



An arbitrary-order Runge–Kutta discontinuous Galerkin approach to reinitialization for banded conservative level sets



Z. Jibben^{a,*}, M. Herrmann^b

^a Computational Physics and Methods (CCS-2), Los Alamos National Laboratory, Los Alamos, NM 87544, USA

^b Department of Mechanical and Aerospace Engineering, Arizona State University, Tempe, AZ 85287, USA

ARTICLE INFO

Article history:

Received 30 December 2016

Received in revised form 4 August 2017

Accepted 16 August 2017

Available online 24 August 2017

Keywords:

Discontinuous Galerkin

Arbitrary order

Conservative level set

Multiphase flow

ABSTRACT

We present a Runge–Kutta discontinuous Galerkin method for solving conservative reinitialization in the context of the conservative level set method [18,19]. This represents an extension of the method recently proposed by Owkes and Desjardins [21], by solving the level set equations on the refined level set grid [9] and projecting all spatially-dependent variables into the full basis used by the discontinuous Galerkin discretization. By doing so, we achieve the full $k + 1$ order convergence rate in the L_1 norm of the level set field predicted for RKDG methods given k th degree basis functions when the level set profile thickness is held constant with grid refinement. Shape and volume errors for the 0.5-contour of the level set, on the other hand, are found to converge between first and second order. We show a variety of test results, including the method of manufactured solutions, reinitialization of a circle and sphere, Zalesak's disk, and deforming columns and spheres, all showing substantial improvements over the high-order finite difference traditional level set method studied for example by Herrmann [9]. We also demonstrate the need for k th order accurate normal vectors, as lower order normals are found to degrade the convergence rate of the method.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

A critical problem in many areas of engineering and computational science is the tracking of interfaces between immiscible media. A common method for capturing these interfaces is the level set (LS) method, originally developed by Osher and Sethian [20]. Level sets have several benefits over volume of fluid (VOF) methods, such as straightforward calculation of normal vectors and curvature. They also allow for high order numerical methods, while VOF methods are currently limited to second order. The possibility of high order convergence rates makes level sets well suited to problems requiring high accuracy, since this can in principle be obtained with a far coarser grid.

However, numerical discretizations of the traditional level set method are well known for their lack of mass conservation, causing volume errors to accumulate at each time step. Several methods have been proposed to address the issue of mass loss [5,28], one of which being the conservative level set (CLS) method by Olsson Kreiss [18] and Olsson, Kreiss, and Zahedi [19]. They take advantage of the solenoidal property of incompressible velocity fields to rewrite the advection equation such that the level set scalar is a conserved quantity. If the level set scalar was the discontinuous Heaviside function and advected via a perfect conservative numerical scheme, no volume loss would occur and droplet volume would be

* Corresponding author.

E-mail address: zjibben@asu.edu (Z. Jibben).

equal to the integral of the level set scalar over the domain. For numerical robustness, they choose the level set profile to mimic a smeared out Heaviside function. As a result, the integral of the level set scalar over the domain only approximates the droplet volume, and conservation is improved as the profile thickness decreases. This, combined with the conservation of the level set scalar, mitigates volume loss compared to the original level set method. However, the level set profile deviates from a smeared Heaviside function due to dissipative errors introduced by the numerical scheme and due to the fact that local iso-surfaces are transported with the local velocity field, steepening or flattening the profile. This necessitates reinitialization to restore the original profile. To address this, they developed a conservative reinitialization equation which converges to a hyperbolic tangent profile, which they choose to represent the smeared Heaviside function. Desjardins, Moureau, and Pitsch [4] showed that oscillatory errors are introduced by reinitialization if the normal is calculated from local variables. Their accurate conservative level set (ACLS) method corrects these errors by instead calculating the normal from the interface geometry alone via a fast marching method. Zhao et al. [34] calculated interface normals from a signed distance function solved by the traditional LS method alongside the CLS method, similarly finding that calculating normals independent of local CLS variables improved their results. Moghadam, Shafieefar, and Panahi [17] introduced the compact conservative level set method, which solves a modified form of the CLS equations with compact finite difference discretizations. Their method showed improved results over the ACLS method [4] by using a compact difference numerical scheme. Recently, McCaslin and Desjardins [16] demonstrated that reinitialization transports the level set interface, introducing errors in regions that are intended to be stationary. To mitigate these errors, they developed a localized reinitialization equation in which the amount of reinitialization performed varies over the spatial domain. Waławczyk [31] used the signed distance function as a mapping function to reformulate discretizations of the conservative reinitialization equation to avoid these interface deformations.

Since there are no known methods for solving the CLS equations exactly for many engineering problems, numerical approaches are needed. Discontinuous Galerkin (DG) methods [23] have recently grown in popularity since they allow for high convergence rates with a local stencil, as opposed to traditional finite difference methods which require large stencil sizes for the same convergence rates. They spatially discretize solution variables with a spectral decomposition within computational cells, allowing for high-order flux terms between neighbors. This makes DG methods both amenable to unstructured grids and very scalable in high performance computing clusters. Runge–Kutta discontinuous Galerkin (RKDG) methods combine Runge–Kutta time stepping with the DG spatial discretization to form a stable, arbitrarily high-order method suitable for nonlinear problems [3].

Atkins and Shu [1] developed a quadrature-free discontinuous Galerkin method, which they found to be more efficient than similar quadrature-based approaches for several linear and non-linear problems. This is due to the fact that the quadrature-free calculation does not perform the operations to evaluate their modal DG representation at quadrature points. We follow an approach similar to theirs, avoiding quadrature where possible. Except where noted, integrals in this paper are evaluated symbolically and tabulated a priori. These integral arrays are sparse, which we take advantage of to avoid unnecessary floating-point operations. In calculations requiring quadrature due to some non-polynomial integrand, we use Gauss–Legendre quadrature rules.

DG formulations of traditional level sets have been developed previously [7,8,14,15,22], however these methods do not take advantage of the mass conservation improvements from CLS. Owkes and Desjardins [21] developed a quadrature-free RKDG-CLS method, but they assume velocity and normal vectors to be constant within a cell rather than also expanding them into the DG basis. We will show that this limits their method to be formally first order only, regardless of the polynomial order of the DG basis function for the level set scalar. This paper presents an extension of their work by projecting all spatially-dependent variables into the DG basis, as is necessary to obtain the full convergence rates predicted by the RKDG method. We also implement the method on the refined level set grid (RLSG) [9] to mitigate computational expense by not performing level set calculations in regions of the domain where they are not essential.

2. Mathematical formulation and methodology

2.1. Conservative level sets

The CLS method developed by Olsson and Kreiss [18] and Olsson, Kreiss, and Zahedi [19] implicitly defines the interface as the 0.5 iso-surface of a level set scalar, G . This scalar is defined in relation to the interface to have a hyperbolic tangent profile:

$$G(\mathbf{x}, t) = \frac{1}{2} \left(\tanh \left(\frac{\phi(\mathbf{x}, t)}{2\varepsilon} \right) + 1 \right) \quad (1)$$

Here, ϕ is the traditional signed distance function to the interface. The width of the profile is determined by ε , generally taken to be half the grid cell width. The interface is advected using the conservative advection equation, assuming the velocity is divergence-free.

$$\frac{\partial G}{\partial t} + \nabla \cdot (G\mathbf{u}) = 0. \quad (2)$$

However, advection does not explicitly maintain the hyperbolic tangent profile, and in fact destroys it. Numerically solving the advection equation introduces dissipative errors due to the total variation diminishing (TVD) method used [18]. Non-0.5-iso-surfaces are also advected by the local velocity, which itself may asymmetrically smear or sharpen the interface. Without correction, the level set scalar will diffuse out as t increases, distorting the shape of the iso-surface. To avoid this problem, reinitialization is performed via Eq. (3) at regular intervals to restore the hyperbolic tangent profile. Reinitialization is evaluated in pseudo-time, ideally until convergence.

$$\frac{\partial G}{\partial \tau} + \nabla \cdot (G(1-G)\hat{\mathbf{n}}) = \nabla \cdot (\varepsilon(\nabla G \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}) \quad (3)$$

Here, $\hat{\mathbf{n}}$ is a vector field describing the normal to the interface. There are a variety of methods for calculating the normal vector, Olsson and Kreiss [18] originally suggesting it be done from the local G field.

$$\hat{\mathbf{n}} = \frac{\nabla G}{|\nabla G|} \quad (4)$$

However, Desjardins, Moureau, and Pitsch [4] found that local variations in the G field can cause wildly varying normal vectors away from the interface. As a result, it is possible for reinitialization to sharpen these small variations to a hyperbolic tangent and create false interfaces. To counteract this, they suggested the accurate conservative level set method (ACLS) which requires that the normal vectors be generated from information in the vicinity of the $G = 0.5$ iso-surface alone. They did this by calculating the normalized gradient to the signed distance function, which was computed using a fast marching method (FMM). However, FMMs have not yet been devised in a DG context with high order. Fast sweeping methods are a possible alternative, however second order DG methods are the highest order that have been achieved to date [13,33], and higher orders are not trivial. An alternative approach to generating arbitrary high order distance function fields would be to calculate the minimum distance at quadrature points by iteratively identifying the closest point constrained to lie on the $G = 0.5$ isosurface using a Newton–Jacobi method, as developed by Saye [26]. This approach is presented in Section 2.4.2 to provide a proof-of-concept demonstration of a RKDG-ACLS method.

2.2. The refined level set grid

The RKDG-CLS method is implemented on the refined level set grid (RLSG) proposed by Herrmann [9]. With the RLSG, the CLS equations are solved on a Cartesian mesh separate from the flow solver, and cells are organized into blocks of a predefined size. To reduce the necessary computational work, the CLS equations are only solved in cells and blocks within a user-defined distance from the interface. Beyond this narrow band, new cells are added or removed as the interface is advected through the computational domain. New cells are given a value of 0 or 1, depending on whether they are inside or outside the interface, and act as Dirichlet condition ghost cells for the narrow band structure. The narrow band structure is possible because advection and reinitialization only need to be solved inside a region surrounding the interface itself. That is, computational effort need not be wasted advecting contours that do not directly affect the 0.5 iso-surface. For details on band generation and parallelization of the refined level set grid, see [9]. Note that banding the mesh affects global conservation of G , since G that is advected outside the bands is lost. However, the conservation of G itself has no physical meaning, since it is the volume enclosed by the $G = 0.5$ iso-surface that is physically conserved in incompressible flows. Section 3.4, Section 3.5, and Section 3.6 discuss this in detail.

2.3. The Runge–Kutta discontinuous Galerkin method

The discontinuous Galerkin method involves spatially discretizing solution variables first into cells, like in the finite volume method. The solution in each cell is then spectrally expanded into some basis, denoted $\{b_i\}$. By doing so, information on derivatives is contained locally. For the purposes of this paper, a regular Cartesian grid is used, although this can be generalized to other mesh types.

The space is defined such that sub-cell coordinates $\xi = [\xi, \eta, \zeta]^T$ are bounded by a cube $\mathcal{K} = [-1, 1]^3$ which defines the edges of a cell. We can map between sub-cell and physical domain Ω coordinates x, y, z via the relations

$$\begin{aligned} \mathcal{K} \rightarrow \Omega: \quad x_\alpha &= x_{\alpha,\kappa} + \xi_\alpha \frac{\Delta x_\alpha}{2} \\ \Omega \rightarrow \mathcal{K}: \quad \xi_\alpha &= \frac{x_\alpha - x_{\alpha,\kappa}}{\Delta x_\alpha/2} \end{aligned} \quad (5)$$

where α denotes a particular direction, Δx_α denotes the cell width in that direction, and $x_{\alpha,\kappa}$ denotes the cell center coordinate. Here, we use the Einstein summation convention on non-Greek indicies.

We choose the normalized Legendre polynomial basis, since their orthonormality makes an identity mass matrix,

$$\int_{\mathcal{K}} b_i b_j \, dV = \delta_{ij}. \quad (6)$$

Table 1
Fill fraction for advection integral arrays.

k	2D simulation integrals			3D simulation integrals		
	Total size	Volume	Surface	Total size	Volume	Surface
1	64	12.5%	50.0%	512	6.25%	25.0%
2	729	10.6%	40.7%	19683	4.30%	16.6%
3	4096	10.1%	35.9%	262144	3.63%	12.9%
4	15625	9.68%	33.6%	1953125	3.25%	11.3%

The basis is constructed by performing the Gram–Schmidt process on the space of 3D monomials $x^\alpha y^\beta z^\eta$. Then, for a maximum monomial degree k , there are $N_f = (k+1)^3$ degrees of freedom per cell. For a given cell κ , solution variables are projected into the basis such that

$$\mathbf{f}(\mathbf{x}, t) = \sum_{i=1}^{\infty} \mathbf{f}_i^\kappa(t) b_i(\xi) \approx \sum_{i=1}^{N_f} \mathbf{f}_i^\kappa(t) b_i(\xi), \quad (7)$$

where N_f is the number of degrees of freedom of our numerical representation of \mathbf{f} (higher order contributions are truncated).

2.3.1. Advection

By taking Eq. (7) and inserting it into Eq. (2) (advection), and upwinding the flux, we arrive at

$$\frac{d\mathbf{g}_n^\kappa}{dt} = u_k^{\kappa,j} g_i^\kappa \frac{2}{\Delta x} \int_{\mathcal{K}} b_k b_i \frac{\partial b_n}{\partial \xi_j} dV + u_k^{\text{up},j} g_i^{\text{up}} \frac{2}{\Delta x} \int_{\partial \mathcal{K}} N_j b_k^{\text{up}} b_i^{\text{up}} b_n dS. \quad (8)$$

We select the upwind direction by the sign of the total flux through that face, given the velocity of the cell on the “+” side of the face. Note that this results in a constant upwind direction for a given face, though in principle this may not be true. In Section 3.1 we find that this results in a decreased convergence rate in the L_∞ norm. This system of coupled ordinary differential equations describes the time evolution of the DG coefficients. It is then integrated in time using an explicit $k+1$ order Runge–Kutta (RK) total variation diminishing (TVD) approach [3,6]. The timestep restriction is given by Cockburn and Shu [3]:

$$\max |\mathbf{u}| \frac{\Delta t}{\Delta x} \leq \frac{1}{2k+1} \quad (9)$$

The integrals in Eq. (8) are entirely in terms of the normalized Legendre polynomial basis functions, and are integrated over the volume or surface of a cube. This makes it possible for them to be evaluated symbolically, as performed by Atkins and Shu [1]. They also happen to be sparse; Table 1 shows the fraction of nonzero elements for these arrays compared to the total number of elements. We take advantage of this by employing a compressed sparse row (CSR) format to avoid the storage and floating point operation cost associated with the many elements with value zero.

2.3.2. Reinitialization

A similar approach is used to discretize reinitialization, Eq. (3). Here, the nonlinear convection term benefits from a local Lax–Friedrichs (LLF) flux approximation due to the frequently intersecting characteristics in the reinitialization equation. We have

$$\widehat{\mathbf{f} \cdot \hat{\mathbf{n}}}^{\text{LLF}}(G^-, G^+) = \frac{1}{2} ((\mathbf{f}(G^-) + \mathbf{f}(G^+)) \cdot \hat{\mathbf{n}} - C(G^+ - G^-)) \quad (10)$$

$$C = \max_{\min(G^-, G^+) \leq s \leq \max(G^-, G^+)} |\hat{\mathbf{n}} \cdot \mathbf{f}'(s)|$$

where the “+” and “−” superscripts indicate the solution on the positive and negative side of the face, respectively. To calculate the coefficient C , we sample the value of $|\hat{\mathbf{n}} \cdot \mathbf{f}'(s)|$ at quadrature points on the cell face, from both the “+” and “−” solutions. C is then set to the maximum of those values.

The LLF approach is not appropriate for the diffusive flux, so we employ the a reconstruction method developed by Luo et al. [12]. This involves projecting the solution from two neighboring cells into one set of coefficients that are associated with a basis extended across the two cells. That is, coefficients associated with neighboring domains \mathcal{K}^+ and \mathcal{K}^- are projected into a single domain $\tilde{\mathcal{K}}$ bisected by the cell face, as shown in Fig. 1. For a given face α , the coefficients for G and $\hat{\mathbf{n}}$ on the shared cell are referred to as \tilde{G} and $\tilde{\hat{\mathbf{n}}}$, and are calculated by projecting each cell’s solution variables into the shared basis, resulting in

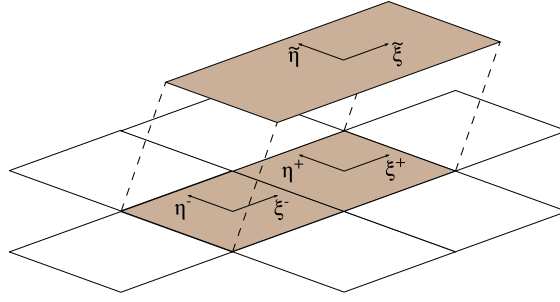


Fig. 1. Two-cell projection in x-direction.

$$\tilde{f}_n^\alpha = f_{i,\kappa}^{\alpha-} \int_{\tilde{\mathcal{K}}^{\alpha-}} b_i \tilde{b}_n^\alpha d\tilde{V}^\alpha + f_{i,\kappa}^{\alpha+} \int_{\tilde{\mathcal{K}}^{\alpha+}} b_i \tilde{b}_n^\alpha d\tilde{V}^\alpha. \quad (11)$$

Here, $\tilde{\mathcal{K}}^{\alpha\pm}$ refers to the \pm half of the domain $\tilde{\mathcal{K}}$. The flux is then evaluated from the coefficients associated with the shared basis along the center-line which coincides with the location of the shared face.

$$\hat{\mathbf{f}}^{\text{recons.}}(G^-, G^+, \hat{\mathbf{n}}^-, \hat{\mathbf{n}}^+) = \mathbf{f}(\tilde{G}, \tilde{\mathbf{n}}). \quad (12)$$

One question worth noting is whether the shared basis $\{\tilde{b}\}$ ought to be enriched. To roughly preserve information from the child cells, we choose the maximum polynomial degree in the shared cell to be equal to the sum of the child cell maximum polynomial degrees plus another degree to approximate the discontinuous jump between cells. This new enriched basis has $k_{\text{en}} = 2k + 1$ maximum polynomial degree and $N_{\text{en}} = (k_{\text{en}} + 1)^d$ degrees of freedom, where d is the number of dimensions in the problem (i.e., 2 or 3 for 2D and 3D, respectively). Test results in Section 3.1 indicate that while enriching the basis improves overall accuracy, it is not necessary for achieving the predicted $k + 1$ convergence rate, compared to the shared cell having the same number of degrees of freedom as the child cells (i.e., $k_{\text{shared}} = k_{\text{child}}$).

With an appropriate change of variables on the derivatives, the shared-cell coefficients are then used to evaluate the diffusive flux. In the shared coordinate space, the cell face simply is at the center of the extended dimension. For example, a face whose normal is aligned with the x -axis sits on $\tilde{\xi} = 0$. Utilizing the reconstruction and LLF flux schemes gives the form of the DG scheme for reinitialization.

$$\begin{aligned} \frac{d\mathbf{g}_n^\kappa}{d\tau} = & n_k^{\kappa,j} g_i^\kappa \frac{2}{\Delta x} \int_{\mathcal{K}} b_k b_i \frac{\partial b_n}{\partial \xi_j} dV - n_k^{\kappa,j} g_i^\kappa g_j^\kappa \frac{2}{\Delta x} \int_{\mathcal{K}} b_k b_i b_j \frac{\partial b_n}{\partial \xi_j} dV \\ & - \varepsilon g_i^\kappa n_k^{\kappa,a} n_l^{\kappa,d} \left(\frac{2}{\Delta x} \right)^2 \int_{\mathcal{K}} \frac{db_i}{d\xi_a} b_k b_l \frac{db_n}{d\xi_d} dV \\ & - \frac{1}{\Delta x} (g_i^{f+} \hat{\mathbf{n}}_k^{f+} + g_i^{f-} \hat{\mathbf{n}}_k^{f-}) \cdot \hat{\mathbf{N}}^f \int_{\partial^f \mathcal{K}} b_i^{f-} b_k^{f-} b_n dS \\ & + \frac{1}{\Delta x} (g_i^{f+} g_j^{f+} \hat{\mathbf{n}}_k^{f+} + g_i^{f-} g_j^{f-} \hat{\mathbf{n}}_k^{f-}) \cdot \hat{\mathbf{N}}^f \int_{\partial^f \mathcal{K}} b_i^{f-} b_j^{f-} b_k^{f-} b_n dS \\ & + \frac{C^f}{\Delta x} \left(g_i^{f+} \int_{\partial^f \mathcal{K}} b_i^{f-} b_n dS - g_i^{f-} \int_{\partial^f \mathcal{K}} b_i^{f-} b_n dS \right) \\ & + \varepsilon \left(\frac{2}{\Delta x} \right)^2 \tilde{g}_i^f \tilde{n}_j^{f,k} \tilde{n}_l^f \cdot \hat{\mathbf{N}}^f (1 - \delta_{kf}/2) \int_{\tilde{\xi}_f=0} \frac{\partial \tilde{b}_i^f}{\partial \tilde{\xi}_k} \tilde{b}_j^f \tilde{b}_l^f b_n d\tilde{S}^f \end{aligned} \quad (13)$$

The coefficients associated with an extended basis, denoted with a tilde (e.g., \tilde{g}_i^f), are calculated via Eq. (11). Just as with advection, reinitialization is time-integrated using an explicit $k + 1$ order RK scheme, similar to the approach by Owkes and Desjardins [21]. The timestep size for the convective term is given by Cockburn and Shu [3], while the restriction from the diffusive term is given by Lörcher, Gassner, and Munz [11]. The timestep size used is the smaller of the two:

$$\Delta\tau \leq \min \left(\frac{\Delta x}{2k+1}, \frac{\beta(k)\Delta x^2}{\varepsilon(2k+1)^2\sqrt{d}} \right) \quad (14)$$

Table 2Stable values for β presented by Lörcher, Gassner, and Munz [11].

k	1	2	3	4	5	6	7
β	1.46	0.80	0.40	0.24	0.16	0.12	0.09

Here, $\beta(k)$ is a function of polynomial order. Several values are shown in Table 2.

Note that, in practice, $\varepsilon \sim \Delta x$ so that $\Delta \tau$ scales with Δx rather than its square. Still, reinitialization is considerably more expensive than advection, and is therefore executed as seldomly as possible. We use the reinitialization factor F introduced by Owkes and Desjardins [21], which relates the amount of reinitialization performed to the advection time step size:

$$\tau_f - \tau_i = F \Delta t \max(|\mathbf{u} \cdot \hat{\mathbf{n}}|) \quad (15)$$

The value of F is selected by the user to decide the amount of reinitialization, when invoked, as a factor of the amount of advection in a given timestep. We select values ranging from $F = 0$ for no reinitialization to $F = 4$ for four times as much reinitialization pseudo-time as the previous advection time step. We also introduce a reinitialization occurrence factor T_r , such that reinitialization is performed only between real time intervals of length T_r . This allows us to invoke reinitialization only seldomly, since we find in practice that it often takes on the order of hundreds of advection steps before the level set profile is sufficiently dissipated to require reinitialization. The affect of T_r and F on solution quality is examined in Section 3.

Note that the integrals in Eqs. (8) and (13) are entirely in terms of the Legendre polynomials, and can therefore be precomputed symbolically and tabulated, making the method quadrature-free.

2.3.3. Slope limiting

Our numerical experiments have shown that reinitialization becomes unstable as a result of the nonlinear compressive term when employing polynomials of degree $k \geq 4$, particularly near regions with intersecting characteristics. That is, near discontinuous direction changes in the normal vector, high order representations of the normal vector exhibit oscillations as a form of Gibbs phenomenon. The oscillations in the normal vector combined with the compressive reinitialization term introduces oscillations and spikes in the G field. To mitigate this effect, a slope limiter is used. As an alternative, it may be possible to locally truncate the polynomial order of the normal vector only, without limiting the G field.

We use the slope limiter introduced by van Leer [29,30] and modified by Cockburn and Shu [3], where high order ($k > 1$) contributions in a cell are discarded if the average slope in that cell exceeds a multiple ν of the slope found through differences between the neighbor averages. In that case, the linear component is then corrected to produce the minimum of the given slopes. That is, in 1D,

$$G_h|_{I_j} = g_0 b_0 + \left(\frac{\Delta x}{2} m \left(\overline{G_{h,x}}|_{I_j}, \nu \frac{\overline{G_{j+1}} - \overline{G_j}}{\Delta x}, \nu \frac{\overline{G_j} - \overline{G_{j-1}}}{\Delta x} \right) / c_1 \right) b_1 + H.O.T. \quad (16)$$

Where $m()$ denotes the minmod function:

$$m(a_1, a_2, a_3) = \begin{cases} s \min_{1 \leq n \leq 3} |a_n| & \text{if } s = \text{sign}(a_1) = \text{sign}(a_2) = \text{sign}(a_3) \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

The average derivative, $\overline{G_{h,x}}|_{I_j}$ is evaluated before limiting as

$$\overline{G_{h,x}}|_{I_j} = \frac{\Delta x}{2} g_i \int_{\mathcal{K}} \frac{\partial b_i}{\partial \xi} dV \quad (18)$$

The quantity c_1 in Eq. (16) is the coefficient in the normalized Legendre polynomial $b_1 = c_1 \xi$. In 2D, $c_1 = \sqrt{3}/2$ while in 3D $c_1 = \sqrt{6}/4$. If the return value of the minmod function is not equal to its first argument, the higher order coefficients are set to zero. Otherwise, they are not changed. This algorithm is repeated in each direction for a 2D or 3D simulation.

The variable ν is user-set, and represents the ratio between the internally-calculated derivative (Eq. (18)) and derivatives calculated from neighbors allowed before limiting is enforced. van Leer originally set the quantity to $\nu = 1$, while Cockburn & Shu use a less restrictive $\nu = 2$. We find that the high curvature of the hyperbolic tangent profile near the $G = 0$ iso-surface results in over-limiting in those regions unless ν is set higher. That is, with $\nu = 2$, we find that high curvature regions of the domain are needlessly limited. In fact, it can be shown that for a hyperbolic tangent profile with the typical $\varepsilon = \Delta x/2$, there will always be regions where the slope limiter is active. For smooth functions, we can use Taylor series and the cell average definition,

$$\overline{f_i} = \int_{\mathcal{K}} f \left(\mathbf{x}_i + \xi \frac{\Delta x}{2} \right) dV \quad (19)$$

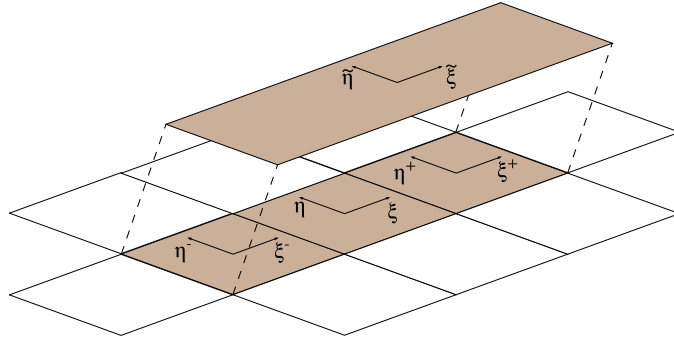


Fig. 2. Three-cell projection in x-direction.

to approximate differences as

$$\frac{\bar{G}_{j+1} - \bar{G}_j}{\Delta x} \approx \bar{G}'_i + \bar{G}''_i \frac{\Delta x}{2}, \quad \frac{\bar{G}_j - \bar{G}_{j-1}}{\Delta x} \approx \bar{G}'_i - \bar{G}''_i \frac{\Delta x}{2} \quad (20)$$

This slope limiter will limit whenever $\bar{G}'_j > \nu \min(|\frac{\bar{G}_{i+1} - \bar{G}_i}{\Delta x}|, |\frac{\bar{G}_i - \bar{G}_{i-1}}{\Delta x}|)$. For regions where G is strictly increasing, this is combined with Eq. (20) and simplified to

$$1 > \nu \left(1 \pm \frac{\Delta x}{2} \frac{\bar{G}''_i}{\bar{G}'_i} \right) \quad (21)$$

Inserting the hyperbolic tangent profile Eq. (1), we find that the limiter will be active whenever

$$\left| \tanh\left(\frac{\phi}{2\varepsilon}\right) \right| > \frac{\nu - 1}{\nu} \quad (22)$$

Therefore, the limiter will apply throughout the region

$$\phi > 2\varepsilon \operatorname{atanh}\left(\frac{\nu - 1}{\nu}\right) \quad (23)$$

The special case where $\nu = 1$, Eq. (23) implies that the limiter will be active everywhere, regardless of profile thickness. When $\nu = 2$ and we use the typical $\varepsilon = \Delta x/2$, the limiter will be active everywhere outside a band of width $\approx 1.1\Delta x$ surrounding the $G = 0.5$ iso-surface. In fact, Eq. (23) implies that for a hyperbolic tangent profile, any choice of ν will force the solution at some distance from the interface to always be limited.

One common way to prevent the slope limiter from limiting in regions it ought not is to use the corrected minmod function, suggested by Shu [27] to prevent limiting near maxima and minima. However, since reinitialization is particularly unstable anywhere there is a discontinuity in the normal vector, local minima and maxima in G are precisely where the limiter is needed.

A second way to prevent the solution from being always limited near the interface is to choose a larger value for ν . We find that for $\varepsilon = \Delta x/2$, $\nu = 2.5$ is large enough to retain high order accuracy near the interface while not so large as to miss the issues near extrema. It should be noted that the limiter is still activated in the vicinity of the interface if there are nearby local minima or maxima. This occurs in particular where the $G = 0.5$ iso-surface has high curvature, for instance in droplets only a few cells wide.

2.4. Normal vector calculation

2.4.1. Normalized local gradient method

The normal vectors that are necessary for reinitializing the level set scalar are calculated by differentiating the local G field, as described in Eq. (4). Although possible to differentiate the field with local DG coefficients, doing so results in a loss of polynomial order. To avoid this, we include information from local neighbors. This approach also has the effect of smoothing out discontinuities between cells.

Each derivative is calculated by projecting the solutions from a 3-cell stencil into a shared basis, similar to the way diffusive flux is handled. The orthonormality condition is then used to find the G coefficients on the extended basis, where the extended domain $\tilde{\mathcal{K}}$ is partitioned into three sub-domains $\tilde{\mathcal{K}}^{\alpha-}$, $\tilde{\mathcal{K}}^{\alpha}$, and $\tilde{\mathcal{K}}^{\alpha+}$, corresponding to the three original domains in the new extended space (see Fig. 2). $g_i^{\alpha\pm}$ refers to coefficients in the cell sharing the $\alpha \pm$ face.

$$\tilde{g}_n^\alpha = g_i^{\alpha-} \int_{\mathcal{K}^{\alpha-}} b_i^{\alpha-} \tilde{b}_n \, d\tilde{V} + g_i \int_{\mathcal{K}^\alpha} b_i \tilde{b}_n \, d\tilde{V} + g_i^{\alpha+} \int_{\mathcal{K}^{\alpha+}} b_i^{\alpha+} \tilde{b}_n \, d\tilde{V} \quad (24)$$

The DG coefficients γ_n^α for $\partial G / \partial x_\alpha$ are then calculated by

$$\gamma_n^\alpha = \tilde{g}_i^\alpha \frac{2}{\Delta x_\alpha} \int_{\mathcal{K}} \frac{\partial \tilde{b}_i}{\partial \xi_\alpha} b_n \, dV \quad (25)$$

Finally, the normal vector is calculated from the gradient of G .

$$n_n^k = \int_{\mathcal{K}} \frac{\gamma_i^k b_i}{\sqrt{\sum_{j=1}^3 (\gamma_k^j b_k)^2}} b_n \, dV \quad (26)$$

The nonlinear expression in the denominator necessitates the use of quadrature rules for Eq. (26). For this, we use Gauss–Legendre quadrature points and weights, with the i -th point and weight for the n -point rule given by

$$P_n(x_i) = 0$$

$$w_i = \frac{2}{(1 - x_i^2)[P'_n(x_i)]^2}$$

Since the integrand in Eq. (26) cannot be represented by a finite-degree polynomial, no finite set of quadrature points will exactly evaluate the integral. For this paper, we find a tensor-product of 5 Gauss–Legendre quadrature points and weights to be sufficiently accurate. This may be arbitrarily changed to suit higher orders than explored here.

2.4.2. Accurate method

One way to calculate normals using the geometry of the $G = 0.5$ isosurface alone is to calculate the normalized gradient of a signed distance function constructed from that isosurface. Desjardins, Moureau, and Pitsch [4] and Owkes and Desjardins [21] estimate the signed distance function using a fast marching procedure, however this is limited to second order convergence rates.

For a high-order estimate of the signed distance function, we use a form of the method by Saye [26], also used by Greene, Nourgaliev, and Schofield [7] to reinitialize a signed distance level set. In [26], Saye describes a method for calculating the distance from an arbitrary point in the domain x_q to the nearest point on the 0-contour of an arbitrary function given at discrete points. For the procedure, a polynomial representation of the function is generated from nearby points. Instead, we use the polynomial basis function already provided by our DG formulation. Given a suitable initial guess, Saye's method invokes Newton's iterative method to find the point at minimal distance to x_q , constrained to lie on the $G = 0.5$ isosurface via a Lagrange multiplier. This method is used to provide a high-order estimate for the signed distance at Gauss–Legendre quadrature points in a band 1 cell space wider than the band in which reinitialization is solved. Finally, Eq. (26) is used to calculate normal vectors from this signed distance function.

3. Results

Here, we present verification of the advection equation, varying polynomial order for both the level set scalar and the velocity. The test shows the overall convergence rate produced by the RKDG method is limited by the variable projected into the smallest polynomial space, motivating the projection of all variables to the same polynomial order. Following this analysis, we present verification tests on the reinitialization equation as well as examine its effect on the level set interface.

3.1. Method of manufactured solutions

The method of manufactured solutions (MMS) was originally developed by Salari and Knupp [25] at Sandia National Laboratory. An excellent overview of the method is given by Roache [24]. The motivation for MMS lies in the difficulty of obtaining analytical solutions to the governing equations in many physical contexts. MMS proposes that instead of attempting to find a solution to the governing PDE, one modifies the PDE to match a solution of ones choosing. That is, the exact solution G_{ex} , along with vector fields \mathbf{u} and $\hat{\mathbf{n}}$ (which here lose their physical meaning) are arbitrarily prescribed. Then, the governing PDE is modified with a source term $Q(\mathbf{x}, t)$, which is computed exactly from the chosen solution variable fields. The source function is then projected into the discontinuous basis. Here, we choose a time-independent solution, initialize the computational domain with $G = 0$, and solve the equations of interest to steady state. We enforce Dirichlet boundary conditions set to chosen exact solution in ghost cells. The converged result is then compared to the chosen time-independent solution. A similar approach is used by Salari and Knupp [25] for the incompressible Navier–Stokes equations, though with the initial condition set within 1% of the exact solution.

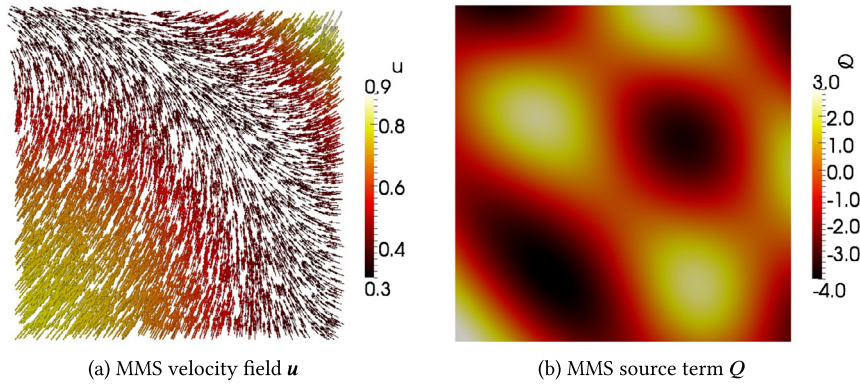


Fig. 3. MMS advection test.

$$F\left(\frac{\partial G}{\partial t}, G, \frac{\partial G}{\partial x}, \dots\right) = 0 \rightarrow F\left(\frac{\partial G}{\partial t}, G, \frac{\partial G}{\partial x}, \dots\right) = Q(\mathbf{x}) \quad (27)$$

$$Q(\mathbf{x}) = F\left(0, G_{\text{ex}}, \frac{\partial G_{\text{ex}}}{\partial x}, \dots\right)$$

It is important to note that MMS tests the numerical method's ability to correctly approximate the various terms of the PDE, but since the PDE itself is modified the solutions do not reflect the physics modeled by the original equation. Attempts to physically interpret MMS results are therefore futile. In our case, MMS does not evaluate the CLS method (since the hyperbolic tangent profile plays no role), but rather the RKDG discretization of the CLS advection and reinitialization equations.

To test the advection equation, Eq. (2), with MMS, it is modified with a source term.

$$\frac{\partial G}{\partial t} + \nabla \cdot (G\mathbf{u}) = Q(\mathbf{x}). \quad (28)$$

The source term is evaluated from the exact solution and prescribed velocity field:

$$Q(\mathbf{x}) = \nabla \cdot (G_{\text{ex}}(\mathbf{x})\mathbf{u}_{\text{ex}}(\mathbf{x})) \quad (29)$$

The following exact solution, prescribed velocity, and resulting source term are defined on a unit-sized 2D domain $[0, 1]^2$.

$$G_{\text{ex}}(x, y) = \frac{1}{2} + \sin(2\pi x) \cos(2\pi y)$$

$$\mathbf{u}_{\text{ex}}(x, y) = \left(\frac{1}{2} - \sin(x^2 + y^2)\right)\hat{\mathbf{x}} + \left(\cos(x^2 + y^2) - \frac{2}{5}\right)\hat{\mathbf{y}}$$

$$\Rightarrow Q(x, y) = -2 \cos(x^2 + y^2)x(1/2 + \sin(2\pi x) \cos(2\pi y))$$

$$+ 2(0.5 - \sin(x^2 + y^2)) \cos(2\pi x) \pi \cos(2\pi y)$$

$$- 2 \sin(x^2 + y^2)y(1/2 + \sin(2\pi x) \cos(2\pi y))$$

$$- 2(\cos(x^2 + y^2) - 0.4) \sin(2\pi x) \sin(2\pi y) \pi \quad (30)$$

The source term is projected into the DG basis using

$$q_i^K = \int_{\mathcal{K}} Q b_i \, dV. \quad (31)$$

These integrals are evaluated using the Gauss–Legendre quadrature rules described earlier. The prescribed velocity is projected into the DG basis similarly.

The solution as it evolves through time via RKDG with $k = 4$ polynomials is shown in Fig. 4 along with the error at the final converged state at $t = 4.3$. Comparing the final error to the velocity field in Fig. 3, it is found that the error predominantly lies in compressive regions where velocity vectors converge.

The errors produced for various maximum polynomial degrees are listed in Table 3. The results show a $k + 1$ order convergence rate in the L_1 norm with only a k th order convergence rate in the L_∞ norm, when both G and \mathbf{u} have degree k . The reduced convergence for L_∞ seems to be attributed to the convergent velocity field, as the error plot shows in Fig. 4, perhaps resulting from the upwind flux along the change in velocity direction.

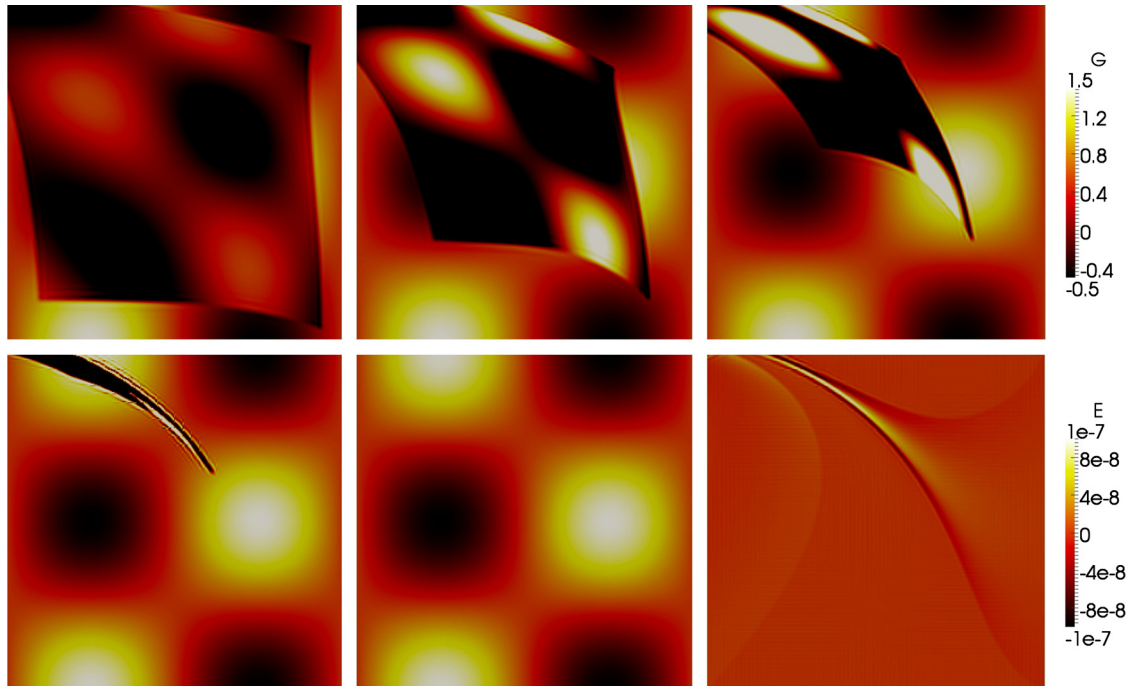


Fig. 4. Solution G of MMS test case for $\Delta x = 1/40$, RKDG-4 for $t = 0.2, 0.5, 1.0, 2.0, 4.3$ time units, and error E at steady state (from top left to bottom right).

Table 3

Error norms of advection MMS test case and their order of convergence under grid refinement for various polynomial orders.

k_g	k_u	Δx	L_∞	Order	L_1	Order	k_g	k_u	Δx	L_∞	Order	L_1	Order
1	0	1/10	7.96e-1	–	6.71e-2	–	3	0	1/10	9.28e-1	–	8.01e-2	–
1	0	1/20	3.00e-1	1.4	3.00e-2	1.1	3	0	1/20	1.20	–0.4	4.30e-2	0.9
1	0	1/40	4.03e-1	–0.4	1.78e-2	0.4	3	0	1/40	6.21e-1	1.0	2.29e-2	0.9
1	0	1/80	2.58e-1	0.6	9.26e-3	0.7	3	0	1/80	4.59e-1	0.4	1.13e-2	1.0
1	1	1/10	1.08e-1	–	2.38e-2	–	3	1	1/10	5.04e-2	–	5.34e-3	–
1	1	1/20	5.72e-2	0.9	7.32e-3	1.7	3	1	1/20	2.40e-2	1.1	1.31e-3	2.0
1	1	1/40	2.18e-2	1.4	2.09e-3	1.8	3	1	1/40	1.05e-2	1.2	3.28e-4	2.0
1	1	1/80	9.96e-3	1.1	5.70e-4	1.9	3	1	1/80	4.39e-3	1.3	8.31e-5	2.0
1	3	1/10	1.38e-1	–	2.51e-2	–	3	3	1/10	4.49e-4	–	7.76e-5	–
1	3	1/20	6.83e-2	1.0	7.44e-3	1.8	3	3	1/20	4.88e-5	3.2	5.69e-6	3.8
1	3	1/40	2.89e-2	1.2	2.06e-3	1.9	3	3	1/40	6.35e-6	2.9	3.79e-7	3.9
1	3	1/80	1.10e-2	1.4	5.45e-4	1.9	3	3	1/80	7.87e-7	3.0	2.47e-8	3.9

Table 3 also shows the resulting errors and convergence rates when G and \mathbf{u} are given different polynomial degrees. It is found that the convergence rate of the solution is limited by the smallest number of degrees of freedom given to a particular variable. That is, the RKDG-CLS method only converges at a rate of $\min(k_g, k_u) + 1$. This particular finding motivates the need for treating both velocity and normal vectors as k -order DG variables.

Reinitialization is also tested with MMS. One important detail with grid convergence studies is that the scalar ε must be held constant, since it appears in the reinitialization equation itself. In practice, it represents the thickness of the hyperbolic tangent profile and its value is changed along with the grid. In practice, ε represents the thickness of the level set scalar and is typically related to the grid size and set to $\Delta x/2$. However, in MMS the G field is entirely disassociated from any notion of an interface and ε only exists as a coefficient to the diffusive term in the reinitialization equation. Since changing ε also changes the governing equation, and therefore the mathematical problem being solved, it must be held constant between grid levels to evaluate the formal order of the RKDG method. In these tests, a value of $\varepsilon = 0.2$ is used since it approximately balances the magnitudes of the convective and diffusive terms. Unfortunately, a constant ε also results in timestep sizes proportional to Δx^2 , meaning that these tests can become expensive quickly. Therefore, we only present $k = 1$ results here and present tests for higher k later.

The reinitialization equation is modified with a source term:

$$\frac{\partial G}{\partial \tau} + \nabla \cdot (G(1 - G)\hat{\mathbf{n}}) = \nabla \cdot (\varepsilon(\nabla G \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}) + Q(\mathbf{x}) \quad (32)$$

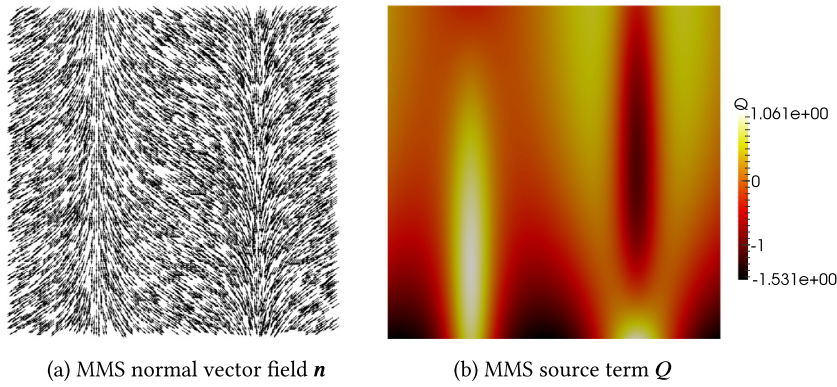


Fig. 5. MMS reinitialization test.

Table 4

2D RKDG-1 reinitialization MMS error norms and their order of convergence under grid refinement.

Δx	L_∞	Order	L_1	Order
1/10	2.54e-2	–	4.73e-3	–
1/20	1.30e-2	1.0	2.24e-3	1.1
1/40	4.88e-3	1.4	6.87e-4	1.7
1/80	1.55e-3	1.7	1.94e-4	1.8

Table 5

Unenriched 2D RKDG-1 reinitialization MMS error norms and their order of convergence under grid refinement.

Δx	L_∞	Order	L_1	Order
1/10	1.27e-1	–	2.42e-2	–
1/20	4.17e-2	1.6	8.80e-3	1.5
1/40	1.71e-2	1.3	3.03e-3	1.5
1/80	5.82e-3	1.6	9.50e-4	1.7

$$Q(\mathbf{x}) = \nabla \cdot (G_{\text{ex}}(1 - G_{\text{ex}})\hat{\mathbf{n}}_{\text{ex}}) - \nabla \cdot (\varepsilon(\nabla G_{\text{ex}} \cdot \hat{\mathbf{n}}_{\text{ex}})\hat{\mathbf{n}}_{\text{ex}}) \quad (33)$$

For evaluating the source term, the normal is calculated from the exact solution chosen for G , using Eq. (4). The simulation calculates the normal using the approach described in Section 2.4.1. In this case, the exact solution to G is set as

$$G_{\text{ex}}(x, y) = \frac{\exp(\frac{1}{4} \sin(2\pi x) - y) - e^{-3/4}}{e^{3/4} - e^{-3/4}}. \quad (34)$$

This function is chosen for its nonlinear behavior, and normal that is defined everywhere in the domain $[-0.5, 0.5]^2$. The resulting source term is too long to write here, but can be easily obtained using computer algebra software. It is shown as a color plot in Fig. 5. Note also that fully enriched diffusive fluxes are used here, and in later tests unless otherwise noted.

The results for this test are shown in Table 4. Convergence rates start low, but approach $k + 1$ in both norms as the grid is refined. The results for unenriched diffusive flux are shown in Table 5. The errors are significantly higher than those for the enriched case, but the convergence rates are similar. The results show error magnitudes of the unenriched case to be approximately those of the enriched case at one higher level of grid refinement.

Reinitialization's 3D capabilities are evaluated with an exact solution chosen as

$$G_{\text{ex}}(x, y, z) = \frac{\exp(\frac{1}{4} \sin(\sqrt{2}\pi(x + z)) - y) - e^{-3/4}}{e^{3/4} - e^{-3/4}}. \quad (35)$$

The results, shown in Table 6, indicate approximately second order convergence rates for $k = 1$ polynomials, following a trend similar to that found in 2D.

3.2. Circle test

To assess reinitialization and the normal calculation algorithm using realistic geometries, a test case was developed which involves a circle of radius R_0 and profile thickness ε_0 placed at the origin of a unit-sized $[-0.5, 0.5]^2$ domain. The normal is calculated using the gradient-based approach of Section 2.4.1. The level set scalar is initialized to

Table 6
3D RKDG-1 reinitialization MMS error norms and their order of convergence under grid refinement.

Δx	L_∞	Order	L_1	Order
1/10	2.24e-2	–	4.52e-3	–
1/20	1.39e-2	0.7	1.51e-3	1.6
1/40	5.02e-3	1.5	4.81e-4	1.7

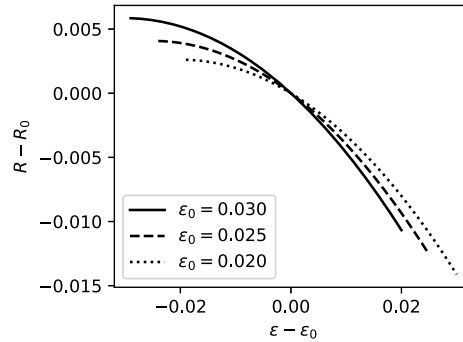


Fig. 6. Change in radius against change in profile thickness for $R_0 = 0.25$.

Table 7
Error norms of circle test and their order of convergence under grid refinement for RKDG-CLS-3. $\varepsilon_0 = 0.025$, $\varepsilon = 0.0125$.

Δx	L_∞	Order	L_1	Order
1/20	2.16e-2	–	2.01e-3	–
1/40	1.04e-3	4.4	2.84e-5	6.1
1/80	6.19e-5	4.1	1.65e-6	4.1
1/160	3.77e-6	4.0	9.84e-8	4.1

$$G(\mathbf{x}) = \frac{1}{2} \left(\tanh \left(\frac{R_0 - \sqrt{x^2 + y^2}}{2\varepsilon_0} \right) + 1 \right). \quad (36)$$

We then solve reinitialization to convergence, which sharpens the interface to thickness ε . Ghost cells are set to the exact solution as the boundary condition.

The exact solution to this problem is a circle of radius R and profile thickness ε . The error is defined as the absolute value of the difference between the exact solution and final state of the level set scalar, and evaluated at quadrature points through the entire domain.

$$G_{\text{ex}}(\mathbf{x}) = \frac{1}{2} \left(\tanh \left(\frac{R - \sqrt{x^2 + y^2}}{2\varepsilon} \right) + 1 \right) \quad (37)$$

$$E = |G_{\text{ex}}(\mathbf{x}) - G(\mathbf{x})|$$

Notice the exact solution has a different radius R than the initial R_0 , due to the fact that the reinitialization equation advects the interface. Advection of the interface performed by reinitialization is also demonstrated by McCaslin and Desjardins [16] through different computational experiments. We can calculate this new radius using the fact the G field is conserved under reinitialization and that the exact solution remains symmetric:

$$\int_0^\infty G(r; R, \varepsilon) r \, dr = \int_0^\infty G(r; R_0, \varepsilon_0) r \, dr \quad (38)$$

Numerically solving this equation for R using a modified Powell method via MINPACK, given a circle of initial radius $R_0 = 0.25$, initial thickness $\varepsilon_0 = 0.025$, and final thickness $\varepsilon = 0.0125$, we find the final radius $R = 0.253063$. Fig. 6 shows the resulting relationship between change in radius and change in profile thickness.

The results of a refinement study for $k = 3$ polynomials are shown in Table 7, which shows the L_∞ and L_1 norms of the error at steady state together with the associated convergence rates. We find each to converge at $k + 1$ order.

We repeat the test using $\varepsilon_0 = \Delta x$, $\varepsilon = \Delta x/2$ to demonstrate the effect of decreasing profile thickness as would be found in practice. The exact circle radius changes along with the profile thickness at every grid level, such that Eq. (38) must

Table 8

Error norms of circle test and their order of convergence under grid refinement for RKDG-CLS-3. $\varepsilon_0 = \Delta x$, $\varepsilon = \Delta x/2$.

Δx	L_∞	Order	L_1	Order
1/20	4.47e−2	–	1.03e−2	–
1/40	1.04e−3	5.4	2.84e−5	8.5
1/80	3.44e−4	1.6	8.67e−6	1.7
1/160	4.02e−4	−0.2	4.51e−6	0.9

Table 9

Error norms of circle normals test and their order of convergence under grid refinement for RKDG-CLS-3.

Δx	Accurate normals				Local-gradient-based normals			
	L_∞	Order	L_1	Order	L_∞	Order	L_1	Order
1/20	1.62e−3		4.21e−4		5.58e−4		1.69e−4	
1/40	9.31e−5	4.1	2.42e−5	4.1	1.94e−5	4.8	5.89e−6	4.8
1/80	1.36e−5	2.8	2.85e−6	3.1	1.08e−6	4.2	3.91e−7	3.9
1/160	7.29e−7	4.2	1.64e−7	4.1	7.09e−8	3.9	2.57e−8	3.9

be solved in each case to find the final radius, using the same process as before. The results of this circle test are shown in Table 8. We find the error to rapidly drop between the coarsest levels, followed by a leveling off as the grid is refined further. As anticipated, no clear convergence rate is found since neither the initial condition, exact solution, nor governing equation are consistent between grid levels. This indicates that arbitrary-order convergence rates can only be expected when the underlying problem is not changed under grid refinement. Furthermore, this demonstrates that the use of a high-order numerical method may not produce higher convergence rates in practical simulations when ε is set proportional to the mesh spacing.

3.3. Circle normals test

The circle described in the previous section is also used to evaluate the performance of the normal algorithms described in Section 2.4. This is done by performing each algorithm on the initial G field from Eq. (36), and comparing the resulting normal field to the exactly known normals for a circle. In this case, the error norms are calculated only within a band of thickness 0.3 surrounding the interface. This is done to avoid the inevitable errors near the discontinuity in the normal vector at the center of the circle. The results are shown in Table 9.

Unsurprisingly, the local gradient-based normals are more accurate than the accurate normals since the G -field does not have erroneous variations throughout due to advection. Both the accurate and direct-local-gradient methods are found to converge at $k + 1$ order.

3.4. Zalesak's disk

Zalesak's disk [32], involves the solid body rotation of a notched disk. A disk of radius 0.15, notch width 0.05, and notch height 0.25 is placed in a unit-sized domain at (0.5,0.75). The disk is then rotated about the origin by the velocity field

$$\mathbf{u}(x, y) = (0.5 - y)\hat{\mathbf{x}} + (x - 0.5)\hat{\mathbf{y}}$$

Normal vectors are calculated using the gradient-based approach described in Section 2.4.1. Here, and in all following test cases, we set $\varepsilon = \Delta x/2$.

This test includes banding from the RLSG. The advection equation is solved in a band of 8 cells on either side of the interface, while transport is solved in a band of 9 cells on either side. These band sizes are shared by the deforming column and sphere test cases. Cells newly added to the band are given a value of 0 or 1, depending on whether they are outside or inside the circle, and act as Dirichlet condition ghost cells.

The impact of banding on conservation of G is examined by calculating the fraction of G lost during the simulation from the initial (G_0) and final (G_f) states.

$$\text{Fraction of } G \text{ lost} = \frac{\int_{\Omega} G_f \, dV - \int_{\Omega} G_0 \, dV}{\int_{\Omega} G_0 \, dV} \quad (39)$$

The results of a study on conservation of G against band size are shown in Table 10. For a mesh filling the entire domain, the fraction of G lost is machine zero, while for our setting of 8 cells 0.0005% of G is lost. However, conservation of G is simply a tool to improve conservation of the volume contained by the $G = 0.5$ iso-surface, and in this sense the loss of G due to banding is not directly relevant for multiphase simulations.

The results of a grid refinement study show that volume errors (Table 13) tend to decrease with grid refinement, although convergence rates can vary drastically between one grid level and another. This is due to the errors present at

Table 10
Zalesak's disk G loss fractions for RKDG-CLS-2,
 $T_r = 1.57$, $F = 2.0$, $\Delta x = 1/100$, after 1 rotation.

Band size	G loss fraction
4	8.76e-4
8	4.96e-6
16	3.58e-11
100	1.22e-13

Table 11
Zalesak's disk volume and shape errors. $T_r = 0$, $F = 0$ cases compared with literature.

	Δx	RKDG-CLS-2		RKDG-CLS-3		RKDG-CLS-4		LS-WENO-5 [9]		ACLS-HOUC-5 [4]	
		E	Order	E	Order	E	Order	E	Order	E	Order
Volume	1/50	3.08e-3	–	4.43e-4	–	1.06e-5	–	4.6e-3	–	7.167e-3	–
	1/100	4.87e-4	2.7	8.29e-5	2.4	9.23e-5	–3.1	4.6e-3	–	3.52e-4	4.3
	1/200	1.43e-4	1.8	2.42e-5	1.8	4.95e-6	4.2	1.0e-3	2.2	8.5e-5	2.1
	1/400	6.65e-5	1.1	1.30e-5	0.9	8.97e-7	2.5	1.3e-4	2.9		
	1/800							7e-5	0.9		
Shape	1/50	3.73e-3	–	1.07e-3	–	4.83e-4	–	1.611e-1	–		
	1/100	1.41e-3	1.4	3.45e-4	1.6	1.66e-4	1.5	4.19e-2	1.9		
	1/200	8.26e-4	0.8	1.47e-4	1.2	5.49e-5	1.6	1.44e-2	1.5		
	1/400	3.91e-4	1.1	6.11e-5	1.3	1.78e-5	1.6	4.5e-3	1.7		
	1/800										

Table 12
Zalesak's disk shape errors and convergence rates for RKDG-CLS- k with reinit. after 1 rotation.

T_r, F	Δx	$k = 2$		$k = 3$		$k = 4$	
		E	Order	E	Order	E	Order
0.0, 0.0	1/50	3.73e-3	–	1.07e-3	–	4.83e-4	–
	1/100	1.41e-3	1.4	3.45e-4	1.6	1.66e-4	1.5
	1/200	8.26e-4	0.8	1.47e-4	1.2	5.49e-5	1.6
	1/400	3.91e-4	1.1	6.11e-5	1.3	1.78e-5	1.6
1.57, 2.0	1/50	2.48e-3	–	1.03e-2	–	7.82e-3	–
	1/100	9.27e-4	1.4	8.42e-4	3.6	1.36e-3	2.5
	1/200	5.86e-4	0.7	2.38e-4	1.8	2.53e-4	2.4
	1/400	3.04e-4	0.9	7.30e-5	1.7	5.89e-5	2.1
0.79, 4.0	1/50	4.46e-3	–	1.28e-2	–	1.29e-2	–
	1/100	9.59e-4	2.2	1.41e-3	3.2	1.64e-3	3.0
	1/200	5.06e-4	0.9	4.56e-4	1.6	3.18e-4	2.4
	1/400	2.04e-4	1.3	2.57e-4	0.8	2.36e-4	0.4

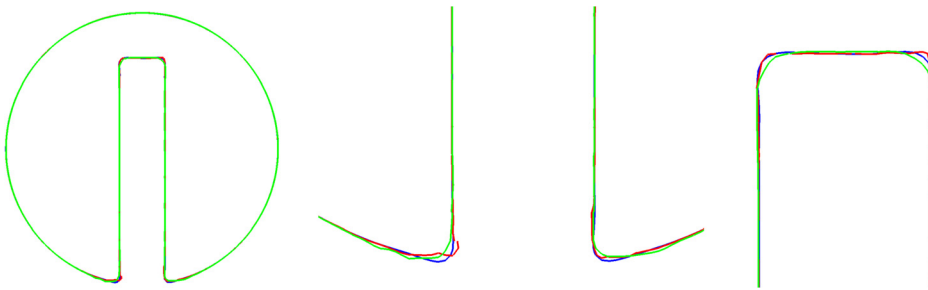


Fig. 7. Interface shape of Zalesak's disk after 1 rotation, $k = 3$, $\Delta x = 1/100$: no reinitialization (blue); $T_r = 1.57$, $F = 2.0$ (red); $T_r = 0.79$, $F = 4.0$ (green). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

the notch corners, which tend to cancel each other out. For comparison, the volume errors reported by Herrmann [9] for a banded fifth order WENO method are shown in Table 11, which are most closely comparable to the $k = 2$ RKDG case without reinitialization on meshes with 4x fewer cells. Volume errors reported by Desjardins, Moureau, and Pitsch [4] for a HOUC-5 ACLS method show (Table 11) a very large error drop between the two coarsest grids, due to the notch in Zalesak's disk closing at the coarsest level. The next grid refinement level shows a second order convergence rate. By comparison, their results most closely resemble the RKDG-CLS-3 case on a one level courser mesh with no reinitialization.

Table 13

Zalesak's disk volume errors and convergence rates for RKDG-CLS-k with reinit. after 1 rotation.

T_r, F	Δx	$k = 2$		$k = 3$		$k = 4$	
		E	Order	E	Order	E	Order
0.0, 0.0	1/50	3.08e-3	–	4.43e-4	–	1.06e-5	–
	1/100	4.87e-4	2.7	8.29e-5	2.4	9.23e-5	–3.1
	1/200	1.43e-4	1.8	2.42e-5	1.8	4.95e-6	4.2
	1/400	6.65e-5	1.1	1.30e-5	0.9	8.97e-7	2.5
1.57, 2.0	1/50	6.00e-4	–	1.87e-2	–	6.46e-3	–
	1/100	6.79e-5	3.1	8.16e-4	4.5	2.35e-4	4.8
	1/200	3.49e-5	1.0	2.13e-5	5.3	4.67e-5	2.3
	1/400	2.62e-5	0.4	1.19e-6	4.2	5.74e-6	3.0
0.79, 4.0	1/50	4.22e-4	–	3.13e-3	–	1.16e-2	–
	1/100	4.24e-4	0.9	3.10e-4	3.3	1.72e-4	6.1
	1/200	1.52e-4	0.6	4.54e-5	2.8	1.52e-5	3.5
	1/400	9.12e-5	0.7	2.71e-5	0.8	2.32e-5	–0.6

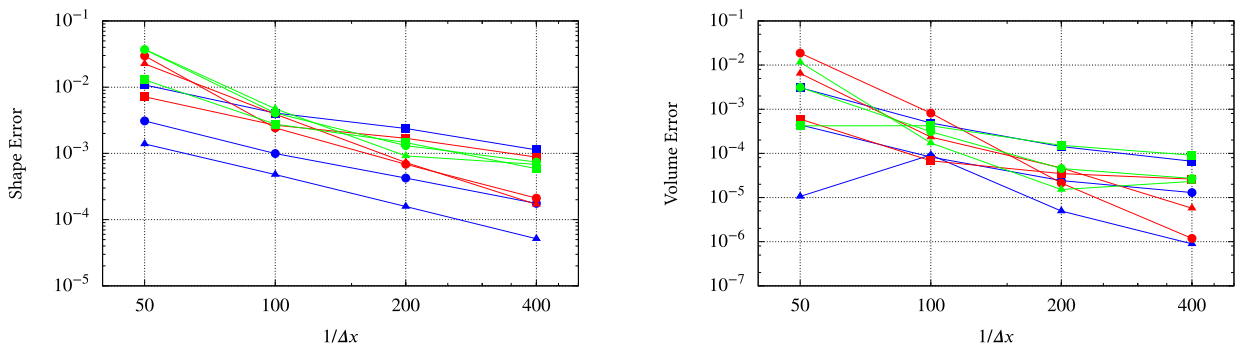
**Fig. 8.** Shape and volume errors for Zalesak's disk. No reinitialization (blue), $T_r = 1.57, F = 2.0$ (red), $T_r = 0.79, F = 4.0$ (green). $k = 2$ (squares), $k = 3$ (circles), $k = 4$ (triangles). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Table 12 summarizes the shape error, defined as

$$E = \frac{\int_A |H(G - \frac{1}{2}) - H(G_{\text{ex}} - \frac{1}{2})| dA}{2L \int_A H(G_{\text{ex}} - \frac{1}{2}) dA} \quad (40)$$

to mimic the definition found in [9]. Here, G_{ex} denotes the exact solution and L denotes the exact shape perimeter (or surface area in 3D). H is the Heaviside function, used to indicate that we are calculating the difference in areas (volumes in 3D) enclosed by the $G = 0.5$ and $G_{\text{ex}} = 0.5$ isosurfaces. It is evaluated employing a recursive cell refinement algorithm using marching triangles to calculate the phase interface position for different RKDG-CLS-k. Overall shape errors are small, however, the convergence rate in this metric appears to approach first order, independent of the order of the employed RKDG basis functions. This appears to be due to the fact that shape errors for the RKDG-CLS methods are confined to the sharp corner regions that represent a discontinuity in the solution gradients and are thus captured with the employed Legendre basis functions at best with first order. Shape errors reported by Herrmann [9] are shown in Table 11. All RKDG shape errors show significant improvement for this test case by comparison, even on many times coarser meshes.

Without reinitialization, it should be noted though that even if the shape error convergence rates appear to be first order for all analyzed k , increasing k with a fixed Δx reduces errors significantly. Reinitialization's affect on Zalesak's disk is analyzed with three amounts of reinitialization shown in Fig. 7, Fig. 8, Table 12, and Table 13. The $T_r = F = 0$ results are calculated with no reinitialization, while $T_r = 1.57, F = 2$ performs reinitialization an amount described by Eq. (15) four times per revolution. The final case doubles both the amount of reinitialization per call and the frequency of reinitialization calls.

In general, it is found that reinitialization increases shape errors. This occurs because reinitialization smooths out the corners of the notched disk, as Fig. 7 shows. It is also found that some reinitialization improves volume conservation, but increasing the amount of reinitialization further increases the volume error. Reinitialization is expected to improve volume conservation, but unphysically varying normal vectors away from the interface cause some G to be gathered outside the disk, slightly reducing the overall volume inside the 0.5 iso-surface. Normal vectors vary because they are dependent on local changes in the G field, rather than on the interface geometry alone. As a result, more frequent reinitialization will increase volume errors. This trend isn't shared by the $k = 4$ case, since it employs limiters during reinitialization that degrade the accuracy in the sharp corners. By comparison, Desjardins, Moureau, and Pitsch [4] find the frequency of reinitialization to have little affect on Zalesak's disk when the ACLS method is used.

Table 14Deforming column volume and shape errors at $t = T$, $T_r = 0$, $F = 0$ cases compared with literature.

	Δx	RKDG-CLS-2		RKDG-CLS-3		RKDG-CLS-4		LS-WENO-5 [9]		RKDG-ACLS-2* [21]	
		E	Order	E	Order	E	Order	E	Order	E	Order
Volume	1/64	6.00e−3	–	8.25e−3	–	5.36e−3	–	–	–	8.9e−3	–
	1/128	4.81e−3	0.3	2.58e−3	1.7	1.65e−3	1.7	3.1e−1	–	4.1e−3	1.1
	1/256	1.43e−3	1.8	7.91e−4	1.7	4.71e−4	1.8	4.6e−2	2.7	4.6e−3	−0.2
	1/512	–	–	–	–	–	–	1.0e−2	2.2	–	–
	1/1024	–	–	–	–	–	–	2.8e−3	1.9	–	–
Shape	1/64	2.69e−2	–	7.81e−3	–	3.43e−3	–	–	–	–	–
	1/128	5.56e−3	2.3	1.47e−3	2.4	7.06e−4	2.3	2.315e−2	–	–	–
	1/256	9.98e−4	2.5	3.30e−4	2.2	1.35e−4	2.4	3.63e−3	2.7	–	–
	1/512	–	–	–	–	–	–	8.4e−4	2.1	–	–
	1/1024	–	–	–	–	–	–	2.4e−4	1.8	–	–

* RKDG-ACLS Owkes and Desjardins [21] does not expand velocity or normals into the DG basis.

3.5. Deforming column

The column or circle in a deformation field problem, also called the vortex test, introduced by Bell, Colella, and Glaz [2] and applied as a level set test problem by Enright et al. [5] tests the ability of the level set method to resolve and maintain ever thinner filaments. A column of radius $R_0 = 0.15$ and center $(0.5, 0.75)^T$ is placed inside a unit sized box. The velocity field is given by the stream function

$$\Psi(\mathbf{x}, t) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos(\pi t/T) \quad (41)$$

with $T = 8$ and first stretches the column into ever thinner filaments that are wrapped around the center of the box, then slowly reverses, and pulls the filaments back into the initial circular shape.

The impact of reinitialization is examined by changing the frequency with which reinitialization is performed, as well as the number of reinitialization iterations performed with each call. A comparison of several schemes is shown in Fig. 9, Table 15, Table 16, and Fig. 11. In this case, reinitialization increases both volume and shape errors, with a jump in error for $k = 4$ tests with reinitialization since the slope limiter is active in these cases. When reinitialization is active, the error diminishes with grid refinement, but increasing the polynomial degree has little affect in most cases. This is likely because small discrepancies in the high order polynomials introduce greater opportunity for error in the normal vectors, and in turn the level set field. The $k = 4$ tests do not necessarily follow this trend since the active slope limiter often truncates cells to a piecewise linear solution. As mentioned in Section 2.3.3, it may be possible to remedy this effect by filtering the normal vector to smooth out variations, rather than limiting the G field. On the other hand, when reinitialization is not active, increasing the polynomial degree consistently reduces the error.

In contrast to Zalesak's disk, the column is deformed, which causes more G to be left in a trail behind the column. Similarly, local variations in the level set scalar cause reinitialization to gather G away from the interface. This often results in what we call streaking (see Fig. 10), and with the added potential for lost G and streaks that deform the interface, both volume errors and shape errors are increased. This result further demonstrates the need for accurately calculated normals.

For comparison, Owkes and Desjardins [21] perform this test with their RKDG-ACLS method. Notably, they show significant improvement in volume errors when reinitialization is performed compared to when it is not, due to their accurate normal calculation. They find increasing reinitialization further also continues to improve volume errors, but only moderately. By comparison, we find reinitialization with normals calculated from the local level set scalar often worsens the results that are otherwise found with no reinitialization at all. However, the volume errors we find with no reinitialization for the same mesh size and polynomial degree are comparable to the errors they find with reinitialization (reprinted in Table 14), since we project velocity and normal vectors to full k order in the DG basis for a more accurate method.

Herrmann [9] reports shape and volume errors for a banded fifth order WENO method, also reprinted in Table 14. The performance of the RKDG method is far superior, as the $k = 2$ case with no reinitialization on a 256^2 mesh produces a smaller volume error than any of the WENO tests performed. His shape error magnitudes most closely resemble either RKDG-CLS-2 with no reinitialization on meshes with 4x fewer cells, or RKDG-CLS-3 with no reinitialization on meshes with 16x fewer cells, albeit at the cost of increasing the degrees of freedom per cell.

3.5.1. Accurate normals

Repeating this test with accurate normals calculated via the method described in Section 2.4.2 reduces the errors produced by reinitialization. Table 17 shows errors and convergence rates for $k = 3$, $T_r = 0.1$, $F = 0.25$ case. The table shows slightly smaller error magnitudes for the accurate normals case. The final interface shapes after full flow reversal are shown in Fig. 12. The local gradient based approach, produces a distorted the interface as local variations in the normal vectors cause reinitialization to collect G into streaks as the column is deformed. These are almost entirely absent in the case with accurate normals. Accurate normals, then, improve the quality of the RKDG-CLS scheme far more than the modest reduction

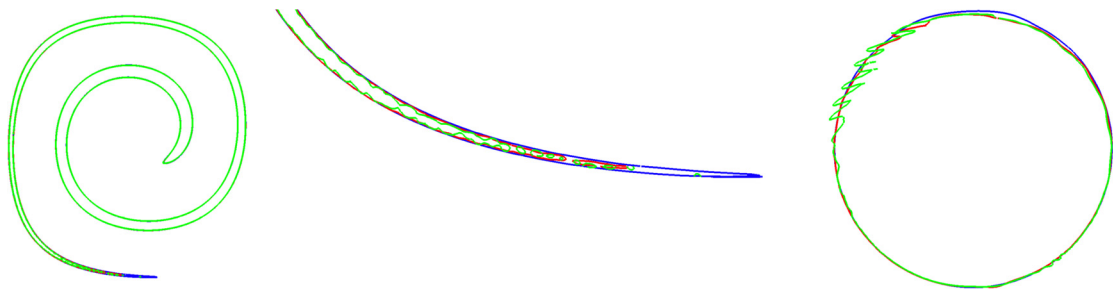


Fig. 9. Interface shape of column in a deformation field for $k = 3$, $\Delta x = 1/128$ at $t = T/2$ (left, middle zoomed) and $t = T$ (right); no reinitialization (blue); $T_r = 0.5$, $F = 2.0$ (red); $T_r = 0.25$, $F = 4.0$ (green). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

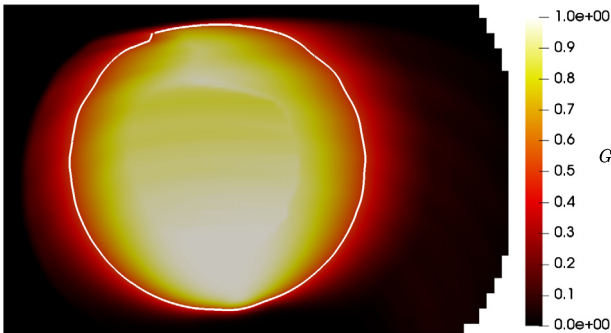


Fig. 10. Streaking exhibited after full flow reversal for $T_r = 0.5$, $F = 2.0$, $k = 2$, $\Delta x = 1/128$. $G = 0.5$ iso-surface indicated with a white line. Note the staircase edge on the right is due to the narrow band structure of the RLSC, i.e. there is no data beyond it to display.

Table 15
Deforming column shape errors and convergence rates for RKDG-CLS-k with reinit. after full flow reversal at $t = T$.

T_r, F	Δx	$k = 2$		$k = 3$		$k = 4$	
		E	Order	E	Order	E	Order
0.0,0.0	1/64	2.69e−2	–	7.81e−3	–	3.43e−3	–
	1/128	5.56e−3	2.3	1.47e−3	2.4	7.06e−4	2.3
	1/256	9.98e−4	2.5	3.30e−4	2.2	1.35e−4	2.4
0.5,2.0	1/64	4.81e−2	–	5.18e−2	–	1.26e−1	–
	1/128	1.12e−2	2.1	1.01e−2	2.4	3.00e−2	2.1
	1/256	2.45e−3	2.2	2.44e−3	2.1	5.31e−3	2.5
0.25,4.0	1/64	4.44e−2	–	6.03e−2	–	2.40e−1	–
	1/128	1.38e−2	1.7	1.50e−2	2.0	7.02e−2	1.8
	1/256	2.34e−3	2.6	2.92e−3	2.4	1.21e−2	2.5

Table 16
Deforming column volume errors and convergence rates for RKDG-CLS-k with reinit. after full flow reversal at $t = T$.

T_r, F	Δx	$k = 2$		$k = 3$		$k = 4$	
		E	Order	E	Order	E	Order
0.0,0.0	1/64	6.00e−3	–	8.25e−3	–	5.36e−3	–
	1/128	4.81e−3	0.3	2.58e−3	1.7	1.65e−3	1.7
	1/256	1.43e−3	1.8	7.91e−4	1.7	4.71e−4	1.8
0.5,2.0	1/64	8.60e−2	–	9.23e−2	–	1.06e−1	–
	1/128	1.82e−2	2.2	1.51e−2	2.6	2.75e−3	5.3
	1/256	3.46e−3	2.4	3.29e−3	2.2	2.08e−3	0.4
0.25,4.0	1/64	6.40e−2	–	8.57e−2	–	2.24e−1	–
	1/128	1.54e−2	2.1	1.67e−2	2.4	1.84e−2	3.6
	1/256	2.51e−3	2.6	3.47e−3	2.3	1.48e−3	3.6

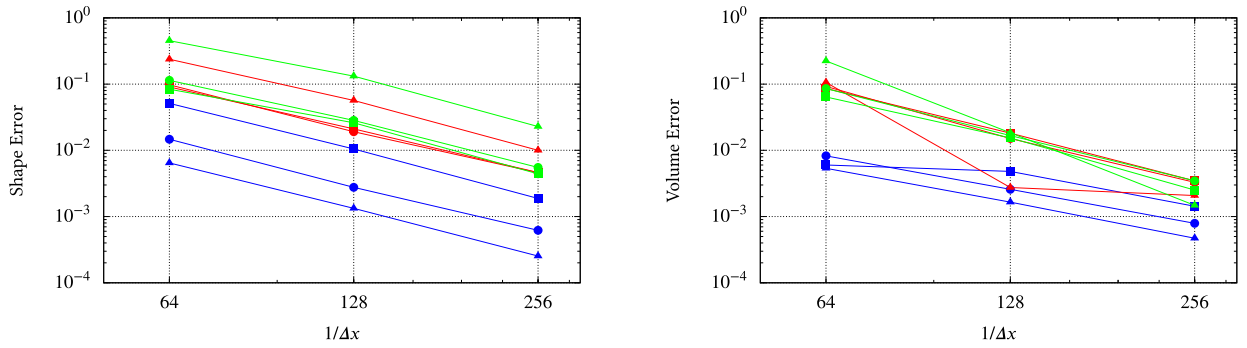


Fig. 11. Shape and volume errors for deforming column. No reinitialization (blue), $T_r = 0.5$, $F = 2.0$ (red), $T_r = 0.25$, $F = 4.0$ (green). $k = 2$ (squares), $k = 3$ (circles), $k = 4$ (triangles). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Table 17

Error comparison for different normal calculation methods on the deforming column test.

Δx	Accurate normals				Local-gradient-based normals			
	Volume error	Order	Shape error	Order	Volume error	Order	Shape error	order
1/64	6.00e-2		6.71e-2		7.41e-2		8.05e-2	
1/128	1.16e-2	2.4	1.45e-2	2.2	1.48e-2	2.3	1.80e-2	2.2
1/256	2.38e-3	2.3	2.93e-3	2.3	2.86e-3	2.4	3.67e-3	2.3

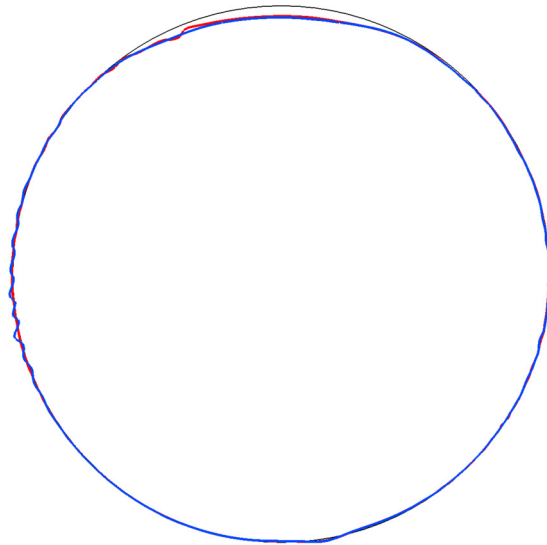


Fig. 12. Deforming column interface shape after full flow reversal for $k = 3$, $T_r = 0.1$, $F = 0.25$. Accurate normals case shown in red; local gradient based normals shown in blue. The thin black line corresponds to the exact solution. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

in volume and shape errors would seem to indicate. Moreover, more egregious errors introduced by local-gradient-based normals, such as the creation of false droplets, occur over much longer timescales than simulated here.

3.6. Deforming sphere

To demonstrate the performance of the RKDG-CLS method in three dimensions, the sphere in a deformation field case proposed by Enright et al. [5] is performed. A sphere of radius $R_0 = 0.15$ is placed at $(0.35, 0.35, 0.35)^T$ inside a unit box, whose time dependent velocity field is given by

$$\begin{aligned}
 u &= 2 \sin^2(\pi x) \sin(2\pi y) \sin(2\pi z) \cos(\pi t/T) \\
 v &= -\sin(2\pi x) \sin^2(\pi y) \sin(2\pi z) \cos(\pi t/T) \\
 w &= -\sin(2\pi x) \sin(2\pi y) \sin^2(\pi z) \cos(\pi t/T),
 \end{aligned} \tag{42}$$

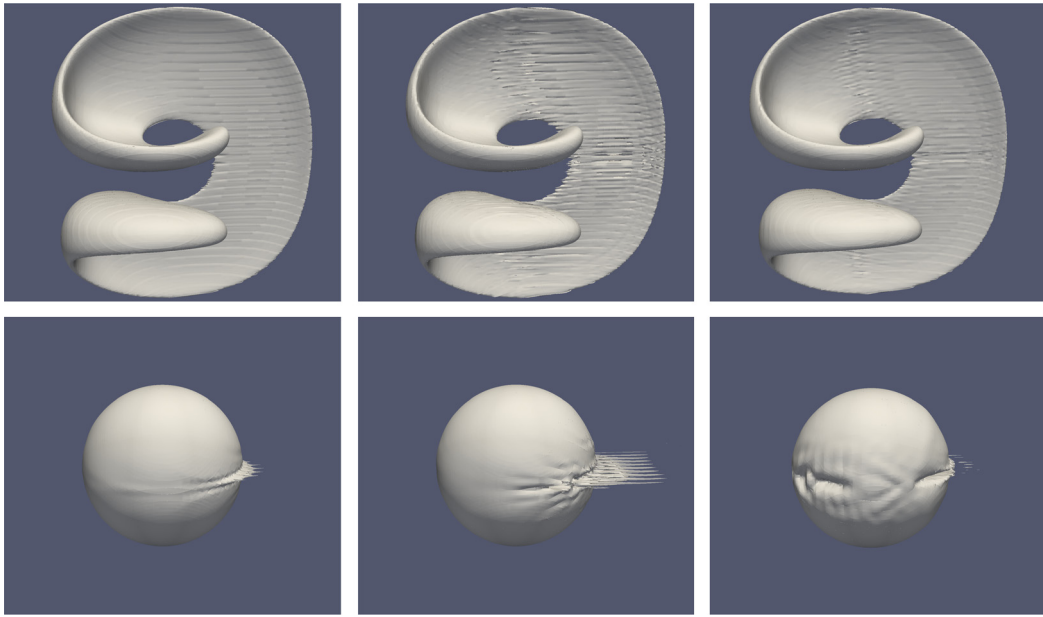


Fig. 13. Sphere in a deformation field interface shape at $t = T/2$ (top row) and $t = T$ (bottom row), all RKDGC-CLS-3 with $\Delta x = 1/64$; from left to right: No reinitialization; $T_r = 1.0, F = 1.0$; $T_r = 0.5, F = 1.0$.

Table 18

Deforming sphere shape errors and convergence rates for RKDG-CLS-k after full flow reversal at $t = T$.

(a) Varying reinit. factor		(b) Varying mesh			(c) Varying DG order	
64^3 mesh, $k = 2$		$T_r = 1.0, F = 1.0, k = 2$			$T_r = 1.0, F = 1.0, 64^3$	
T_r, F	E	Δx	E	Order	k	E
0.0,0.0	7.38e−2	1/32	1.20e−1	–	1	1.87e−1
1.0,1.0	1.26e−1	1/64	1.26e−1	−0.1	2	1.26e−1
0.5,1.0	8.59e−2	1/128	1.01e−1	0.3	3	5.63e−2

with $T = 3$. The normal vectors are calculated using the gradient-based approach.

Jibben and Herrmann [10] show results for the deforming sphere using a LS-WENO-5 method with a $\Delta x = 1/128$ cell width, finding a 27.4% mass loss after one revolution. By comparison RKDG-CLS-3 is far superior, losing only 0.38% of its mass after one revolution on a four times coarser mesh with $\Delta x = 1/32$.

Fig. 13 shows the interface shape at $t = T/2$, the time of maximum deformation, and $t = T$ after full flow reversal, for RKDG-CLS-3 using $\Delta x = 1/64$, varying the amount of reinitialization. In all cases, the thin sheet remains intact at full extension and the original shape is qualitatively restored after full flow reversal, though partially deformed. Parts (a) of Table 18 and Table 19 show the volume and shape errors for these cases. As in previous tests, reinitialization accentuates variations in the level set scalar field, distorting the interface and increasing shape and volume errors. With reinitialization present, shape errors converge particularly slowly with grid refinement, although even in these cases increasing the polynomial degree appreciably reduces the error. It is expected that 3D accurate normals would address this issue, similar to the 2D case.

Table 18 and Table 19 also show that without reinitialization, both grid refinement and increasing polynomial degree decrease shape and volume errors. Table 20 and Table 21 show the results of a convergence study for RKDG-CLS-2 and −3 with no reinitialization. In both cases the convergence is less than first order, though the error decreases with the use of higher polynomial degrees. This is likely due to the fact that at full extension, the interface forms a thin sheet, exhibiting discontinuities in the gradient of G close to the $G = 0.5$ iso-surface.

4. Summary

We have presented a Runge–Kutta discontinuous Galerkin method and demonstrated $k + 1$ order convergence rates with manufactured solutions test cases. This requires that all variables be expanded into a k th order polynomial basis, otherwise the order drops to the order of the smallest polynomial basis plus one. Test cases mimicking physical flow show the CLS method will produce deformed interfaces as a result of reinitialization advecting the interface, in agreement with both McCaslin and Desjardins [16] and Desjardins, Moureau, and Pitsch [4]. In many cases in which reinitialization is active, it is difficult to calculate formal convergence rates for the method. The fact that here the governing equations

Table 19Deforming sphere volume errors and convergence rates for RKDG-CLS-k after full flow reversal at $t = T$.

(a) Varying reinit. factor		(b) Varying mesh			(c) Varying DG order	
64^3 mesh, $k = 2$		$T_r = 1.0, F = 1.0, k = 2$			$T_r = 1.0, F = 1.0, 64^3$	
T_r, F	E	Δx	E	Order	k	E
0.0,0.0	3.12e−3	1/32	1.35e−1	–	1	1.23e−1
1.0,1.0	3.17e−2	1/64	3.17e−2	2.1	2	3.17e−2
0.5,1.0	3.63e−2	1/128	2.50e−3	3.7	3	1.40e−2

Table 20Deforming sphere shape errors and convergence rates for RKDG-CLS-k without reinit. after full flow reversal at $t = T$.

Δx	$k = 2$		$k = 3$	
	E	Order	E	Order
1/32	1.19e−1	–	6.03e−2	–
1/64	7.38e−2	0.7	3.15e−2	0.9
1/128	6.11e−2	0.3	2.02e−2	0.6

Table 21Deforming sphere volume errors and convergence rates for RKDG-CLS-k without reinit. after full flow reversal at $t = T$.

Δx	$k = 2$		$k = 3$	
	E	Order	E	Order
1/32	4.91e−2	–	3.83e−3	–
1/64	3.12e−3	4.0	2.12e−3	0.9
1/128	1.94e−3	0.7	1.61e−3	0.4

change at each mesh level, due to coupling the profile thickness ε to the mesh spacing, may contribute to the difficulty in calculating convergence rates. For Zalesak's disk, we find that reinitialization worsens shape errors, except at $k = 2$ where we observe modest improvement. Conversely, we find reinitialization volume errors are significantly improved at $k = 2$ and significantly worsened at $k = 4$. For $k = 3$, modest reinitialization improves the overall convergence rate while greater reinitialization worsens the results at every grid level. For the deforming column, we find reinitialization to worsen all shape and volume errors. This is partially due to the fact that reinitialization exacerbates interface deformations if the normal vector is calculated from the local level set field. The use of arbitrary-order accurate normals improves the quality of the interface by mitigating deformations attributed to reinitialization, resulting in improved volume and shape errors.

Acknowledgements

This work was supported by the National Science Foundation grant CBET-1054272 and the Department of Energy at Los Alamos National Laboratory under contract DE-AC52-06NA25396. Calculations were performed using the ASU Advanced Computing Center Saguaro and Ocotillo clusters.

References

- [1] H.L. Atkins, C.-W. Shu, Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations, *AIAA J.* 36 (5) (1998) 775–782.
- [2] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier Stokes equations, *J. Comput. Phys.* 85 (1989) 257–283.
- [3] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, *J. Sci. Comput.* 16 (2001) 173–261.
- [4] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, *J. Comput. Phys.* 227 (2008) 8395–8416.
- [5] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, *J. Comput. Phys.* 183 (2002) 83–116.
- [6] S. Gottlieb, On high order strong stability preserving Runge–Kutta and multi step time discretizations, *J. Sci. Comput.* 25 (2005) 105–128.
- [7] P. Greene, R. Nourgaliev, S. Schofield, Marker Re-Distancing (MRD) algorithm for high-fidelity interface tracking on arbitrary meshes, in: *ASME 2017 Fluids Engineering Division Summer Meeting*, 2017.
- [8] J. Grooss, J.S. Hesthaven, A level set discontinuous Galerkin method for free surface flows, *Comput. Methods Appl. Mech. Eng.* 195 (2006) 3406–3429.
- [9] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, *J. Comput. Phys.* 227 (2008) 2674–2706.
- [10] Z. Jibben, M. Herrmann, An arbitrary high-order conservative level set Runge–Kutta discontinuous Galerkin method for capturing interfaces, in: *ILASS Americas 25th Annual Conference on Liquid Atomization and Spray Systems*, 2013.
- [11] F. Lörcher, G. Gassner, C.-D. Munz, An explicit discontinuous Galerkin scheme with local time-stepping for general unsteady diffusion equations, *J. Comput. Phys.* 227 (2008) 5649–5670.
- [12] H. Luo, L. Luo, R. Nourgaliev, V.A. Mousseau, N. Dinh, A reconstructed discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids, *J. Comput. Phys.* 229 (2010) 6961–6978.
- [13] S. Luo, A uniformly second order fast sweeping method for Eikonal equations, *J. Comput. Phys.* 241 (2013) 104–117.

- [14] E. Marchandise, J.-F. Remacle, A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows, *J. Comput. Phys.* 219 (2) (2006) 780–800.
- [15] E. Marchandise, J.-F. Remacle, N. Chevaugeon, A quadrature-free discontinuous Galerkin method for the level set equation, *J. Comput. Phys.* 212 (1) (2006) 338–357.
- [16] J. McCaslin, O. Desjardins, A localised re-initialization equation for the conservative level set method, *J. Comput. Phys.* 262 (2014) 408–426.
- [17] A.M. Moghadam, M. Shafieefar, R. Panahi, Development of a high-order level set method: compact conservative level set (CCLS), *Comput. Fluids* 129 (2016) 79–90.
- [18] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, *J. Comput. Phys.* 210 (2005) 225–246.
- [19] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow II, *J. Comput. Phys.* 225 (2007) 785–807.
- [20] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1988) 12–49.
- [21] M. Owkes, O. Desjardins, A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows, *J. Comput. Phys.* 249 (2013) 275–302.
- [22] F. Pochet, K. Hillewaert, P. Geuzaine, J.-F. Remacle, É. Marchandise, A 3D strongly coupled implicit discontinuous Galerkin level set-based method for modeling two-phase flows, *Comput. Fluids* 87 (2013) 144–155.
- [23] W.H. Reed, T.R. Hill, Triangular Mesh Methods for the Neutron Transport Equation, Tech. rep. LA-UR-73-479, Los Alamos National Laboratory, 1973.
- [24] P.J. Roache, Code verification by the method of manufactured solutions, *J. Fluids Eng.* 124 (2002) 4–10.
- [25] K. Salari, P. Knupp, Code Verification by the Method of Manufactured Solutions, Tech. rep. SAND2000-1444, Sandia National Laboratory, 2000.
- [26] R. Saye, High-order methods for computing distances to implicitly defined surfaces, *Commun. Appl. Math. Comput. Sci.* 9 (1) (2014) 107–141.
- [27] C.-W. Shu, TVB uniform high-order schemes for conservation laws, *Math. Comput.* 46 (1987) 105–121.
- [28] M. Sussman, E.G. Puckett, A coupled level set and volume of fluid method for computing 3d and axisymmetric incompressible two-phase flows, *J. Comput. Phys.* 162 (2000) 301–337.
- [29] B. van Leer, Towards the ultimate conservation difference scheme, II, *J. Comput. Phys.* 14 (1974) 361–376.
- [30] B. van Leer, Towards the ultimate conservation difference scheme, V, *J. Comput. Phys.* 32 (1979) 1–136.
- [31] T. Waclawczyk, A consistent solution of the re-initialization equation in the conservative level-set method, *J. Comput. Phys.* 299 (2015) 487–525.
- [32] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.* 31 (1979) 335–362.
- [33] Y.-T. Zhang, S. Chen, F. Li, H. Zhao, C.-W. Shu, Uniformly accurate discontinuous Galerkin fast sweeping methods for Eikonal equations, *SIAM J. Sci. Comput.* 33 (2011) 1873–1896.
- [34] L. Zhao, J. Mao, X. Bai, X. Liu, T. Li, J. Williams, Finite element implementation of an improved conservative level set method for two-phase flow, *Comput. Fluids* 100 (2014) 138–154.