

# A parabolic level set reinitialisation method using a discontinuous Galerkin discretisation

Thomas Adams, Nicholas McLeish, Stefano Giani\*, William M. Coombs

Department of Engineering, Durham University, South Road, Durham, DH1 3LE, UK

## ARTICLE INFO

**Article history:**  
Available online 7 March 2019

**Keywords:**  
Level set  
Reinitialisation  
Discontinuous Galerkin

## ABSTRACT

Level set reinitialisation is a part of the level set methodology which allows one to generate, at any point during level set evolution, a level set function which is a signed distance function to its own zero isocontour. Whilst not in general a required condition, maintaining the level set function as a signed distance function is often desirable as it removes a known source of numerical instability. This paper presents a novel level set reinitialisation method based on the solution of a nonlinear parabolic PDE. The PDE is discretised using a symmetric interior penalty discontinuous Galerkin method in space, and an implicit Euler method in time. Also explored are explicit and semi-implicit time discretisations, however, numerical experiments demonstrate that such methods suffer from severe time step restrictions, leading to prohibitively large numbers of iterations required to achieve convergence. The proposed method is shown to be high-order accurate through a number of numerical examples. More specifically, the presented experimental orders of convergence align with the well established optimal convergence rates for the symmetric interior penalty method; that is the error in the  $L^2$  norm decreases proportionally to  $h^{p+1}$  and the error in the DG norm decreases proportionally to  $h^p$ .

© 2019 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The level set method is a popular technique used for representing and tracking evolving interfaces with many applications interesting to engineers and computer scientists. To name just a few, these applications include; fluid dynamics [1], shape optimisation [2], computer vision [3] and biomechanics [4]. A review of 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 [3], and Osher and Fedkiw [5]. The aim of the work 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. This is an extension of earlier work which presented a reinitialisation method based on the DG solution of an elliptic problem [6], by investigating a reformulation of the elliptic problem in that paper as a nonlinear parabolic problem. The main novelty of this article is the presentation of, as far as the authors are aware, the first solution to what is referred to as the parabolic reinitialisation problem [7]. Furthermore, the proposed method is the second of two reinitialisation methods, discretised using DG methods, capable of arbitrarily high-order accuracy, and is demonstrably optimally convergent.

\* Corresponding author.

E-mail addresses: [thomas.d.adams@durham.ac.uk](mailto:thomas.d.adams@durham.ac.uk) (T. Adams), [nick.tmcleish@gmail.com](mailto:nick.tmcleish@gmail.com) (N. McLeish), [stefano.giani@durham.ac.uk](mailto:stefano.giani@durham.ac.uk) (S. Giani), [w.m.coombs@durham.ac.uk](mailto:w.m.coombs@durham.ac.uk) (W.M. Coombs).

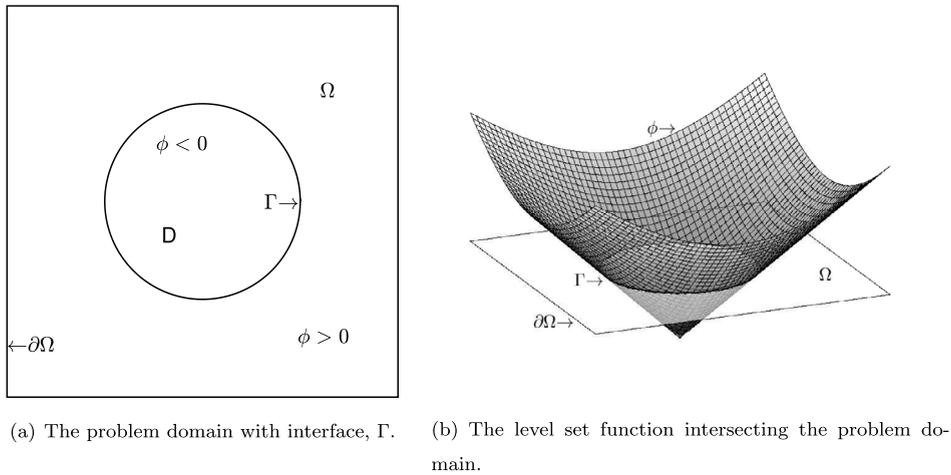


Fig. 1. Graphical representation of a discrete level set function on a square domain.

The remainder of this introduction consists of the following subsections. Section 1.1 provides an introduction to the level set method, Section 1.2 provides a brief review of the pertinent level set reinitialisation literature and finally Section 1.3 provides an introduction to, and brief literature review of, DG methods.

### 1.1. Level set method

The idea behind the level set method, [8], is to use a real scalar valued function,  $\phi$ , called a level set function to divide a problem domain,  $\Omega$ , into a number of subdomains and the interface,  $\Gamma$ , between these subdomains. In general, there will be two subdomains defined, at any given point, by the sign of the value of the level set function, and an interface between the two subdomains defined as the zero isocontour of the level set function. This can be stated as

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

where  $D$  is a subdomain implied by the level set function. This is also shown graphically using an example circular interface modelled using the level set method in Fig. 1.

Once initialised, the level set function can be then be tracked or otherwise evolved in time, through the solution of a scalar transport problem, sometimes called the level set equation. The level set equation can be stated as

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

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

When evolving a level set function through the solution of (2), the level set interface can only be transported along its unit outward normal. Due to this, it is a natural choice to initialise the level set function as a signed distance function to the interface. A level set function which is a signed distance function to its zero isocontour (i.e. the interface) is defined as,  $\phi = \pm \text{dist}(\mathbf{x}, \Gamma)$ , where  $\text{dist}(\mathbf{x}, \Gamma)$  is the minimum distance from any point in the domain,  $\mathbf{x}$ , to the interface,  $\Gamma$ , with the sign of the function defined as positive for  $\phi \in \Omega \setminus D$  and negative for  $\phi \in D$ , using the notation from (1). One important property of a signed distance function, sometimes called the *signed distance property*, is that a signed distance function will satisfy the Eikonal equation, which can be stated as

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

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

For a given velocity field,  $\mathbf{b}$ , the level set function is unlikely to satisfy the signed distance property, (3), for all time,  $t_e$ , when solving the level set equation, (2). Whilst it is not a requirement that the level set function must maintain satisfaction of this property, it is often preferred, as large variations in the gradient of the level set function have been shown to cause numerical instability during the solution of the level set equation, [9]. In an attempt to avoid this behaviour, which he labelled the ‘tentpole’ phenomenon, Chopp introduced the idea of reinitialisation, [10], by which, between iterations of the level set evolution problem, the level set function can be reinitialised as a signed distance function to the current interface. Reinitialisation thus is an ancillary technique forming part of the level set methodology which helps ensure numerical stability, by making it possible that for all time,  $t_e$ , the level set function is, as defined by the user, a ‘good’ approximation of a signed distance function.

### 1.2. Level set reinitialisation

Level set reinitialisation is a technique by which for a given level set function,  $\phi^0$ , one generates a new level set function,  $\phi$ , 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 (3), relative to the following Dirichlet boundary condition

$$\phi = 0 \quad \text{on } \Gamma(\phi^0). \tag{4}$$

There are of course a number of existing methods of reinitialisation, and a more exhaustive review of the reinitialisation literature, especially as it relates to DG, can be found in [6]. For brevity we mention here just the most relevant papers. One such paper is that of Li et al. [11], in which an *all-in-one* level set method is presented, meaning that the level set equation, (2), itself was modified to include a penalty term enforcing the signed distance constraint which can be stated as

$$\frac{\partial \phi}{\partial t_e} = \underbrace{b|\nabla \phi|}_{\text{advection term}} + \underbrace{\alpha \nabla \cdot \left( \nabla \phi - \frac{\nabla \phi}{|\nabla \phi|} \right)}_{\text{signed distance constraint}}, \tag{5}$$

where  $\alpha$  is a penalty parameter.

The signed distance constraint from the formulation in (5) was considered separately by Basting et al. in [7,12]. In particular, they present what we will call the *parabolic reinitialisation problem*, which can be stated as

$$\frac{\partial \phi}{\partial t_r} = \nabla \cdot \left( \nabla \phi - \frac{\nabla \phi}{|\nabla \phi|} \right), \tag{6}$$

where  $t_r$  is pseudotime as related to the reinitialisation problem. Basting and Kuzmin choose not to solve (6) but instead reformulate the parabolic reinitialisation problem as an elliptic problem, so as to avoid pseudotime stepping. Using a penalty method to enforce the boundary condition, (4), the elliptic reinitialisation problem can then be stated as

$$\nabla \cdot \left( \nabla \phi - \frac{\nabla \phi}{|\nabla \phi|} \right) + \gamma \phi = 0, \tag{7}$$

where  $\gamma$  is a penalty parameter. Basting and Kuzmin then solved this quasilinear elliptic formulation by first linearising the problem using a Picard iteration and then discretising spatially using a continuous Galerkin finite element method.

It was the formulation in (7), which led to the work presented in [6], by which an DG spatial discretisation is applied to the elliptic reinitialisation problem. In that article a number of issues are addressed concerning singularities in the solution and the application of boundary conditions, such that it is possible to demonstrate an optimally convergent method. With the knowledge gained in solving the elliptic reinitialisation problem, we now choose to investigate DG semi-discretisations for the full parabolic problem, which will form the work to be presented in this article. As far as we are aware, this article presents the first attempt at a solution of the parabolic reinitialisation problem, (6).

### 1.3. Discontinuous Galerkin methods

Originally developed for solving first-order hyperbolic problems [13], in 1974, 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. The advantages of this decoupling between adjacent elements are numerous, and include; the methods’ formal high-order accuracy, their high level of parallelisability, their ability to easily incorporate *hp*-adaptivity, their ability to deal with complex geometries, as well as their nonlinear stability and ability to deal with discontinuous solutions, see [14–16]. For these reasons, since the 1970s, the literature surrounding DG methods has become extensive and as such for a more exhaustive review, we refer the reader to [17] and the references therein, choosing instead here to focus only on the literature pertinent to the following discussion. Furthermore, we note the abundant advantages mentioned above as reasons for the interest in the development of a DG level set methodology. The particular interest of the authors is to exploit some of these advantages especially the high-order accuracy, adaptivity, and parallelisability, to develop an interface tracking method to be used to solve topology optimisation problems by which the underlying physics (which drives the optimisation problem) will also be discretised using DG methods. Despite these aims however it should be noted that the method presented in this article, is completely agnostic and is applicable to any level set method requiring reinitialisation.

The reinitialisation method to be presented and discussed in this article will take the form of a parabolic equation with non-monotone nonlinear diffusion, that is diffusion of the form

$$\nabla \cdot (\rho(u)\nabla u), \tag{8}$$

where the diffusion function,  $\rho(u)$ , does not satisfy the monotonicity condition, that is, for  $\rho \in C(\Omega \times [0, \infty))$

$$m_\rho(t - s) \leq \rho(t)t - \rho(s)s \leq M_\rho(t - s) \quad \text{for } t \geq s \geq 0, \tag{9}$$

for some positive constants  $m_\rho$  and  $M_\rho$ , see [18] for details. Arnold, [19], in 1982, first analysed, what we would now call a symmetric interior penalty DG (SIPG) semi-discretisation of nonlinear parabolic boundary value problems and found optimal order error estimates in the  $L^2$  and energy norms, for a sufficiently large penalty parameter. One issue with interior penalty (IP) DG methods of the time, such as that of Arnold [19], Baker [20] and Wheeler [21], was that convergence, stability and matrix conditioning were all dependent on the correct choice of the penalty parameter and furthermore, none of these methods were locally conservative. These issues were addressed by using a nonsymmetric formulation, based on the method of Baumann–Oden (BO), [22], and through the determination of an appropriate choice for the general penalty parameter, to form the nonsymmetric interior penalty DG (NIPG) method, by Rivière, Wheeler and Girault, [23]. Rivière and Wheeler, [24], then applied NIPG, along with an implicit time scheme, to nonlinear parabolic problems and derived error estimates in the  $L^\infty(L^2)$  and  $L^2(H^1)$  norms.

When dealing with quasilinear diffusion, for the general case, i.e. diffusion of the form

$$\nabla \cdot \mathcal{D}(u, \nabla u), \quad (10)$$

the (now) standard NIPG and SIPG schemes, see [25], cannot be applied directly, as one of the stabilisation terms which arises can become impossible to linearise in both the trial and the test functions, see [26] for details. In such a case, an incomplete interior penalty DG (IIPG) method can be useful as was analysed by Dolejší [26]. For monotone quasilinear diffusion, i.e. where the diffusion coefficient satisfies the condition (9), a new DG scheme was developed by Houston et al. [18], which presented a new stabilisation term which is linear in both the test and trial functions. For non-monotone quasilinear diffusion of the form, (8), SIPG and NIPG methods were analysed by Gudi and Pani in [27], who derive *a priori* error estimates in the broken  $H^1$  norm for both SIPG and NIPG, which are shown to be equivalent to NIPG as applied to the linear problem. Furthermore, optimal estimates are derived in the  $L^2$  norm for both SIPG and superpenalised NIPG on uniform regular grids. This was extended to nonlinear parabolic equations with diffusion of the form (8) by Song [28], who investigated the use of SIPG, NIPG and IIPG spatial discretisations with an explicit Euler temporal discretisation for the full discretisation of nonlinear parabolic equations, and derived error estimates in the  $L^\infty(L^2)$  and  $L^2(H^1)$  norms. Song then went on to perform the same analysis for nonlinear parabolic problems discretised with IPDG methods in space and implicit schemes in time [29], deriving optimal *a priori* error estimates in the  $L^\infty(L^2)$  and  $L^2(H^1)$  norms.

In this work it is the authors' preference to use an SIPG spatial discretisation, [25]. There are a number of reasons why SIPG can be considered preferable to other DG discretisations, which are outlined in the Conclusion of [30]. We highlight the well established optimal rates of convergence for SIPG, which are not found for the nonsymmetric DG methods (which includes NIPG, IIPG and BO), 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), [31]).

The rest of the paper is organised as follows. In Section 2, the proposed reinitialisation problem is derived and a number of discretisations proposed. Section 3 presents a comparison of these proposed formulations through the solution of a number of example problems. Finally, the article is then concluded in Section 4.

## 2. Discontinuous Galerkin parabolic level set reinitialisation

Section 2 consists of the following subsections. Section 2.1 presents the mathematical preliminaries required for the discussion of the adopted DG methods. Section 2.2 presents the derivation of the strong formulation of the parabolic reinitialisation problem. Section 2.3 presents a discussion on how to appropriately enforce the Dirichlet boundary condition on an immersed implicit interface. Section 2.4 presents the proposed spatial semi-discretisation using IPDG methods. Section 2.5 and its various subsections present the proposed temporal discretisation schemes, and thus fully discrete formulations of the problem. Finally, Section 2.6 presents a discussion on the adoption of a narrow band approach.

### 2.1. Interior penalty discontinuous Galerkin method preliminaries

Let  $\mathcal{T}_h$  denote any partition of a domain,  $\Omega$ , into nonoverlapping quadrilateral elements,  $\tau$ , with element size,  $h$ , such that the computational domain can be defined,  $\Omega = \cup_{\tau \in \mathcal{T}_h} \tau$ , with boundary vertices on  $\partial\Omega$ . It can be noted that in general,  $\Omega \subset \mathbb{R}^d$ , and although in this article all numerical examples are restricted to  $d = 2$ , there would be no added difficulty in extending the work presented here to higher dimensions. The skeleton of the mesh,  $S$ , is defined as the set of all interior edges, that is  $S = \cup_{\tau \in \mathcal{T}_h} \partial\tau \setminus \partial\Omega$ . The unit outward normal on the boundary,  $\partial\tau$ , of a given element,  $\tau$ , is denoted as  $\hat{\mathbf{n}}$ . For any partition  $\mathcal{T}_h$  of  $\Omega$ , with elements of maximum polynomial degree,  $p$ , the DG finite element space is defined as

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

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. Parabolic reinitialisation method: strong form

As first presented by Basting and Kuzmin in [7], the parabolic reinitialisation method aims to solve the level set reinitialisation problem by minimising the least squares residual of the Eikonal equation, (3), that is

$$\min_{\phi \in H^1(\Omega)} \left( \int_{\Omega} \frac{1}{2} (|\nabla\phi| - 1)^2 \, d\mathbf{x} \right). \tag{12}$$

As discussed in [6], the Gateaux derivative of such a functional (12) becomes singular as  $|\nabla\phi| \rightarrow 0$ , and as such it is preferable to modify the objective functional to avoid this. As such, the minimisation problem to be solved here can be stated as

$$\min_{\phi \in H^1(\Omega)} \left( \int_{\Omega} R(|\nabla\phi|) \, d\mathbf{x} \right), \tag{13}$$

where

$$R(|\nabla\phi|) = \begin{cases} \frac{1}{2} (|\nabla\phi| - 1)^2 & \text{if } |\nabla\phi| > 1, \\ \frac{(|\nabla\phi|)^3}{3} - \frac{(|\nabla\phi|)^2}{2} + \frac{1}{6} & \text{if } |\nabla\phi| \leq 1. \end{cases} \tag{14}$$

With the inclusion of the Dirichlet boundary condition (4), it can be seen that the signed distance function to the interface minimises (12) and (13), as in such a case the value of the functional would be equal to zero, which is clearly a global minimum. As such at least one minimiser to the problem exists. A well known method for minimising an objective functional is to find the steady-state solution to the gradient flow equation, see [32], that is for functional,  $R(|\nabla\phi|)$ ,

$$\frac{\partial\phi}{\partial t} + \frac{\partial R}{\partial\phi} = 0. \tag{15}$$

The Gateaux derivative of the functional  $R(\phi)$  can be stated

$$\delta R(\phi, v) = \int_{\Omega} d(|\nabla\phi|) \nabla\phi \cdot \nabla v \, d\mathbf{x}, \tag{16}$$

with diffusion function,  $d(|\nabla\phi|)$ , which can be stated as

$$d(|\nabla\phi|) = \begin{cases} 1 - \frac{1}{|\nabla\phi|} & \text{if } |\nabla\phi| > 1, \\ |\nabla\phi| - 1 & \text{if } |\nabla\phi| \leq 1. \end{cases} \tag{17}$$

Combining the above with appropriate boundary conditions leads to a strong formulation of the parabolic reinitialisation problem which can be expressed as

$$\begin{aligned} \frac{\partial\phi}{\partial t} + \nabla \cdot (d(|\nabla\phi|) \nabla\phi) &= 0 & \text{in } \Omega, \\ \phi &= 0 & \text{on } \Gamma(\phi^0) \text{ and} \\ d(|\nabla\phi|) (\nabla\phi \cdot \hat{\mathbf{n}}) &= 0 & \text{on } \partial\Omega. \end{aligned} \tag{18}$$

The first equation forming (18) is a parabolic equation, with quasilinear diffusion. As the diffusion will be positive where  $|\nabla\phi| > 1$  and negative where  $|\nabla\phi| < 1$ , the unique solution is a signed distance function satisfying the equilibrium point  $|\nabla\phi| = 1$ , [7]. The signed distance function itself is uniquely defined by the homogeneous Dirichlet boundary condition along the pre-reinitialisation level set interface,  $\Gamma(\phi^0)$ . There is also a homogeneous Neumann boundary condition on the natural boundary by which the gradient of the solution at the domain boundary must also satisfy the Eikonal equation.

### 2.3. Dirichlet boundary conditions on immersed implicit interfaces

The enforcement of a Dirichlet boundary condition, such as that stated in (18), on an immersed implicit interface is not a trivial issue. The reader is referred to [6] for a detailed investigation into a number of techniques for enforcing immersed boundary conditions, as well as associated methods of integration on implicit interfaces. Here the focus instead will be on outlining the conclusions drawn therein and thus the preferred method.

The evidence presented in [6] suggests that, given the aim of developing a high-order accurate and sufficiently general method (i.e. discounting methods involving remeshing), the most appropriate method for enforcing the Dirichlet boundary condition in (18) is to use the method of Lagrange multipliers. The main concern when using such a method, is that the Lagrange multipliers,  $\lambda$ , must be chosen from an appropriate interpolation space,  $L$ . The space must be rich enough to

contain the approximate solution, and thus ensure sufficient accuracy, but not so rich as to overconstrain the solution and cause boundary locking or spurious oscillations. Experimental evidence shown in [6] suggests that this can be achieved by choosing the Lagrange multiplier space as follows

$$L = \text{span}\{\mathbb{1}^\tau\}, \tag{19}$$

$\tau \in \mathcal{T}_h^\Gamma$

where

$$\mathcal{T}_h^\Gamma = \{\tau \in \mathcal{T}_h : \tau \cap \Gamma(\phi^0) \neq \emptyset\}, \tag{20}$$

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

$$\mathbb{1}^\tau(\mathbf{x}) := \begin{cases} 1 & \text{if } \mathbf{x} \in \tau, \\ 0 & \text{if } \mathbf{x} \notin \tau. \end{cases} \tag{21}$$

Choosing the space in such a way means that the homogeneous Dirichlet boundary condition on the interface will be enforced by adding one constraint per element intersected by the interface, by which the average movement of the interface over that element will be zero.

For a discussion on methods for integrating on implicit interfaces, such as integrals of the type  $\int_{\Gamma(\phi)} \cdot \, ds$ , readers are once again referred to [6]. Here we use the method of Müller et al. [33] to compute integrals of this kind. This method, referred to as *Müller’s method* from here on, computes integrals over implicit interfaces by constructing a new quadrature rule over the area of the element (or the volume of the element in 3D) which approximates the integral along the implicit interface. It does this by solving a moment fitting problem over each element cut by the interface which could be stated as follows

$$\sum_{i=1}^N \mathbf{f}_j(\mathbf{x}_i) \cdot \mathbf{n}_\phi w_i = \int_\Gamma \mathbf{f}_j \cdot \mathbf{n}_\phi \, ds. \tag{22}$$

Choosing the location of the integration points,  $\mathbf{x}_i$ , to be equal to the standard Gauss point locations, allows the nonlinear problem to be reduced to a problem which is linear in the quadrature weights,  $w_i$ . Then by employing the use of divergence-free basis functions,  $\mathbf{f}_j$ , the application of the divergence theorem to the RHS of (22), transforms the integral over the unknown implicit interface to an integral over a known portion of the element edges (or faces in 3D). This then allows one to generate efficiently a quadrature rule for accurate integration over an implicit interface. It is noted here that for all numerical experiments presented in this article, the test functions against which the new quadrature rule is computed, using Müller’s method, are chosen to be tenth order, and thus the quadrature rule itself is chosen such that there are  $11^2$  integration points per element. The reason for choosing such an order is to remove as much as possible any error associated with Müller’s method from the numerical experiments as the aim here is to test the efficacy only of the reinitialisation methods. For practical situations it is left to the user’s discretion to determine an appropriate balance here between accuracy and efficiency.

#### 2.4. Parabolic reinitialisation method: spatial semi-discretisation

Whilst retaining sufficient generality, applying an IPDG discretisation in space leads to a variational formulation of the problem (18) to be stated as follows; find  $\phi_h(t)$  and  $\lambda(t)$  from  $t = (0, \infty)$  in  $V_{DG}$  and  $L$  respectively, such that

$$\left( \frac{\partial \phi_h}{\partial t}, v \right)_\Omega + D_\theta(\phi_h; \phi_h, v) + \langle \lambda, v \rangle_{\Gamma(\phi^0)} = 0, \quad \forall v \in V_{DG}, \tag{23}$$

and

$$\langle \phi_h, \zeta \rangle_{\Gamma(\phi^0)} = 0, \quad \forall \zeta \in L, \tag{24}$$

where the notation  $(a, b)_\Omega$  denotes  $\int_\Omega ab \, d\mathbf{x}$ , and  $\langle a, b \rangle_{\partial\Omega}$  denotes  $\int_{\partial\Omega} ab \, ds$ . We define the form  $D_\theta(\rho; u, w)$  which is bilinear in  $u$  and  $w$  as

$$D_\theta(\rho; u, w) = (d(\rho)\nabla u, \nabla w)_\Omega - \langle \{d(\rho) \nabla u\}, \llbracket w \rrbracket \rangle_S - \theta \langle \{d(\rho) \nabla w\}, \llbracket u \rrbracket \rangle_S + \mu \langle \llbracket u \rrbracket, \llbracket w \rrbracket \rangle_S, \tag{25}$$

where the parameter  $\theta$  may take the value  $-1, 0$  or  $1$ . Where  $\theta = -1$  the resultant discretisation is NIPG, where  $\theta = 0$  the resultant discretisation is IIPG, and where  $\theta = 1$  the resultant discretisation is SIPG. The discontinuity penalisation parameter,  $\mu$ , is 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 for quasilinear diffusion problems, please refer to [18]. The jump and average operators denoted by  $\llbracket \cdot \rrbracket$  and  $\{ \cdot \}$  respectively, are as defined in [25] and are reproduced here as follows: for an arbitrary scalar valued function,  $\psi$ , and vector valued function,  $\Psi$ , on adjacent elements,  $\tau^+$  and  $\tau^-$ , which share an edge

$$\llbracket \psi \rrbracket = \begin{cases} (\psi^+ - \psi^-)\hat{\mathbf{n}}^+ & \text{on } \partial\tau \setminus \partial\Omega, \\ \psi^+\hat{\mathbf{n}}^+ & \text{on } \partial\tau \cap \partial\Omega, \end{cases} \quad \{ \Psi \} = \begin{cases} (\Psi^+ + \Psi^-)/2 & \text{on } \partial\tau \setminus \partial\Omega, \\ \Psi^+ & \text{on } \partial\tau \cap \partial\Omega. \end{cases} \tag{26}$$

Choosing  $\{v_j\}_{j=1}^{N_h}$  to form a basis of  $V_{DG}$ , and  $\{\zeta_j\}_{j=1}^{N_{LM}}$  to form a basis of  $L$  one can write

$$\phi_h = \sum_{j=1}^{N_h} \phi_j v_j, \quad \lambda = \sum_{j=1}^{N_{LM}} \lambda_j \zeta_j, \tag{27}$$

where  $N_h$  denotes the number of solution variables per element, and  $N_{LM}$  denotes the number of Lagrange multipliers per element. This leads to a linear system of nonlinear ordinary differential equations (ODE) which can be stated as

$$B\phi'(t) + K_\theta\phi(t) + A^\top\lambda(t) = 0, \tag{28}$$

$$A\phi(t) = 0, \tag{29}$$

where  $B = (b_{ij})$  is the mass matrix, and  $K_\theta = (k_{\theta,ij})$  is the stiffness matrix, and  $A = (a_{ij})$  is a matrix with entries defined as

$$b_{ij} = (v_j, v_i)_\Omega, \quad k_{\theta,ij} = D_\theta(\phi_h, v_j, v_i) \quad \text{and} \quad a_{ij} = (v_j, \zeta_i)_{\Gamma(\phi^0)}. \tag{30}$$

2.5. Parabolic reinitialisation method: full discretisations

Explicit [28], semi-implicit [34] and fully implicit [24,29] time discretisations have all found use when solving parabolic problems with nonlinear diffusive terms discretised spatially using IPDG methods. Explicit methods, whilst known for their simplicity, often demonstrate severe time step restrictions when combined with IPDG spatial discretisations and have thus found limited use. However, as is noted in [28], explicit discretisations could become competitive due to their ability to be parallelised. Implicit methods, are usually preferred for their superior stability, however, the requirement to solve a nonlinear system at each time step could similarly be considered prohibitively expensive. Semi-implicit methods, which use a suitable combination of explicit and implicit linearisations, can demonstrate improved stability in comparison with explicit methods, whilst only requiring a linear system to be solved at each time step, however, the efficacy of such an approach is of course problem dependent, and difficult to analyse *a priori*. Since it is not immediately obvious which is most suitable for the given problem all three types of temporal discretisation will be investigated. In particular, in this paper three different full discretisations will be explored: an SIPG in space, explicit forward Euler (EE) in time discretisation (SIPG-EE); an SIPG in space, semi-implicit (SI) method in time (SIPG-SI); and an SIPG in space, implicit backwards Euler (IE) in time discretisation (SIPG-IE).

2.5.1. Fully explicit

For both forward and backward Euler methods, we will use the same notation for the discretisation of the time derivative, that is

$$\frac{\partial\phi_h}{\partial t} = \frac{\phi_h^n - \phi_h^{n-1}}{\Delta t}, \tag{31}$$

where  $\Delta t$  denotes the time step, and  $\phi^n = \phi(t^n)$  denotes the value of  $u$  at the  $n$ th time step, i.e.  $t^n = n\Delta t$ . Treating the diffusive terms explicitly, the SIPG-EE full discretisation can thus be stated as follows; find  $\phi_h^n \in V_{DG}$ , and  $\lambda^n \in L$ , as  $t^n \rightarrow \infty$  such that

$$\left(\frac{\partial\phi_h}{\partial t}, v\right)_\Omega + (\lambda^n, v)_{\Gamma(\phi^0)} = -D_1(\phi_h^{n-1}; \phi_h^{n-1}, v), \quad \forall v \in V_{DG} \tag{32}$$

and

$$(\phi_h^n, \zeta)_{\Gamma(\phi^0)} = 0, \quad \forall \zeta \in L. \tag{33}$$

Using the matrix notation introduced in Section 2.4, the linear system to be solved at each time step can thus be stated as

$$\begin{bmatrix} B & \Delta t A^\top \\ \Delta t A & 0 \end{bmatrix} \begin{Bmatrix} \phi^n \\ \lambda^n \end{Bmatrix} = \begin{Bmatrix} (B - \Delta t K_1)\phi^{n-1} \\ 0 \end{Bmatrix}. \tag{34}$$

2.5.2. Semi-implicit

The idea behind semi-implicit methods is to deal with the nonlinear parts of the problem explicitly, whilst dealing with the linear parts implicitly. As such various semi-implicit formulations of (23)–(24) are possible. Here we choose to apply a Picard linearisation to the nonlinear diffusion coefficient,  $d(|\nabla\phi|)$ , such that it is a function of the solution at the previous time step,  $t^{n-1}$ . A backwards Euler method can then be applied to the resulting system of linear ODE's. The SIPG-SI full discretisation, can thus be stated: find  $\phi_h^n \in V_{DG}$ , and  $\lambda^n \in L$ , as  $t^n \rightarrow \infty$  such that

$$\left(\frac{\partial\phi_h}{\partial t}, v\right)_\Omega + D_1(\phi_h^{n-1}; \phi_h^n, v) + (\lambda^n, v)_{\Gamma(\phi^0)} = 0, \quad \forall v \in V_{DG} \tag{35}$$

and

$$\langle \phi_h^n, \zeta \rangle_{\Gamma(\phi^0)} = 0, \quad \forall \zeta \in L. \tag{36}$$

Once again using the matrix notation introduced in Section 2.4, the linear system to be solved at each time step can be stated as

$$\begin{bmatrix} B + K_1 \Delta t & \Delta t A^\top \\ \Delta t A & 0 \end{bmatrix} \begin{Bmatrix} \phi^n \\ \lambda^n \end{Bmatrix} = \begin{Bmatrix} B\phi^{n-1} \\ 0 \end{Bmatrix}. \tag{37}$$

2.5.3. Fully implicit

Using the backwards Euler method to discretise the system (23)–(24) allows one to state the corresponding variational formulation, SIPG-IE, as: find  $\phi_h^n \in V_{DG}$ , and  $\lambda^n \in L$ , as  $t^n \rightarrow \infty$  such that

$$\left( \frac{\partial \phi_h}{\partial t}, v \right)_\Omega + D_1(\phi_h^n; \phi_h^n, v) + \langle \lambda^n, v \rangle_{\Gamma(\phi^0)} = 0, \quad \forall v \in V_{DG} \tag{38}$$

and

$$\langle \phi_h^n, \zeta \rangle_{\Gamma(\phi^0)} = 0, \quad \forall \zeta \in L. \tag{39}$$

The resulting system is still nonlinear and thus needs to be solved at each time step using an appropriate iterative solver. For simplicity we choose here to use a quasi-Newton scheme which can be stated as follows

$$\begin{Bmatrix} \phi^{n,k+1} \\ \lambda^{n,k+1} \end{Bmatrix} = \begin{Bmatrix} \phi^{n,k} \\ \lambda^{n,k} \end{Bmatrix} - \begin{bmatrix} \left[ \frac{\partial F(\phi^{n,k}, \lambda^{n,k})}{\partial \phi} \right] \\ \left[ \frac{\partial G(\phi^{n,k}, \lambda^{n,k})}{\partial \phi} \right] \end{bmatrix} \begin{bmatrix} \left[ \frac{\partial F(\phi^{n,k}, \lambda^{n,k})}{\partial \lambda} \right] \\ 0 \end{bmatrix}^{-1} \begin{Bmatrix} F(\phi^{n,k}, \lambda^{n,k}) \\ G(\phi^{n,k}, \lambda^{n,k}) \end{Bmatrix}. \tag{40}$$

where  $F = (F_i)$  and  $G = (G_i)$  are the residual vectors associated with (38) and (39) respectively, with elements given by

$$F_i(\phi^{n,k}, \lambda^{n,k}) = \frac{1}{\Delta t} (\phi^{n,k} - \phi^{n,0}, v_i)_\Omega + D_1(\phi^{n,k}; \phi^{n,k}, v_i) + \langle \lambda^{n,k}, v_i \rangle_{\Gamma(\phi^0,0)}. \tag{41}$$

and

$$G_i(\phi^{n,k}, \lambda^{n,k}) = \langle \phi^{n,k}, \zeta_i \rangle_{\Gamma(\phi^0,0)}. \tag{42}$$

The notation  $\phi^{n,k}$  denotes the  $k$ th Newton iteration, at the  $n$ th time step, and thus  $\phi^{n,0} = \phi^{n-1,\infty}$ . The constituents of the Jacobian matrix,  $J$ , that is  $\frac{\partial F}{\partial \phi}$ ,  $\frac{\partial G}{\partial \phi}$  and  $\frac{\partial F}{\partial \lambda}$ , are computed using a first order finite difference method, i.e.

$$\left[ \frac{\partial F_i(\phi^{n,k}, \lambda^{n,k})}{\partial \phi_j} \right]_{ij} = \frac{F_i(\phi_j^{n,k} + \delta\phi, \lambda^{n,k}) - F_i(\phi_j^{n,k}, \lambda^{n,k})}{\delta\phi}, \tag{43}$$

and

$$\left[ \frac{\partial F_i(\phi^{n,k}, \lambda^{n,k})}{\partial \lambda_j} \right]_{ij} = \left[ \frac{\partial G_i(\phi^{n,k}, \lambda^{n,k})}{\partial \lambda_j} \right]_{ij}^\top = \frac{F_i(\phi_j^{n,k}, \lambda^{n,k} + \delta\lambda) - F_i(\phi_j^{n,k}, \lambda^{n,k})}{\delta\lambda}, \tag{44}$$

where  $\delta\phi$  and  $\delta\lambda$  are two small constants, chosen for all examples in this work as  $\delta\phi = \delta\lambda = 10^{-12}$ . Furthermore, again for all examples presented in this work, the Newton loop is considered to have converged when the Euclidean norm of the vector  $\{J^{-1}[F \ G]^\top\}$  is less than  $10^{-10}$ .

2.6. Narrow band

It is a known phenomenon that rates of convergence using the SIPG, are dependent on the Sobolev regularity of the solution, [35]. For a level set interface,  $\Gamma(\phi)$ , which has at least one loop surrounding a simply-connected subdomain,  $D$ , the signed distance function to that interface will always contain a singularity; this can be seen, for example, with the apex of the conic level set function in Fig. 1. This means that convergence rates will always be limited to the linear case, unless one can remove the singular part of the solution from the domain. As discussed in [6], one way that this can be achieved is through the combination of the use of a narrow band [3] approach and a ‘sufficiently refined’ mesh near to the interface. It is noted here that for the numerical examples presented in Section 3, a narrow band approach will be adopted, the specifics of which will be noted for each given example problem. It is further noted here that the boundary conditions stated in (18) naturally extend to the narrow band. That is, since the narrow band will always contain all elements intersected by the interface, the Dirichlet condition remains unchanged from that presented in Sections 2.3 and 2.4. Finally, whilst the set of edges comprising the Neumann boundary will change, the boundary condition imposed on these edges remains homogeneous.

### 3. Numerical examples

The numerical examples section comprises the following subsections. Section 3.1 defines the various error measures used to evaluate the proposed discretisations. Section 3.2 presents the first numerical experiment, an investigation into the appropriate choice of time step for the three time discretisations. Then based on the conclusions drawn from the first experiment, Sections 3.3, 3.4, and 3.5 present  $h$ -convergence studies for three different example problems using just the SIPG-IE discretisation.

#### 3.1. Error measures

For the numerical examples to follow; in the case that an analytical solution is known, the error between the computed solution which satisfies the convergence criterion,  $\phi_h$ , and the analytical solution,  $\phi$ , is given in the  $L^2$  norm which can be stated as,

$$E_{L^2}^2 = \int_{\Omega} (\phi_h - \phi)^2 \, d\mathbf{x}, \quad (45)$$

and the DG norm which can be stated as,

$$E_{DG}^2 = \int_{\Omega} (\nabla(\phi_h - \phi))^2 \, d\mathbf{x} + \mu \int_S \llbracket \phi_h - \phi \rrbracket^2 \, d\mathbf{x}. \quad (46)$$

Established optimal convergence rates for SIPG semi-discretisations in the  $L^2$  norm are known to be  $h^{p+1}$ , and in the DG norm,  $h^p$ , [25], assuming the problem is sufficiently smooth. It is shown in [35], that for a problem which lacks this sufficient smoothness, the convergence rates fall back equal to the linear case for all  $p$ .

In the case that an analytical solution does not exist, there are two additional error measures which can be used to evaluate the efficacy of a reinitialisation method. The first is referred to as the *signed distance error measure*, which measures globally, the degree to which the computed solution satisfies the Eikonal equation, that is

$$E_{SD}^2 = \int_{\Omega} (|\nabla\phi_h| - 1)^2 \, d\mathbf{x}. \quad (47)$$

This signed distance error measure computes the difference between measures of the gradient of the solution, and therefore acts similarly to the  $H^1$  seminorm. 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. Experimental evidence supporting this is presented in [6].

The second of these, which will be referred to as the *interface error measure*, 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

$$E_{\text{int}}^2 = \int_{\Gamma(\phi^0)} \phi_h^2 \, d\mathbf{x}. \quad (48)$$

In all cases the interface error measure will be computed using Müller's method, see Section 2.3.

#### 3.2. Investigation into critical time step

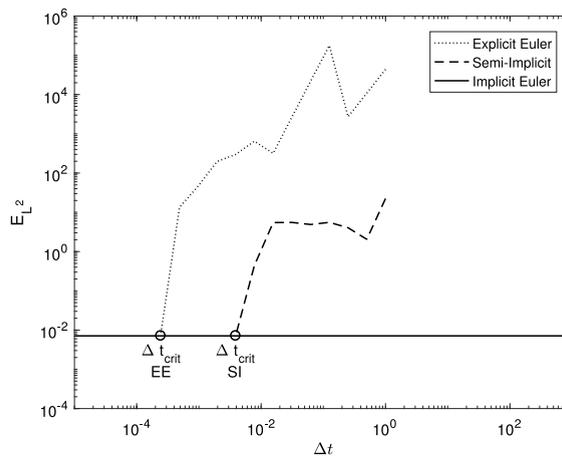
The first experiment will form an investigation into appropriate choices of time step,  $\Delta t$ , for the three time discretisations. In order to investigate this, a level set function with a circular interface, defined as

$$\phi^0 = x^2 + y^2 - 1, \quad (49)$$

is  $L^2$  projected onto the domain  $\Omega = (-2, 2)^2 \setminus (-0.4, 0.4)^2$ , which is then reinitialised using each of the three discretisations. The resulting signed distance function to be returned can be described by the analytical solution

$$\phi = \sqrt{x^2 + y^2} - 1. \quad (50)$$

The problem will be computed on a sequence of Cartesian meshes of bilinear, that is  $p = 1$ , square elements of size,  $h = 0.8, 0.4, 0.2, 0.1$ . For each mesh, the three methods will attempt to reinitialise the level set function as a signed distance function to the interface, for a sequence of time steps of magnitude,  $\Delta t^m = 2^{-(m-1)}$ , for  $m \in \mathbb{N}$ . For each time step,  $\Delta t^m$  the solver will run until either: the solution diverges, defined by the condition  $E_{SD}^n > 100$ , the solution stagnates, defined by the condition,  $E_{SD}^n < E_{SD}^{n-1}$  &&  $E_{SD}^{n-1} > E_{SD}^{n-2}$ , or the solution converges, defined by  $|E_{SD}^n - E_{SD}^{n-1}| < 10^{-8}$ . The first time step with which the method converges will be considered the critical time step,  $\Delta t_{\text{crit}}$ . For this experiment, the analytical solution is known to be singular at the origin. This is the reason for removing the region  $(-0.4, 0.4)^2$  from the domain, as it ensures that the problem is sufficiently smooth everywhere, and is a reasonable, if naive, approximation of a narrow band approach which could be successfully applied in more practical situations.



**Fig. 2.** Variation of error in the  $L^2$  norm of the converged solution with time step, for all three discretisations on a fixed mesh with  $h = 0.2$  and  $p = 1$ .

**Table 1**

Critical time steps with associated errors and computational expense for the three discretisations of the parabolic formulation, as well as the elliptic formulation. For SIPG-IE,  $\Delta t$  was set to 1000.

Method	$\Delta t_{crit}$	Number of iterations	Time per iteration (s), (Relative time)	Total time (s), (Relative time)	$E_{L2}$	$E_{SD}$
SIPG-EE	2.44e-4	40 475	1.17e-1, (1.28e-2)	4713.40, (130.05)	8.05e-3	6.51e-2
SIPG-SI	3.91e-3	3 101	2.60e-1, (2.87e-2)	806.45, (22.25)	7.25e-3	6.51e-2
SIPG-IE	-	4	9.06, (1)	36.24, (1)	7.10e-3	6.51e-2
Elliptic [6]	-	295	1.21e-1, (1.34e-2)	35.55, (0.98)	7.40e-3	6.51e-2

We begin the discussion by looking at the results for a fixed mesh. In this case we choose the mesh where  $h = 0.2$ , the results of which are shown in Fig. 2 with details given in Table 1. Fig. 2 shows what one might expect from the three time discretisations. For the fully implicit discretisation, it can be seen in Fig. 2 that the method is stable across the full range of tested time steps. For this experiment the maximum time step considered  $\Delta t^1 = 1$  was satisfactory and as such larger values were also considered for the implicit discretisation i.e.  $\Delta t = 10, 100, 1000$ . It was found that in all such cases the implicit method converged to the same error values up to the number of significant figures given in Table 1, with the number of iterations required reducing from 25 at  $\Delta t = 1$  to just 4 at  $\Delta t = 1000$ . As such the critical time step for the implicit discretisation was never reached in this experiment, and thus all results are given for  $\Delta t = 1000$ . For the fully explicit discretisation, Fig. 2 shows that the method always diverges for values of the time step which do not strictly satisfy the CFL condition, and is stable for any value smaller. For the semi-implicit discretisation, Fig. 2 shows that for time steps greater than some threshold value the method diverges, however there is also a region, in this case where  $10^{-2} < \Delta t < 0.5$ , whereby the solution stagnates and oscillates between multiple poor approximations. Once the critical time step is reached, the semi-implicit method is stable and furthermore it can be seen in Fig. 2 that the critical time step for the semi-implicit discretisation is larger than for the explicit discretisation.

Table 1 gives more specific information pertaining to the critical time step for the fixed mesh. It can be seen in Table 1 that all three of the proposed formulations converge to the same value of  $E_{SD}$  with slight variations in the  $E_{L2}$ . This is because the convergence (and divergence) criteria are functions of the signed distance error, and as such all solutions which do converge, converge to the same value of  $E_{SD}$ . Also included in Table 1 is a result using the elliptic reinitialisation method. As will be seen throughout the numerical examples section, the computed error to which the parabolic and elliptic methods converge is approximately equal in all cases.

Table 1 also provides a comparison of the relative computational expense. It can be seen that the critical time step for both the fully explicit and semi-implicit time schemes results in very large numbers of iterations required to achieve convergence. Such large numbers of iterations mean that in terms of computational time a semi-implicit discretisation for this problem was around 22 times more expensive than an implicit method, and the explicit method around 130 times more expensive. A result using the elliptic reinitialisation method is also presented and it can be seen that the time taken is roughly equivalent to the SIPG-IE parabolic formulation. It can be noted that all of the timing data presented in Table 1 was obtained using the tic function in MATLAB R2016b. All of the information was collected from a machine with an Intel i7-7700HQ CPU at 2.80 GHz. It should be noted that the relative time per iteration was calculated as the time taken to compute the time stepping loop to convergence divided by the number of iterations, normalised by the average time taken to compute one iteration of the SIPG-IE loop. It should also be noted that the resolution of the experiment is

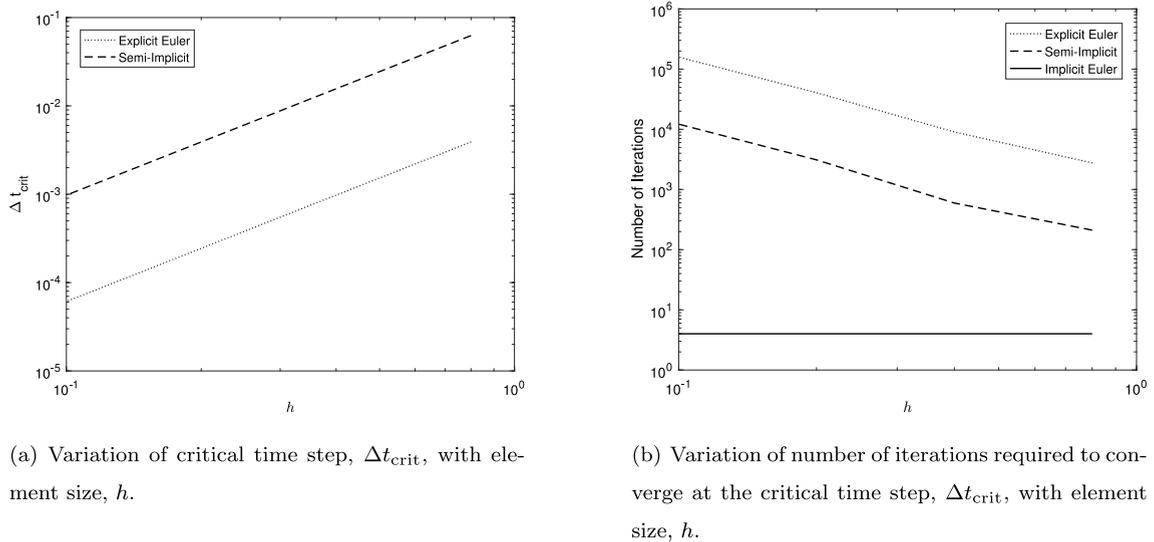


Fig. 3. Variation of critical time step and number of iterations for the three discretisations, for the circular interface reinitialisation problem.

fairly coarse in terms of the time steps considered, however, this coarseness does little to affect the main result of such an experiment.

Fig. 3 shows how the critical time step is related to  $h$  and further the effect of this on the number of iterations required to converge using the critical time step. As mentioned above the critical time step was never reached across the range of values tested for the SIPG-IE method and as such the number of iterations presented in Fig. 3(b) using SIPG-IE is for all  $h$  set to  $\Delta t = 1000$ . The ability to choose this value as a large constant as  $h$  varies, allows the number of iterations required to converge to also remain constant as  $h$  varies, for this problem. This fact further compounds the noted issues with expense of both the SIPG-EE and SIPG-SI formulations stated earlier, whereby at least for the forward Euler method the critical time step is known to be proportional to  $h^2$ , [28].

At this point it is clear that the comparative expense of the fully implicit discretisation is orders of magnitude less than either the explicit or semi-implicit discretisations. This is true despite the nonlinear solve required, due to the restrictions on time step and thus the large number of iterations required to reach a steady state for both the SI and EE discretisations. Given that there is no benefit accuracy-wise, as demonstrated by Fig. 2, it is thus recommended by the authors that for problems of this kind, one should adopt the implicit time discretisation. Thus, for the remaining problems we choose to focus only on the SIPG-IE formulation.

### 3.3. $h$ -convergence study: Circular interface

Given the equivalency in terms of accuracy, but the much reduced expense for the SIPG-IE method, we will now compute  $h$ -convergence studies using just the SIPG-IE discretisation for varying polynomial orders. For the first test, the same initial conditions as in Section 3.2 will be repeated i.e. a level set function with a circular interface defined by (49) is  $L^2$  projected onto the domain  $\Omega = (-2, 2)^2 \setminus (-0.4, 0.4)^2$ . The relative position of the interface and the domain boundaries can be seen in Fig. 4. The initial level set function is then reinitialised such that the analytical function describing the solution is that stated in (50). This is computed 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,  $p = 1, 2, 3$ . It should be noted for all meshes considered the time step is chosen as  $\Delta t = 100$ . The stopping criterion defining convergence in pseudotime is the same as in the previous experiment, that is  $|E_{SD}^n - E_{SD}^{n-1}| < 10^{-8}$ . Given that this is a relatively simple problem, which is smooth everywhere, it should be reasonable to expect that the proposed method demonstrates optimal rates of convergence in all of the norms discussed in Section 3.1. The convergence data collected, is presented in Fig. 5.

Fig. 5 demonstrates experimental orders of convergence congruent with the expected optimal convergence rates in all of the relevant norms including a convergence rate of  $h^{p+1}$  in the  $L^2$  norm and interface error, and a convergence rate of  $h^p$  in the DG norm and signed distance error. It can be noted that the quoted experimental order of convergence is computed in all cases using the difference between the error values where  $h = 0.4$  and  $h = 0.05$ . Results for the same example problem computed using the elliptic reinitialisation method are presented in [6], where it can be seen that the reported errors and quoted rates of convergence are essentially equivalent for both the elliptic reinitialisation method and the proposed parabolic reinitialisation method.

It is noted that for fixed time step  $\Delta t = 100$ , the number of iterations required for almost all of the meshes is equal to 5. The only exception to this was for the meshes with  $p = 3$  and  $h = 0.4$  and  $h = 0.05$ , where 4 and 6 iterations

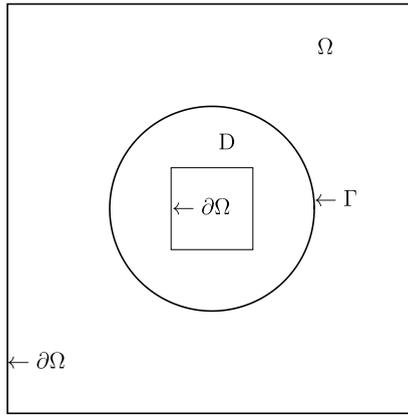
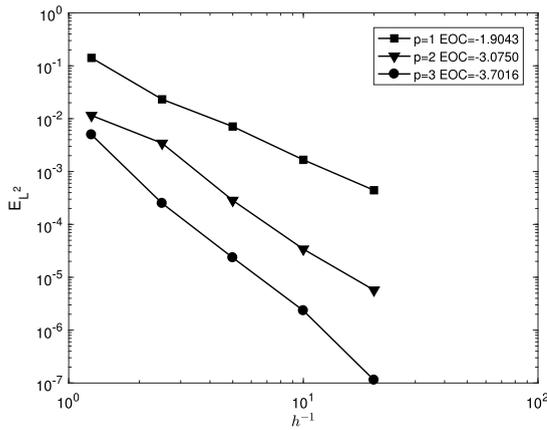
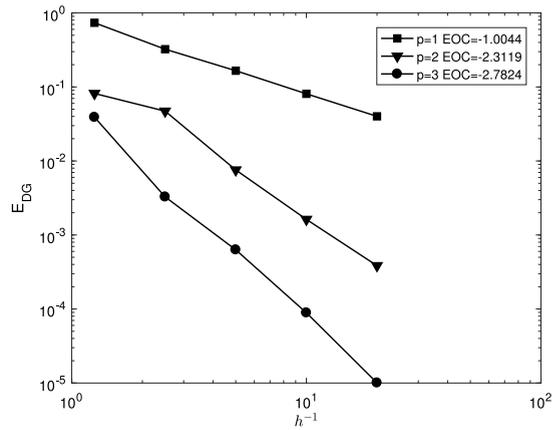


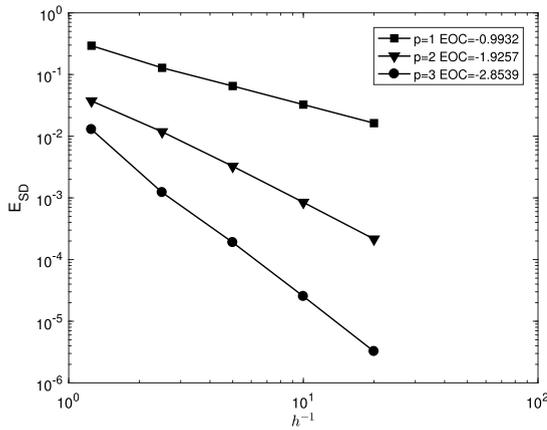
Fig. 4. Domain configuration for the circular interface problem where  $\Omega = (-2, 2)^2 \setminus (-0.4, 0.4)^2$ .



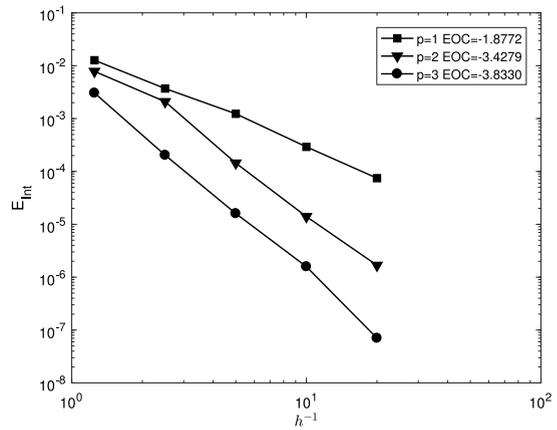
(a) Convergence in the  $L^2$  norm.



(b) Convergence in the DG norm.



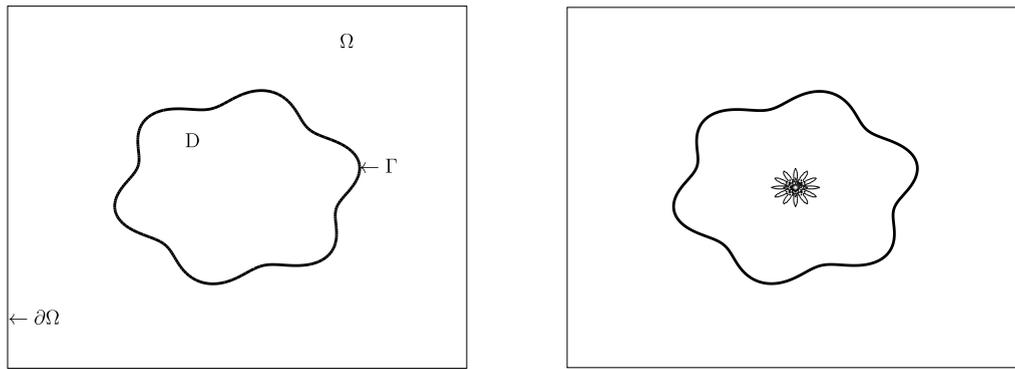
(c) Convergence using the signed distance error measure.



(d) Convergence using the interface error measure.

Fig. 5. Error data and convergence rates for the circular interface problem in the domain  $\Omega = (-2, 2)^2 \setminus (-0.4, 0.4)^2$ .

respectively were required to satisfy the convergence criterion. This does suggest that the number of iterations required may grow with  $h^{-1}$  and  $p$ . As shown in Table 1, 295 iterations are required to satisfy the convergence criterion using the



(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.

Fig. 6. Domain configuration for the smooth star interface problem where  $\Omega = (-2, 2)^2$ .

elliptic method where  $p = 1$  and  $h = 0.2$ , which lead to comparable run times for that given mesh, however, for the mesh where  $p = 3$  and  $h = 0.05$ , the elliptic method converges in just 5 iterations. In such a case it would be cheaper to use the elliptic formulation than the parabolic method, although it is difficult to approximate *a priori* the required number of iterations to satisfy the convergence criterion for either the parabolic or elliptic reinitialisation methods which makes it difficult to choose in which situation one would be preferable over the other in terms of computational expense.

### 3.4. *h*-convergence study: Smooth star interface

A second *h*-convergence study is presented for a slightly more complex interface without an analytical solution, and with an automatic narrow band. For this second test, an initial level set function with a *smooth star* interface, which can be described by the equation

$$\phi^0 = (x^2 + y^2) - \left(1 + 0.2 \sin\left(6 \arctan\left(\frac{y}{x}\right)\right)\right), \tag{51}$$

is  $L^2$  projected onto the domain  $\Omega = (-2, 2)^2$ . The relative position of this interface, and the natural boundary as well as the approximate position of the singular parts of the solution can be seen in Fig. 6. This initial level set function is then reinitialised for a series of Cartesian meshes of uniform polynomial order,  $p = 1, 2, 3$ , with square elements of sizes  $h = 0.4, 0.2, 0.1, 0.05$ . In this case, the narrow band is defined automatically for each mesh through the following condition; remove from the mesh any element which has a minimum absolute nodal value greater than two times the size of the smallest element,  $h_{min}$ . As such the size of the domain will be slightly different for each value of  $h$ . Furthermore, as there is not an analytical function which can describe the signed distance function to the interface of the level set function initialised as (51), the convergence data to be presented is only given using the signed distance and interface error measures. It should be noted for all meshes considered the time step is chosen as  $\Delta t = 1$ . The stopping criterion defining convergence in pseudotime is defined as  $|E_{SD}^n - E_{SD}^{n-1}| < 10^{-8}$ .

As the narrow band is now defined by the criterion stated above, it is possible (especially for the coarser meshes) that a signed distance function to the interface within that narrow band contains a singularity. Given that this is the case, one might reasonably expect for any set of meshes within which the solution is singular, the rate of convergence to be limited to the equivalent linear rate of convergence in the relevant norms. Once the mesh has been refined such that the criterion for the narrow band becomes sufficiently narrow to remove all singular parts of the solution from the domain, then one should expect the method to once again demonstrate optimal convergence. The computed convergence data is presented in Fig. 7.

Fig. 7(a) shows that beyond the first two data for each polynomial order, a convergence rate of  $h^p$  can be observed using the signed distance error measure which is equivalent to the previous example. In the region of the first datum, i.e  $h = 0.4$ , the mesh is not sufficiently refined and the narrow band is not sufficiently narrow, so as to remove the singularities from the solution. As expected the existence of any singularities within the computational domain means that convergence rates will be limited to equal to the linear case for all  $p$ , as explained in Section 2.6. This is the reason for our advocacy for measures such as narrow bands in the context of this work. It can be noted that the quoted experimental order of convergence is computed in all cases using the difference between the error values where  $h = 0.2$  and  $h = 0.05$ .

Fig. 7(b) shows convergence data in the interface error measure. Similar behaviour of the error using this norm is noted also in [6], whereby for any problem beyond the simple circular interface the error at any  $p$  for a given  $h$  is almost

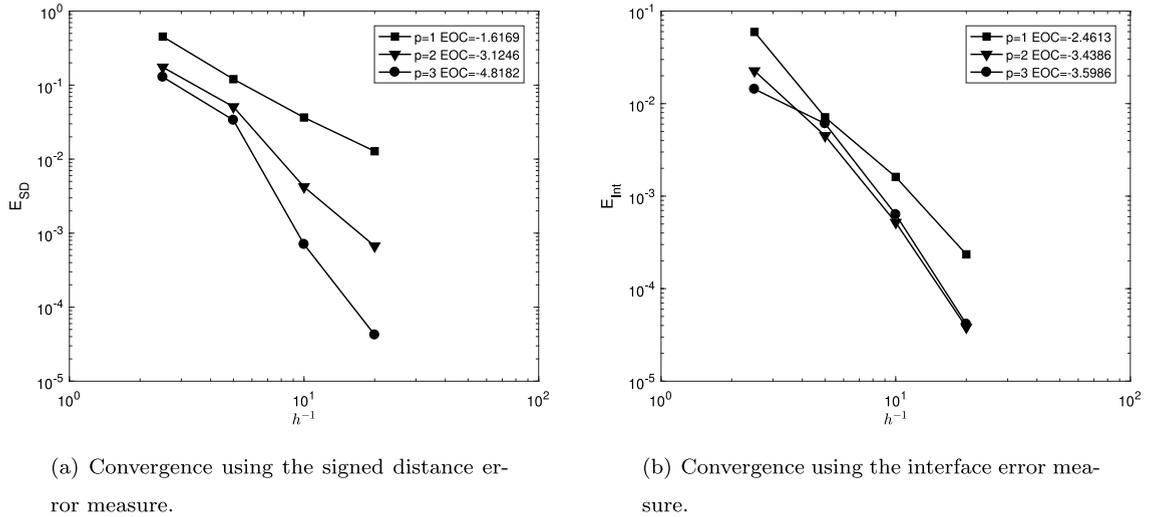


Fig. 7. Error data and convergence rates for the smooth star interface problem in the domain  $\Omega = (-2, 2)^2$ , with automatic narrow band.

equivalent. Again, it is difficult to explain this behaviour, and our comments are limited to the following; the error at the interface is much smaller with the Lagrange multiplier approach than observed with competitive methods in the literature, see [7] for example which presents data for multiple popular reinitialisation methods. Furthermore, we can note that the error seems to be related to element size,  $h$ , whereby one can expect the  $L^2$  error along the interface to decrease proportionally to at least  $h^2$  for all  $p$ .

In [6], the same example problem was computed using the elliptic reinitialisation method. The reported errors using the signed distance error measure are essentially equivalent for the two methods. This is less true for the interface error measure, however, we note that the general trend is the same and for the reasons stated above otherwise difficult to analyse.

For this problem there is slightly more variation in the number of iterations required to satisfy the convergence criterion. This number does not seem to scale with  $h$  although we note that this is likely due to the variations in the domain size due to the automatic narrow band. Furthermore, we note that the average number of iterations for all of the meshes tested is 14 with a maximum number of required iterations equal to 52, for the mesh where  $p = 3$  and  $h = 0.2$ . The same problem computed using the elliptic method also has a somewhat random distribution of number of iterations required to converge, with an average of 563 iterations. As there is no pattern in the required number of iterations for a given level set function on a given mesh, it is again difficult *a priori* to determine whether or not it is advantageous in terms of computational expense to choose one formulation over the other. This echoes the comparison made for the previous numerical experiment in Section 3.3

### 3.5. $h$ -convergence study: Multiple interfaces

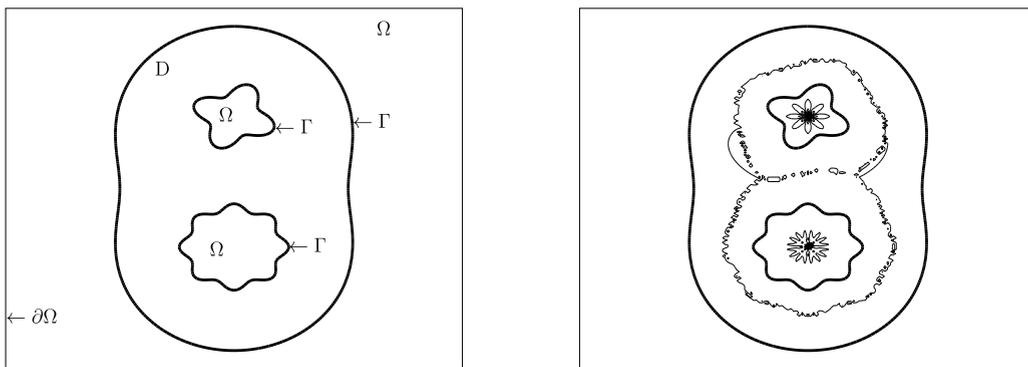
A final numerical example contains an even more complex interface, which might more closely resemble an interface one might encounter in a practical situation; that is multiple nested interfaces of various curvatures. In this case, the initial level set function is defined as the maximum value of one of three analytical functions; that is

$$\phi^0 = \max(q_k), \quad k = 1, 2, 3, \tag{52}$$

where

$$\begin{aligned} q_1 &= 1.5 \left( \sqrt{x^2 + y^2} - \left( 1 + 0.8 \sin \left( \arctan \left( \frac{y}{x} \right) \right)^2 \right) \right), \\ q_2 &= -2 \left( \sqrt{x^2 + y^2} - \left( 0.3 - 0.075 \sin \left( 4 \arctan \left( \frac{y - 0.8}{x} \right) \right) \right) \right), \\ q_3 &= -2 \left( \sqrt{x^2 + y^2} - \left( 0.48 - 0.08 \sin \left( 4 \arctan \left( \frac{y - 0.65}{x} \right) \right) \right) \right). \end{aligned} \tag{53}$$

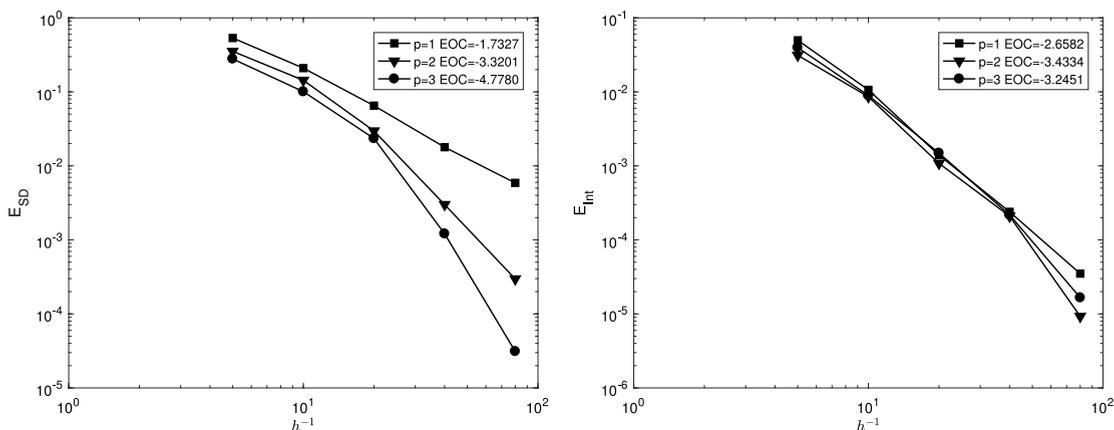
This is sampled at the Gauss points and then  $L^2$  projected onto the domain  $\Omega = (-2, 2)^2$ . The original configuration of this mesh can be seen in Fig. 8(a). It is noted that whilst these curves have been chosen somewhat arbitrarily, considerations were made such that across the domain the problem has a range of gradients and curvatures to be dealt



(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.

Fig. 8. Domain configuration for the multiple interface problem where,  $\Omega = (-2, 2)^2$ .



(a) Convergence using the signed distance error measure.

(b) Convergence using the interface error measure.

Fig. 9. Error data and convergence rates for the multiple interface problem in the domain  $\Omega = (-2, 2)^2$ , with automatic narrow band.

with. This initial level set function is then reinitialised for a series of Cartesian meshes of uniform polynomial order,  $p = 1, 2, 3$ , with square elements of size  $h = 0.2, 0.1, 0.05, 0.025, 0.0125$ . This particular problem again does not have a solution which can be described by an analytical equation and as such only the signed distance and interface errors will be reported in this case. Furthermore, an automatic narrow banding will be applied to the problem restricting the domain size by the following rule: remove from the mesh any element which has a minimum absolute nodal value greater than four times the size of the smallest element,  $h_{min}$ .

In computing the solution to the example problem, described by (52) and (53), using the SIPG-IE discretisation, it was found that stability was determined by an appropriate choice of the time step,  $\Delta t$ . In particular, it was the coarsest meshes (i.e. where  $h_{min}$  is large) tested which were the least stable, with the required time step decreasing with increasing polynomial order. The reason for this is therefore likely that the singularities present in the solution (which are quite severe for this example as can be seen in Fig. 8(b)) are poorly resolved by the mesh leading to overshoots and oscillations for larger time steps. This does provide evidence suggesting that the SIPG-IE discretisation of problem, (18), is only conditionally stable. As such for this numerical example, the time step is chosen such that  $\Delta t = 0.01$ , for all meshes. We do note here however, that for the denser meshes, particularly those in the region where optimal rates of convergence are demonstrated, larger time steps do once again become viable. For the presented error data the stopping criterion defining convergence in pseudotime is defined as  $|E_{SD}^n - E_{SD}^{n-1}| < 10^{-8}$ .

The error data shown in Fig. 9 echoes the previous example problem shown in Section 3.4. That is, for meshes where the narrow banding is not sufficient to remove the singular parts of the solution the signed distance error measure converges

with a rate equal to linear convergence for all polynomial orders. Beyond this point, convergence of this error aligns with the theoretically optimal. The quoted rates of convergence for the signed distance error are actually greater than the expected  $h^p$ , but again this is likely due to the differing domain sizes based on the automatic narrow banding rule. Similarly, the interface error measure gives approximately equal values for all polynomial orders for a given element size and converges with a rate of at least  $h^2$  for all polynomial orders. It should be noted that the quoted experimental order of convergence in all cases is computed using the values for the meshes where  $h = 0.05$  and  $h = 0.0125$ .

#### 4. Conclusions

After experimentation with a number of time discretisations for the solution to the parabolic reinitialisation problem it was determined that the relative expense of both explicit and semi-implicit time schemes would be prohibitively large for any practical scenario. As such an appropriate practical solution is presented through discretisation of the parabolic reinitialisation problem using a symmetric interior penalty discontinuous Galerkin method in space and implicit Euler method in time. Numerical experiments show that the method is capable of high-order accuracy and experimental orders of convergence are equivalent to optimal orders of convergence for SIPG discretisations. As in [6], optimal convergence is dependent upon an as of yet poorly defined, ‘sufficiently narrow’ narrow band and ‘sufficiently refined’ mesh. This is what has prompted the next stage of research which is the development of error estimators, and  $hp$ -refinement and narrow band strategies for DG level set methods.

#### Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council, UK (grant EP/M507854/1). All data created during this research is openly available at [doi:10.15128/r2x346d417r](https://doi.org/10.15128/r2x346d417r).

#### References

- [1] J. Grooss, J.S. Hesthaven, A level set discontinuous Galerkin method for free surface flows, *Comput. Methods Appl. Mech. Engrg.* 195 (25) (2006) 3406–3429.
- [2] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, *J. Comput. Phys.* 194 (1) (2004) 363–393.
- [3] J.A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, second ed., in: Cambridge Monographs on Applied and Computational Mathematics, vol. 3, Cambridge University Press, Cambridge, U.K. ; New York, 1999.
- [4] E. Maitre, T. Milcent, G.-H. Cottet, A. Raoult, Y. Usson, Applications of level set methods in computational biophysics, *Math. Comput. Modelling* 49 (11–12) (2009) 2161–2169.
- [5] S. Osher, R.P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, in: Applied Mathematical Sciences, vol. 153, Springer, New York, 2003.
- [6] T. Adams, S. Giani, W.M. Coombs, A high-order elliptic PDE based level set reinitialisation method using a discontinuous Galerkin discretisation, *J. Comput. Phys.* 379 (2019) 373–391.
- [7] C. Basting, D. Kuzmin, A minimization-based finite element formulation for interface-preserving level set reinitialization, *Computing* 95 (S1) (2013) 13–25.
- [8] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1) (1988) 12–49.
- [9] W. Mulder, S. Osher, J.A. Sethian, Computing interface motion in compressible gas dynamics, *J. Comput. Phys.* 100 (2) (1992) 209–228.
- [10] D.L. Chopp, Computing minimal surfaces via level set curvature flow, *J. Comput. Phys.* 106 (1) (1993) 77–91.
- [11] 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.
- [12] C. Basting, D. Kuzmin, J.N. Shadid, Optimal control for reinitialization in finite element level set methods, *Internat. J. Numer. Methods Fluids* 84 (5) (2017) 292–305.
- [13] P. LeSaint, P. Raviart, On a finite element method for solving the neutron transport problem, *Math. Aspects Finite Elem. Partial Differential Equations* (1974) 89–123.
- [14] B. Cockburn, Discontinuous Galerkin methods for convection-dominated problems, in: M. Griebel, D.E. Keyes, R.M. Nieminen, D. Roose, T. Schlick, T.J. Barth, H. Deconinck (Eds.), *High-Order Methods for Computational Physics*, Vol. 9, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 69–224.
- [15] B. Cockburn, G.E. Karniadakis, C.-W. Shu, The development of discontinuous Galerkin methods, in: M. Griebel, D.E. Keyes, R.M. Nieminen, D. Roose, T. Schlick, B. Cockburn, G.E. Karniadakis, C.-W. Shu (Eds.), *Discontinuous Galerkin Methods*, Vol. 11, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 3–50.
- [16] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, *J. Sci. Comput.* 16 (3) (2001) 173–261.
- [17] D.N. Arnold, F. Brezzi, B. Cockburn, D. Marini, Discontinuous Galerkin methods for elliptic problems, in: *Discontinuous Galerkin Methods*, in: Lecture Notes in Computational Science and Engineering, Springer, Berlin, Heidelberg, 2000, pp. 89–101.
- [18] P. Houston, J. Robson, E. Süli, Discontinuous Galerkin finite element approximation of quasilinear elliptic boundary value problems I: The scalar case, *IMA J. Numer. Anal.* 25 (4) (2005) 726–749.
- [19] D.N. Arnold, An interior penalty finite element method with discontinuous elements, *SIAM J. Numer. Anal.* 19 (4) (1982) 742–760.
- [20] G.A. Baker, Finite element methods for elliptic equations using nonconforming elements, *Math. Comp.* 31 (137) (1977) 45–49.
- [21] M.F. Wheeler, An elliptic collocation-finite element method with interior penalties, *SIAM J. Numer. Anal.* 15 (1) (1978) 152–161.
- [22] J. Oden, I. Babuška, C.E. Baumann, A discontinuous  $hp$  finite element method for diffusion problems, *J. Comput. Phys.* 146 (2) (1998) 491–519.
- [23] B. Rivière, M.F. Wheeler, V. Girault, Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems, Part I, *Comput. Geosci.* 3 (3–4) (1999) 337–360.
- [24] B. Rivière, M.F. Wheeler, A discontinuous Galerkin method applied to nonlinear parabolic equations, in: M. Griebel, D.E. Keyes, R.M. Nieminen, D. Roose, T. Schlick, B. Cockburn, G.E. Karniadakis, C.-W. Shu (Eds.), *Discontinuous Galerkin Methods*, Vol. 11, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 231–244.

- [25] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, *SIAM J. Numer. Anal.* 39 (5) (2002) 1749–1779.
- [26] V. Dolejší, Analysis and application of the IIPG method to quasilinear nonstationary convection–diffusion problems, *J. Comput. Appl. Math.* 222 (2) (2008) 251–273.
- [27] T. Gudi, A.K. Pani, Discontinuous Galerkin methods for quasi-linear elliptic problems of nonmonotone type, *SIAM J. Numer. Anal.* 45 (1) (2007) 163–192.
- [28] L. Song, Fully discrete interior penalty discontinuous Galerkin methods for nonlinear parabolic equations, *Numer. Methods Partial Differential Equations* 28 (1) (2012) 288–311.
- [29] L. Song, G.-M. Gie, M.-C. Shiue, Interior penalty discontinuous Galerkin methods with implicit time-integration techniques for nonlinear parabolic equations, *Numer. Methods Partial Differential Equations* 29 (4) (2013) 1341–1366.
- [30] P. Castillo, Performance of discontinuous Galerkin methods for elliptic PDEs, *SIAM J. Sci. Comput.* 24 (2) (2002) 524–547.
- [31] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, *SIAM J. Numer. Anal.* 35 (6) (1998) 2440–2463.
- [32] F. Santambrogio, Euclidean, metric, and wasserstein gradient flows: an overview, *Bull. Math. Sci.* 7 (1) (2017) 87–154.
- [33] B. Müller, F. Kummer, M. Oberlack, Highly accurate surface and volume integration on implicit domains by means of moment-fitting, *Internat. J. Numer. Methods Engrg.* 96 (8) (2013) 512–528.
- [34] V. Dolejší, Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows, *Commun. Comput. Phys.* 4 (2) (2008) 231–274.
- [35] S. Giani, E.J.C. Hall, An a posteriori error estimator for hp-adaptive discontinuous Galerkin methods for elliptic eigenvalue problems, *Math. Models Methods Appl. Sci.* 22 (10) (2012) 1250030.