange multiplier approach, with an interpolation space consisting of piecewise constant functions. The method of Müller *et al.* [27] is used to compute the integral along the interface, with the maximum order of the divergence free basis functions, $\boldsymbol{g'}$, equal to 10. This is much higher than that required in practice, for the problems to be presented, however, it allows as much as possible one to remove the error associated with the mesh/problem dependency of the integration method and thus better evaluate the reinitialisation method.

3.2. Circular interface

The first test case presented is that of a circular interface defined initially by a level set function, ϕ^0, which can be described analytically as

$$\phi^0 = x^2 + y^2 - 1, \tag{47}$$

in the domain $\Omega = (-2, 2)^2$. The corresponding signed distance function, and therefore the analytical solution to the problem can thus be stated

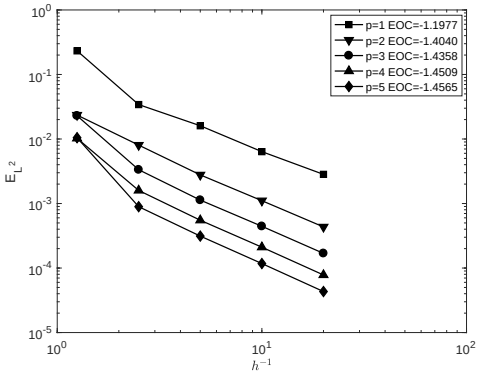
$$\phi = \sqrt{x^2 + y^2} - 1. \tag{48}$$

For this problem, the zero isocontour of the level set function can also be described analytically as follows, for $0 \leq \theta \leq 2\pi$,

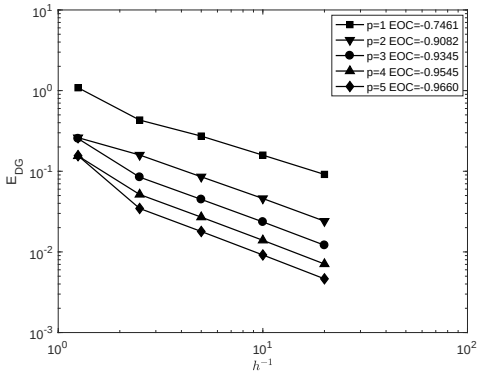
$$\begin{aligned} x &= \cos(\theta), \\ y &= \sin(\theta). \end{aligned} \tag{49}$$

As such the interface error measure will be computed using the trapezium rule, to remove any error associated with the methods for integrating over an implicit surface.

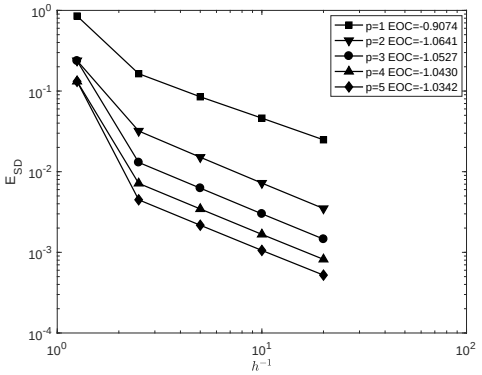
An h-convergence study is performed by computing the reinitialisation of the level set function, initialised as the L^2 projection of (47), on a sequence of Cartesian meshes with square elements of size, $h = 0.8, 0.4, 0.2, 0.1, 0.05$, for meshes of uniform polynomial order, $p = 1, 2, 3, 4, 5$. Error measures will be computed in each of the norms defined in Section 3.1. The analytical solution for this problem, as defined in (48), is singular at the origin, and one should expect for this problem a convergence rate in the L^2 norm of h^2, for all p, a convergence rate in the DG norm and signed distance error norms of h^1, and a convergence rate of $\ln(h^{-1})h^2$. The results of the h-convergence study are shown in Figure 7 and demonstrate that beyond the initial pre-asymptotic datum the



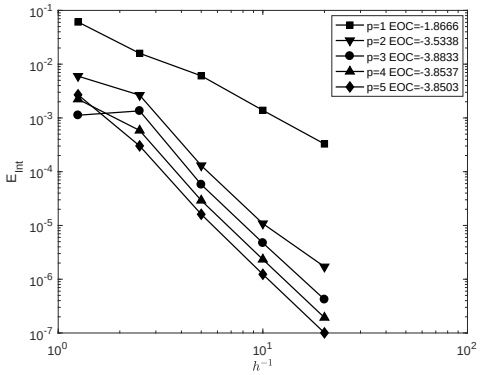
(a) Convergence in the L^2 norm.



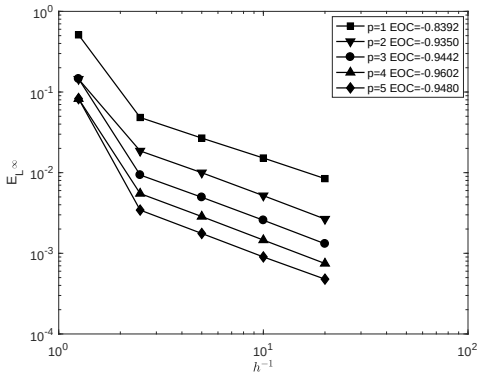
(b) Convergence in the DG norm.



(c) Convergence using the signed distance error measure.



(d) Convergence using the interface error measure.



(e) Convergence in the L^∞ norm.

Figure 7: Error data and convergence rates for the circular interface problem in the domain $\Omega = (-2, 2)^2$.

experimental orders of convergence, using the four aforementioned error measures, are congruent with those expected for a non-smooth problem.

The convergence rate using the interface error measure does show an increase between $p = 1$ and $p = 2$, but remains constant beyond that point. For the purposes of our discussion it is useful to observe that the presence of a singularity in the mesh, constrains the rate at which the L^2 error at the interface decreases when using high-order elements.

3.3. Circular interface with narrow band

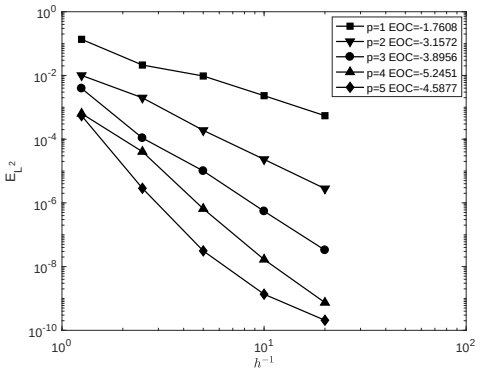
For the previous example problem, the analytical solution is known to be singular, and thus the computed experimental orders of convergence are limited. In order to demonstrate optimal rates of convergence one needs to change the domain such that everywhere within the domain the solution is smooth, which, as discussed in Section 2.3, can be achieved through the use of a narrow band approach. For this somewhat trivial example, the position of the singularity is known to be at the origin and thus a naive implementation of a narrow band approach, is to simply repeat the previous experiment in the domain, $\Omega = (-2, 2)^2 \backslash (-0.4, 0.4)^2$, such that the singularity at the origin is removed.

The same h-convergence study is computed on the new domain leading to the results shown in Figure 8. As expected, removing the origin from the problem domain, allows the solution to be smooth enough everywhere to display optimal convergence rates in all of the relevant norms. This includes a convergence rate using the interface error measure of h^{p+1}, which suggests that one might expect this to be the optimal rate of convergence for this error measure. It can be noted that the quoted orders of convergence for all measures and polynomial orders are computed using the difference between the results for $h = 0.4$ and $h = 0.05$.

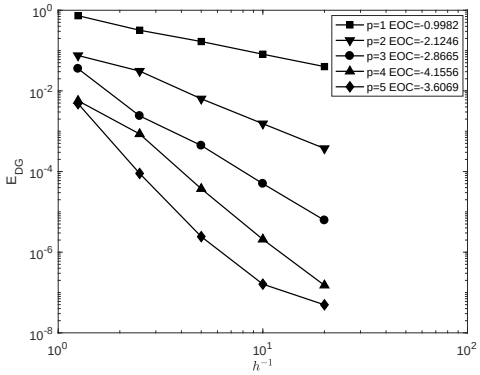
It should also be noted here that for this example the number of iterations taken to reach the convergence criterion is often few, for this simple example; for the mesh with $h = 0.05$ and $p = 5$, just 6 iterations are required.

3.4. Smooth star shaped interface

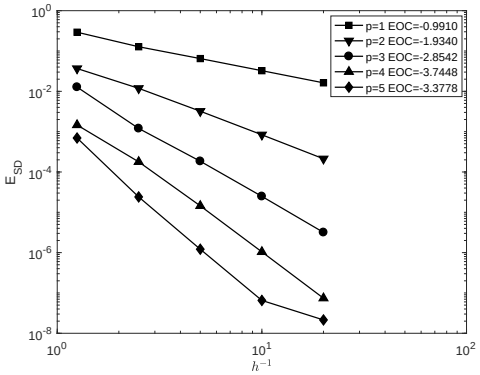
The remaining examples will be of a more arbitrary nature than the simple circle example, thus the rule determining the width of the narrow band will be defined as follows: remove from the mesh any element which has a minimum absolute nodal value greater than four times the size



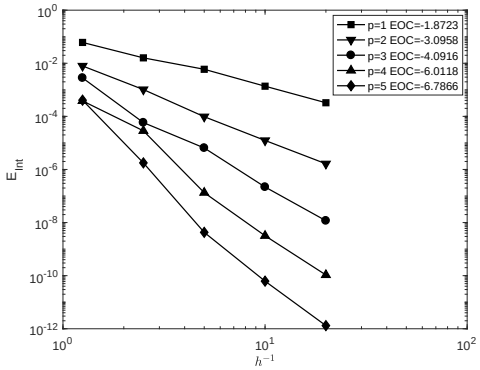
(a) Convergence in the L^2 norm.



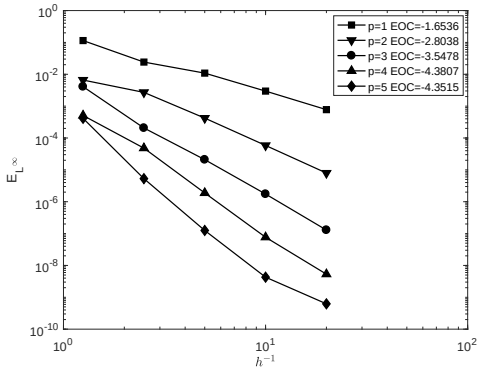
(b) Convergence in the DG norm.



(c) Convergence using the signed distance error measure.

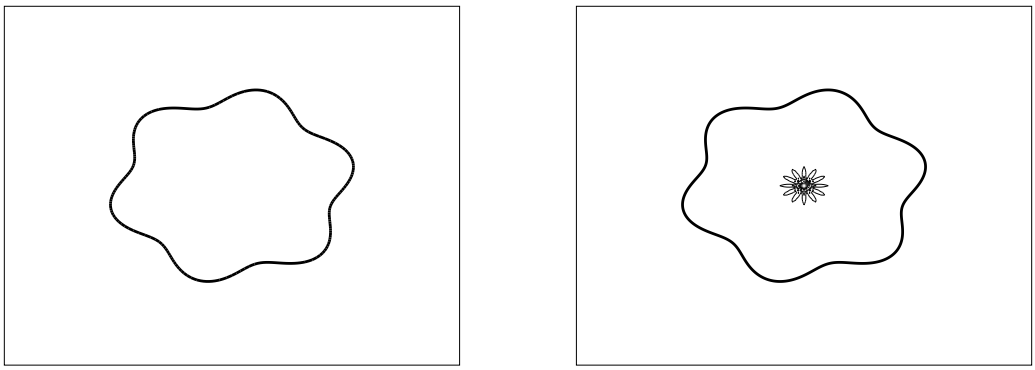


(d) Convergence using the interface error measure.



(e) Convergence in the L^∞ norm.

Figure 8: Error data and convergence rates for the circular interface problem in the domain $\Omega = (-2, 2)^2 \backslash (-0.4, 0.4)^2$.



(a) The level set function's zero isocontour.

(b) Contour plot of the gradient showing where in the full mesh the level set function is singular, relative to the zero isocontour.

Figure 9: Domain configuration for the smooth six pointed star where, $\Omega = (-2, 2)^2$.

of the smallest element, h_{min}. This will also mean that analytical solutions to the problems will be unknown and as such the convergence data presented will be using the signed distance and interface error measures only. The interface error will be computed using the method of Müller [27] instead of the trapezium rule, and as such the error computed will be a measure of the movement of the interface from it's initial projection as opposed to the distance from the analytical solution (although in practice, calculating the error in these two ways gives similar results except for the coarsest meshes tested).

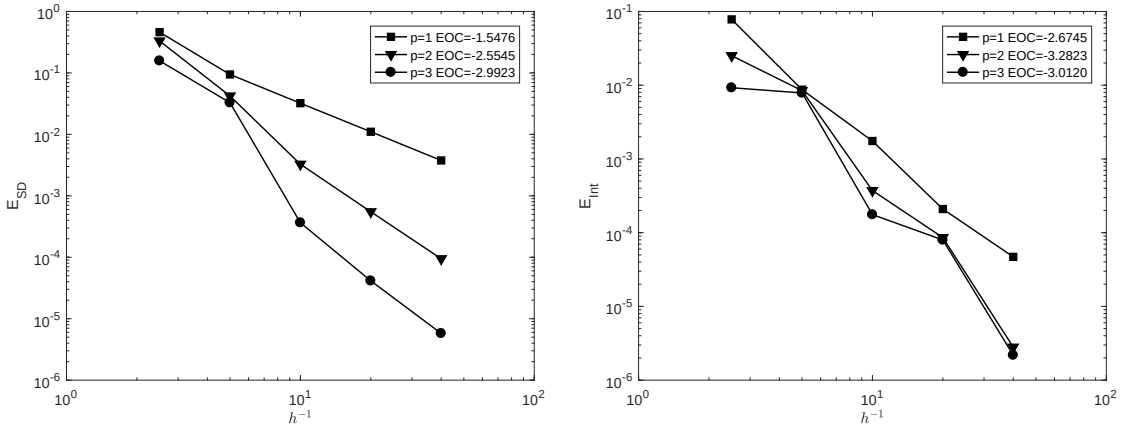
The first of the arbitrary interfaces will be a smooth six pointed star interface, shown in Figure 9(a), which has an initial level set function which can be defined everywhere by

$$\phi = x^2 + y^2 - \left(1 + 0.2 \sin\left(6 \arctan\left(\frac{y}{x}\right)\right)\right), \tag{50}$$

on a domain of maximum size $\Omega = (-2, 2)^2$, however for a given element size, h, the narrow band within which the reinitialisation problem is solved will be a subset of the full domain.

In this case, an h-convergence study will be computed on a sequence of Cartesian meshes with square elements of size $h = 0.4, 0.2, 0.1, 0.05, 0.025$, for meshes of uniform polynomial order, $p = 1, 2, 3$.

Figure 10(a) shows the convergence data for the smooth star problem using the signed distance error measure. The first two data points for all polynomial orders show linear convergence, this



(a) Convergence using the signed distance error measure.

(b) Convergence using the interface error measure.

Figure 10: Error data and convergence rates for the smooth star interface problem, with narrow band, in the domain $\Omega = (-2, 2)^2$.

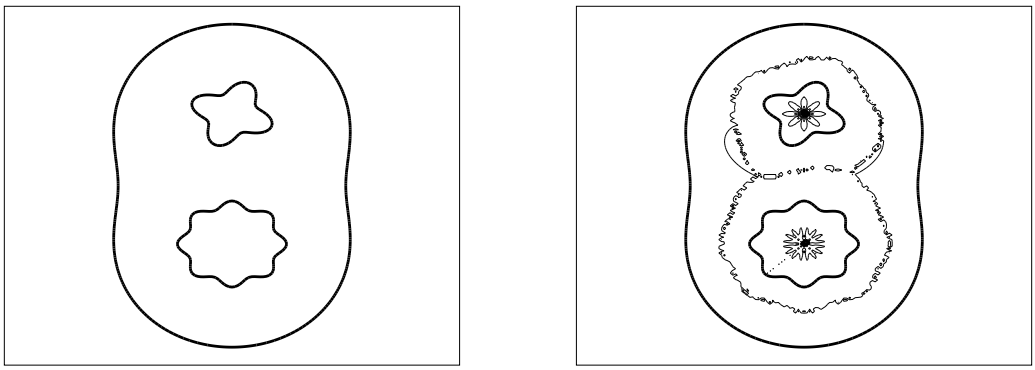
is because the criterion defining the narrow band, is yet to be sufficient to remove the part of the mesh which is singular, see Figure 9(b). As h becomes smaller, the narrow band becomes narrower and the singular part of the solution is no longer part of the mesh, allowing for optimal rates of convergence. The quoted orders of convergence for all measures and polynomial orders are computed using the difference between the results for $h = 0.2$ and $h = 0.025$.

The rate of convergence for the interface error increases slightly between the meshes of linear and quadratic elements, however increasing the polynomial order of the elements beyond that, no longer results in an increase in the accuracy of the solution at the interface, despite the improving gradient solution.

3.5. Multiple arbitrary interfaces

The final example to be presented consists of multiple nested interfaces, which more closely resembles an arbitrary level set function which one might encounter in practice. In particular, the initial level set function at a point is defined as the maximum value of one of three analytical functions, i.e.

$$\phi^0 = \max(q_k), \quad k = 1, 2, 3. \tag{51}$$



(a) The level set function's zero isocontour.

(b) Contour plot of the gradient showing where in the full mesh the level set function is singular, relative to the zero isocontour.

Figure 11: Domain configuration for the multiple interface problem where, $\Omega = (-2, 2)^2$.

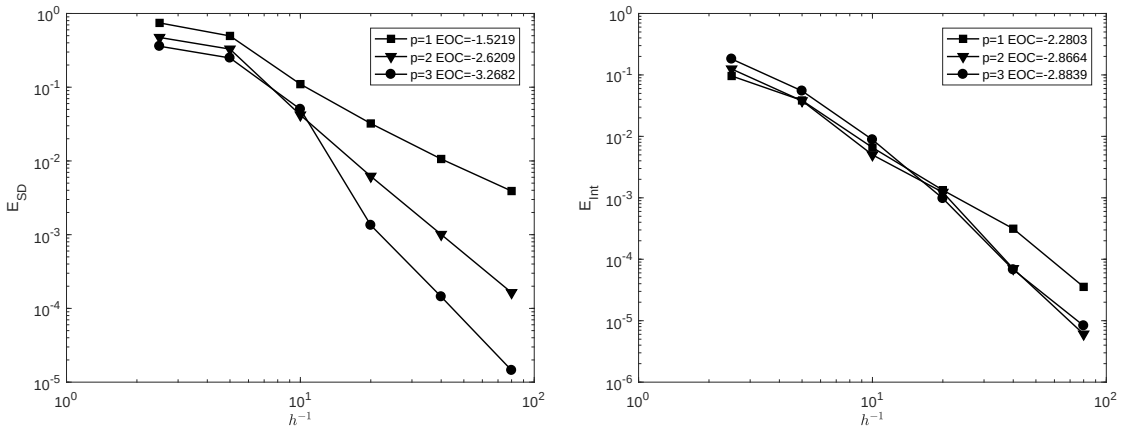
where

$$\begin{aligned} q_1 = &\quad 1.5\left(\sqrt{x^2+y^2} - \left(1 + 0.8\sin\left(\arctan\left(\tfrac{y}{x}\right)\right)^2\right)\right) \\ q_2 = &\quad -2\left(\sqrt{x^2+y^2} - \left(0.3 - 0.075\sin\left(4\arctan\left(\tfrac{y-0.8}{x}\right)\right)\right)\right) \\ q_3 = &\quad -2\left(\sqrt{x^2+y^2} - \left(0.48 - 0.08\sin\left(4\arctan\left(\tfrac{y-0.65}{x}\right)^2\right)\right)\right) \end{aligned} \tag{52}$$

The original configuration of this mesh can be seen in Figure 11(a). These curves have been chosen somewhat arbitrarily, however, considerations were made such that across the domain, the problem has a range of gradients and curvatures to be dealt with.

An h-convergence study is computed on a sequence of Cartesian meshes with square elements of size $h = 0.4, 0.2, 0.1, 0.05, 0.025, 0.0125$, for meshes of uniform polynomial order $p = 1, 2, 3$. As for the previous example, there isn't an analytical solution available for the problem and so the convergence results are given using only the signed distance and interface error measures.

Looking at the signed distance error measure in Figure 12(a) it can once again be seen that until the mesh is sufficiently refined and therefore the narrow band sufficiently narrow, there are singularities present in the solution and the experimental order of convergence for all polynomial orders, p, is equivalent to the linear case. That is the case for all meshes with element size, $h \leq 10^{-1}$. Beyond this point, optimal convergence rates in this error measure can be observed. The quoted orders of convergence for all polynomial orders are computed for this example using the difference between the results for $h = 0.05$ and $h = 0.0125$.



(a) Convergence using the signed distance error measure.

(b) Convergence using the interface error measure.

Figure 12: Error data and convergence rates for the multiple interface problem, with narrow band, in the domain $\Omega = (-2, 2)^2$.

The interface error is displayed in Figure 12(b). It shows almost equivalent errors for a given element size, h, regardless of polynomial order, p, with a small increase in accuracy between $p = 1$ and $p = 2$ which was also the case for the previous example. As has been the case for all of the presented examples, it is difficult to explain the behaviour of this error measure for this problem. As such we restrict our comments to the following; for all examples the demonstrated movement of the level set function at the interface is small (in comparison to other reinitialisation methods), and furthermore can be decreased predictably by controlling the element size with order $\sim h^2$.

For this example, it should also be noted that for the denser higher-order meshes, the number of iterations required to satisfy the convergence criterion grows large, for this problem when $p = 3$ it takes an average of 920 iterations. However, it can also be noted that, for the most dense, high-order mesh tested, it takes just 5 iterations to improve the gradient solution by 3 orders of magnitude, and 34 iterations for an improvement of 4 orders of magnitude. This suggests that in practical applications of the method, it would be up to the user to decide where to strike the balance between expense and accuracy.

4. Conclusions

A practical method for level set reinitialisation using an SIPG discretisation has been presented, based on the elliptic reinitialisation method originally presented by Basting and Kuzmin, [1]. The proposed method is able to demonstrate optimal convergence in the relevant norms and overcomes a number of issues found with other similar reinitialisation techniques. This is achieved through the adoption of a Lagrange multiplier technique, with an appropriate interpolation space, for imposing a Dirichlet boundary condition on an immersed implicit boundary; through a reformulation of the problem by the introduction of a new objective functional driving the problem; and through the adoption of a narrow band approach. This reinitialisation method can be combined with a much simplified level set transport problem, to create a full DG level set methodology. It was demonstrated that a combination of sufficient refinement and a narrow band approach allow one to return optimal convergence rates, as such future work will focus on the development of error estimators and strategies for driving mesh adaptivity, based on the reinitialisation problem.

5. Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (grant EP/M507854/1). All data created during this research is openly available at `doi:10.15128/r1cn69m412x`.

References

[1] C. Basting, D. Kuzmin, A minimization-based finite element formulation for interface-preserving level set reinitialization, Computing 95 (S1) (2013) 13–25.

[2] J. Grooss, J. S. Hesthaven, A level set discontinuous Galerkin method for free surface flows, Computer Methods in Applied Mechanics and Engineering 195 (25) (2006) 3406–3429.

[3] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, Journal of Computational Physics 194 (1) (2004) 363–393.

[4] J. A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, 2nd Edition, no. 3

in Cambridge monographs on applied and computational mathematics, Cambridge University Press, Cambridge, UK, 1999.

[5] E. Maitre, T. Milcent, G.-H. Cottet, A. Raoult, Y. Usson, Applications of level set methods in computational biophysics, Mathematical and Computer Modelling 49 (11-12) (2009) 2161–2169.

[6] S. Osher, R. P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, no. v. 153 in Applied mathematical sciences, Springer, New York, 2003.

[7] 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.

[8] W. Mulder, S. Osher, J. A. Sethian, Computing interface motion in compressible gas dynamics, Journal of Computational Physics 100 (2) (1992) 209–228.

[9] D. L. Chopp, Computing Minimal Surfaces via Level Set Curvature Flow, Journal of Computational Physics 106 (1) (1993) 77–91.

[10] D. N. Arnold, F. Brezzi, B. Cockburn, L. D. Marini, Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems, SIAM Journal on Numerical Analysis 39 (5) (2002) 1749–1779.

[11] J. Cheng, X. Liu, T. Liu, H. Luo, A Parallel, High-Order Direct Discontinuous Galerkin Method for the Navier-Stokes Equations on 3D Hybrid Grids, Communications in Computational Physics 21 (05) (2017) 1231–1257.

[12] B. Cockburn, C.-W. Shu, Runge–Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems, Journal of Scientific Computing 16 (3) (2001) 173–261.

[13] P. Castillo, Performance of discontinuous Galerkin methods for elliptic PDEs, SIAM Journal on Scientific Computing 24 (2) (2002) 524–547.

[14] B. Rivière, M. F. Wheeler, V. Girault, Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. Part I, Computational Geosciences 3 (3-4) (1999) 337–360.

[15] C. E. Baumann, J. T. Oden, A discontinuous hp finite element method for the Euler and Navier-Stokes equations, International Journal for Numerical Methods in Fluids 31 (1) (1999) 79–95.

[16] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM Journal on Numerical Analysis 35 (6) (1998) 2440–2463.

[17] M. Sussman, P. Smereka, S. Osher, A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow, Journal of Computational Physics 114 (1) (1994) 146–159.

[18] M. Sussman, E. Fatemi, An Efficient, Interface-Preserving Level Set Redistancing Algorithm and Its Application to Interfacial Incompressible Fluid Flow, SIAM Journal on Scientific Computing 20 (4) (1999) 1165–1191.

[19] R. Mousavi, Level Set Method for Simulating the Dynamics of the Fluid-Fluid Interfaces: Application of a Discontinuous Galerkin Method, Ph.D. thesis (2014).

[20] A. Karakus, T. Warburton, M. H. Aksel, C. Sert, A GPU accelerated level set reinitialization for an adaptive discontinuous Galerkin method, Computers & Mathematics with Applications 72 (3) (2016) 755–767.

[21] J. Gomes, O. Faugeras, Reconciling Distance Functions and Level Sets, Journal of Visual Communication and Image Representation 11 (2) (2000) 209–223.

[22] M. Weber, A. Blake, R. Cipolla, Sparse finite elements for geodesic contours with level-sets, Computer Vision-ECCV 2004 (2004) 391–404.

[23] C. Li, C. Xu, C. Gui, M. D. Fox, Level set evolution without re-initialization: A new variational formulation, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference On, Vol. 1, IEEE, 2005, pp. 430–436.

[24] C. Li, C. Xu, C. Gui, M. D. Fox, Distance Regularized Level Set Evolution and Its Application to Image Segmentation, IEEE Transactions on Image Processing 19 (12) (2010) 3243–3254.

[25] T. Utz, F. Kummer, M. Oberlack, Interface-preserving level-set reinitialization for DG-FEM, International Journal for Numerical Methods in Fluids 84 (4) (2017) 183–198.

[26] P. Houston, J. Robson, E. Süli, Discontinuous Galerkin finite element approximation of quasi-
685 linear elliptic boundary value problems I: The scalar case, IMA Journal of Numerical Analysis 25 (4) (2005) 726–749.

[27] B. Müller, F. Kummer, M. Oberlack, Highly accurate surface and volume integration on implicit domains by means of moment-fitting, International Journal for Numerical Methods in Engineering 96 (8) (2013) 512–528.

690 [28] I. Babuška, U. Banerjee, J. E. Osborn, Survey of meshless and generalized finite element methods: A unified approach, Acta Numerica 12 (2003) 1–125.

[29] A. J. Lew, G. C. Buscaglia, A discontinuous-Galerkin-based immersed boundary method, International Journal for Numerical Methods in Engineering 76 (4) (2008) 427–454.

[30] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung
695 von Teilräumen, die keinen Randbedingungen unterworfen sind, Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg 36 (1) (1971) 9–15.

[31] I. Babuška, The finite element method with Lagrangian multipliers, Numerische Mathematik 20 (3) (1973) 179–192.

[32] G. Brandstetter, S. Govindjee, A high-order immersed boundary discontinuous-Galerkin
700 method for Poisson's equation with discontinuous coefficients and singular sources, International Journal for Numerical Methods in Engineering 101 (11) (2015) 847–869.

[33] H. J. Barbosa, T. J. Hughes, Boundary Lagrange multipliers in finite element methods: Error analysis in natural norms, Numerische Mathematik 62 (1) (1992) 1–15.

[34] P.-O. Persson, Mesh generation for implicit geometries, PhD Thesis, Massachusetts Institute
705 of Technology (2005).

[35] C. S. Peskin, Flow patterns around heart valves: A numerical method, Journal of Computational Physics 10 (2) (1972) 252–271.

[36] R. Saye, High-Order Quadrature Methods for Implicitly Defined Surfaces and Volumes in Hyperrectangles, SIAM Journal on Scientific Computing 37 (2) (2015) A993–A1019.

[37] B. Engquist, A.-K. Tornberg, R. Tsai, Discretization of Dirac delta functions in level set methods, Journal of Computational Physics 207 (1) (2005) 28–51.

[38] J. Bremer, Z. Gimbutas, V. Rokhlin, A Nonlinear Optimization Procedure for Generalized Gaussian Quadratures, SIAM Journal on Scientific Computing 32 (4) (2010) 1761–1788.

[39] S. Giani, E. J. C. Hall, An a posteriori error estimator for hp-adaptive discontinuous galerkin methods for elliptic eignvalue problems, Mathematical Models and Methods in Applied Sciences 22 (10) (2012) 1250030.

[40] J. Strain, Tree Methods for Moving Interfaces, Journal of Computational Physics 151 (2) (1999) 616–648.

[41] Pointwise Error Estimates of Discontinuous Galerkin Methods with Penalty for Second-Order Elliptic Problems, SIAM Journal on Numerical Analysis 42 (3) (2005) 1146–1166.