1998-99 Annual Report to the Texas Water Development Board on Research Activities of the Center for Subsurface Modeling

Clint Dawson, Mary F. Wheeler and Jennifer Proft
The University of Texas at Austin *

October 11, 1999

1 Introduction

The Texas Water Development Board (TWDB) supported research activities at the Center for Subsurface Modeling, The University of Texas at Austin (UT Austin), during the fiscal year 1998-99. The principal investigators on this project were Clint Dawson and Mary F. Wheeler. Jennifer Proft, a graduate student in Computational and Applied Mathematics at UT Austin, was also involved in the project. During this fiscal year, we accomplished the following;

- Included an iterative solver capability in the TWDB codes TxBLEND 2D and TxBLEND 3D.
- Developed a finite volume-based code for transport equations on unstructured, two-dimensional grids.

Below we describe in more detail the methodologies and computer codes developed. All codes and related papers have been delivered to Junji Matsumoto at the TWDB.

2 TxBLEND Modifications

During the past year, we replaced the direct solvers in TxBLEND 2D and TxBLEND 3D with NSPCG iterative routines in order to decrease the overall computational time. NSPCG is a Fortran package to solve large sparse systems of linear equations using iterative methods, developed by the Center for Numerical Analysis at The University of Texas at Austin. These routines as well as a complete description of the iterative methods and their usage can be found online at http://rene.ma.utexas.edu/CNA/NSPCG/

*Center for Subsurface Modeling, SHC 413, University of Texas, Austin, TX 78712

Changes to the TxBLEND code are clearly delineated within the files *txblend.f* (2D version) and *txblend3d.f* (3D version). Some additional variables have been defined for use by the NSPCG routines:

Variables defined in TxBLEND	
iiaz, jjaz	integer arrays for sparse storage to solve Z (H)
iiac, jjac	integer arrays for sparse storage to solve C
zmatrix, cmatrix	real arrays of coefficient matricies in sparse storage
zcoef, ccoef	integer arrays to describe coefficient matricies
iwkspc, iwkspz	integer arrays for NSPCG workspace
nw, inw, mdim	integers of NSPCG-specific parameters
nzz, nzc, maxnzs	integers of NSPCG-specific parameters
ubar,mdim,xp,xip	reals of NSPCG-specific parameters
wksp	real array for NSPCG workspace
iparm	integer array of NSPCG-specific parameters
rparm	real arrays of NSPCG-specific parameters
level	integer to control level of NSPCG feedback
ier	integer to describe error conditions encountered
val	real number to provide initial guess for solution
tokeepc, tokeepz	real arrays to store parameters overwritten
ntp, maxntp	integers to describe tapering elements (3d only)
include	integer array to define matrix assembly (3d only)

Within the TxBLEND code itself, certain lines have been commented out to prevent interference with the iterative solvers, and external routines specific to the NSPCG package have been declared.

In order to replace the direct solvers in the 2D version, construction of parameters and/or calls to iterative solvers are located in three places within the TxBLEND code. Concentration variable C is solved prior to entering the outer loop, and once at the end of each outer loop iteration. The coefficient matrix for surface elevation variable Z (H) is constructed the first time through the outer loop, and thereafter only if the wet/dry condition has changed. Utilizing this decomposition, the Z variable is solved each "Picard" cycle within each outer loop iteration succeeding the construction of the right-hand side.

Likewise, in order to replace the direct solvers in the 3D version, construction of parameters and/or calls to iterative solvers are located in two places within the TxBLEND code. The coefficient matrix for surface elevation variable Z (H) is constructed the first time through the outer loop, and thereafter only if the wet/dry condition has changed. Utilizing this decomposition, the Z variable is solved each "Picard" cycle within each outer loop iteration succeeding the construction of the right-hand side. Concentration variable C is solved for each level of each iteration, providing the bulk of the computational time.

2.1 Additional Subroutines in TxBLEND

The NSPCG subroutines contained in files *nspcg1.f*, *nspcg2.f*, *nspcg3.f*, *nspcg4.f* and *nspcg5.f* have been added to the 2D and 3D TxBLEND packages. These routines will iteratively solve a sparse matrix system using single precision floating point operations. We have incorporated the Jacobi preconditioner and the Orthogonalized Conjugate Gradient Method (ODIR) accelerator to solve the nonsymmetric matricies arising from the calculation of the concentration in the convection-diffusion equation. Similarly, the Jacobi preconditioner and the Conjugate Gradient (CG) accelerator solve the symmetric matricies arising from the calculation of the surface elevation in the generalized wave continuity equation. Any additional subroutines not defined within file *assembler.f* are contained within the NSPCG package.

The file *assembler.f* containing routines *assemblenon*, *assemblesym*, *errorcontrol*, *sbini*, *sbsij*, *sbendn*, *store* and *restore* has been added to the 2D TxBLEND package in order to arrange matrix entries into the sparse storage coordinate format utilized by NSPCG. It effectively replaces the previous routine *assmbl1.f*.

The file *assembler.f* containing routines *includearray*, *geologicalfeatures*, *assemblenon*, *assemblesym*, *errorcontrol*, *sbini*, *sbsij*, *sbendn*, *store* and *restore* has been added to the 3D TxBLEND package in order to arrange matrix entries into the sparse storage coordinate format utilized by NSPCG. It effectively replaces the previous routines *assmbl1.f* and *assmbl3.f*.

Additional Subroutines	
INCLUDEARRAY	Delineates circumstances under which nodes are known, unknown, or defined within a tapering element (3d only)
GEOLOGICALFEATURES	Modifies matrix for known nodes due to river inflow, tidal boundaries, or placement in bathymetry (3d only)
ASSEMBLENON	Assembles local element matrix into nonsymmetric global structure
ASSEMBLESYM	Assembles local element matrix into symmetric global structure
ERRORCONTROL	Handles errors returned from NSPCG
SBINI	Initializes coefficient arrays iia, jja, matrix, and iwksp
SBSIJ	Sets individual entries in the matrix and builds a linked list representation of the global structure
SBENDN	Restructures the linked list into final sparse storage form
STORE	Stores parameters overwritten by NSPCG
RESTORE	Restores parameters overwritten by NSPCG

2.2 TxBLEND Usage

Before calling the NSPCG iterative solver package, the coefficient matrix must first be stored in sparse matrix coordinate format. To do so, the subroutines described in section 3 have been incorporated. The right-hand-side determines the unique solution to the matrix equation.

A single call to NSPCG begins the iterative solution cycle after the parameters have been initialized. For a complete description of the NSPCG calling sequence and parameters, refer to the NSPCG User's Guide at http://rene.ma.utexas.edu/CNA/NSPCG/usernsp

Specifically, parameter arrays *iparm* and *rparm* tailor the iterative solver behavior and may be modified subsequent to being initialized by subroutine DFAULT. In particular, *iparm(3)* controls the level of output provided by NSPCG and can be useful for debugging.

iparm(3)	< 0	no output
	$= 0$	fatal error messages only (default)
	$= 1$	warning messages and minimum output
	$= 2$	reasonable summary
	$= 3$	parameter values and informative comments
	$= 4$	approximate solution after every iteration

Also of use may be *rparm(1)* to control the stopping test value, which may be modified by NSPCG according to the relative machine precision. *rparm(1)* (zeta) is the stopping test value or approximate relative accuracy desired in the final computer solution. Iteration terminates when the stopping test is less than zeta. If the method does not converge in the maximum allowed iterations, zeta is reset to an estimate of the relative accuracy achieved. [Default 10^{-6}]

NSPCG calling parameter *ier* is an error flag to indicate the type of error encountered during iteration and possibly stop execution. A zero value on output indicates no error occurred; a negative value on output indicates a fatal error was detected during the iteration, while a positive value on output indicates a warning error was detected during the iteration. NSPCG will output all errors and warnings to the output file defined by TxBLEND parameter W.

3 Development of a two-dimensional finite volume transport code and application to Galveston Bay simulation

The Runge Kutta Discontinuous Galerkin (RKDG) scheme is a locally conservative finite element (finite volume) method of formal high-order accuracy for advection equations developed in a series of papers by Cockburn and Shu [4, 1, 2, 3, 6, 5]. As described in TICAM report 99-16, Dawson and Proft have applied the method with piecewise linear approximating spaces to transport equations on triangular grids [7]. Here we describe the method, discuss the structure of a code implementing the method, and present numerical results for the code on simulation data from Galveston Bay.

3.1 RKDG Formulation

We consider the following conservation law:

$$\begin{aligned} (\xi c)_t + \nabla \cdot (\mathbf{u}c) = & \quad 0 \qquad on\, \Omega \times (0,T], \\ c(\mathbf{x},0) = & \quad c_0(\mathbf{x}) \quad on\, \Omega \times \{0\} \end{aligned} \tag{1}$$

where Ω is a bounded domain in $I\!R^2$ with Lipschitz boundary $\partial\Omega$. The vector $\mathbf{u}(x,t)$ represents a given velocity field. In porous media applications, ξ is the porosity of the medium, while in shallow water applications it is the water column height. We assume ξ and $\mathbf{u}$ are related by a continuity equation:

$$\xi_t + \nabla \cdot \mathbf{u} = 0. \tag{2}$$

We discretize Ω into triangles K. Following [7], we multiply (1) by a test function $w_h \in W_h = \{w_h \in L^\infty(\Omega) : w_h \mid_K \in P^1(K), \forall K \in \Omega\}$ where $P^1(K)$ denotes the space of polynomials on triangle K of degree at most 1. Integrating by parts over each triangle, we obtain the weak formulation: find $c_h \in W_h$ such that $\forall w_h \in W_h$,

$$\frac{d}{dt}\int_K \xi\, c_h(x,t)\, w_h(x)\, d\mathbf{x} \quad = \quad Ta - Tb, \tag{3}$$

where

$$T_a \quad = \quad \int_K c_h(x,t)\, \mathbf{u}(x,t) \cdot \nabla w_h(x) dx \tag{4}$$

$$T_b \quad = \quad \sum_{e \in \partial K} \int_e c_h(x,t)\, \mathbf{u}(x,t) \cdot \mathbf{n}_{e,K}\, w_h(x)\, d\Gamma, \tag{5}$$

where the summation in T_b is over all edges in the mesh, and $\mathbf{n_{e,K}}$ is the unit outward normal to ∂K. The resulting equation can be put in ODE form and discretized in time using a total variation diminishing Runge-Kutta integration [8, 9]. Because c_h is discontinuous across element boundaries we evaluate c_h on ∂K using upwinding, based on the sign of $\mathbf{u} \cdot \mathbf{n}_{e,K}$. If ∂K is part of the boundary of Ω and the flux $\mathbf{u} \cdot \mathbf{n}$ is negative ($\mathbf{n}$ = outward unit normal to $\partial\Omega$, we specify c_h as an inflow concentration, otherwise this part of the boundary is an outflow/noflow boundary. To further enhance the stability of the method and eliminate possible spurious oscillations in the approximate solution, a local slope limiting operator is introduced in the time-marching algorithm. The resulting form is

$$\frac{d}{dt}(\xi c_h) = M_K^{-1}\Big(T_a - T_b\Big) \equiv L_h(c_h), \tag{6}$$

where M_K^{-1} is the inverse of the mass matrix.

3.2 RKDG method applied to Galveston Bay simulation

The advection scheme described above has been incorporated into a hydrodynamic and transport simulator. In this simulator we have used the ADCIRC code to compute elevations and velocities. The elevations and velocities are then postprocessed so that the discrete continuity equation

$$\int_K \left(\frac{\xi^k - \xi^{k-1}}{\Delta t} + \nabla \cdot \mathbf{u^k} \right) dx = 0 \tag{7}$$

is satisfied over each element K and each hydrodynamic time step $\Delta t = t^k - t^{k-1}$.

The Fortran-90 driver for this simulator is contained in the file *tracer.f90.* The driver reads in appropriate ADCIRC data and calls the RKDG subroutine *hyperbolic* contained in file *hyperbolic.f90.* All other files in this package are supporting routines for the RKDG method. We have tested this simulator on a Galveston Bay data set.

After reading in grid data resulting from ADCIRC, the driver will initialize the concentration of the tracer. For this simulation, the concentration is defined initially to be 0.0 everywhere in the triangular grid except at one element within the shipping channel perturbed to be 1.0 for the duration of the simulation. Simulation parameters specified by the user are then defined before looping over each outer time step. For each step Δt, the driver will compute various parameters before advecting forward in time by calling the RKDG scheme using linear basis functions (*hyperbolic*) and by calling a scheme using constant basis functions (*advect*) for comparison. The resulting RKDG solution for each step is written to a file for graphical output. The driver algorithm is as follows:

```
Read in grid data
Initialize concentration of tracer
Define simulation parameters
Do $k = 1$, numsteps

        Compute parameters

                Read in volume and flux data at time level $t^{k-1}$
                Compute CFL condition for the RKDG scheme
                Check the mass balance

          Advect solutions forward to time level $t^k$

                Call advect, a constant in space scheme
                Call hyperbolic, the RKDG linear in space scheme

          Output solution

                Project RKDG solution to nodal points for tecplot
                Write ascii data to files

    End do
```

Our primary concern is in obtaining valid higher-order results as a solution to the advection equation. The use of linear basis functions discontinuous across each element in the RKDG method will produce less diffusive, more accurate results than will the scheme utilizing constant basis functions. For stability, the advective time step must satisfy a CFL condition. Thus, although the user has control over how long the simulation will run (Δt*numsteps*), the inner loop time step is defined within the program itself. Note that the outer loop time step Δt is defined by the ADCIRC data.

File *hyperbolic.f90* contains the subroutine *hyperbolic* defining the RKDG method. The subroutine is passed an initial time $t0 = t^{k-1}$ and loops over each inner time step until final time $tf = t^k$. The subroutine will initialize some structures and L^2 project the initial condition from P^0, the space of polynomials of degree at most zero, to P^1, the space of polynomials of degree at most one, before begining the Runge-Kutta integration loop over each inner advecting time step. The solution for each inner time step is a weighed averaging of approximations evaluated at the begining of each step and halfway between each step. Upon completion of the inner loop, the integral average over each element is returned to the driver for output. The RKDG algorithm is as follows:

```
Initialize structures
$L^2$ project initial condition to $P^1$
Slopelimit approximation $c_h$
Do from $t0$ to $tf$

        Compute $L_h$ at beginning of time step

                Compute term $Ta$
                Compute term $Tb$
                Multiply by $M_K^{-1}$

        Obtain first approximation $c_h$
        Slopelimit approximation $c_h$
        Compute $L_h$ half way through time step

                Compute term $Ta$
                Compute term $Tb$
                Multiply by $M_K^{-1}$

        Obtain second approximation $c_h$
        Slopelimit approximation $c_h$

    End do
```

3.3 RKDG Useage

This package entails twelve Fortran-90 files, a makefile for compilation, and ADCIRC data files:

RKDG Files	
tracer.f90	Contains driver routine and P^0 solution routine
out_tecplot.f90	Contains routines to project solution to nodal points and output in Tecplot FEM format
val_par.f90	Contains definitions for global values and parameters
hyperbolic.f90	Contains RKDG driver routine to compute P^1 solution
l2proj_slopelmt.f90	Contains L^2 projection and slopelimiting routines
computeLh.f90	Contains routine for computing $L_h(c_h, w_h)$
computeTa.f90	Contains routine for computing term Ta defined over interior
computeTb.f90	Contains routine for computing term Tb defined over edges
methods.f90	Contains supporting routines for the RKDG scheme
userDefined.f90	Contains specified boundary conditions
integrate.f90	Contains supporting routines for all integration
flux.f90	Contains definitions of velocity and flux terms
adcgrid.dat	ADCIRC grid data
mcflux.dat	Mass conservative flux data
elemvols.dat	Element volume data
watcol.dat	Water column data

With appropriate parameters defined by the user, the program will output the concentration values at each nodal point in the mesh for each outer loop timestep to file *output.plt*. The solution is in Tecplot FEM format and can be immediately read into Tecplot for simulation. To modify the boundary conditions, the user may set Dirichlet conditions for particular boundary elements via subroutines *setBoundaryElements* and *forceBC* in file *userDefined.f90*. The default is to assume zero inflow concentration.

Within file *tracer.f90*, the user defines simulation parameters to control the amount of feedback during a run as well as the length of time to compute. These user specified parameters include:

Parameters	
numsteps	Integer number of outer time steps to be taken during simulation period where finaltime = Δt*numsteps (max 192)
callhyp	Logical flag to call RKDG scheme for solution
calladv	Logical flag to call P^0 scheme for solution
feedback	Logical flag to display current solution at each outer time step to standard out
tol	Real value to define the tolerance of the lowest value displayed as feedback (does not affect solution written to *output.plt*

The current problem set to run in the RKDG package is defined by a zero initial condition and tracer injection to one element within the shipping channel. This problem can be changed by modifying files *hyperbolic.f90* and *tracer.f90*. Results presented below for this problem demonstrate the solution after 6, 12, 18 and 24 hours of simulated time. The tracer presents a rather sharp front as it flows into the bay with the current, drifts back through the shipping channel, and begins to spread further into the bay area with the current.

References

[1] B. COCKBURN and C.W. SHU. Tvb runge-kutta local projection discontinuous galerkin finite element method for scalar conservation laws ii: general framework. *Math. Comp.*, 52:411–435, 1989.

[2] B. COCKBURN and C.W. SHU. Tvb runge-kutta local projection discontinuous galerkin finite element method for scalar conservation laws iii: one dimensional systems. *Comput. Phys.*, 84:90–113, 1989.

[3] B. COCKBURN and C.W. SHU. Tvb runge-kutta local projection discontinuous galerkin finite element method for scalar conservation laws iv: the multidimensional case. *Math. Comp.*, 54:545–581, 1990.

[4] B. COCKBURN and C.W. SHU. The runge-kutta local projection p^1-discontinuous galerkin method for scalar conservation laws. M^2AN, 25:337–361, 1991.

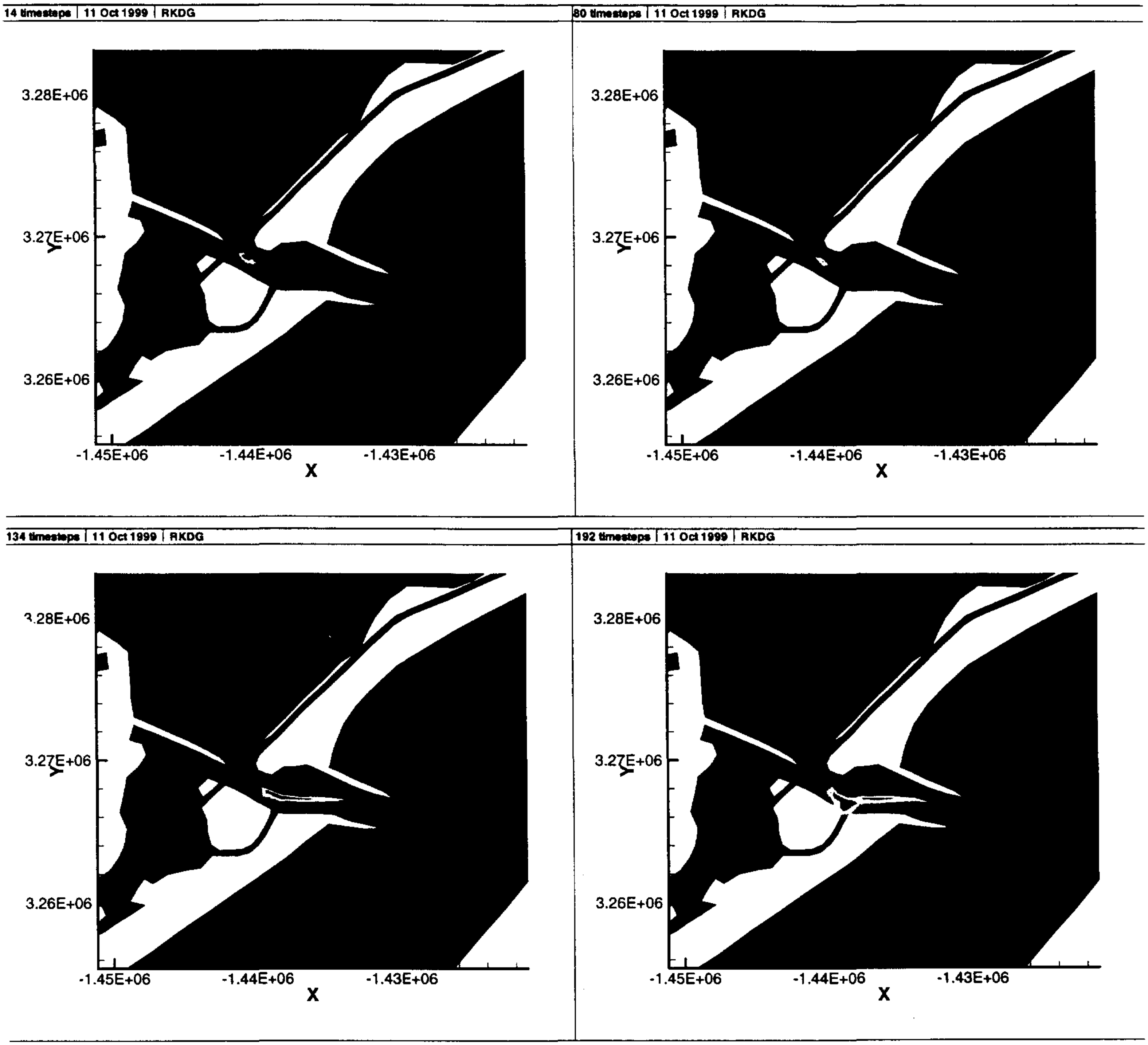
[5] B. COCKBURN and C.W. SHU. The local discontinuous galerkin method for time-dependent convecton-diffusion systems. *SIAM J. Numer. Anal., submitted*, 1997.

[6] B. COCKBURN and C.W. SHU. Tvb runge-kutta local projection discontinuous galerkin finite element method for scalar conservation laws v: multidimensional systems. *J. of Comput. Phys.*, 141:199–224, 1998.

[7] C. DAWSON and J. PROFT. Adaptive stencil and discontinuous galerkin methods for transport equations on triangular grids. *TICAM Report 99-16*, May 1999.

[8] C.W. SHU. Total-variation-diminishing time discretizations. *SIAM J. Sci. Stat. Comput.*, 9:1073–1084, 1988.

[9] C.W. SHU and S. OSHER. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.


14 timesteps | 11 Oct 1999 | RKDG
3.28E+06
3.27E+06
3.26E+06
Y
-1.45E+06
-1.44E+06
-1.43E+06
X
80 timesteps | 11 Oct 1999 | RKDG
3.28E+06
3.27E+06
3.26E+06
Y
-1.45E+06
-1.44E+06
-1.43E+06
X
134 timesteps | 11 Oct 1999 | RKDG
3.28E+06
3.27E+06
3.26E+06
Y
-1.45E+06
-1.44E+06
-1.43E+06
X
192 timesteps | 11 Oct 1999 | RKDG
3.28E+06
3.27E+06
3.26E+06
Y
-1.45E+06
-1.44E+06
-1.43E+06
X

Adaptive Stencil and Discontinuous Galerkin Methods
for Transport Equations on Triangular Grids

by

Jennifer Kay Proft, M.S.C.A.M.
The University of Texas at Austin, 1999
SUPERVISOR: Clint Dawson

Abstract

In this report we describe two higher order methods for solving first-order linear and nonlinear hyperbolic equations on two dimensional domains. Numerical results for constant and variable coefficient linear advection problems are presented.

1 Introduction

In this report we describe numerical methods for solving first-order linear and nonlinear hyperbolic equations on two dimensional domains discretized into triangular elements. There are special difficulties associated with solving these unsteady, time-dependent problems such as spontaneous shock formation and steep gradients even in the presence of smooth initial conditions.

We focus on high-resolution schemes for advection problems that achieve higher-order accuracy while suppressing spurious oscillations. Implementation of an operator splitting technique renders these schemes useful for the solution of advection-dominated diffusion equations as well.

In particular, we investigate two locally conservative second order methods for the solution of hyperbolic conservation laws by comparing numerical convergence rates and by examining the results of a variable coefficient linear advection problem. An Adaptive Flux Stencil finite volume method developed by Durlofsky, Engquist, and Osher [22] is compared to a Runge-Kutta Discontinuous Galerkin finite element method developed in a series of papers by Cockburn and Shu [12, 9, 10, 11, 14, 13].

This report is outlined as follows. In the next section we discuss physical models for advection-diffusion equations and hyperbolic conservation laws. We also discuss splitting techniques and present a model advection problem. In section 3 we describe the Discontinuous Galerkin method, and in section 4 we describe the Adaptive Flux Stencil method. Finally, in section 5 we present some numerical results for the methods applied to two-dimensional test problems.

1.1 Background

One of the earliest so called "high-resolution" schemes was introduced by Boris and Book in [5] in the context of a flux-limiting algorithm, and a wide class was analyzed by Sweby [37]. Leveque defines these methods as possessing the following features [30]:

- At least second order accuracy on smooth solutions and in smooth regions of a solution even when discontinuities are present elsewhere.
- Sharp resolution of discontinuities without excessive smearing.
- The absence of spurious oscillations in the computed solution.

- An appropriate form of consistency with the weak form of the conservation law, required if we hope to converge to weak solutions.
- Nonlinear stability bounds that, together with consistency, allow us to prove convergence as the grid is refined.
- A discrete form of the entropy condition, allowing us to conclude that the approximations converge to the physically correct weak solution.

The development of high-resolution methods is non-trivial due to the mathematical and numerical difficulties of these types of problems. Because discontinuities may exist in the solution, we work with a weak form of the conservation law. However, it becomes necessary to apply physical knowledge of the problem in order to obtain the correct entropy-satisfying solution, since there often exists more than one weak solution to the conservation law with the same initial data. Straightforward application of first-order accurate methods generally result in a solution profile that is very smeared in regions near a discontinuity, while standard second-order accurate methods result in solutions that are highly oscillatory near the discontinuity. Subsequently, methods which are at least second order accurate and limit the oscillation of the solution were constructed.

The original development of successful slope- and flux-limiting methods was in one space dimension [28, 32, 37]. Multidimensional methods must incorporate additional behavior arising in two and three-dimensional problems. Typically, the geometry can be complicated and the structure of the discontinuities is more complex in these problems. The Riemann problem in one dimension, for example, can be solved in closed form, and discontinuity curves in the (x, t) plane are simply straight lines passing through the origin. However, in two dimensions only some special cases of the Riemann problem have been solved analytically.

A class of numerical methods developed in [25, 26, 27, 35, 36], called essentially non-oscillatory schemes (ENO), construct a piecewise linear interpolant of the flux function that is the least oscillatory over various choices of interpolating polynomials. This is the main idea behind the Triangular Based Adaptive Flux Stencil scheme developed by Durlofsky *et al* in [22] that we examine. Durlofsky subsequently applied this technique to the modeling of two phase flow through porous media in [21].

Recently, other multidimensional extensions of one-dimensional algorithms have been developed in, for example, [4, 15]. Alternative flux-limiting methods that possess a total variation diminishing or total variation boundedness property (TVD, TVB) as described below are a current research topic as well. In particular, we examine the Runge-Kutta Discontinuous Galerkin method developed in a series of papers by Cockburn and Shu [12, 9, 10, 11, 14, 13].

LeSaint and Raviart [29], first introduced the Discontinuous Galerkin finite element method for solving the neutron transport equation, which is a linear version of the hyperbolic conservation law. The method was rendered explicit in time by Chavent and Salzano [7] and was applied to hyperbolic problems by Chavent and Cockburn [6]. Subsequent papers by Cockburn and Shu generalized this scheme to systems and to multidimensions while incorporating a practical slope limiter. We will examine the Runge-Kutta Discontinuous Galerkin method developed by Cockburn and Shu for advection problems and compare it to the Adaptive Flux method developed by Durlofsky, Engquist and Osher.

1.2 Description of Results

We implement these two methods and compare our results on two test problems. The Discontinuous Galerkin method demonstrates convergence rates approaching 1.55 in the L^2 norm and 1.86 in the L^1 norm for a linear advection problem. The Adaptive Flux method demonstrates convergence rates approaching 1.60 and 1.72 respectively. For both methods, the results of the rotating cone problem exhibit accurate and well-resolved solution profiles for 2048 elements.

Both methods are straightforward to implement numerically and give good results insofar as they approach the optimal rate of convergence and accurately resolve a solution profile with sharp gradients. The computational efficiency of one method over another is not significantly distinguishable. The extremely local computational domain render them suitable for efficient parallel implementation. The methods can easily handle complicated geometries as they are applicable to unstructured triangular meshes. The main advantage of these methods over others is their high parallelizability and higher order accuracy, rendering them suitable for advection-dominated flow computations.

2 Preliminaries

2.1 Notation and Definitions

Here we define the notation we used throughout this report. Let Ω be a polygonal bounded domain in $I\!R^d, d = 1, 2$, or 3 with Lipschitz boundary. Recall the usual Sobolev space definitions:

- $L^p(\Omega) = \{v : \Omega \to I\!R : (\int_\Omega |v(x)|^p dx)^{1/p} < \infty\}$;
- $L^\infty(\Omega) = \{v : \Omega \to I\!R : max_{x\in\Omega}|v(x)| \leq \infty\}$;
- $L^1_{loc}(\Omega) = \{v : v \in L^1_K \, \forall \, compact \, K \subset int \, \Omega\}$;
- $W^{m,p}(\Omega) = \{v : v \in L^1_{loc}(\Omega) : \|f\|_{W^{m,p}(\Omega)} < \infty\}$;

and the norms associated with these spaces:

- $\|f\|_{L^p(\Omega)} = (\int_\Omega |f(x)|^p dx)^{1/p}$;
- $\|f\|_{L^2(\Omega)} = (\int_\Omega |f(x)|^2 dx)^{1/2}$;
- $\|f\|_{L^\infty(\Omega)} = \text{ess sup}_{x\epsilon\Omega}|f(x)|$;
- $\|f\|_{W^{m,p}(\Omega)} = (\sum_{|\alpha|\leq m} \|D^\alpha f\|^p_{L^p(\Omega)})^{1/p}, \quad p < \infty$;

Define the time-space norms:

- $\|f\|_{L^q(L^p)} = \left(\int_0^T \|f(\cdot,t)\|^q_{L^p(\Omega)} dt\right)^{1/q}$;
- $\|v\|_{L^\infty(L^1)} = \max_{0\leq t\leq T} \|v(\cdot,t)\|_{L^1}$;

Define the $L^2(\Omega)$ inner product as:

$$(f,g)_\Omega = \int_\Omega fg \; dx.$$

To distinguish integration over lines, we define:

$$< f, g >_{\partial\Omega} = \int_{\partial\Omega} fg \; dx.$$

Recall the Cauchy-Schwartz inequality:

$$\int |vw| \leq \|v\| \cdot \|w\|,$$

and Young's inequality:

$$ab \leq \frac{1}{2\varepsilon}a^2 + \frac{\varepsilon}{2}b^2, \quad a, b, \varepsilon \in I\!R, \varepsilon > 0.$$

2.2 Physical Models

Numerous physical applications give rise to advection-diffusion equations. Operator splitting techniques such as those discussed in the next section render analysis of advection equations pertinent to the analysis of advection-diffusion equations as well. The standard transport equation is an advection-diffusion equation based on conservation of mass [3]. Advection and hydrodynamic dispersion control the flux into and out of an elemental volume, while source/sink terms of solute mass within a volume are the result of chemical or biochemical reactions. The physical process of advection is solute movement attributed to transport by flow; the process of hydrodynamic dispersion is a result of mechanical mixing and molecular diffusion [23]. This model arises in transport phenomena applications such as secondary oil recovery, tumor cell growth, and contaminant transport.

For example, the principal differential equation that describes transport of dissolved reactive constituents in saturated isotropic porous media is [3, 19]:

$$\phi u_t + \nabla \cdot (\mu u - D\nabla u) = \phi f + q\tilde{u}, \quad (\mathrm{x}, t) \in \Omega \times (0, T], \tag{1}$$

for Ω a bounded domain in $I\!R^d$, $d = 1$, 2, or 3, $T > 0$, where

$$\tilde{u} = \begin{cases} u_I & q > 0, \\ u & q < 0 \end{cases} \tag{2}$$

and we define q to be a sum of point sources q^+ and sinks q^- (Dirac measures). In porous media applications, u represents the concentration of some chemical component, ϕ is the porosity of the medium and may also include adsorption effects, μ is the Darcy velocity and D is a diffusion/dispersion tensor, assumed to be symmetric and at worst positive semi-definite.

In this report we are primarily concerned with the analysis and approximation of the advection equation modeling hyperbolic conservation laws. Additional physical processes that give rise to these equations include gas dynamics and fluid dynamics. In particular, the Euler equations of gas dynamics model interesting physical problems such as the shock tube problem, the flow of air around a vehicle, meteorology weather-prediction problems, and are related to the shallow water equations.

2.3 Splitting Techniques

Here we describe time-splitting techniques which can be applied to advection-diffusion-type equations, whereby advection and diffusion are approximated by different solution procedures. One particular splitting for the transport equation results in the approximation of a hyperbolic and a parabolic equation at each time step: given $U^n \in W_h \in L^2(\Omega)$, solve the hyperbolic equation

$$\phi \hat{u}_t + \nabla \cdot (\mu \hat{u}) = \phi f + q^+ u_I \quad on\, \Omega \times (t^n, t^{n+1}]. \tag{3}$$

The solution generated at this step, $\hat{U}$, is the initial condition for the parabolic equation

$$\begin{aligned} u_t^* + \nabla \cdot g^* &= q^- u^* \quad on\, \Omega \times (t^n, t^{n+1}] \\ g^* &= -D \nabla u^*. \end{aligned} \tag{4}$$

The solution generated here will approximate $u^{n+1} = u(t^{n+1})$.

A number of methods for advection-dominated diffusion equations have been based on the above splitting. For example, Douglas and Russell developed the Modified Method of Characteristics [20] based on combining a characteristic method for equation (3) with a Galerkin finite element method for (4).

In [16, 17, 18], Dawson examined Upwind-Mixed methods to explicitly approximate the advective terms using an upwind method and implicitly approximate diffusive terms using a mixed finite element method. Recently, Dawson and Aizinger [19] extended this analysis by applying the Discontinuous Galerkin method developed by Cockburn and Shu [11] and the Enhanced Mixed finite element method developed by Arbogast *et al* [1] to the transport equation. They analyzed the standard transport equation utilizing higher order approximating spaces, a positive semi-definite diffusion coefficient, and physically realistic boundary conditions.

Recently, Arbogast and Wheeler developed the Characteristic Mixed method in [2], based on combining a characteristic method for equation (3) with a mixed method for equation (4). Additionally, Cockburn and Shu have developed the local Discontinuous Galerkin method which approximates the advection and the diffusion equation by different Discontinuous Galerkin procedures [13].

2.4 Model Advection Problem

In this report we focus on analysis and approximation of the advection equation specifically; to this end, we define the model advection problem. Consider the following hyperbolic conservation law:

$$\begin{aligned} u_t + \nabla \cdot \mathbf{f}(u) &= 0 \quad on\, \Omega \times (0,T], \qquad (5) \\ u(\mathbf{x},0) &= u_0(\mathbf{x}) \quad on\, \Omega \times \{0\} \\ u(\mathbf{x},t) &= \gamma \quad on\, (0,T) \times \partial\Omega_- \end{aligned}$$

where Ω is a polygonal bounded domain in $I\!R^2$ with Lipschitz boundary and $\gamma \in L^\infty((0,T] \times \partial\Omega)$. Let $\mathbf{n}$ denote the unit outward normal to $\partial\Omega = \partial\Omega_- \cup \partial\Omega_+$, where $\partial\Omega_-$ denotes the inflow boundary and $\partial\Omega_+$ denotes the outflow boundary.

3 Runge-Kutta Discontinuous Galerkin Advection Approach

The Runge-Kutta Discontinous Galerkin method is a highly parallelizable method of formal high-order accuracy and efficiency for advection equations. It is discontinuous in both time and in space, and is applicable to rectangular and triangular structured and unstructured grids. This method is formally $(k+1)$th order accurate for piecewise polynomials of degree k utilized to define the finite element space, and is entropy-satisfying total variation bounded (TVB), hence converges to the correct weak solution.

3.1 Formulation

To formulate the method we follow [11] and first discretize (5) in space via the Discontinuous Galerkin finite element method. The resulting equation can be put in ODE form as $\frac{d}{dt}u_h = L_h(u_h, \gamma_h)$ and discretized in time using the TVD Runge-Kutta integration introduced in [12]. Finally, a local projection $\Lambda\Pi_h$ is applied to the intermediate values of the Runge-Kutta time discretization in order to enforce stability.

We define some notation to be used in this section. Let $\mathcal{F} = \{\tau_h\}$ be a family of regular triangulations K discretizing Ω. That is, there exists a constant $\sigma > 0$ such that

$$\frac{h_K}{\rho_K} \geq \sigma, \quad \forall K \in \tau_h, \quad \forall \tau_h \in \mathcal{F},$$

where h_K is the diameter of K, and ρ_K is the diameter of the largest ball included in K.

Let $|e|$ denote the length of edge e and $|K|$ denote the area of triangle K. Let $\{t^n\}_{n=1}^N$ be a partition of $(0, T]$ where $\Delta t^n = t^{n+1} - t^n, n = 0, \ldots, N-1$. We define u_h^- as the value obtained from the interior of the element K and u_h^+ as the value obtained from the exterior of the element K:

$$\begin{aligned} u_h^- &= lim_{s\to 0^-}\, u(\mathbf{x} + s\mathbf{n}_{e,K}) \\ u_h^+ &= \begin{cases} \gamma_h(x,t) & if\, \mathbf{x} \in \partial\Omega, \\ lim_{s\to 0^+}\, u(\mathbf{x} + s\mathbf{n}_{e,K}) & otherwise \end{cases} \end{aligned} \tag{6}$$

where $\mathbf{n}_{e,K}$ denotes the outward unit normal to the edge $e \in \partial K$.

Define the finite dimensional space

$$V_h = V_h^k = \{v \in L^\infty(\Omega) : v\,|_K \in P^k(K), \forall K \in \tau_h\} \tag{7}$$

where $P^k(K)$ denotes the space of polynomials in K of degree at most k. Denote by ∂V_h the space of functions of $L^\infty(\partial\Omega)$ which are traces of functions of $V_h(\Omega)$.

Multiplying by test functions v and integrating by parts, we obtain the following weak formulation of (5): find $u \in H^1(K)$ such that $\forall v \in H^1(K)$,

$$\frac{d}{dt}\int_K u(x,t)\, v(x)\, dx - \int_K \mathbf{f}(u(x,t)) \cdot \nabla v(x)\, dx + \sum_{e\in\partial K} \int_e \mathbf{f}(u(x,t)) \cdot \mathbf{n}_{e,K}\, v(x)\, d\Gamma = 0, \tag{8}$$

We replace the integrals by appropriate quadrature rules. The quadrature rule for the interior of the element should be exact for polynomials of degree $2k$ and the edge rule should be exact for polynomials of degree $2k+1$ for k the degree of the approximation space P^k. Theorem 3.1 justifies the use of these quadrature rules to preserve the $\mathcal{O}(k+1)$th accuracy of the method:

$$\int_K \mathbf{f}(u(\mathbf{x},t)) \cdot \nabla v(\mathbf{x})\, dx \approx \sum_{l=1}^{L} \omega_l\, \mathbf{f}(u(\mathbf{x}_{K,l}, t)) \cdot \nabla v(\mathbf{x}_{K,l})\, |K|. \tag{9}$$

$$\int_e \mathbf{f}(u(\mathbf{x},t)) \cdot \mathbf{n}_{e,K}\, v(\mathbf{x})\, d\Gamma \approx \sum_{l=1}^{L} \omega_l\, \mathbf{f}(u(\mathbf{x}_{e,l}, t)) \cdot \mathbf{n}_{e,K}\, v(\mathbf{x}_{e,l})\, |e|, \tag{10}$$

We replace the flux function $\mathbf{f}(u(\mathbf{x},t))\cdot\mathbf{n}_{e,K}$ by the numerical flux function $h_{e,K}(\mathbf{x},t) = h_{e,K}(u_h(\mathbf{x}^-,t), u_h(\mathbf{x}^+,t))$, where $h_{e,K}$ is any two-point Lipschitz flux which is monotone in the scalar case and is an exact or approximate Riemann solver in the system case. It is consistent with $\mathbf{f}(u)\cdot\mathbf{n}_{e,K}$,

$$h_{e,K}(u,u) = \mathbf{f}(u)\cdot\mathbf{n}_{e,K}, \tag{11}$$

and conservative,

$$h_{e,K}(u^-,u^+) + h_{e,K'}(u^-,u^+) = 0, \quad K^{'}\cap K = e. \tag{12}$$

The semidiscrete weak formulation of (5) is: find $u_h \in V_h$ such that for $v_h \in V^h$,

$$\frac{d}{dt}\int_K u_h(\mathbf{x},t)\,v_h\,dx - \sum_{l=1}^{L}\omega_l\,\mathbf{f}(u_h(\mathbf{x}_{K,l},t))\cdot\nabla v_h(\mathbf{x}_{K,l})\,|K|$$
$$+\sum_{e\in\partial K}\sum_{l=1}^{L}\omega_l\,h_{e,K}(\mathbf{x}_{e,l},t)\,v_h(\mathbf{x}_{e,l})\,|e|. = 0 \quad \forall v_h\in V_h, \quad \forall K\in\tau_h \tag{13}$$

After inverting the mass matrix analytically, these equations can be rewritten in ODE form as $\frac{d}{dt}u_h = L_h(u_h,\gamma_h)$, where γ_h is the L^2 projection of γ into ∂V_h:

$$L_h : \Upsilon_h \times \partial\Upsilon_h \rightarrow V_h, \frac{d}{dt}(u_h(t),v_h) = (L_h(u_h,\gamma_h),v_h), \quad \forall v_h\in V_h,\ t\in(0,T], \tag{14}$$

for $\Upsilon_h = \{w:\Omega\rightarrow I\!R : w|_K \in C^0(K), \forall K\in\tau_h\}$, and $\partial\Upsilon_h = \{w:\partial\Omega\rightarrow I\!R : w|_e \in C^0(e), \forall e\in\partial\tau_h\}$, where $(\cdot,\cdot)$ is the $L^2(\Omega)$ inner product.

We discretize in time this system with a method that is at least $(k+1)$th order accurate to preserve the order of convergence of the finite element method. To do so, we use the TVD Runge-Kutta time discretization introduced in [34, 35]. To enhance the stability of the method and eliminate possible spurious oscillations in the approximate solution, a local slope limiting operator $\Lambda\Pi_h$ is introduced in the time-marching algorithm as follows. Let $\hat{P}_h$ denote the L^2-projection into the space V_h:

- Set $u_h^0 = \hat{P}_h(u_0)$;
- For $n = 0,\ldots,N-1$ compute u_h^{n+1} as follows:
  - Set $u_h^{(0)} = u_h^n$;

- For $i = 1, \ldots, m+1$ compute the intermediate values:

$$u_h^{(i)} = \Lambda\Pi_h \left(\sum_{l=0}^{i-1} \alpha_{i,l} u_h^{(l)} \; + \; \beta_{i,l} \delta t^n L_h(u_h^{(l)}, \gamma_h(t^n + d_l \delta t^n)) \right);$$

- Set $u_h^{n+1} = u_h^{(m+1)}$

where m = order of approximation. The slope-limiting operator $\Lambda\Pi_h$ is constructed in such a way that the following properties are satisfied:

- Local conservation of mass: for every element K of triangulation τ_h,

$$\int_K \Lambda\Pi_h u_h = \int_K u_h.$$

- Limiting: on each element K, the gradient of $\Lambda\Pi_h$ is no larger than that of u_h.

We discuss a maximum principle that this operator satisfies in section 3.3.

Remark: In the case of piecewise constants, the Discontinous Galerkin method reduces to a finite volume, monotone scheme in the scalar case. Thus, the discretization by this method can be considered as a high-order accurate extension of finite volume, monotone schemes.

3.2 Error Estimates

The operator L_h is a discrete approximation of $-\nabla \cdot \mathbf{f}(u)$, one that defines the construction of a higher-order approximate solution. In the methods we examine, the quality of the approximation to the solution of the hyperbolic conservation law differs only in this construction. For polynomials of degree k, we expect a formal accuracy of $\mathcal{O}(k+1)$ for the Discontinuous Galerkin method. This order of accuracy is dependent upon the error in the constructed approximation L_h. Thus, we demonstrate the estimate of the error in the constructed terms as developed by Cockburn and Shu in [11]. This estimate also justifies our use of the above quadrature rules. Our objective is to give sufficient conditions on the quadrature scheme which ensure that the effect of numerical integration does not decrease the formal convergence order of the method.

In the estimate we will utilize the following tools:

Lemma 3.1 *Let $E_\Omega(\phi) = \int_\Omega \phi - \sum_{l=1}^{L} \omega_l \, \phi(\mathbf{x}_l) \, |\Omega|$, and suppose that $E_\Omega(\phi) = 0, \forall \phi \in P^r(\Omega)$. Then $\exists$ a constant C such that $\forall \psi \in P^s(\Omega)$,*

$$|E_\Omega(g\psi)| \leq C|\Omega|h^{r+1-s}|\psi|_{L^\infty(\Omega)}|g|_{W^{r+1-s,\infty}(\Omega)}, \tag{15}$$

where $h = diam(\Omega)$.

Proof: A consequence of the Bramble-Hilbert Lemma [8].

Lemma 3.2 *Let $\hat{P}_h$ denote the L^2 projection into the space $P^k(K)$ where $(\hat{P}_h g - g, v) = 0$ for $g, v \in V$. Assuming the triangulation τ_h is regular,*

$$||\hat{P}_h g - g|| \leq C(k)h^{k+1}|g|_{W^{k+1,\infty}}. \tag{16}$$

Proof: A well known approximation theory result [8].

Lemma 3.3 *For $w_h \in V_h(K)$,*

$$|w_h|^2_{L^\infty(K)} \leq C\frac{1}{|K|}|w_h|^2_{L^2(K)}, \tag{17}$$

where C depends on the dimension of $V_h(K)$ which is assumed to be uniformly bounded from above.

Proof: A consequence of equivalence of norms in finite dimensional spaces [8].

We now present the estimate of the quality of the approximation, L_h defined in (14).

Theorem 3.1 *Let $\mathbf{f}(u) \in W^{k+2,\infty}(\Omega)$, and set $\gamma = trace(u)$. Assume that the family of triangulations $\mathcal{F} = \{\tau_h\}$ is regular. That is, that there exists a constant $\sigma > 0$ such that*

$$\frac{h_K}{\rho_K} \geq \sigma, \quad \forall K \in \tau_h, \quad \forall \tau_h \in \mathcal{F},$$

where h_K is the diameter of K, and ρ_K is the diameter of the largest ball included in K. Then, if $P^k(K) \subset V(K), \quad \forall K \in \tau_h$:

$$\| \, L_h(u,\gamma) + \nabla \cdot \mathbf{f}(u) \, \|_{L^\infty(\Omega)} \leq Ch^{k+1}|\mathbf{f}(u)|_{W^{k+2,\infty}}.$$

Proof: We follow [11]:

$$\begin{aligned}
&\|L_h(u_h,\gamma_h) + \nabla\cdot\mathbf{f}(u)\|_{L^\infty(K)} && (18)\\
&\quad\le\ \|\nabla\cdot\mathbf{f}(u) - \hat{P}_h(\nabla\cdot\mathbf{f}(u))\|_{L^\infty(K)}\ +\ \|L_h(u,\gamma)\ +\ \hat{P}_h(\nabla\cdot\mathbf{f}(u))\|_{L^\infty(K)}
\end{aligned}$$

We bound the first term of the right-hand side via (16)

$$\|\nabla\cdot\mathbf{f}(u) - \hat{P}_h(\nabla\cdot\mathbf{f}(u))\|_{L^\infty(K)} \le Ch^{k+1}|\nabla\cdot\mathbf{f}(u)|_{W(K)^{k+1,\infty}}$$

and proceed to estimate the second term. By (17) we have

$$\begin{aligned}
&\|L_h(u,\gamma)\ +\ \hat{P}_h(\nabla\cdot\mathbf{f}(u))\|^2_{L^\infty(K)} && (19)\\
&\le C\frac{1}{|K|}\|L_h(u,\gamma)\ +\ \hat{P}_h(\nabla\cdot\mathbf{f}(u)))\|^2_{L^2(K)}\\
&= C\frac{1}{|K|}\int_K |L_h(u,\gamma)\ +\ \hat{P}_h(\nabla\cdot\mathbf{f}(u))|\,|v_h|\,dx
\end{aligned}$$

where $v_h = (L_h(u,\gamma) + \hat{P}_h(\nabla\cdot\mathbf{f}(u))$. To estimate this term we use the definition of the L^2-projection and of L_h, to obtain, since $v_h \in P^k(K)$,

$$\begin{aligned}
&C\frac{1}{|K|}\int_K (L_h(u,\gamma)\ +\ \hat{P}_h(\nabla\cdot\mathbf{f}(u)))\,v_h\,dx && (20)\\
=\ & C\frac{1}{|K|}\int_K (L_h(u,\gamma)\ +\ \nabla\cdot\mathbf{f}(u))\,v_h\,dx\\
=\ & C\frac{1}{|K|}\int_K L_h(u,\gamma)\,v_h\,dx\\
& -C\frac{1}{|K|}\int_K \mathbf{f}(u)\cdot\nabla v_h\,d\Gamma\ +\ C\frac{1}{|K|}\sum_{e\in\partial K}\int_e \mathbf{f}(u)\cdot\mathbf{n}_{e,K}\,v_h\,d\Gamma\\
=\ & C\frac{1}{|K|}E_K(\mathbf{f}(u)\cdot\nabla v_h)\ +\ C\frac{1}{|K|}\sum_{e\in\partial K}E_e(\mathbf{f}(u)\cdot\mathbf{n}_{e,K}\,v_h)
\end{aligned}$$

where

$$\begin{aligned}
E_K(g) &= \int_K g - \sum_{l=1}^{L}\omega_l\,g(\mathbf{x}_{l,K})|K|, && (21)\\
E_e(g) &= \int_e g - \sum_{l=1}^{L}\omega_l\,g(\mathbf{x}_{l,e})|e|. && (22)
\end{aligned}$$

We have $\mathbf{f}(u) \in W^{k+2,\infty}(\Omega)$ and $\nabla v_h \in P^{k-1}$. By hypothesis, integration over the interior of the triangle is exact for polynomials of degree $2k$, so (16) implies

$$\begin{aligned} C\frac{1}{|K|}|E_K(\mathbf{f}(u) \cdot \nabla v_h)| &\leq Ch^{k+2}|\mathbf{f}(u)|_{W^{k+2,\infty}(K)}\,|\nabla v_h|_{L^\infty(K)} \\ &\leq Ch^{k+2}|\mathbf{f}(u)|_{W^{k+1,\infty}(K)}\,|v_h|_{L^\infty(K)}. \end{aligned}$$

Similarly, integration over the edges is exact for polynomials of degree $2k+1$, and we also have

$$\begin{aligned} C\frac{1}{|K|}|E_e(\mathbf{f}(u) \cdot \mathbf{n}_{e,K}\, v_h)| &\leq C\frac{|e|}{|K|}h^{k+2}|\mathbf{f}(u) \cdot \mathbf{n}_{e,K}|_{W^{k+2,\infty}(e)}\,|v_h|_{L^\infty(K)} \\ &\leq Ch^{k+1}|\mathbf{f}(u)|_{W^{k+2,\infty}(K)}\,|v_h|_{L^\infty(K)} \end{aligned}$$

by regularity of the triangulation. Since $v_h = L_h(u,\gamma) + \hat{P}_h(\nabla \cdot \mathbf{f}(u))$, we thus have

$$\begin{aligned} &||L_h(u,\gamma) \;+\; \hat{P}_h(\nabla \cdot \mathbf{f}(u))||^2_{L^\infty(K)} \qquad (23) \\ &\leq C\frac{1}{|K|}||L_h(u,\gamma) \;+\; \hat{P}_h(\nabla \cdot \mathbf{f}(u)))||^2_{L^2(K)} \\ &\leq Ch^{k+1}|\mathbf{f}(u)|_{W^{k+2,\infty}(K)}\,|L_h(u,\gamma) + \nabla \cdot \mathbf{f}(u)|_{L^\infty(K)}. \end{aligned}$$

3.3 Maximum principle

It remains a current research topic to develop high resolution methods of hyperbolic conservation laws; that is, high-order schemes that converge in a non-oscillatory manner to the correct weak solution. Obtaining sharp, non-oscillating solutions is a function of the local slope limiting operator $\Lambda\Pi_h$, while convergence to the unique ***physically valid*** weak solution is enforced by the entropy-satisfying numerical flux function $h(u_h(\mathbf{x},t))$. However, to initially obtain convergence of the approximation to *any* weak solution of the conservation law, the method must satisfy several properties pertaining to hyperbolic conservation laws [30]. We define the following concepts to demonstrate that the Runge-Kutta Discontinuous Galerkin method satisfies these conditions for convergence.

DEFINITION (*Total Variation and Total Variation Stable*)

The total variation of a numerical approximation in two dimensions is

$$\begin{aligned} TV(u_h) &= limsup_{\epsilon\to 0}\frac{1}{\epsilon}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|u_h(x+\epsilon,y)-u_h(x,y)|\,dx\,dy \\ &\quad + limsup_{\epsilon\to 0}\frac{1}{\epsilon}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|u_h(x,y+\epsilon)-u_h(x,y)|\,dx\,dy \end{aligned}$$

A numerical method is total variation stable (TV-stable) if for $t_n \leq T$, we have $u_h^n \in K$ where $K = \{u : \|u\|_{L_1(L_1)} < \infty,\ TV(u) \leq C < \infty,$ and $supp(u) \subset M$, for M a compact set in $I\!R^2\}$

Lemmä 3.4 *Suppose a numerical method is in conservation form with a Lipschitz-continuous numerical flux, consistent with some scalar conservation law. If the method is TV-stable, then the method is convergent.*

Remark: This convergence is in the sense that for approximation u_h, $dist(u_h, W) \to 0$ as $\delta t \to 0$ where $W = \{w : w(\mathbf{x},t)$ *is a weak solution to the conservation law*$\}$.

Provided the total variation of the approximation over all time is uniformly bounded by the total variation of the initial data, TV-stability and hence convergence is ensured. This is the definition of a total variation diminishing (TVD) method. We would like to obtain convergence of a high-order accurate method. However, in two dimensions there is a strong incompatibility between TVD-compactness and high order accuracy; Goodman and Leveque prove in [24] that any TVD scheme is at most first order accurate.

In [33], Shu introduced a modification called Total Variation Boundedness (TVB) in which the variation is allowed to increase by $\mathcal{O}(\delta t)$ each time step while maintaining total variation stability. However, to prove the TVB property in two dimensions is a rather difficult task even for the simplest monotone scheme if arbitrary triangulations are considered. We instead rely on corrected limiters and ensure a maximum-value principle of the approximation for high order-accurate methods induced by the local slope limiting operator $\Lambda\Pi_h$ [9]:

Theorem 3.2 *For the hyperbolic conservation law (5), the Runge-Kutta Discontinuous Galerkin method, with numerical flux function h and TVD Runge-*

Kutta time discretization, satisfies a maximum principle under the Courant-Friedrichs-Levy condition:

$$\frac{\delta t}{\delta x} max|\frac{\partial h(u)}{\partial u}| \leq \frac{1}{2k+1}$$

for k = 1,2 (second and third order schemes).

4 Adaptive Flux Stencil Advection Approach

We now describe the finite volume method developed by Durlofsky, Engquist, and Osher in [22, 21]. The Triangle Based Adaptive Stencil method is an efficient, formally second order scheme which is applicable to an unstructured triangular grid. It achieves greater than first-order accuracy through the use of a local adaptive flux interpolation procedure which results in computational efficiency. The method is total variation bounded; an extension developed by Liu [31] strictly satisfies the maximum principle.

4.1 Formulation

To formulate the method we present the general finite volume approach followed by the limiting procedure. We define the average of u over triangle K to be

$$\bar{u} = \frac{1}{|K|}\left(\int_K u\,dx\right)$$

where for $\mathbf{x}_{centroid}$ the centroid of triangle K we have $|\bar{u} - u(\mathbf{x}_{centroid})| = O(|K|) = O(|h|^2)$.

Integrating (5) over triangle K yields

$$\frac{\partial}{\partial t}\int_K u\,dx = -\int_K (\nabla \cdot \mathbf{f}(u))dx \tag{24}$$

Applying the divergence theorem to the right-hand side of (24) yields

$$\begin{aligned} \frac{\partial}{\partial t}\bar{u} = & - \frac{1}{|K|}(\int_{e_1} \mathbf{f}(u) \cdot \mathbf{n}_{e_1,K}\,d\Gamma \\ & + \int_{e_2} \mathbf{f}(u) \cdot \mathbf{n}_{e_2,K}\,d\Gamma \\ & + \int_{e_3} \mathbf{f}(u) \cdot \mathbf{n}_{e_3,K}\,d\Gamma) \end{aligned} \tag{25}$$

where $e_1 \cup e_2 \cup e_3 = \partial K$. We will discretize in space first by approximating

$$u_h(t) \approx \bar{u}(t).$$

We replace the flux function $\mathbf{f}(u(\mathbf{x},t))\cdot\mathbf{n}_{e_i,K}$ by the numerical flux function $h_{e_i,K}(\mathbf{x},t) = h_{e_i,K}(u_h^-, u_h^+)$, where $h_{e_i,K}$ is any two-point Lipschitz continuous monotone flux defined on edge e_i of triangle K. We integrate over each line segment via the midpoint rule which is exact for linear functions. Our approximation becomes

$$\begin{aligned} \frac{d}{dt}u_h(t) = & - \frac{1}{|K|}(h_{e_1,K}(u^-,u^+)\,|e_1| \\ & + h_{e_2,K}(u^-,u^+)\,|e_2| \\ & + h_{e_3,K}(u^-,u^+)\,|e_3|). \end{aligned} \tag{26}$$

Remark: By the divergence theorem, the following holds:

$$h_{e1,K}(c,c)\,|e_1| \;+\; h_{e2,K}(c,c)\,|e_2| \;+\; h_{e3,K}(c,c)\,|e_3| \;=\; 0 \quad \forall c = constant$$

To obtain higher order accuracy, we preprocess the initial data so that for each triangle K, a linear interpolating function $\Pi_h u$ is constructed over adjacent triangles so that

$$|\Pi_h u(\mathbf{x}_{e_i}^-,t) - \Pi_h u(\mathbf{x}_{e_i}^+,t)| = O(|K|) = O(|h|^2).$$

The semidiscrete formulation of (5) is: find u_h such that

$$\begin{aligned} \frac{d}{dt}u_h(\mathbf{x},t) = & - \frac{1}{|K|}(h_{e_1,K}(\Pi_h u^-,\Pi_h u^+)\,|e_1| \\ & + h_{e_2,K}(\Pi_h u^-,\Pi_h u^+)\,|e_2| \\ & + h_{e_3,K}(\Pi_h u^-,\Pi_h u^+)\,|e_3|). \end{aligned} \tag{27}$$

Let us define the right-hand side of this equation as $R_h(u_h,\gamma_h)$ to write in ODE form as $\frac{d}{dt}u_h = R_h(u_h,\gamma_h)$.

We now describe the construction of the high-order interpolation approximation which incorporates the limiting procedure as well. For each triangle $|K|$, there exist three candidates $\Pi_h u_i$ depending on the nearest neighbor elements. Provided triangle K is not on a boundary, we generate $\Pi_h u_1$ from the three data points:

$$\begin{array}{ll} (\mathbf{x}_{centroid}, u_h), & on\, triangle\, K \\ (\mathbf{x}_{centroid}, u_h), & on\, triangle\, K_{e_1} \\ (\mathbf{x}_{centroid}, u_h), & on\, triangle\, K_{e_2} \end{array}$$

where we define K_{e_1} to be the triangle adjacent to K on side e_1. Thus, we utilize the data value at the centroid of the corresponding triangle to construct the unique linear functional passing through these three points. We obtain $\Pi_h u_2$ and $\Pi_h u_3$ similarly utilizing K_{e_2}, K_{e_3} and K_{e_3}, K_{e_1} respectively. If the triangle K is on a boundary, we utilize the value $(\mathbf{x}, u_h)$ on edge e_i.

At this point, three possible interpolants exist and a limited version of $\Pi_h u_i$ must be selected from these. To accomplish this, we first compute the magnitude of the gradient of each interpolant $|\nabla \Pi_h u_i|$ and select the $\Pi_h u_i$ for which this gradient is maximized, subject to the restriction that no overshoot or undershoot occurs at any of the three triangle boundaries. Given $\Pi_h u$, we integrate (27) in time utilizing the same Runge-Kutta procedure as described above for the Discontinuous Galerkin method.

Instead, we may choose to incorporate a modification to the selection of the interpolant as described in [31]. This Modified Adaptive Flux method is slightly more compressive than the original, yet strictly achieves the maximum principle in all cases.

4.2 Error Estimate

The operator R_h is a discrete approximation of $-\nabla \cdot \mathbf{f}(u)$, one that defines the construction of a linear approximate solution. By the midpoint formula for integrals, this approximation is weakly second-order accurate insofar as the formal truncation error is within $\mathcal{O}(h^2)$ at the midpoint of each edge.

4.3 Maximum principle

As described above, we will rely on a monotone numerical flux function and satisfaction of a maximum principle in order to justify convergence of our method to the correct entropy-satisfying solution to the hyperbolic conservation law. Assuming we incorporate the limiting modification [31], we will satisfy the following maximum principle:

Theorem 4.1 *For the hyperbolic conservation law (5), the Triangle Based Adaptive Stencils method, with a monotone flux function and TVD Runge-Kutta time discretization, satisfies a maximum principle under the Courant-Friedrichs-Levy condition:*

$$\delta t\sigma(sup|\frac{\partial h(u)}{\partial u}|) \leq \frac{1}{3}.$$

where $\sigma = sup(\{K\}/|K|)$, which is evaluated over all triangles $K \in \Omega$. $\{K\}$ and $|K|$ denote the perimeter and area of triangle K, respectively.

Proof: See [31].

5 Numerical Results

We present results for the convergence of the two methods as well as solution contours to demonstrate accuracy. In particular, we show some numerical results for a convergence test problem and for the rotating cone problem. To compare the two methods described, we restrict the Discontinuous Galerkin method to the case of piecewise linears where $k = 1$.

In our numerical tests, we utilize the Godunov flux function. Given interior and exterior states u^- and u^+, the Riemann solution is determined by

$$h_{e,K}(u_h^-, u_h^+) = \begin{cases} min_{u^- \leq u \leq u^+} \mathbf{f}(u) \cdot \mathbf{n}_{e,K} & if\, u^- \leq u^+, \\ max_{u^+ \leq u \leq u^-} \mathbf{f}(u) \cdot \mathbf{n}_{e,K} & else. \end{cases} \tag{28}$$

We consider the following linear test problem

$$u_t - \nabla \cdot (au) = 0 \quad on\, (0,1) \times (0,1) \times (0,T], \tag{29}$$

where $a = (1,1)^t$ with initial and boundary conditions chosen so that

$$u(x,y,t) = sin(2\pi(x-t)) \cdot sin(2\pi(y-t)).$$

The errors $||u - u_h||_{L^2}$ and rates of convergence at $T = .05$ are given in Table 1. Table 2 exhibits the $||u - u_h||_{L^1}$ errors. Note that the rates of convergence are cumulative least-squares.

TABLE 1: L^2 errors and convergence rates for linear test problem.

	Discontinuous Galerkin		Adaptive Flux		Modified Adaptive Flux	
elements	L^2 error	rate	L^2 error	rate	L^2 error	rate
32	1.59e-01		7.41e-3		8.44e-3	
128	4.58e-02		3.09e-3		3.69e-3	
512	1.28e-02	1.82	1.12e-3	1.36	1.59e-3	1.20
2048	6.13e-03	1.59	3.30e-4	1.40	5.90e-4	1.27
8192	2.02e-03	1.55	8.72e-5	1.60	2.72e-4	1.26

TABLE 2: L^1 errors and convergence rates for linear test problem.

	Discontinuous Galerkin		Adaptive Flux	
elements	L^1 error	rate	L^1 error	rate
32	1.27e-1		1.30e-2	
128	3.40e-2		4.12e-3	
512	1.11e-2	1.76	1.29e-3	1.67
2048	2.91e-3	1.82	3.79e-4	1.70
8192	7.58e-4	1.86	1.10e-4	1.72

We also consider the rotating cone problem which is of the same form as (29) but incorporates variable coefficients:

$$a = [(y - 0.5), (0.5 - x)]^t.$$

The initial condition is a cone of maximum height 1 and radius 0.15, centered at (0.75, 0.5). The exact solution is counterclockwise rotation of the initial condition about (0.5, 0.5). The solution after a quarter revolution and after a half revolution is shown in Figure 1 for the Discontinuous Galerkin method with 2048 elements. The solution for the Adaptive Flux Stencil method with 2048 elements is shown in Figure 2. Figure 3 depicts 3-dimensional mesh plots of the methods at one-half revolution. The maximum value of u after one-half revolution for each method is given in Table 3.

TABLE 3: Maximum value of u for rotating cone test problem.

Discont. Galerkin		Adaptive Flux	
maximum u quarter rev	maximum u half rev	maximum u quarter rev	maximum u half rev
.856	.821	.795	.712

FIGURE 1. Discontinuous Galerkin contour plot of rotating cone test problem at one-quarter revolution (left) and one-half revolution (right).

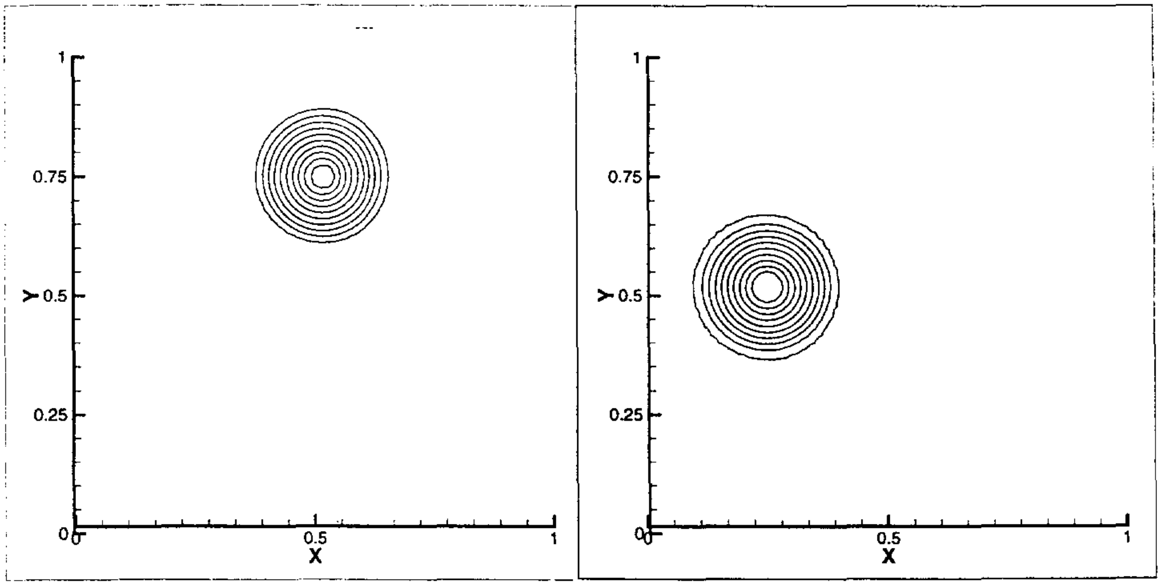


FIGURE 2. Adaptive Flux contour plot of rotating cone test problem at one-quarter revolution (left) and one-half revolution (right).

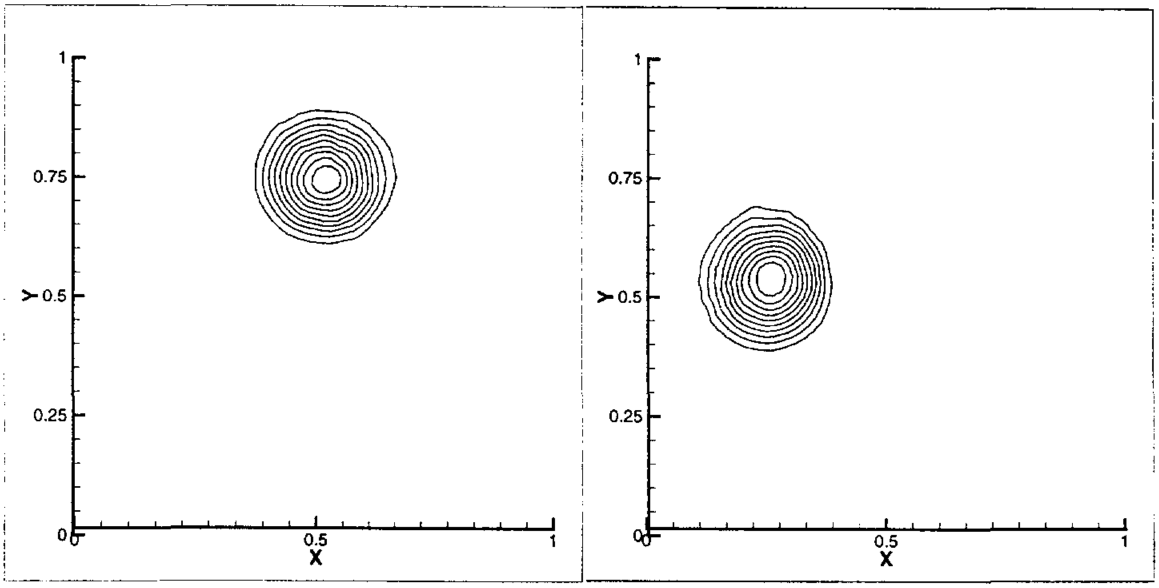
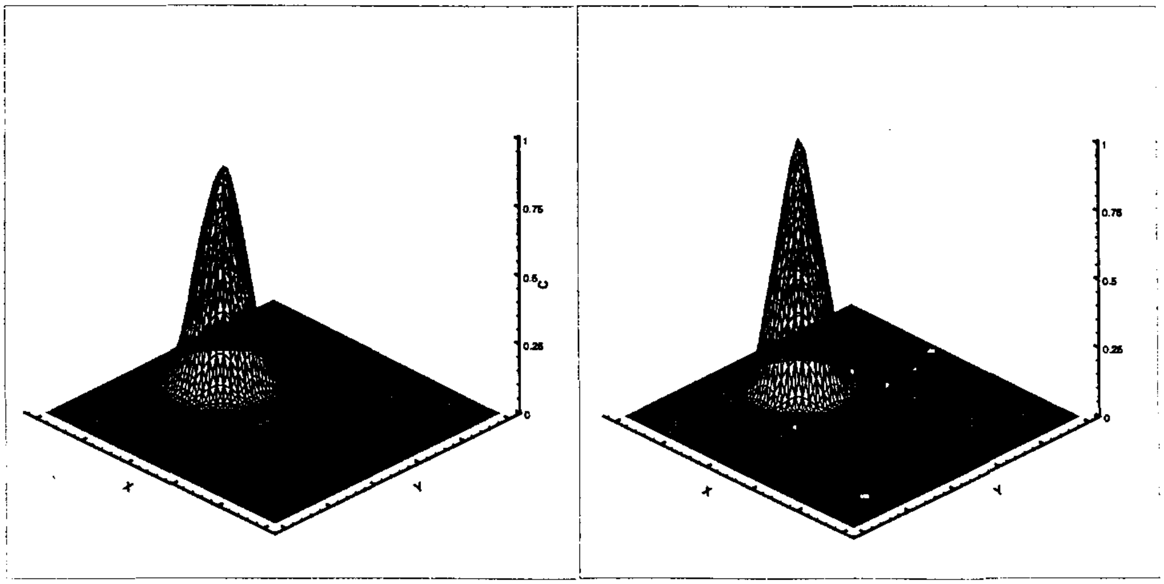


FIGURE 3. 3D mesh plots of Adaptive Flux (left) and Discontinuous Galerkin (right) solutions to rotating cone test problem at one-half revolution.



6 Conclusions

In this report we have considered the Runge-Kutta Discontinuous Galerkin method and for the Adaptive Flux Stencil method for hyperbolic conservation laws. We have implemented these methods on two-dimensional triangular structured domains and now analyze our results.

Numerical results for the linear test problem reveal similar rates of convergence for both methods and a correspondingly lower absolute error for the Adaptive Flux Stencil scheme. The cumulative least-squares convergence rate approaches 1.55 in the L^2 norm and 1.86 in the L^1 norm for the Discontinuous Galerkin method. The corresponding rates of convergence for the Adaptive Flux method approach 1.60 in the L^2 norm and 1.72 in the L^1 norm. The Modified Adaptive Flux Stencil method is slightly more compressive than the unmodified version in order to strictly enforce the maximum principle. This is reflected in the slightly lower convergence rate of approximately 1.25 in the L^2 norm for this method.

For the rotating cone problem, the methods demonstrate high order accuracy without the addition of spurious oscillations or overly diffusive solutions. After one-half revolution, the Discontinuous Galerkin method better preserved the height of the cone (.836) than did the Adaptive Flux method (.795), although both gave good results. A typical first-order approximation to this problem can diffuse the height of the cone to only .48, reduced from 1 in the initial condition.

Both methods are straightforward to implement numerically and give good results insofar as they approach the optimal rate of convergence and accurately resolve a solution profile with sharp gradients. The computational efficiency of one method over another is not significantly distinguishable. The extremely local computational domain render the methods suitable for efficient parallel implementation.

The flexibility of the Discontinuous Galerkin method in its ability to construct higher than first order approximations makes it more conducive to higher-order investigations and adaptive finite element problems. The corresponding extension of the Adaptive Flux method to greater than first order approximations would be prohibitive, primarily due to the number of degrees of freedom needed to uniquely define the quadratic interpolant in two dimensions.

Both methods can easily handle complicated geometries as they are applicable to unstructured triangular meshes; the Discontinuous Galerkin method is applicable to rectangular elements as well. The main advantage of these methods over others is their high parallelizability and higher order accuracy, rendering them suitable for advection-dominated flow computations.

References

[1] T. ARBOGAST, C.N. DAWSON, P.T. KEENAN, M.F. WHEELER, and I. YOTOV. Enhanced cell-centered finite differences for elliptic equations on general geometry. *SIAM J. Sci. Comput.*, 19(2):404–425, 1998.

[2] T. ARBOGAST and M.F. WHEELER. A characteristics-mixed finite element method for advection-dominated transport problems. *SIAM J. Numer. Anal.*, 32:404–424, 1995.

[3] J. BEAR. *Dynamics of Fluids in Porous Media.* Dover, New York, 1972.

[4] J.B. BELL, C.N. DAWSON, and G. SHUBIN. An unsplit, higher-order godunov method for scalar conservation laws in multiple dimensions. *J. Comput. Phys.*, 1, 1984.

[5] J.P. BORIS and D.L. BOOK. Flux-corrected transport i. shasta, a fluid transport algorithm that works. *J. of Comp. Phys.*, 11:38–69, 1973.

[6] G. CHAVENT and B. COCKBURN. The local projection p^0p^1-discontinuous galerkin finite element method for scalar conservation laws. *IMA Preprint Series 341, University of Minnesota*, 1987.

[7] G. CHAVENT and G. SALZANO. A finite element method for the 1d water flooding problem with gravity. *J. Comput. Phys.*, 45:307–344, 1982.

[8] P. CIARLET. *The Finite Element Method for Elliptic Problems.* North-Holland, Amsterdam, 1975.

[9] B. COCKBURN and C.W. SHU. Tvb runge-kutta local projection discontinuous galerkin finite element method for scalar conservation laws ii: general framework. *Math. Comp.*, 52:411–435, 1989.

[10] B. COCKBURN and C.W. SHU. Tvb runge-kutta local projection discontinuous galerkin finite element method for scalar conservation laws iii: one dimensional systems. *Comput. Phys.*, 84:90–113, 1989.

[11] B. COCKBURN and C.W. SHU. Tvb runge-kutta local projection discontinuous galerkin finite element method for scalar conservation laws iv: the multidimensional case. *Math. Comp.*, 54:545–581, 1990.

[12] B. COCKBURN and C.W. SHU. The runge-kutta local projection p^1-discontinuous galerkin method for scalar conservation laws. *M^2AN*, 25:337–361, 1991.

[13] B. COCKBURN and C.W. SHU. The local discontinuous galerkin method for time-dependent convecton-diffusion systems. *SIAM J. Numer. Anal., submitted*, 1997.

[14] B. COCKBURN and C.W. SHU. Tvb runge-kutta local projection discontinuous galerkin finite element method for scalar conservation laws v: multidimensional systems. *J. of Comput. Phys.*, 141:199–224, 1998.

[15] P. COLELLA. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.*, 171, 1990.

[16] C.N. DAWSON. Godunov-mixed methods for advective flow problems in one space dimension. *SIAM J. Num. Anal.*, 28:1282–1309, 1991.

[17] C.N. DAWSON. Godunov-mixed methods for advection-diffusion equations in multidimensions. *SIAM J. Num. Anal.*, 30:1315–1332, 1993.

[18] C.N. DAWSON. High resolution upwind-mixed finite element methods for advection-diffusion equations with variable time-stepping. *Num. Meth. for PDEs*, 11:525–538, 1995.

[19] C.N. DAWSON and V. AIZINGER. Upwind-mixed methods for transport equations. *TICAM Report 98-18*, 1998.

[20] J.JR. DOUGLAS and T.F. RUSSELL. Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Numer. Anal.*, 19:871–885, 1982.

[21] L.J. DURLOFSKY. A triangle based mixed finite element– finite volume technique for modeling two phase flow through porous media. *J. Comput. Phys.*, 105(2):252–266, 1993.

[22] L.J. DURLOFSKY, B. ENGQUIST, and S. OSHER. Triangle based adative stencils for the solution of hyperbolic conservation laws. *J. Comput. Phys.*, 98:64–73, 1992.

[23] R.A. FREEZE and J.A. CHERRY. *Groundwater.* Prentice Hall, 1979.

[24] J.B. GOODMAN and R.J. LEVEQUE. On the accuracy of stable schemes for 2d scalar conservation laws. *Math. Comp.*, 45:15–21, 1985.

[25] A. HARTEN. On a class of high-resolution total-variation-stable finite-difference schemes. *SIAM J. Numer. Anal.*, 21:1–23, 1984.

[26] A. HARTEN, B. ENGQUIST, S. OSHER, and S. CHAKRAVARTHY. Some results on uniformly high-order accurate essentially nonoscillatory schemes. *Appl. Numer. Math.*, 2:347–377, 1986.

[27] A. HARTEN, B. ENGQUIST, S. OSHER, and S. CHAKRAVARTHY. Uniformly high order accurate non-oscillatory schemes, iii. *J. Comput. Phys.*, 71:231–303, 1987.

[28] P.D. LAX and B. WENDROFF. Systems of conservation laws. *Comm. Pure Appl. Math.*, 13:217–237, 1960.

[29] P. LESAINT and P.A. RAVIART. On a finite element method for solving the neutron transport equation. *Mathematical Aspects of Finite Elements in Partial Differential Equations (C. de Boor, Ed.), Academic Press*, pages 89–145, 1974.

[30] R.J. LEVEQUE. *Numerical Methods for Conservation Laws.* Birkhauser, 1990.

[31] X.D. LIU. A maximum principle satisfying modification of triangle based adaptive stencils for the solution of scalar hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 30(3):701–716, 1993.

[32] P.L. ROE. Approximate riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

[33] C.W. SHU. Tvb uniformly high-order schemes for conservation laws. *Math. Comp.*, 49:105–121, 1987.

[34] C.W. SHU. Total-variation-diminishing time discretizations. *SIAM J. Sci. Stat. Comput.*, 9:1073–1084, 1988.

[35] C.W. SHU and S. OSHER. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.

[36] C.W. SHU and S. OSHER. Efficient implementation of essentially non-oscillatory shock capturing schemes ii. *J. Comput. Phys.*, 83, 1989.

[37] P.K. SWEBY. High resolution schemems using flux limiters for hyperbolic conservation laws. *SIAM J. Num. Anal.*, 21:995–1011, 1984.

```
paros% ls
3D              ContE           Landcut         SunPatch        WDryOld
CCB.DE1         Corpus          Lavaca          Tecplot         WDryOld1
CCBNEP          DoublE          Nueces          TxB1            Westrnk
CCChan          GFD             Quoddy4         TxBLEND.Galv89  hardcopy.ps
CNueS           GridsJM1        SALdata         TxEMP.Corpus    lost+found
CONTI           GridsJM2        Sabine          TxEMP.lcra      temp.ps
Chippada        GridsSabine     Sabine2         UT-Dawson       temp9
Conjugate       JiChun          Smodel          WDry
paros% cd UT-Dawson
paros% ls -l
total 7490
-rw-r--r--   1 matsu    staff        180 Sep 10 15:09 ReadMe
-rw-r--r--   1 matsu    staff    1853440 Sep 10 15:06 tartxblend
-rw-r--r--   1 matsu    staff    1955840 Sep 10 15:06 tartxblend3d
paros% view ReadMe
paros% lp ReadMe
request id is HP5-912 (1 file)
paros% pwd
/envi4/UT-Dawson
paros%
```

computer programs
came from UT-Austin

tartxblend is a tar file for TxBLEND-2D with iterative solver. tartxblend3d if for 3D version. Jennifer Proft/Clint Dawson developed these programs under TWDB's research contract.

ReadMe in /envi4/UT-Dawson