ERDC/CHL CHETN-XIII-2
January 2008

Implementation of Discontinuous Galerkin Methods for the Level Set Equation on Unstructured Meshes

by Matthew W. Farthing and Christopher E. Kees

PURPOSE: Level set methods are often used to capture interface behavior in two-phase, incompressible flow models. While level set techniques for structured computational grids have been widely investigated, approaches for unstructured meshes are less mature. This report details the formulation and implementation of a discontinuous Galerkin-based approach that is suitable for unstructured meshes and offers potential gains in accuracy and efficiency over more traditional level set techniques.

INTRODUCTION: Flow of air and water around solid objects can be modeled by assuming the phases are separated by sharp interfaces and that each fluid subdomain is governed by the Navier-Stokes equations. The resulting model requires resolution of the flow field and the location and evolution of the interfaces. There are many techniques available for approximating flow in each subdomain and many ways to resolve the interfaces. Level set methods represent one such class of techniques for capturing interface behavior that has been applied successfully to problems in fields from computational geometry to fluid mechanics (Osher and Fedkiw 2001, Sethian 2001). Among other things, the appeal of level set methods can be attributed to the generality of their formulation, ability to resolve interfaces accurately while allowing for topological changes, and relatively straightforward extension to higher dimensional problems (Sethian 1999).

Level set methods for structured grids are fairly mature (Sethian 2001, Osher and Fedkiw 2001). Although extensions for general interface propagation problems as well as two-phase flow have been considered (Barth and Sethian 1998, Sethian and Vladimirsky 2000, Nagrath et al. 2005, Smolianski 2005), approaches for unstructured meshes are less mature. With this in mind, we are interested in the design and implementation of level set techniques for air/water flow that naturally apply to unstructured tetrahedral meshes and are well suited to distributed computing architectures.

Standard level set methodology builds upon an accurate discretization for a linear hyperbolic partial differential equation (PDE) or an equivalent Hamilton-Jacobi formulation (Sethian 2001). It also relies heavily on an effective reinitialization technique that must be employed at intermediate times to recover the accuracy of the level set representation of the fluid front (Sussman and Fatemi 1999).

In this report we focus on the implementation of an alternative strategy that employs a Runge-Kutta discontinuous Galerkin (RKDG) discretization (Cockburn and Shu 2001) and exploits the fluid incompressibility assumption (Marchandise et al. 2006). With sufficiently high-order approximations, this method potentially offers accurate solutions with good mass conservation, while obviating the reinitialization step. It also lends itself to fully explicit time integration and a quadrature-free implementation that is readily parallelizable (Atkins and Shu 1996, Baggag et al. 1999).

FORMULATION: We begin with a physical domain Ω in which an interface $\Gamma(t)$ evolves over a time interval $[0, T]$. To characterize $\Gamma(t)$, we adopt a level set formulation and define implicitly a function $\phi(\mathbf{x}, t)$ such that $\phi(\mathbf{x}, t) = 0$ corresponds to $\Gamma(t)$. In this case, the propagation of $\Gamma(t)$

with normal speed u_n can be expressed as (Osher and Fedkiw 2001, Sethian 2001)

$$\frac{\partial \phi}{\partial t} + u_n \|\nabla \phi\| = 0, \text{ for } \mathbf{x}, t \in \Omega \times (0, T] \tag{1}$$

with the initial data, $\phi(\mathbf{x}, 0) = \phi^0(\mathbf{x})$, chosen so that $\phi^0(\mathbf{x}) = 0$ on $\Gamma(0)$. We further require that $\phi^0(\mathbf{x})$ be a signed distance (i.e., $\phi^0(\mathbf{x}) = \pm d$ where d is the distance to Γ), although this is not strictly necessary (Olsson and Kreiss 2005).

Successful level set approximations require accurate solution of Equation 1 and a suitable velocity, $\mathbf{u}$, defined throughout Ω that gives the correct front propagation speed, u_n, on Γ (Sethian 2001). In addition, these methods typically require an efficient approach for initializing $\phi(\mathbf{x}, t = t^m)$ at given time instances, t^m, that ensures $\phi(\mathbf{x}, t^m)$ is sufficiently smooth over Ω but yet maintains $\phi(\mathbf{x}, t^m) = 0$ on $\Gamma(t^m)$ (Sussman and Fatemi 1999). As mentioned in the introduction, the emphasis here is on one configuration of discrete approximations and solution algorithms that attempts to address the above requirements and may hold promise for simulating multiphase, incompressible fluid flow.

Conservative level set equation formulation: The approaches considered rely on discontinuous Galerkin (DG) spatial discretizations to obtain accurate numerical solutions for ϕ on unstructured meshes. DG approximations for ϕ can be obtained in at least two ways. Equation 1 can be solved directly using a general RKDG formulation for Hamilton-Jacobi equations (Hu and Shu 1999, Li and Shu 2005). Or, we can take advantage of the fact that our area of interest is incompressible, multiphase flow in order to apply methods for solving conservative, linear advection problems. We follow Marchandise et al. (2006) and adopt the latter approach here, since it allows us to use tools we have previously developed for solving PDEs with DG methods (Li et al. 2007).

First, Equation 1 is equivalent to solving

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{2}$$

for an appropriately defined $\mathbf{u}$. In the case of two-phase flow, we can identify $\mathbf{u}$ with the fluid velocity so that $\mathbf{u} \cdot \mathbf{n} = u_n$, where $\mathbf{n} = \nabla\phi / \|\nabla\phi\|$ is the unit normal to the interface Γ (Chang et al. 1996). Equation 2 can then be rearranged as

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = \phi \nabla \cdot \mathbf{u} \tag{3}$$

which gives

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \tag{4}$$

when the fluid is incompressible (i.e., $\nabla \cdot \mathbf{u} = 0$) (Marchandise et al. 2006).

Element weak formulation: Equation 4 is just the linear advection equation in conservative form, and a weak formulation follows in a standard way (Cockburn and Shu 2001). Given a triangulation, $\mathcal{M}^h$, of Ω and an element $\mathcal{E} \in \mathcal{M}^h$, an approximate solution is sought in the space

$$V_h = \{v_h \in L^\infty(\Omega) : v_h|_{\mathcal{E}} \in V_h(\mathcal{E}), \forall \mathcal{E} \in \mathcal{M}^h\} \tag{5}$$

where we denote the local discrete test and trial space as $V_h(\mathcal{E})$. Multiplying Equation 4 by a test function and integrating by parts over an element $\mathcal{E}$, we have

$$\int_{\mathcal{E}} v_h \frac{\partial \phi}{\partial t} \, \mathrm{d}x = \int_{\mathcal{E}} \phi \mathbf{u} \cdot \nabla v_h \, \mathrm{d}x - \int_{\partial \mathcal{E}} v_h \phi \mathbf{u} \cdot \mathbf{n}_{\mathcal{E}} \, \mathrm{d}s, \ \forall v_h \in V_h(\mathcal{E}) \tag{6}$$

where $\mathbf{n}_{\mathcal{E}}$ is the unit outer normal on $\mathcal{E}$. Inserting a trial solution $\phi_h \in V_h(\mathcal{E})$ gives

$$\int_{\mathcal{E}} v_h \frac{\partial \phi_h}{\partial t} \, \mathrm{d}x = \int_{\mathcal{E}} \phi_h \mathbf{u} \cdot \nabla v_h \, \mathrm{d}x - \int_{\partial \mathcal{E}} v_h \phi_h \mathbf{u} \cdot \mathbf{n}_{\mathcal{E}} \, \mathrm{d}s, \ \forall v_h \in V_h(\mathcal{E}) \tag{7}$$

Since the underlying spaces are discontinuous, the flux on internal element boundaries is multiply defined. We then replace the outer flux in the last term of the right-hand side of Equation 7 with a numerical, upwinded flux

$$\int_{\mathcal{E}} v_h \frac{\partial \phi_h}{\partial t} \, \mathrm{d}x = \int_{\mathcal{E}} \phi_h \mathbf{u} \cdot \nabla v_h \, \mathrm{d}x - \int_{\partial \mathcal{E}} v_h \phi_h^{up} \mathbf{u} \cdot \mathbf{n}_{\mathcal{E}} \, \mathrm{d}s, \ \forall v_h \in V_h(\mathcal{E}) \tag{8}$$

where

$$\begin{aligned} \phi^{up} &= \begin{cases} \phi^- & \mathbf{u} \cdot \mathbf{n}_{\mathcal{E}} \geq 0 \\ \phi^+ & \text{otherwise} \end{cases} \\ \phi^- &= \lim_{\epsilon \to 0^-} \phi(\mathbf{x} + \epsilon \mathbf{n}_{\mathcal{E}}, t) \\ \phi^+ &= \lim_{\epsilon \to 0^+} \phi(\mathbf{x} + \epsilon \mathbf{n}_{\mathcal{E}}, t) \end{aligned} \tag{9}$$

That is, ϕ^{up} is simply the value of ϕ taken from the upwind element at an element interface.

DISCRETE APPROXIMATION: Equation 8 represents a semi-discrete system on each element of $\mathcal{M}^h$ with coupling across elements introduced through element boundary fluxes. In the following, we detail an RKDG discrete approximation for Equation 8 on affine, simplicial meshes, which is appealing in its simplicity.

Finite element approximation: We assume that there is a consistently defined, affine mapping $\mathbf{F}$ from a reference element, $\hat{\mathcal{E}}$, for all $\mathcal{E} \in \mathcal{M}^h$, where $\hat{\mathcal{E}}$ is the reference simplex in $\mathbb{R}^d$, $d = 1, 2, 3$ (see Figure 1). A local basis for $\mathcal{E}$ is denoted $\{N_i\}$ for $i = 1, \ldots, n_p$, where

$$N_i = \hat{N}_i \circ \mathbf{F}^{-1}$$

and $\{\hat{N}_i\}$ is a basis for $P^k(\hat{\mathcal{E}})$, the space of polynomials on $\hat{\mathcal{E}}$ of degree k or lower. n_p is the dimension of $P^k(\hat{\mathcal{E}})$. To define the local shape functions, $\{\hat{N}_i\}$, we consider the standard nodal Lagrangian polynomials on $\hat{\mathcal{E}}$ (see Figure 2 for the $P^3(\hat{\mathcal{E}})$ case). We label the corresponding set of nodal locations $\{\hat{p}_i\}$ with $\hat{N}_i(\hat{\mathbf{p}}_j) = \delta_{ij}$, for $i, j = 1, \ldots, n_p$, In general, we follow the convention of using a hat (ˆ) to signify quantities associated with a reference element and write a trial solution as $\phi_h = \sum_{j=1}^{n_p} \phi^j N_j$. For convenience, we also define $\mathbf{f} = \phi \mathbf{u}$ and

$$f_{\mathcal{E}}^{up} = \phi_h^{up} \mathbf{u} \cdot \mathbf{n}_{\mathcal{E}} \tag{10}$$

as the upwinded, normal flux along the boundary of $\mathcal{E}$.

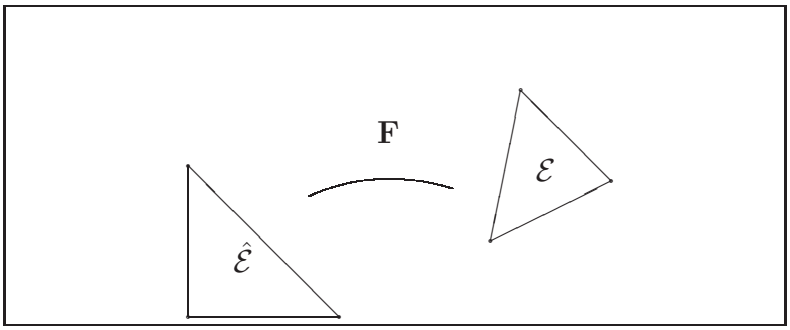

Figure 1. reference element, $\hat{\mathcal{E}}$ and mapping $\mathbf{F} : \hat{\mathcal{E}} \rightarrow \mathcal{E}$

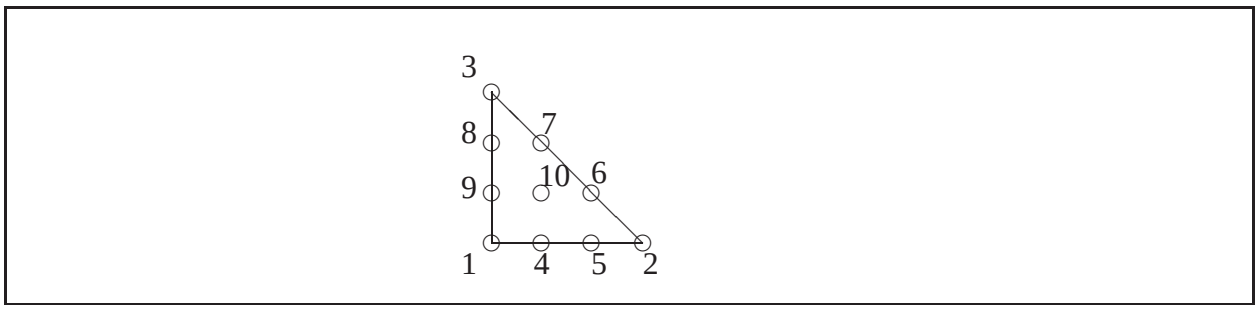

Figure 2. Standard Lagrangian nodal locations on reference element $\hat{\mathcal{E}}$ for P^3 (triangles)

Quadrature-free approximation: DG methods typically require more degrees of freedom than their standard C^0 Galerkin counterparts and also require the evaluation of both volume and surface integrals over each element. These factors may lead to increased operation counts and storage requirements for DG methods, particularly when numerical quadrature is used to evaluate element integrals. On the other hand, a quadrature-free implementation, in which element and boundary integrals are evaluated analytically, can reduce operation counts and storage requirements for a DG discretization of Equation 8 significantly (Atkins and Shu 1996). Although the standard approach for DG methods is to use numerical integration (Cockburn and Shu 1998), a quadrature-free implementation for Equation 8 is straightforward. In essence, only two basic modifications are necessary to accommodate the quadrature-free approach. The first is a projection of $\mathbf{f}$ in $[V_h(\mathcal{E})]^d$, and the other is a representation of numerical fluxes on element boundaries (Marchandise et al. 2006).

Before describing this quadrature-free RKDG approach, we introduce some additional notation. An element boundary is written as e, and the global set of element boundaries in $\mathcal{M}^h$ is $\{e_i\}$. That is, $\{e_i\}$ is the set of vertices in $\mathcal{M}^h$ in one spatial dimension, the set of edges in $\mathcal{M}^h$ for two-dimensional problems, and faces in three dimensions. A tilde ($\tilde{}$) is used to distinguish quantities associated with element boundaries.

In the following, we restrict ourselves to conforming meshes and introduce a local trial space $\tilde{V}_h(e)$ on each $\mathbb{R}^{d-1}$ element boundary, e, contained in $\mathcal{M}^h$ (Atkins and Shu 1996). For lack of better notation, we label the basis for $\tilde{V}_h(e)$ as $\{\tilde{N}_i\}$, for $i = 1, \ldots, \tilde{n}_p$, and the $\mathbb{R}^{d-1}$ reference simplex

as $\hat{e}$ ($\hat{e}$ is a point for $d = 1$). We define $\tilde{N}_i$ as before

$$\tilde{N}_i = \widehat{\tilde{N}}_i \circ \mathbf{G}^{-1}$$

where $\mathbf{G}$ is a consistently chosen map from $\hat{e}$ to e and $\{\widehat{\tilde{N}}_i\}$, $i = 1, \ldots, \tilde{n}_p$ is a basis for $P^k(\hat{e})$. A natural fit is to use a Lagrangian basis for $P^k(\hat{e})$ with associated nodes $\hat{\tilde{\mathbf{p}}}_j \in \mathbb{R}^{d-1}$.

For every element, we require a mapping from the elemental space $V_h(\mathcal{E})$ to $\tilde{V}_h(e)$ for each $e \in \partial\mathcal{E}$. Accordingly, we introduce the trace mapping $\mathcal{T}_l^{\mathcal{E}} : V_h(\mathcal{E}) \rightarrow \tilde{V}_h(e_l)$

$$\mathcal{T}_l^{\mathcal{E}}(N_j) = \sum_{i=1}^{\tilde{n}_p} T_{l,ij}^{\mathcal{E}} \tilde{N}_i \tag{11}$$

$$T_{l,ij}^{\mathcal{E}} = N_j(\tilde{\mathbf{p}}_i), \quad \tilde{\mathbf{p}}_i = \mathbf{G}(\hat{\tilde{\mathbf{p}}}_i) \tag{12}$$

In other words, for each boundary face, e_l, of an element $\mathcal{E}$, we have a local matrix $\mathbf{T}_l^{\mathcal{E}} \in \mathbb{R}^{\tilde{n}_p \times n_p}$ that takes members of the element trial/test space to a $d-1$ dimensional space defined on the boundary face that is independent of the element's local coordinate system. Similarly, the transpose of $\mathbf{T}_l^{\mathcal{E}}$ can be used to map from $\tilde{V}_h(e_l)$ to $V_h(\mathcal{E})$.

To define $\mathbf{f}_h \in [V_h(\mathcal{E})]^d$, we use either a coordinate-wise L^2 projection

$$\int_{\mathcal{E}} f_h^k N_i \, \mathrm{d}x = \int_{\mathcal{E}} \phi_h u^k N_i \, \mathrm{d}x, \; i = 1, \ldots, n_p, \; k = 1, \ldots, d \tag{13}$$

or simple nodal interpolation

$$f_h^{k,j} = \phi^j u^k(\mathbf{p}_j), \; j = 1, \ldots, n_p, \; k = 1, \ldots, d \tag{14}$$

A unique upwinded numerical flux $f^{up} \in \tilde{V}_h(e)$ is also necessary for each $e \in \mathcal{M}^h$. We set

$$f_e^{up} = \sum_{j=1}^{\tilde{n}_p} f_e^{up,j} \tilde{N}_j \tag{15}$$

with $f_e^{up,j} = \phi^{up,j} \mathbf{u} \cdot \mathbf{n}_e(\tilde{\mathbf{p}}_j)$. The unit outer normal, $\mathbf{n}_e$, is arbitrarily chosen to point from "left" to "right," so that it will be $\pm$ the unit outer normal for the elements neighboring e. The value of ϕ_h^{up} is defined using the left and right traces

$$\phi^{up,j} = \phi_h^{up}(\tilde{\mathbf{p}}_j) = \begin{cases} \tilde{\phi}_h^L(\tilde{\mathbf{p}}_j) & \mathbf{u} \cdot \mathbf{n}_e \geq 0 \\ \tilde{\phi}_h^R(\tilde{\mathbf{p}}_j) & \text{otherwise} \end{cases} \tag{16}$$

The left and right traces are defined simply by mapping the local element degrees of freedom, $\boldsymbol{\phi}_{\mathcal{E}} \in \mathbb{R}^{n_p}$, from the neighboring elements ($\mathcal{E}^L$ and $\mathcal{E}^R$), to the corresponding degrees of freedom for $\tilde{V}_h(e)$

$$\begin{aligned} \tilde{\boldsymbol{\phi}}^L &= \mathbf{T}_l^L \boldsymbol{\phi}_L \\ \tilde{\boldsymbol{\phi}}^R &= \mathbf{T}_{l'}^R \boldsymbol{\phi}_R \end{aligned} \tag{17}$$

where $\tilde{\phi}^{L/R} \in \mathbb{R}^{\tilde{n}_p}$. The subscripts l and l' are local numberings for e for the left and right neighboring elements, respectively.

Inserting the new notation into Equation 8, we have

$$\sum_{j=1}^{n_p} \frac{\partial \phi^j}{\partial t} \int_{\mathcal{E}} N_i N_j \, \mathrm{d}x = \int_{\mathcal{E}} \mathbf{f}_h \cdot \nabla N_i \, \mathrm{d}x - \sum_{l=1}^{d+1} \int_{e_l} N_i f^{up}_{\mathcal{E},l} \, \mathrm{d}s \tag{18}$$

for $i = 1, \ldots, n_p$ and $f^{up}_{\mathcal{E},l} = f^{up}_{e_l} \mathbf{n}_{e_l} \cdot \mathbf{n}_{\mathcal{E}}$. The three spatial integrals in Equation 18 can now be calculated analytically over each element, and the corresponding semi-discrete system can be integrated in time using a method of lines approach and a class of Runge-Kutta time discretizations (Marchandise et al. 2006). Calculation of the spatial integrals for two-dimensional triangular meshes is detailed in Appendix A.

Time integration: To integrate Equation 18, we use a class of explicit, strong stability preserving (SSP) Runge-Kutta schemes. The SSP property is based on the assumption that, for a given problem, a forward Euler time discretization is stable under a given norm (and suitable time step constraint). An SSP method is then one that maintains this stability with possibly different approximation order and time step constraint (Gottlieb et al. 2001). For our purposes, we can consider linear ordinary differential equation (ODE) systems, since the mass matrix in Equation 18 is invertible and limiting is not used (Cockburn and Shu 2001, Marchandise et al. 2006). That is, we have an ODE system

$$\frac{d\mathbf{y}}{dt} = \mathbf{L}\mathbf{y} \tag{19}$$

An s-stage family of Runge-Kutta methods that are SSP for Equation 19 can be written

$$\begin{aligned} \mathbf{y}^0 &= \mathbf{y}^n \\ \mathbf{y}^m &= \mathbf{y}^{m-1} + \Delta t^{n+1} \mathbf{L}\mathbf{y}^{m-1}, \text{ for } m = 1, \ldots, s-1 \\ \mathbf{y}^s &= \sum_{k=0}^{s-2} \alpha_{s,k} \mathbf{y}^k + \alpha_{s,s-1} \left(\mathbf{y}^{s-1} + \Delta t^{n+1} \mathbf{L}\mathbf{y}^{s-1} \right) \\ \mathbf{y}^{n+1} &= \mathbf{y}^s \end{aligned} \tag{20}$$

for suitably chosen coefficients $\alpha_{s,k}$, $k = 0, \ldots, s-1$ (Gottlieb et al. 2001). A DG discretization of order p should be stable for Equation 20 with the Courant-Friedrichs-Lewy (CFL) condition

$$\Delta t < \frac{h}{c(2p+1)} \tag{21}$$

where h is the element size and $c \geq \|\mathbf{u}\|$ for all $\mathcal{E} \in \mathcal{M}^h$ (Cockburn and Shu 2001, Marchandise et al. 2006). For the numerical experiments below, we use the $k+1$*th* version of Equation 20 along with Equation 21 when the spatial approximation is based on P^k elements. The corresponding coefficients can be found in Table 3.1 of Gottlieb et al. (2001).

NUMERICAL EXPERIMENTS: To evaluate our quadrature-free RKDG approach, we consider several classical test problems for propagating interfaces with a specified velocity (Rider and Kothe 1995, Sussman and Fatemi 1999, Pilliod and Puckett 2004, Olsson and Kreiss 2005, Marchandise et al. 2006).

Test problems: The physical domain for the first test problem (PA) is $\Omega = [0,1] \times [0,1]$, and the velocity field is $\mathbf{u} = [2\pi(y - 1/2), 2\pi(1/2 - x)]$. The exact solution is a cone of radius $r_0 = 1/8$. A parameterization for the exact solution, ϕ^{ex}, is

$$\phi^{ex}(x,y,t) = \begin{cases} (1 + \cos(\pi\bar{x}/r_0))\,(1 + \cos(\pi\bar{y}/r_0))\,/4, & \|\bar{\mathbf{r}}\| < r_0 \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

where $\bar{x} = x - x_c$, $\bar{y} = y - y_c$, and

$$x_c = \sin(2\pi t)/4 + 1/2,\ y_c = \cos(2\pi t)/4 + 1/2$$

The second problem (PB) is set on $\Omega = [0,1] \times [0,1]$ with a velocity field given by

$$\begin{aligned} u^x &= \cos(\pi t/8)\sin(2\pi y)\sin^2(\pi x) \\ u^y &= -\cos(\pi t/8)\sin(2\pi x)\sin^2(\pi y) \end{aligned} \tag{23}$$

The initial condition is a disk of radius 0.15 centered at $(0.5, 0.75)$, with an initial signed distance function $d_0 = (x - 0.5)^2 + (y - 0.75)^2 - 0.15^2$. The solution should return to the initial condition at $T = 8$.

The final example (PC) is on $\Omega = [0,100] \times [0,100]$ with a velocity field

$$\mathbf{u} = (\pi(50 - y)/314, \pi(x - 50)/314)^T$$

The initial condition is Zalesak's slotted disk with a radius of 15, width of 5, and slot length of 15 (Sussman and Fatemi 1999, Marchandise et al. 2006).

Illustrative results: For each problem, a regular triangulation of Ω was formed with N_x triangles along the x axis, and N_y triangles along the y axis. Simulations were performed with varying orders of approximation from $k = 1, \ldots, 4$. A target Courant number, Cr, was chosen close to the maximum allowed for each k as given by Equation 21. A dual processor Mac G5 (2 GHz) with 2 GB RAM was used for the computations. The methods were implemented in `C++` and compiled with `gcc` version 3.3 and `-O` optimization.

Table 1 summarizes the computations performed for PA, while Table 2 contains the corresponding L^1 errors, CPU times, and mass errors. The mass error here is simply defined to be the normalized difference between the mass in the numerical and analytical solutions. In Table 2, N_d is the total number of degrees of freedom on each mesh. Figure 3 shows the initial condition and solution at $T = 0.5$ for a P^2 approximation on a grid with 64×64 triangles.

Table 3 summarizes the computations performed for PB, and Table 4 contains the corresponding L^1 errors, total mass errors, and CPU times. Figure 4 shows the zero contour of the numerical solution for a P^2 approximation on a 64×64 regular grid. The exact solution for PB is identical to the initial condition. The right plot illustrates the significant deformation that the solution experiences before returning to the initial state at $T = 8$. Note the difference in scales for the two plots.

The simulations performed for PC are summarized in Table 5, while Figure 5 illustrates the relative accuracy for first- through fourth-order approximations on the same 128×128 grid. The difference in accuracy is most evident around corners, where the lower order methods are smeared significantly (Sussman and Fatemi 1999, Marchandise et al. 2006).

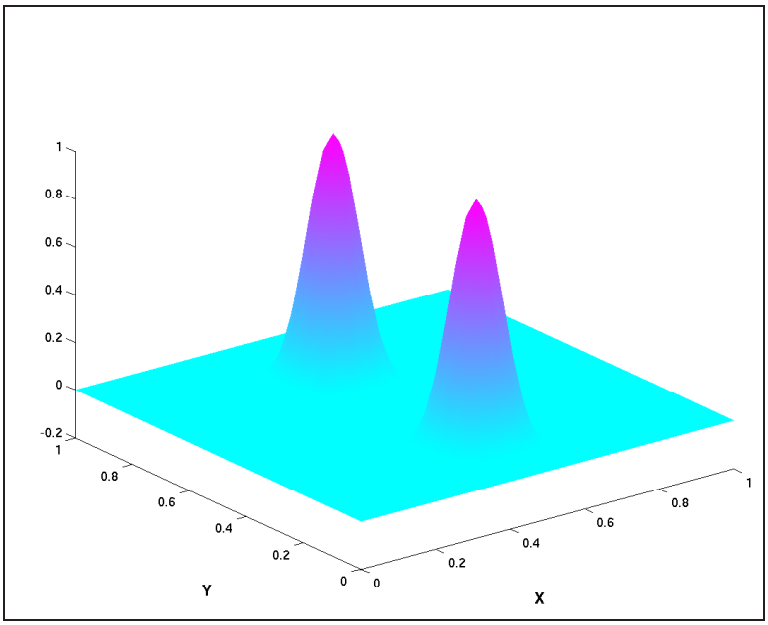

Figure 3. Solution PA.2, $T = 0.5$

Table 1. Simulation summary PA

Run	Problem	P^k	N_x	N_y	N_d	Cr
PA.1	PA	1	128	128	98304	0.3
PA.2	PA	2	64	64	49152	0.18
PA.3	PA	3	32	32	20480	0.128
PA.4	PA	4	16	16	7680	0.1

Table 2. Simulation results PA

Run	L_1 Error	Mass Error	CPU [s]
PA.1	1.3097×10^{-4}	0.000	3.214×10^{2}
PA.2	1.1110×10^{-4}	0.000	2.048×10^{2}
PA.3	1.5038×10^{-4}	5.600×10^{-6}	8.237×10^{1}
PA.4	3.3699×10^{-4}	2.230×10^{-5}	2.805×10^{1}

Table 3. Simulation summary PB

Run	Problem	P^k	N_x	N_y	Cr
PB.1	PB	1	128	128	0.3
PB.2	PB	2	64	64	0.18
PB.3	PB	3	64	64	0.128
PB.4	PB	4	32	32	0.1

DISCUSSION: The numerical results above are preliminary. They indicate that a quadrature-free RKDG approach can accurately resolve propagating interfaces for simple rotating velocity fields and more complex two-dimensional flows as in PB (Rider and Kothe 1995). The results show that higher order approximations are more accurate for the same spatial resolution. In some

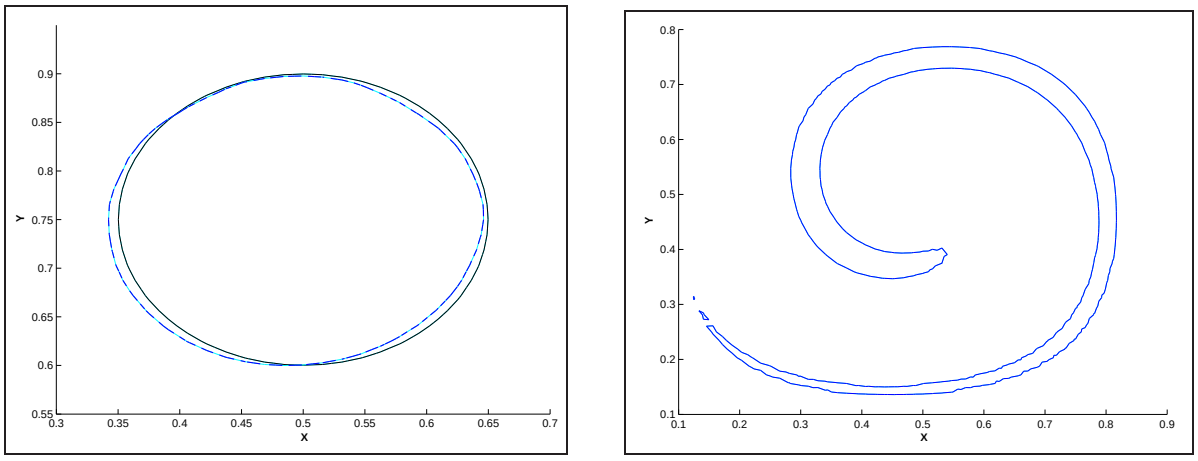

Figure 4. PB.2. ϕ^0 (black), ϕ (blue) at $T = 8$ [left]. ϕ at $T = 1.8$ [right]

Table 4. Simulation results PB

Run	L_1 Error	Mass Error	CPU [s]
PB.1	2.6219×10^{-3}	0.000	1.416×10^{3}
PB.2	1.6250×10^{-3}	0.000	8.199×10^{2}
PB.3	1.0616×10^{-3}	0.000	2.566×10^{3}
PB.4	1.6091×10^{-3}	1.000×10^{-6}	8.113×10^{2}

Table 5. Simulation summary PC

Run	Problem	P^k	N_x	N_y	Cr
PC.1	PC	1	128	128	0.3
PC.2	PC	2	128	128	0.18
PC.3	PC	3	128	128	0.128
PC.4	PC	4	128	128	0.1

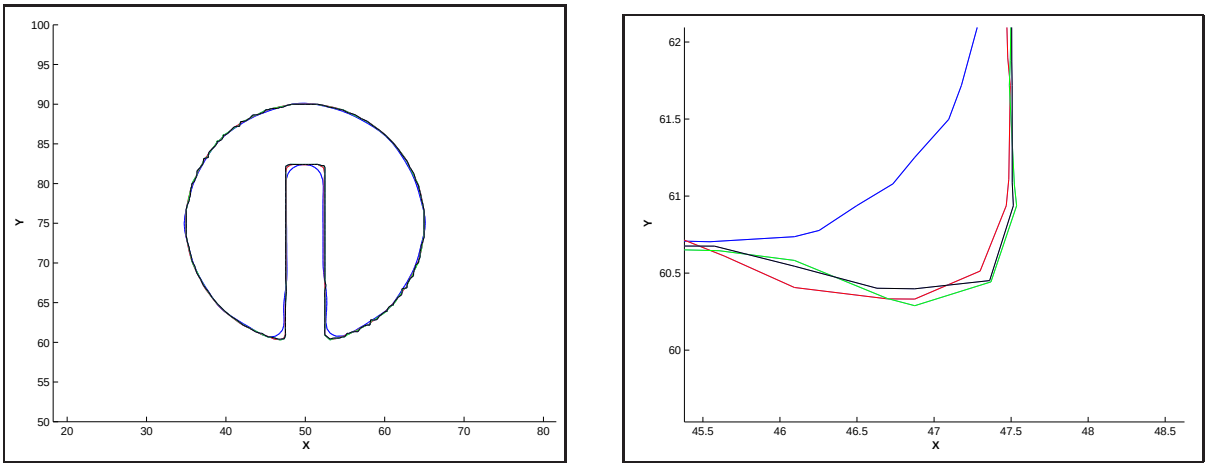

Figure 5. PC, $T = 628$. Zero contour, P^1 (blue), P^2 (red), P^3 (green), and P^4 (black)

cases, similar accuracy can be achieved for much coarser meshes with higher order approximations and an overall reduction in computational effort. However, this work does not constitute a rigorous evaluation of computational efficiency. Moreover, the methods' performance needs to be evaluated for problems in which there is nontrivial coupling between the interface location and fluid flow.

ADDITIONAL INFORMATION: This CHETN is a product of the High Fidelity Vessel Effects Work Unit of the Navigation Systems Research Program being conducted at the U.S. Army Engineer Research and Development Center, Coastal and Hydraulics Laboratory. Questions about this technical note can be addressed to Dr. Christopher Kees (Voice: 601-634-3110, email: `christopher.e.kees@erdc.usace.army.mil`). For information about the Navigation Systems Research Program, please contact the Navigation Systems Program Manager, Dr. John Hite (Voice: 601-634-2402, email: `John.E.Hite@erdc.usace.army.mil`). This technical note should be cited as follows:

> Farthing, M., and C. Kees. 2008. *Implementation of discontinuous Galerkin methods for the level set equation on unstructured meshes*. Coastal and Hydraulics Engineering Technical Note ERDC/CHL CHETN XIII-2. Vicksburg, MS: U.S. Army Engineer Research and Development Center.

REFERENCES

Atkins, H. L., and C. W. Shu. 1996. Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations. *AIAA Journal* 36:775–782.

Baggag, A., H. Atkins, and D. Keyes. 1999. *Parallel implementation of the discontinuous Galerkin method*. Technical Report NASA/CR-1999-209546, ICASE Report No. 99-35.

Barth, T. J., and J. A. Sethian. 1998. Numerical schemes for Hamilton-Jacobi and level set equations on triangulated domains. *Journal of Computational Physics* 145:1–40.

Chang, Y., B. Hou, T.Y. Merriman, and S. Osher. 1996. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics* 124:449–464.

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

Cockburn, B., and C. W. Shu. 2001. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing* 16(3):173–247.

Gottlieb, S., C. W. Shu, and E. Tadmor. 2001. Strong stability-preserving high-order time discretization methods. *SIAM Review* 43(1):89–112.

Hu, C., and C.-W. Shu. 1999. A discontinuous Galerkin finite element method for Hamilton-Jacobi equations. *SIAM Journal on Scientific Computing* 21(2):666–690.

Li, F., and C.-W. Shu. 2005. Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton-Jacobi equations. *Applied Mathematics Letters* 18:1204–1209.

Li, H., M. W. Farthing, C. N. Dawson, and C. T. Miller. 2007. Local discontinuous Galerkin approximations to Richards' equation. *Advances in Water Resources* 30:555–575.

Marchandise, E., J. F. Remacle, and N. Chevaugeon. 2006. A quadrature-free discontinuous Galerkin method for the level set equation. *Journal of Computational Physics* 212:338–357.

Nagrath, S., K. E. Jansen, and R. T. Lahey. 2005. Computation of incompressible bubble dynamics with a stabilized finite element level set method. *Computer Methods in Applied Mechanics and Engineering* 194:4565–4587.

Olsson, E., and G. Kreiss. 2005. A conservative level set method for two phase flow. *Journal of Computational Physics* 210:225–246.

Osher, S., and R. P. Fedkiw. 2001. Level set methods: An overview and some recent results. *Journal of Computational Physics* 169:463–502.

Pilliod, J., and E. Puckett. 2004. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics* 199(2):465–502.

Rider, W., and D. Kothe. 1995. Stretching and tearing interface tracking methods. In *12th AIAA Computational Fluid Dynamics Conference*, 2:806–816. San Diego, CA.

Sethian, J. A. 1999. *Level set methods and fast marching methods: Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. New York: Cambridge University Press.

Sethian, J. A. 2001. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. *Journal of Computational Physics* 169:503–555.

Sethian, J. A., and A. Vladimirsky. 2000. Fast methods for the Eikonal and related Hamilton-Jacobi equations on unstructured meshes. *Proceedings of the National Academy of Science* 97(11):5699–5703.

Smolianski, A. 2005. A finite element/level-set/operator splitting (FELSOS) approach for computing two-fluid unsteady flows with free moving interfaces. *International Journal for Numerical Methods in Fluids* 48:231–269.

Sussman, M., and E. Fatemi. 1999. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing* 20(4):1165–1191.

APPENDIX A. ELEMENT INTEGRAL CALCULATIONS: To illustrate the calculations necessary to evaluate Equation 18 analytically, we consider a two-dimensional example. The element matrix on the left-hand side of Equation 18 is the standard mass matrix

$$M_{ij} = \int_{\mathcal{E}} N_i N_j \, \mathrm{d}x = \int_{\hat{\mathcal{E}}} \hat{N}_i \hat{N}_j |\det J| \, \mathrm{d}\hat{x}\mathrm{d}\hat{y} = |\det J| \hat{M}_{ij} \tag{24}$$

where $\mathbf{J}$ is the Jacobian of $\mathbf{F}$. The value of $\mathbf{J}$ is constant on each element, since $\mathbf{F}$ is affine, and $\widehat{\mathbf{M}} = (\hat{M}_{ij}) \in \mathbb{R}^{n_p \times n_p}$ is only a function of the reference element and the local shape functions chosen.

The first integral on the right-hand side of Equation 18 is also straightforward to calculate. Recalling that $\nabla N_i = \mathbf{J}^{-t} \widehat{\nabla} \hat{N}_i$ and setting $\mathbf{f}_h = [f_h^x \; f_h^y]^t$, we have

$$\int_{\mathcal{E}} \mathbf{f} \cdot \nabla N_i \, \mathrm{d}x = |\det J| \sum_{j=1}^{n_p} g^{\hat{x},j} D_{ij}^{\hat{x}} + |\det J| \sum_{j=1}^{n_p} g^{\hat{y},j} D_{ij}^{\hat{y}} \tag{25}$$

$$g^{\hat{x},j} = f^{x,j} J_{11}^{-1} + f^{y,j} J_{12}^{-1} \tag{26}$$

$$g^{\hat{y},j} = f^{x,j} J_{21}^{-1} + f^{y,j} J_{22}^{-1} \tag{27}$$

$$D_{ij}^{\hat{x}} = \int_{\hat{\mathcal{E}}} \frac{\partial \hat{N}_i}{\partial \hat{x}} \hat{N}_j \, \mathrm{d}\hat{x}\mathrm{d}\hat{y} \tag{28}$$

$$D_{ij}^{\hat{y}} = \int_{\hat{\mathcal{E}}} \frac{\partial \hat{N}_i}{\partial \hat{y}} \hat{N}_j \, \mathrm{d}\hat{x}\mathrm{d}\hat{y} \tag{29}$$

Using the local trace mapping (transposed) to write

$$N_i = \sum_{j=1}^{\tilde{n}_p} T_{l,ji}^{\mathcal{E}} \tilde{N}_j \tag{30}$$

the boundary integral in Equation 18 can be written in terms of the element boundary flux degrees of freedom, $\{f_e^{up,j}\}$ for $j = 1, \ldots, \tilde{n}_p$ and local element matrices

$$\int_{e_l} N_i f_{\mathcal{E},l}^{up} \, \mathrm{d}s = a_l^{\mathcal{E}} |e_l| \sum_{j=1}^{\tilde{n}_p} B_{l,ij}^{\mathcal{E}} f_{e_l}^{up,j} \tag{31}$$

$$B_{l,ij}^{\mathcal{E}} = \sum_{k=1}^{n_p} \hat{M}_{l,ik} T_{l,jk}^{\mathcal{E}} \tag{32}$$

$$\hat{M}_{l,ij} = \int_{\hat{e}_l} \hat{N}_i \hat{N}_j \, \mathrm{d}\hat{s} \tag{33}$$

Here, $a_l^{\mathcal{E}} = \mathbf{n}_{\mathcal{E}} \cdot \mathbf{n}_{e_l}$ accounts for the current element's outer normal orientation relative to the unique global orientation assigned to the boundary face.

For each $\mathcal{E} \in \mathcal{M}^h$, we can define element degree of freedom vectors $\boldsymbol{\phi}_{\mathcal{E}}$, $\mathbf{g}_{\mathcal{E}}^{\hat{x}}$, and $\mathbf{g}_{\mathcal{E}}^{\hat{y}}$ in $\mathbb{R}^{n_p}$ and for each element boundary, e_l, a vector of numerical flux degrees of freedom, $\mathbf{f}_l^{up} \in \mathbb{R}^{\tilde{n}_p}$. We then have the local elemental equation

$$\widehat{\mathbf{M}}\frac{\partial \boldsymbol{\phi}_{\mathcal{E}}}{\partial t} = \mathbf{D}^{\hat{x}}\mathbf{g}_{\mathcal{E}}^{\hat{x}} + \mathbf{D}^{\hat{y}}\mathbf{g}_{\mathcal{E}}^{\hat{y}} - a_l^{\mathcal{E}} \sum_{l=1}^{d+1} \frac{|e_l|}{|\det J|}\mathbf{B}_l^{\mathcal{E}}\mathbf{f}_l^{up} \tag{34}$$

APPENDIX B. P^2 LOCAL MATRICES FOR TRIANGLES: Finally, we reproduce the local element matrices for a triangular mesh with a P^2 local approximation space. The barycentric coordinates on $\hat{\mathcal{E}}$ are

$$\begin{aligned} \lambda_0 &= 1 - \hat{x} - \hat{y} \\ \lambda_1 &= \hat{x} \\ \lambda_2 &= \hat{y} \end{aligned} \tag{35}$$

The P^2 shape functions are then

$$\begin{aligned} \hat{N}_0 &= \lambda_0(2\lambda_0 - 1) \\ \hat{N}_1 &= \lambda_1(2\lambda_1 - 1) \\ \hat{N}_2 &= \lambda_2(2\lambda_2 - 1) \\ \hat{N}_3 &= 4\lambda_0\lambda_1 \\ \hat{N}_4 &= 4\lambda_1\lambda_2 \\ \hat{N}_5 &= 4\lambda_0\lambda_2 \end{aligned} \tag{36}$$

The nodal locations associated with $\{\hat{N}_i\}$, $i = 0, \ldots, 5$ are illustrated in Figure 6. Note that, in the matrix definitions below, the edges are numbered counterclockwise around $\hat{\mathcal{E}}$ starting with the $\hat{x}$ axis.

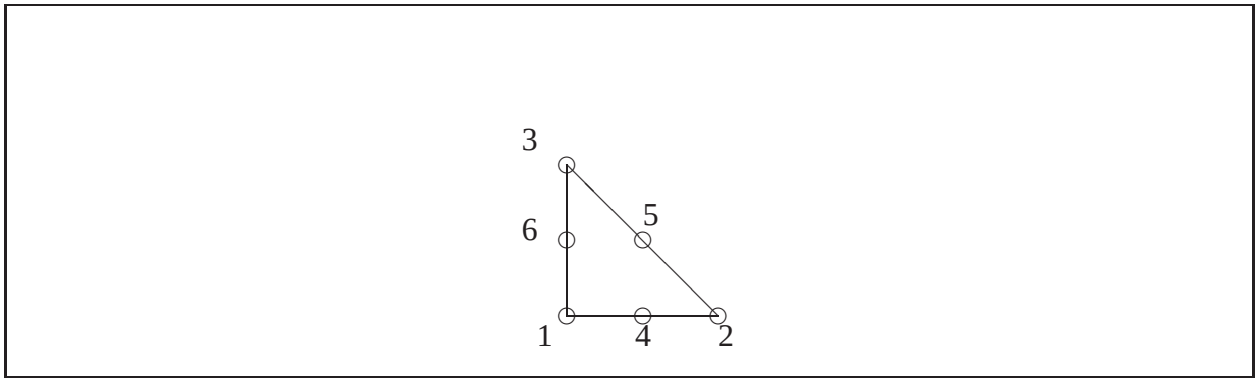

Figure 6. Standard Lagrangian nodal locations on reference triangle $\hat{\mathcal{E}}$ for P^2

$$\widehat{\mathbf{M}} = \begin{bmatrix} \frac{1}{60} & -\frac{1}{360} & -\frac{1}{360} & 0 & -\frac{1}{90} & 0 \\ -\frac{1}{360} & \frac{1}{60} & -\frac{1}{360} & 0 & 0 & -\frac{1}{90} \\ -\frac{1}{360} & -\frac{1}{360} & \frac{1}{60} & -\frac{1}{90} & 0 & 0 \\ 0 & 0 & -\frac{1}{90} & \frac{4}{45} & \frac{2}{45} & \frac{2}{45} \\ -\frac{1}{90} & 0 & 0 & \frac{2}{45} & \frac{4}{45} & \frac{2}{45} \\ 0 & -\frac{1}{90} & 0 & \frac{2}{45} & \frac{2}{45} & \frac{4}{45} \end{bmatrix} \tag{37}$$

$$\mathbf{D}^{\hat{x}} = \begin{bmatrix} -\frac{1}{15} & \frac{1}{30} & \frac{1}{30} & -\frac{1}{10} & \frac{1}{30} & -\frac{1}{10} \\ -\frac{1}{30} & \frac{1}{15} & -\frac{1}{30} & \frac{1}{10} & \frac{1}{10} & -\frac{1}{30} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{10} & -\frac{1}{10} & 0 & 0 & -\frac{2}{15} & \frac{2}{15} \\ -\frac{1}{30} & -\frac{1}{30} & \frac{1}{15} & \frac{2}{15} & \frac{4}{15} & \frac{4}{15} \\ \frac{1}{30} & \frac{1}{30} & -\frac{1}{15} & -\frac{2}{15} & -\frac{4}{15} & -\frac{4}{15} \end{bmatrix} \tag{38}$$

$$\mathbf{D}^{\hat{y}} = \begin{bmatrix} -\frac{1}{15} & \frac{1}{30} & \frac{1}{30} & -\frac{1}{10} & \frac{1}{30} & -\frac{1}{10} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{30} & -\frac{1}{30} & \frac{1}{15} & -\frac{1}{30} & \frac{1}{10} & \frac{1}{10} \\ \frac{1}{30} & -\frac{1}{15} & \frac{1}{30} & -\frac{4}{15} & -\frac{4}{15} & -\frac{2}{15} \\ -\frac{1}{30} & \frac{1}{15} & -\frac{1}{30} & \frac{4}{15} & \frac{4}{15} & \frac{2}{15} \\ \frac{1}{10} & 0 & -\frac{1}{10} & \frac{2}{15} & -\frac{2}{15} & 0 \end{bmatrix} \tag{39}$$

$$\widehat{\mathbf{M}}_0 = \begin{bmatrix} \frac{2}{15} & -\frac{1}{30} & 0 & \frac{1}{15} & 0 & 0 \\ -\frac{1}{30} & \frac{2}{15} & 0 & \frac{1}{15} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{15} & \frac{1}{15} & 0 & \frac{8}{15} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{40}$$

$$\widehat{\mathbf{M}}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{15} & -\frac{1}{30} & 0 & \frac{1}{15} & 0 \\ 0 & -\frac{1}{30} & \frac{2}{15} & 0 & \frac{1}{15} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{15} & \frac{1}{15} & 0 & \frac{8}{15} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{41}$$

$$\widehat{\mathbf{M}}_2 = \begin{bmatrix} \frac{2}{15} & 0 & -\frac{1}{30} & 0 & 0 & \frac{1}{15} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{30} & 0 & \frac{2}{15} & 0 & 0 & \frac{1}{15} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{15} & 0 & \frac{1}{15} & 0 & 0 & \frac{8}{15} \end{bmatrix} \tag{42}$$