ELSEVIER

Advances in Water Resources 24 (2001) 29–48

Advances in Water Resources

www.elsevier.com/locate/advwatres

A comparison of high-resolution, finite-volume, adaptive–stencil schemes for simulating advective–dispersive transport

Matthew W. Farthing, Cass T. Miller *

Department of Environmental Sciences and Engineering, Center for the Advanced Study of the Environment, University of North Carolina, Chapel Hill, NC 27599-7400, USA

Received 1 February 2000; received in revised form 8 May 2000; accepted 16 May 2000

Abstract

We investigate a set of adaptive–stencil, finite-volume schemes used to capture sharp fronts and shocks in a wide range of fields. Our objective is to determine the most promising methods available from this set for solving sharp-front advective–dispersive transport problems. Schemes are evaluated for a range of initial conditions, and for Peclet and Courant numbers. Based upon results from this work, we identify the most promising schemes based on efficiency and robustness.

1. Introduction

The processes of solute and energy transport in natural systems are most often described using an advective–dispersive equation (ADE) with additional terms for sources, reactions, and interphase exchange of mass or energy [34]. For many problems, especially those involving multiple species, ADE solutions can be a substantial part of the computational effort involved for a given flow and transport simulator. Economical solution of this equation is still elusive for cases in which sharp fronts in space and/or time develop. Dozens of approaches have appeared in the literature in a wide variety of fields, including water resources and environmental engineering [2,29,66], chemical and petroleum engineering [10,54], atmospheric science [43], applied mathematics, and gas dynamics [11,37].

Most current subsurface transport codes use fixed-grid Eulerian methods, method of characteristics, or particle tracking approaches – each of which has its own limitations. For example, conventional low-order finite-difference or finite-element methods are widely used, but known to be overly diffusive, subject to phase error, undershoot, peak depression, and oscillations [37]. Combining the advantages of Eulerian and Lagrangian approaches, method of characteristic approaches can be efficient for solving relatively simple advective-dominated problems. These approaches, however, can pose mass conservation problems, are not well suited for problems with multiple sources and non-linear mass transfer terms, and can be diffusive and oscillatory [3,19,38,44].

Over the last decade, simulators in the water resources field have employed so-called high-resolution methods more and more for discretizing the advective portion of the ADE [6,9,20,25,30,35,40,51,52,61,67]. By high-resolution methods, we mean a class of adaptive–stencil methods, usually explicit in time, for resolving sharp fronts. This class has been actively investigated in the fields of gas dynamics and applied mathematics over the last two decades [14,15,18,22,24,28,32,42,45,48,49,53,57,59,64].

Some issues should be considered before directly applying high-resolution schemes from the gas dynamics and applied math literature to the ADE. Much of the work in these fields is primarily concerned with how well the schemes resolve solutions arising from non-linear equations or systems of equations (e.g., Burgers' equation or the Euler equations) whose fronts are usually combinations of self-sharpening shocks, rarefaction waves, and contact discontinuities in the presence of negligible physical dispersion [11,28,49]. In many respects, the ADE is much simpler and often serves as a useful initial test problem. However, the ability to resolve sharp fronts for linear advective–dispersive problems is not the chief criterion for determining the overall

* Corresponding author.
E-mail addresses: matthew_farthing@unc.edu (M.W. Farthing), casey_miller@unc.edu (C.T. Miller).

0309-1708/01/$ - see front matter
PII: S0309-1708(00)00027-0

Notation

C	concentration
$\tilde{C}$	higher-order approximation (ULT), Eq. (45)
C^{b}	extrapolated constraint bound (ULT), Eq. (46)
C^{L}	cell reconstruction approached from left side of cell face (PPM)
C^{R}	cell reconstruction approached from right side of cell face (PPM)
C^{p}	higher-order approximation (PPM), Eq. (49)
D	dispersion coefficient
F_{l}	lower-order numerical flux, Eq. (18)
F_{h}	higher-order numerical flux, Eq. (18)
$\mathscr{M}$	minmod operator, Eq. (24)
N_x	number of cells along x-axis
Pe	mesh Peclet number ($Pe = u\Delta x/D$)
S	numerical slope, Eq. (17)
S_{f}	sign of higher-order flux correction (FCT), Eq. (33)
a	starting spatial coordinate for semi-ellipse profile, Table 4
b	final spatial coordinate for semi-ellipse profile, Table 4
i_{c}	center cell number for semi-ellipse profile, Table 4
i_{f}	final cell number for semi-ellipse profile, Table 4
i_{s}	initial cell number for semi-ellipse profile, Table 4
i_{w}	half cell-width for semi-ellipse profile, Table 4
q_k	kth ENO stencil (WENO), Eqs. (64)–(66)
t	time coordinate
u	interstitial velocity
x	spatial coordinate

Greek symbols

Δ_j	second-order difference, Eq. (13)
ΔC	general higher-order flux correction, Eq. (19)
Δf	higher-order flux correction (FCT), Eq. (34)
Ω	spatial domain under consideration
α_k	weighting fraction for kth ENO stencil (WENO), Eq. (68)
β_k	smoothness monitor for kth ENO stencil (WENO), Eqs. (69)–(71)
β_{c}	compression coefficient (COX), Eq. (38)
β_{u}	compression coefficient (COX), Eq. (38)
δ_j	first-order difference ($\delta_j = C_j - C_{j-1}$)
$\bar{\delta}_j$	centered first-order difference, Eq. (27)
$\delta^*_{j+1/2}$	first-order difference ($\delta^*_{j+1/2} = C^{\mathrm{L}}_{j+1/2} - C^{\mathrm{R}}_{j-1/2}$) (PPM)
δf	limited higher-order flux correction (FCT), Eq. (32)
η_j	term ($\eta_j = C_j - \sigma^*_{j+1/2}/2$) (PPM)
θ_k	ideal weight for kth ENO stencil (WENO)
λ	Courant–Friedrichs–Lewy number ($\lambda = u\Delta t/\Delta x$)
ξ	normalized spatial coordinate ($\xi = (x - x_{j-1/2})/\Delta x \in [x_{j-1/2}, x_{j+1/2}]$)
$\sigma^*_{j+1/2}$	sum ($\sigma^*_{j+1/2} = C^{\mathrm{L}}_{j+1/2} + C^{\mathrm{R}}_{j-1/2}$) (PPM)
σ_j	sum ($\sigma_j = C_{j+1} + C_j$)
τ_{s}	local spatial truncation error
φ	flux-limiter function, Eq. (20)
ω	weighting parameter (COX)
ω_k	final weighting applied to kth ENO stencil (WENO), Eq. (67)

Subscripts and superscripts

j	spatial grid cell identifier (subscript)
n	time level identifier (superscript)

Abbreviations

ADE	advective–dispersive equation
ASO	alternating split operator
CFL	Courant–Friedrichs–Lewy number
COX	Cox and Nishikawa transport scheme
CPU	total elapsed CPU time
ENO	essentially non-oscillatory
FCT	flux-corrected transport
HARM	harmonic transport scheme
LIU	Liu et al. transport scheme
MMOD	minmod transport scheme
MUSCL	monotone upstream-centered scheme for conservation laws
PPM	piecewise parabolic method
Ref	primary literature source(s) for method, Table 1
SBEE	superbee transport scheme
TV	total variation, Eq. (1)
TVD	total variation diminishing, Eq. (2)
ULT	ultimate transport scheme
UNO	uniformly non-oscillatory
UNO2	second-order uniformly non-oscillatory transport scheme
WENO	weighted essentially non-oscillatory
ZDL	Zaidel transport scheme
1-PT	one-point upstream weighting
2-PT	two-point upstream weighting
flop	floating point operation
`hpm`	hardware performance monitor

quality of a scheme in the gas dynamics or applied mathematics fields.

The objectives of this work are:

1. to review evolving classes of high-resolution finite-volume methods that hold promise for the water resources field;
2. to summarize a significant set of high-resolution methods;
3. to compare the efficiency and robustness of a broad set of methods for solving the ADE;
4. to recommend approaches for applications and extension to other cases.

2. Background

The challenge for modeling sharp fronts in advective-dominated problems is to resolve the front over relatively few computational cells without introducing spurious oscillations in the solution. The scheme should provide higher-order accuracy in smooth regions of the solution as well. Godunov's original scheme (simple upwinding) [12] produces a monotone solution from monotone initial data but introduces excessive numerical diffusion, losing the sharp nature of the physical solution. Godunov also showed that any linear scheme of second or higher-order necessarily introduces oscillations in the numerical solution. Modern high-resolution schemes have circumvented this barrier because they are non-linear. That is, they use an approximation that is at least second-order in smooth regions but then adapt their discretization to add sufficient numerical diffusion around sharp fronts to avoid oscillations while still maintaining good resolution. Early examples of modern high-resolution schemes can be found in the work of Van Leer [53], Zalesak [63], Colella [8], and Sweby [48].

These methods can be categorized in several different ways. For example, one can divide adaptive schemes based on the principle, algebraic or geometric, used to determine intercell values [57,64]. Detailed explanations of these approaches are available in the literature [11,28,49,50]. Briefly, geometric schemes can be viewed as extensions of Godunov's original method [12] that employ adaptive higher-order reconstructions to approximate cell interface values. The basic idea behind algebraic schemes is to combine low- and higher-order fluxes. The higher-order flux provides better resolution than the low-order solution, while the scheme reverts to the low-order flux to prevent oscillations around sharp fronts. Generally, this is done by summing the low-order flux with an "anti-diffusive" correction, which is just a limited difference between higher- and lower-order fluxes. The limiting procedure enforces some algebraic constraint on the numerical solution. In a great many schemes, this condition is that the total variation

$$\mathrm{TV}(C) = \sum_j |C_{j+1} - C_j| \tag{1}$$

of the solution is non-increasing [27,48,64], i.e.

$$\mathrm{TV}(C^{n+1}) \leqslant \mathrm{TV}(C^n). \tag{2}$$

Appropriately, these schemes are known as total variation diminishing (TVD).

The notion of total variation leads to another classification of high-resolution methods that is closely tied to their accuracy at smooth local extrema. TVD schemes necessarily degrade to first-order at local extrema [11]. In [18], the TVD criterion was replaced by the uniformly non-oscillatory (UNO) property. UNO schemes insure that the number of local extrema does not increase rather than enforcing the stricter TVD condition, which requires that local extrema are damped at each time step. As a result, the second-order UNO scheme presented in [18] maintains its formal accuracy at smooth local extrema. The notion of essentially non-oscillatory (ENO) methods presented in [17] represents a further generalization. ENO schemes allow local extrema to be increased at times but guarantee that the total variation increase is on the order of the grid spacing.

$$\mathrm{TV}(C^{n+1}) \leqslant \mathrm{TV}(C^n) + \mathrm{O}(\Delta x^{p+1}) \quad \text{for } p > 0. \tag{3}$$

As a result, ENO schemes also maintain higher-order accuracy around smooth extrema.

Schemes can also be categorized according to the order of interpolation and extrapolation used to approximate the intercell face values. To a certain extent, classification along order of interpolation and extrapolation reflects the reviewed schemes' complexity and traces the historical development of methods in the field.

Before the introduction of adaptive–stencil methods, the only monotonicity-preserving schemes were first-order. The first modern, adaptive high-resolution methods [4,53] sought to employ piecewise linear approximations wherever possible while maintaining the monotonicity of a solution. From there, the next step was the development of schemes such as the piecewise parabolic method (PPM) [8] and versions of flux-corrected transport (FCT) [63], which used higher-order approximations while reverting to lower order to maintain monotonicity.

The next logical extensions were the UNO and ENO schemes, which were specifically designed to employ reconstructions of arbitrary order without reverting to lower-order approximations. The relevance of interpolation and extrapolation is based on the simple view that a successful scheme is one that consistently provides accurate approximation for intercell face values. Of course, the quality of an approximation is influenced by several factors besides the formal order. For example, the information used for constructing these values should come from the same side of a discontinuity. This

idea has a simple physical interpretation from the principle of spreading information along characteristics for pure advection processes. Even though interpolation may be more accurate for intercell reconstruction, extrapolation should be employed when a discontinuity falls onto an interpolating stencil. It follows that successful adaptive methods should also include efficient, robust detection of steep gradients and discontinuities in order to guide the switch between interpolation and extrapolation.

One could also distinguish these methods based on the time discretization used. For the most part, adaptive–stencil schemes are explicit in time. Implicit versions can, however, be found in [51,58,60]. Most schemes match the accuracy of the temporal and spatial discretizations [50]. For example, a well-known second-order Lax–Wendroff type discretization is used in many second-order TVD schemes [48,53]. Third- and fourth-order TVD Runge–Kutta time discretizations have been commonly employed in ENO schemes [14,31,45].

As mentioned above, high-resolution schemes have been available in the water resources literature for some time [9,30,40] and are becoming more common [35,51,61,67]. The introduction of adaptive-stencil schemes into the water resources field has mirrored their development in the original fields. Second-order TVD schemes were used early on [9,40,51], while later works have tended to use higher-order TVD or ENO schemes [35,61,67]. However, a limited number of schemes has been considered in the water resources field, and no comparison of these schemes exists. If the current trend towards higher-order methods continues, it seems appropriate that the most promising schemes from this class be identified to guide future work.

There has been a series of comparisons of selected schemes for individual test problems in other fields [14,39,48,64,57]. These comparisons are insufficient for our purposes, since either the test problems are inapplicable or too few approaches are considered, making comparison among methods difficult. For example, [57] provides a wide survey of shock capturing methods (including TVD, FCT, and ENO schemes) but only considers Burgers' equation with initial data that led to solutions with shock waves and rarefaction fans. Also, these studies do not usually address computational efficiency, which is relevant concern, since these methods involve adaptive discretizations.

For the water resources field, the most useful comparison considered numerical efficiency for a one-dimensional non-linear ADE problem arising from transport with non-linear sorption (Langmuir isotherm) [39]. While the physical dispersion (diffusion) insures a smooth solution, the sharp front in this problem is self-sharpening. That is, even initially smooth profiles can develop steep gradients. This type of problem poses somewhat different challenges than linear transport with contact discontinuities, where no sharpening mechanism competes with numerical and physical dispersion. In addition, only a small subset of high-resolution schemes described above was considered: a Taylor Galerkin, an Eulerian–Lagrangian, and a random choice scheme.

3. Approach

3.1. Scope

As outlined in Section 2, high-resolution methods can be grouped in several ways including the use of the TVD, UNO, and ENO classes. Within these three categories, a wide range of specific schemes have been advanced, which vary in the order of both spatial and temporal truncation error of the approximation.

Because of the large number of schemes that exist, limits must be drawn for any comparative investigation, especially because we intend to perform a thorough comparison of the schemes' efficiency and robustness. First, we limit our investigation to one-dimensional approaches. This is reasonable, since one-dimensional methods are often extended to multiple dimensions within an operator splitting framework [7,36,49], and schemes that are ineffective in one dimension are not apt to be competitive in higher dimensions.

Second, we limit the scope of the approaches considered in this work by restricting ourselves to approximations which are, in general, less than or equal to third order; many of the schemes that we consider are second-order in space and/or time. This restriction limits the schemes considered in both number and in complexity of the stencils needed to form the approximations. While a detailed consideration of higher-order methods is a reasonable goal for future work, higher-order accuracy is achieved for such schemes only for sufficiently smooth solutions. Rather, the largest potential benefit from using higher-order schemes for a discontinuous solution is not more rapid convergence in some error norm but sharper resolution of the discontinuity [23].

3.2. Formulation

Our test problem is the one-dimensional ADE

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - u\frac{\partial C}{\partial x}, \quad \text{for } x \in \Omega \times t \in [0, T] \tag{4}$$

with boundary conditions

$$C(0,t) = C_0, \tag{5}$$

$$\left.\frac{\partial C}{\partial x}\right|_{x_1,t} = 0 \tag{6}$$

and initial condition

$$C(x,0) = C^0(x), \tag{7}$$

where C is the concentration, t the time, x the spatial coordinate, D the dispersion coefficient, $u > 0$ the velocity, and $\Omega \in [0, x_1]$ is the spatial domain.

Following a typical cell-centered finite-volume, or equivalently in one dimension, a finite-difference discretization of Eq. (4), C_j^n will represent an approximation of the cell average of $C(x,t)$ over the jth cell $[x_{j-1/2}, x_{j+1/2}]$ at time t^n

$$C_j^n = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} C(x,t^n)\,\mathrm{d}x \tag{8}$$

where Δx and Δt are the spatial and time increments, $x_j = j\Delta x$ the center of the jth cell of the uniform grid, $x_{j-1/2}$ and $x_{j+1/2}$ are cell boundaries or faces, and $t^n = n\Delta t$.

3.3. Algorithm

To decouple the approximation of the dispersive and advective portions of the ADE, we use an alternating split operator (ASO) approach, which has a truncation error of order Δt^2 [1]. In the ASO algorithm, a three-step approach is used to advance a time level. The first step is the solution of the linear dispersion equation over a half time step, $\Delta t/2$:

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2}, \quad t \in \{t^n, t^n + \Delta t/2\} \tag{9}$$

with the solution at t^n as the initial condition. The second step is the solution of the linear advection equation over a full time step, Δt, using the solution from Eq. (9) as the initial condition

$$\frac{\partial C}{\partial t} = -u\frac{\partial C}{\partial x}, \quad t \in \{t^n, t^n + \Delta t\} \tag{10}$$

and the third step again solves the linear dispersion equation over a half time step

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2}, \quad t \in \{t^n + \Delta t/2, t^n + \Delta t\}, \tag{11}$$

using the solution from (10) for its initial value.

We approximate Eqs. (9) and (11) using a standard centered discretization in space and a Crank–Nicolson discretization in time, which are both second-order approximations. For Eq. (9), for example, these approximations yield

$$C_j^{n+1/2} = C_j^n + \frac{D\Delta t}{4\Delta x^2}\left(\Delta_j^{n+1/2} + \Delta_j^n\right), \tag{12}$$

where

$$\Delta_j^n = C_{j+1}^n - 2C_j^n + C_{j-1}^n. \tag{13}$$

3.4. Temporal discretization

A variety of explicit time discretizations are used by the schemes considered in this work. With one exception, they are all subject to a stability constraint on the time step of, $u\Delta t \leqslant \Delta x$. The exception is the scheme introduced by Liu et al. [30], which has the tighter bound of $u\Delta t \leqslant \Delta x/2$. Most of the methods, including the traditional second-order TVD schemes [27,48], are based on a time discretization originally found in the fixed-stencil Lax–Wendroff scheme. Basically, a time correction is added to cancel the leading error term from a forward Euler time discretization. This correction achieves second-order accuracy, assuming the underlying spatial discretization is second-order accurate [27,28,50]. There are a number of ways to present this temporal approximation, including a two-step form

$$C_{j+1/2}^{n+1/2} = C_{j+1/2}^n - \lambda\left(C_{j+1/2}^n - C_j^n\right), \tag{14}$$

$$C_j^{n+1} = C_j^n - \lambda\left(C_{j+1/2}^{n+1/2} - C_{j-1/2}^{n+1/2}\right). \tag{15}$$

When combined into one step, this leads to

$$C_j^{n+1} = C_j^n - \lambda\delta_{j+1/2}^n + \lambda^2\left(\delta_{j+1/2}^n - \delta_j^n\right), \tag{16}$$

where

$$\delta_j^n = (C_j - C_{j-1}), \quad \lambda = u\Delta t/\Delta x$$

is the Courant–Friedrichs–Lewy (CFL) number, and $C_{j+1/2}^n$ is a cell interface concentration value for $x_{j+1/2} = (j + 1/2)\Delta x$. The Lax–Wendroff type discretization for Eq. (4) is simple and can be applied to a wide variety of spatial discretization schemes used to compute the cell face concentrations. We note that the two-step form is written in terms of a conservative difference, Eq. (15) [49]. This property is, in fact, shared by each discretization we consider.

3.5. Spatial discretization

For the methods considered in this work, the approach used to determine intercell values, $C_{j+1/2}$, plays the central role in the spatial discretization of Eq. (4). Several classes of these schemes for computing face values were reviewed in Section 2. The schemes we consider can be placed into four classes:

1. Fixed-stencil schemes, which use the same approximation at all interior locations.
2. Traditional adaptive second-order TVD schemes, which select among linear stencils and have a spatial truncation error $\tau_s \leqslant \mathrm{O}[(\Delta x)^2]$.
3. Adaptive extensions of second-order TVD schemes, which select among variable-order stencils such that $\tau_s \leqslant \mathrm{O}[(\Delta x)^3]$.
4. Adaptive UNO or ENO schemes for which the spatial truncation error is at least second-order accurate ($\tau_s \geqslant \mathrm{O}[(\Delta x)^2]$).

The local truncation error results cited here are valid only for sufficiently smooth regions. Again, there are

several factors including the formal order of a specific approximation that dictate the quality and accuracy of a scheme for a given range of sharp front problems. The set of spatial discretization schemes considered is summarized in Table 1. The orders shown are the minimum and maximum formal, local truncation error (in space and time) for smooth regions. In order to simplify our notation, we omit the time index, n, where possible.

Spatial discretization methods considered in the fixed-stencil and second-order TVD groups can be put into and interpreted in multiple forms. Specifically, the two-step time discretization, Eqs. (14) and (15), is often combined with either a slope- or flux-limiting approach. From a geometric perspective, the slope-limiter form replaces a piecewise constant representation of the solution from Godunov's method with a piecewise linear representation. In the flux-limiter form, intercell fluxes are computed so as to preserve some property such as assuring the solution is TVD. For the linear problem we focus on here, either of these two conceptual approaches can be used as a framework for the two groups of methods noted above. In the slope-limiter form, the cell interface values can be expressed as

$$C_{j+1/2} = C_j + S_j \Delta x/2, \tag{17}$$

where S_j is the slope, or concentration gradient, for the jth cell. The approach used to assign S_j varies with the scheme. For linear advection, determining the correct interface value then reverts to tracing back along characteristics ($x = ut$) in the upstream direction.

In flux-limiter form, the cell interface values can be expressed as

$$uC_{j+1/2} = F_l + (F_h - F_l), \tag{18}$$

where F_l is a low-order flux, and F_h is a higher-order flux [28]. For the linear transport problem with constant velocity that we consider here, we give the concentration form for the case in which the low-order concentration is computed using one-point upstream weighting. It can be expressed as

$$C_{j+1/2} = C_j + \Delta C_{j+1/2}, \tag{19}$$

where $\Delta C_{j+1/2}$ is a higher-order correction, which depends upon the scheme.

The flux-limiter schemes are often written in the form

$$C_{j+1/2} = C_j + \varphi_j \delta_{j+1}/2, \tag{20}$$

where $\delta_{j+1} = C_{j+1} - C_j$.

Sweby [48] specified regions that the function φ_j should occupy in order to be TVD as well as TVD and second-order. The various members of this family differ only in the portion of this TVD region that they occupy. Detailed descriptions of these schemes and limiters can be found in many places, e.g., [28,48,64].

For the problem and schemes considered here, the equivalence of the slope-limiter and flux-limiter approaches clearly requires that

$$S_j \Delta x/2 = \Delta C_{j+1/2} = \varphi_j \delta_{j+1}/2. \tag{21}$$

3.5.1. Fixed-stencil schemes

Fixed-stencil schemes have been commonly used in the water resources literature [13,41,47,55,65] and are included here as a point of reference for those unfamiliar with the adaptive–stencil schemes that are the primary focus of this work. We consider two fixed-stencil methods: the one- and two-point upstream weighted schemes. These schemes use the same stencil and approximation regardless of the local features of the numerical solution. We present both schemes using the two-step temporal discretization given in Eqs. (14) and (15). Of course, 1-PT is only first-order accurate in time, since its underlying spatial approximation is first-order and the assumptions for the time correction are not met.

The one-point upstream weighting (1-PT) scheme is

$$C_{j+1/2} = C_j. \tag{22}$$

Table 1
Spatial discretization schemes

Scheme	Description	Class	Order (x)	Order (t)	Ref.
1-PT	One-point upstream weighting	TVD	1	1	[21]
2-PT	Two-point upstream weighting	–	2	2	[21]
MMOD	Minmod	TVD	1,2	1,2	[15,28]
MUSCL	Muscl	TVD	1,2	1,2	[49,53]
SBEE	Superbee	TVD	1,2	1,2	[28,42]
HARM	Harmonic	TVD	1,2	1,2	[48,53]
FCT	Flux-corrected transport	TVD	1,2	1,2	[33]
LIU	Liu et al.	TVD	1,3	1,2	[30]
COX	Cox	TVD	1,3[a]	1,3[a]	[9]
ULT	Ultimate	TVD	1,3	1,3	[27]
PPM	Piecewise parabolic	TVD	1,3[b]	1,3	[8,14]
UNO2	Uniformly non-oscillatory	UNO	2	2	[18]
ZDL	Zaidel	ENO	3	2	[62]
WENO	Weighted ENO	ENO	3,5	3	[24]

[a] With appropriate parameter choice $\omega = (1 + \lambda)/3$, otherwise order is 2.
[b] In limit $\Delta t \to 0$ order is 4.

The 1-PT scheme is monotonic, TVD, and when $\lambda = 1$ it provides the exact solution for advection of concentration along characteristics for the strictly hyperbolic problem (point-to-point transfer). 1-PT has a spatial truncation error of $\tau_s = O(\Delta x)$. The well-known cost of the one-point upstream weighting scheme is excessive numerical diffusion [49].

The two-point upstream weighting (2-PT) scheme is

$$C_{j+1/2} = C_j + \delta_j/2. \tag{23}$$

The 2-PT scheme has oscillatory behavior in the vicinity of contact discontinuities and steep gradients (see [43]) – a fact shown to be true in general for any fixed-stencil method that has $\tau_s \geqslant O[(\Delta x)^2]$ [21]; the 2-PT scheme has $\tau_s = O[(\Delta x)^2]$.

3.5.2. Second-order TVD schemes

The schemes considered in this section are based on the use of adaptive stencils where intercell values are computed as linear approximations of cell averages. They are second-order accurate on smooth solutions and degrade to first-order accuracy at local extrema, a feature common to TVD schemes [11]. They are second-order accurate in time over sufficiently smooth regions (away from local extrema) and, with the exception of FCT, employ the standard two-step time discretization given by Eqs. (14) and (15).

The MMOD scheme [15,22,28,64] is a common, simple scheme of the form

$$C_{j+1/2} = C_j + \mathscr{M}(\delta_{j+1}, \delta_j)/2, \tag{24}$$

where the two-term minmod function is

$$\mathscr{M}(m_1, m_2) = [\operatorname{sign}(m_1) + \operatorname{sign}(m_2)] \times \min(|m_1|, |m_2|)/2. \tag{25}$$

MUSCL uses the second-order, centered slope except when this choice introduces a new extremum [50,53]

$$C_{j+1/2} = C_j + \mathscr{M}(\delta_{j+1}, \delta_j, \bar{\delta}_j/4), \tag{26}$$

where

$$\bar{\delta}_j = \delta_{j+1} + \delta_j = C_{j+1} - C_{j-1} \tag{27}$$

and the three-term minmod function is

$$\mathscr{M}(m_1, m_2, m_3) = [\operatorname{sign}(m_1) + \operatorname{sign}(m_2)] \times \min(|m_1|, |m_2|, |m_3|)/2. \tag{28}$$

Superbee (SBEE) [42] occupies the upper bound of the second-order TVD region [48] and can be shown to reduce to

$$C_{j+1/2} = C_j + [\operatorname{sign}(\delta_{j+1}) + \operatorname{sign}(\delta_j)] \times \max[\min(|\delta_{j+1}|, 2|\delta_j|), \min(2|\delta_{j+1}|, |\delta_j|)]/4. \tag{29}$$

Harmonic [53] (HARM) is a second-order TVD scheme with a smoother limiter function than SBEE:

$$\Delta C_{j+1/2} = \begin{cases} (\delta_{j+1}\delta_j)/(\delta_{j+1} + \delta_j) & \text{if } \delta_{j+1}\delta_j > 0 \\ 0 & \text{if } \delta_{j+1}\delta_j \leqslant 0. \end{cases} \tag{30}$$

Flux-corrected transport (FCT) [4,5] is a TVD scheme. A general algorithm, it can be applied using a variety of specific spatial and temporal discretization approaches [33,63]. In general, the FCT algorithm consists of:

1. computing a low-order solution that is guaranteed to be monotonic;
2. computing a higher-order correction;
3. computing upper and lower bounds on the correction using the low-order solution;
4. limiting the higher-order correction to ensure conformance with the bounds on the solution; and
5. advancing a time step by computing the sum of the low-order solution and the limited corrector.

We implemented a version of FCT that is second-order in space and time [33]. This version was superior to the original scheme [57] which exhibited a staircasing pattern for linear advection of a semi-ellipse profile. FCT is often expressed as a two-step procedure, since the higher-order correction is limited against values of a preliminary low-order solution at the current time step. A simpler way of presenting the FCT scheme for our problem, however, is in the form of one-point upstream weighting modified with an up-and-downstream limited higher-order correction

$$C_j^{n+1} = C_j^n - \lambda\delta_j^n - \lambda(\delta f_{j+1/2} - \delta f_{j-1/2}) \tag{31}$$

$$\delta f_{j+1/2} = S_{fj+1/2} \max\Big\{0, \min\Big[|\Delta f_{j+1/2}|, S_{fj+1/2}\big((1-\lambda)\delta_{j+2}^n + \lambda\delta_{j+1}^n\big), S_{fj+1/2}\big((1-\lambda)\delta_j^n + \lambda\delta_{j-1}^n\big)\Big]\Big\}, \tag{32}$$

$$S_{fj+1/2} = \operatorname{sign}(\Delta f_{j+1/2}), \tag{33}$$

$$\Delta f_{j+1/2} = \Big[(2-3\lambda)\delta_j^n + \lambda\delta_{j-1}^n\Big]/4. \tag{34}$$

3.5.3. Adaptive, variable-order TVD extensions

These methods represent extensions to classic second-order TVD schemes. They introduce higher-order approximations, while still reverting to lower-order reconstructions to maintain monotonicity.

Liu et al. [30] introduced a scheme (LIU) that advances the flux-limiter approach in [48], using a cubic stencil from [26] and a CFL-dependent limiting procedure

$$C_{j+1/2} = C_j + [\operatorname{sign}(\delta_{j+1}) + \operatorname{sign}(\delta_j)] \min[2|\delta_{j+1}|, |\delta_{j+1}|(2/\lambda - 2), (2|\delta_{j+1}| + |\delta_j|)/3, 2|\delta_j|, |\delta_j|(2/\lambda - 2)]/4. \tag{35}$$

The LIU scheme employs a slightly different time discretization

$$C_j^{n+1} = C_j^n - \lambda\delta_{j+1/2}^n + \lambda^2\Delta_j^n/2 \tag{36}$$

which is second-order when $C_{j+1/2}$ is at least second-order. When the upstream or downstream stencils are chosen ($C_{j+1/2} = C_j$ or $C_{j+1/2} = C_j + \delta_{j+1}$), comparing Eqs. (14) and (15) with Eq. (36) using simple modified equation analysis shows that the magnitude of the leading, diffusive (or anti-diffusive) error terms is larger with LIU's time discretization. The leading error terms, however, are smaller with the third-order stencil.

The scheme introduced by Cox and Nishikawa [9] (COX) uses a flux-limiter that is a weighted sum of central and upwind differences of the form

$$C_{j+1/2} = C_j + [\omega\mathscr{M}(\beta_u\delta_{j+1}, \delta_j) + (1-\omega)\mathscr{M}(\delta_{j+1}, \beta_c\delta_j)]/2, \tag{37}$$

where $\omega \in [0,1]$ is a weighting factor, and β_u, and β_c are CFL-dependent compression parameters.

COX attempts to improve the traditional second-order TVD methodology by making advantageous choices for the parameters. The optimal values of the parameters β_u, β_c, and ω depend on the dimensionality of the problem and whether or not dispersion is included [9]. For example, if $\omega = (1+\lambda)/3$, then the COX scheme achieves formal third-order spatial accuracy in smooth regions. In the presence of physical dispersion, the following parameter values were suggested

$$\begin{aligned} \beta_c &= 1 + \frac{1}{(1-\omega)}\left(\frac{2-\lambda}{\lambda}\right) - \frac{4}{Pe(1-\lambda)(1-\omega)}, \\ \beta_u &= 1 + \frac{1}{\omega}\left(\frac{1+\lambda}{1-\lambda}\right) + \frac{2}{Pe(1-\lambda)\omega}, \end{aligned} \tag{38}$$

where $Pe = u\Delta x/D$ is the mesh Peclet number. While we use these values for our analysis, we note that they are not optimal for other cases, such as multidimensional problems with variable velocity fields [9].

Leonard [27] analyzed a family of upwind and centered difference schemes from low- to high-order and extended the TVD analysis of Sweby [48] to a more general, less restrictive form, and gave a variety of algorithms. We investigated the simplified ultimate limiter scheme (ULT) based upon the cubic QUICKEST stencil and a universal limiter that is independent of the stencil used, but dependent upon the CFL number [27]. A similar approach has also recently been implemented in the popular MT3DMS code [67]. The universal limiter is extended to be non-linear in both the solution, C_j^n, and the Courant number. With this approach, one discards the higher-order stencil only if the value it generates violates CFL-dependent constraints designed to maintain monotonicity (assuming the solution is monotonic in a given region). The ULT approach generates a solution at $C_{j+1/2}^{n+1/2}$ directly, which is in turn used in (15). When the QUICKEST stencil is used, this approximation is third-order in both space and time [27].

The ULT scheme implemented in this work for determining $C_{j+1/2}^{n+1/2}$ with constant positive velocity is described algorithmically as:

(i) if $|\Delta_j| \leqslant 0.6|\bar{\delta}_j|$, then the solution is monotonic and sufficiently smooth, so use

$$C_{j+1/2}^{n+1/2} = \tilde{C}_{j+1/2}^{n+1/2} \tag{39}$$

(ii) else if $|\Delta_j| \geqslant |\bar{\delta}_j|$, then the solution is non-monotonic, so set

$$C_{j+1/2}^{n+1/2} = C_j^n \tag{40}$$

(iii) else if $\bar{\delta}_j > 0$, then the solution is a monotonic increasing function, so enforce the constraint

$$C_{j+1/2}^{n+1/2} = \max\left(\tilde{C}_{j+1/2}^{n+1/2}, C_j^n\right) \tag{41}$$

and then

$$C_{j+1/2}^{n+1/2} = \min\left(C_{j+1/2}^{n+1/2}, C_{j+1/2}^{b}, C_{j+1}^n\right) \tag{42}$$

(iv) else since $\bar{\delta}_j < 0$, the solution is a monotonic decreasing function, so enforce the constraint

$$C_{j+1/2}^{n+1/2} = \min\left(\tilde{C}_{j+1/2}^{n+1/2}, C_j^n\right) \tag{43}$$

and then

$$C_{j+1/2}^{n+1/2} = \max\left(C_{j+1/2}^{n+1/2}, C_{j+1/2}^{b}, C_{j+1}^n\right), \tag{44}$$

where the third-order QUICKEST approximation is

$$\tilde{C}_{j+1/2}^{n+1/2} = [3(\sigma_{j+1} - \lambda\delta_{j+1}) - (1-\lambda^2)\Delta_j]/6 \tag{45}$$

and the extrapolated constraint bound is

$$C_{j+1/2}^{b} = C_{j-1}^n + \delta_j/\lambda \tag{46}$$

and

$$\sigma_{j+1} = C_{j+1} + C_j. \tag{47}$$

The PPM [8,14] improves upon MUSCL by introducing a quadratic reconstruction over each cell of the form

$$C^n(x) = C_{j-1/2}^R + \xi\left[\delta_{j+1/2}^* + 6(1-\xi)\eta_j\right], \tag{48}$$

where

$$\delta_{j+1/2}^* = C_{j+1/2}^L - C_{j-1/2}^R,$$

$$\eta_j = C_j - \sigma_{j+1/2}^*/2,$$

$$\sigma_{j+1/2}^* = C_{j+1/2}^L + C_{j-1/2}^R,$$

$$\xi = (x - x_{j-1/2})/\Delta x \in [x_{j-1/2}, x_{j+1/2}],$$

and the superscripts R and L indicate the limits approached from the right and left side of the cell face, respectively.

The PPM algorithm for a positive velocity is:

(i) Form the higher-order, limited interpolated value as

$$C^{p}_{j+1/2} = \sigma_{j+1}/2 - (S_{j+1} - S_j)/6, \tag{49}$$

where

$$S_j = 2\mathscr{M}(\delta_{j+1}, \delta_j, \bar{\delta}_j/4). \tag{50}$$

(ii) Assign $C^{\mathrm{L}}_{j+1/2} = C^{p}_{j+1/2}$ and $C^{\mathrm{R}}_{j-1/2} = C^{p}_{j-1/2}$.

(iii) Ensure that no new local extrema at cell centers are generated from the reconstruction

if $(C^{\mathrm{L}}_{j+1/2} - C_j)(C_j - C^{\mathrm{R}}_{j-1/2}) \leqslant 0$ then

$$C^{\mathrm{R}}_{j-1/2} = C_j, \tag{51}$$

$$C^{\mathrm{L}}_{j+1/2} = C_j. \tag{52}$$

(iv) Make sure that the reconstruction does take on extreme values beyond C_j and C_{j+1} [50],

else if $\delta^{*}_{j+1/2}\eta_j > (\delta^{*}_{j+1/2})^2/6$, then

$$C^{\mathrm{R}}_{j-1/2} = 3C_j - 2C^{\mathrm{L}}_{j+1/2} \tag{53}$$

(v) else if $\delta^{*}_{j+1/2}\eta_j < -(\delta^{*}_{j+1/2})^2/6$, then

$$C^{\mathrm{L}}_{j+1/2} = 3C_j - 2C^{\mathrm{R}}_{j-1/2}. \tag{54}$$

(vi) Using the constrained values for $C^{\mathrm{R}}_{j-1/2}$ and $C^{\mathrm{L}}_{j+1/2}$, form

$$C^{n+1/2}_{j+1/2} = C^{\mathrm{L}}_{j+1/2} - \lambda\left(\delta^{*}_{j+1/2} - 2(3 - 2\lambda)\eta_j\right)/2. \tag{55}$$

This value for $C^{n+1/2}_{j+1/2}$ is then used directly in Eq. (15). In general, the final PPM approximation is third-order accurate in space and time for smooth regions. In the limit as $\Delta t \to 0$, however, it is fourth-order accurate [8]. The PPM scheme can also be extended to the case in which a cell falls over a contact discontinuity. This is similar to the subcell resolution approach in [16]. Additionally, the detection of the discontinuity requires the use of problem-dependent parameters. As with artificial compression and subcell resolution with the WENO scheme, we do not use this approach in our comparisons.

3.5.4. Adaptive, uniformly higher-order schemes

These methods extend the classic second-order TVD schemes to higher-order approximations without reverting to lower-order reconstructions. We implemented a UNO method [18] which insures that the number of local extrema does not increase. It employs the time discretization in Eqs. (14) and (15) together with a linear reconstruction of the form

$$C_{j+1/2} = C_j + \mathscr{M}[\delta_{j+1} - \mathscr{M}(\Delta_{j+1}, \Delta_j)/2, \delta_j + \mathscr{M}(\Delta_j, \Delta_{j-1})/2]/2. \tag{56}$$

The Zaidel scheme (ZDL) [62] implemented uses the time discretization in Eq. (16) with either third-order interpolation or extrapolation stencil based upon a test of smoothness. This yields

$$C_{j+1/2} = \begin{cases} (3C_{j+1} + 6C_j - C_{j-1})/8 & \text{if } |\delta_{j+1}| \leqslant \max(|\delta_j|, |\delta_{j-1}|), \\ (15C_j - 10C_{j-1} + 3C_{j-2})/8 & \text{if } |\delta_{j+1}| > \max(|\delta_j|, |\delta_{j-1}|). \end{cases} \tag{57}$$

Weighted ENO schemes [24] (WENO) are designed to perform well on vector computing platforms and to provide higher-order accuracy in smooth regions. The WENO scheme implemented here can be summarized using a method of lines approach

$$\frac{\partial C}{\partial t} = L(C), \tag{58}$$

$$L(C) = -u\delta_{j+1/2}/\Delta x \tag{59}$$

and a TVD Runge–Kutta temporal integration scheme [45,46] of the form

$$C^{(1)} = C^n + \Delta t L(C^n), \tag{60}$$

$$C^{(2)} = C^n + \Delta t[L(C^n) + L(C^{(1)})]/4, \tag{61}$$

$$C^{n+1} = C^n + \Delta t[L(C^n) + L(C^{(1)}) + 4L(C^{(2)})]/6. \tag{62}$$

There are various advantages and disadvantages of method of lines approach compared to other temporal integration methods that we have used. For example, Eq. (16) is relatively simple and cheap. When $\lambda = 1$, the first step provides the point-to-point transfer property. On the other hand, decoupling the spatial and temporal discretizations can be useful when moving to higher dimensions [28]. It also makes it relatively easy to increase the temporal accuracy of the overall scheme. However, the Runge–Kutta discretization requires two more flux evaluations than Eq. (16). These flux evaluations represent most of the computational work associated with these methods.

WENO schemes use a convex combination of the r candidate ENO stencils, where r is the order of the ENO base for the weighted scheme. The weights for the stencils are chosen so that in smooth regions the combination approaches a discretization that is formally $(2r-1)$th order accurate. The WENO scheme implemented in this work is technically fifth-order accurate in ideal situations. At a discontinuity, the weights are designed to neglect stencils that contain the discontinuity. In this way, the behavior of the base ENO scheme is recreated. For $u > 0$,

$$C_{j+1/2} = \sum_{k=0}^{2} w_k q_k, \tag{63}$$

where

$$q_0 = (11C_j - 7C_{j-1} + 2C_{j-2})/6, \tag{64}$$

$$q_1 = (2C_{j+1} + 5C_j - C_{j-1})/6, \tag{65}$$

$$q_2 = (-C_{j+2} + 5C_{j+1} + 2C_j)/6, \tag{66}$$

$$w_k = \alpha_k/(\alpha_2 + \alpha_1 + \alpha_0), \tag{67}$$

$$\alpha_k = \frac{\theta_k}{(\epsilon + \beta_k)^p}. \tag{68}$$

Here, θ_k is the weight for the kth ENO stencil required in order to achieve fifth-order accuracy. The values are 0.1, 0.6, and 0.3 for $k = 0, 1, 2$, respectively. Following [24], we use $\epsilon = 1.0 \times 10^{-6}$ and $p = 2$ in our computations. The smoothness monitors, β_k, are

$$\beta_0 = (3\delta_j - \delta_{j-1})^2/4 + 13(\Delta_{j-1})^2/12, \tag{69}$$

$$\beta_1 = \bar{\delta}_j^2/4 + 13(\Delta_j)^2/12, \tag{70}$$

$$\beta_2 = (\delta_{j+2} - 3\delta_{j+1})^2/4 + 13(\Delta_{j+1})^2/12. \tag{71}$$

Techniques such as subcell resolution [16] and artificial compression [56] can improve resolution of contact discontinuities for WENO schemes [24]. This technique, however, requires the use of problem-dependent parameters. Moreover, neither subcell resolution nor artificial compression are restricted to ENO schemes [8,63]. For these reasons, we restrict ourselves to the base scheme in our comparisons.

4. Results

The goal of this work is to compare the efficiency of the candidate set of methods described above for solving advective–dispersive transport problems. We are also interested in evaluating the robustness, or the level of performance across a range of problems, of the candidate methods. We define efficiency as the computational effort needed to achieve a given accuracy of a numerical solution. Such comparisons can be approached in a variety of ways. We consider two main approaches:

1. An analytical comparison of algorithms based upon the order of accuracy in space and time, operations needed to advance a time step, and the relative cost of those operations.
2. A significant set of numerical experiments for a range of initial conditions and spatial and temporal discretizations.

4.1. Algorithm analysis

As with any numerical efficiency comparison, one would expect the computational expense associated with the various methods to be sensitive to their implementation. This is especially true for the adaptive schemes we consider whose speed is dictated largely by the process used to pick a given stencil based on the numerical solution at the previous time step. For instance, the TVD schemes presented above rely on a limiting procedure, such as the minmod function $\mathscr{M}$, in which the scheme goes through a selection process for some relevant quantity when the solution is monotonic around the current cell face. Otherwise, the approximation reverts to one-point upstream. There are a number of different ways to implement this limiting process that appear in the literature. Most differ only in the way they determine whether or not the solution is monotonic around a given cell. If the solution is locally non-monotonic, the extra computational work associated with choosing slopes or fluxes is not required, and some expense can be saved. Using a conditional, or "if" statement, to distinguish the monotonic region is potentially more difficult to optimize, however. This can lead to poorer performance when the solution is monotonic and diminish the potential savings gained by avoiding the full limiting procedure for the non-monotonic region. In either case, the relative performance of a scheme will depend on the choice of implementation for this limiting process. The speed of a given implementation will, in turn, depend on the compiler and platform used for a code and how well they optimize the specific intrinsic operations used in the limiting process.

To provide an even comparison, we used the same basic limiting approach and implementation (wherever applicable) for each scheme. Before choosing which scheme to implement, we compared the performance of four different approaches on three different computational platforms (machine/compiler combinations), which are summarized in Table 2.

The four candidate implementations we tested can be illustrated using the MUSCL scheme. For the first three, the MUSCL slope $S_j\Delta x$ would be

$$S_j\Delta x = \begin{cases} 2\,\mathrm{sign}(\delta_{j+1})\min[|\delta_{j+1}|, |\delta_j|, |\bar{\delta}_j|/4] & \text{if } \delta_{j+1}\delta_j > 0 \\ 0 & \text{if } \delta_{j+1}\delta_j \leqslant 0, \end{cases} \tag{72}$$

$$S_j\Delta x = [\mathrm{sign}(\delta_{j+1}) + \mathrm{sign}(\delta_j)] \times \min[|\delta_{j+1}|, |\delta_j|, |\bar{\delta}_j|/4], \tag{73}$$

$$S_j\Delta x = 2\,\mathrm{sign}(\delta_{j+1}) \times \max\{0, \min[|\delta_{j+1}|, \mathrm{sign}(\delta_{j+1})\ \delta_j, |\bar{\delta}_j|/4]\}. \tag{74}$$

The fourth approach uses no sign functions and can be given in algorithm form by

Table 2
Computational platforms

Hardware	Operating system	Compiler/opt. level
Cray T-90	UNICOS (10.0.0.2)	Cray CF90 (3.2.1.0)/03
HP C160 (PA-RISC 2.2)	HPUX (10.20)	HP f77 (10.10)/02
Pentium III (450 mHz)	LINUX (2.2.5)	g77 (egcs-2.91.66)/02

$$
\begin{aligned}
&\text{(i) If } \delta_{j+1}\delta_j \leqslant 0, \text{ then } S_j\Delta x = 0\\
&\text{(ii) Otherwise } S_j\Delta x = 2\delta_j\\
&\quad \text{(a) if } |\delta_{j+1}| < |\delta_j|, \text{ then } S_j\Delta x = 2\delta_{j+1}\\
&\quad \text{(b) if } |\delta_{j+1} + \delta_j|/2 < |S_j\Delta x|,\\
&\qquad \text{then } S_j\Delta x = (\delta_{j+1} + \delta_j)/2
\end{aligned}
\tag{75}
$$

The method given by Eq. (72) [8] and Eq. (75) avoid the extra steps in the MUSCL limiting procedure if the solution is non-monotonic, while Eq. (73) [28] and (74) [9] do not. In implementation Eq. (75) the minimum slope is chosen based upon "if" statements rather than relying on the sign and min intrinsics.

First, a pair of simple trials were performed on HPUX C160 and Pentium III workstations. The above calculations for $S_j\Delta x$ were performed on an array of random values. In the first test, $S_j\Delta x$ was calculated using values for δ_j and δ_{j+1} that agreed in sign for each cell. In the second test, 10% of the calculations were performed on values that disagreed in sign. In general, we expect the number of cells for which the solution is non-monotonic to be a small fraction of the total domain, since this is defined using the solution value over only three cells. It is certainly the case for the test profiles we use in our investigation. The calculations were performed a total of 10 000 times for a vector of length 100 000, and the resultant CPU times were recorded using the FORTRAN subroutine `itime` in the U77 library.

To test the performance on a vector platform, a similar test was run on a Cray T-90 vector supercomputer (see Table 2). For this test, the ADE equation was solved using the MUSCL scheme for a 1000 cell simulation, and $\lambda = 0.05$. The total CPU time accumulated in the MUSCL loop was recorded using the hardware performance monitor (`hpm`).

The four implementations were comparable in performance for the Pentium III, while Eq. (75) was between 3.75 and 4.6 times faster than all other methods for both tests on the HP C160. As would be expected, the two implementations without conditionals Eqs. (73) and (74) were superior for the vector platform, obtaining 667 and 746 megaflops, respectively. Implementation Eq. (72) still performed well, however (568 megaflops). The version that relied solely on conditionals, Eq. (75), lagged far behind (234 megaflops).

The best implementation clearly depends on the platform. The only approach that was obviously inadequate for any of the tested platforms was Eq. (75) on the Cray T-90. Since it was fairly consistent in our tests, finishing no worse than second for any run, implementation Eq. (73) was used in our formulation and in the efficiency comparisons below. The exceptions are the HARM and ULT schemes. Implementation Eq. (72) was used with HARM to avoid a potential divide by 0, while ULT contains a filter for monotonicity (if $|\Delta_j^n| < |\bar{\delta}_j|$) which is significantly different in flavor from the other four we considered. Modifying this would have significantly altered Leonard's original algorithm.

In our work, the final measure of computational expense is the total CPU time required to complete a given run, but we also provide a careful analysis of the number and type of operations required by each scheme we consider. This analysis complements the CPU time measurements in a number of ways. It expresses the work associated with the various schemes through a set of basic, common operations. This provides a framework that is largely independent of platform and implementation (of these core operations) through which the computational expense of the various schemes can be compared. It also clarifies the implementation of the schemes and provides a rational basis for the total CPU time results we show.

Table 3 presents the total number of operations required to advance the advective solution over a time step for an interior cell j. Interior cells are those for which enough neighbor cells exist to apply the widest stencil for a given scheme. Whenever a stencil crosses a physical

Table 3
Summary of operation counts

Scheme	$x \pm y$	$x * y$	x/y	$\lvert x\rvert$	Max/min[a]	Sign	If	Total
1-PT	2	1	0	0	0	0	0	3
2-PT	4	2	0	0	0	0	0	6
MMOD	5	4	0	1	1(2)	1	0	12
MUSCL	6	4	0	2	1(3)	1	0	14
SBEE	5	6	0	1	3(2)	1	0	16
HARM	5	3	1	0	0	0	1	10
FCT	6	8	0	1	1(2), 1(3)	1	0	18
LIU	7	11	0	1	1(5)	1	0	21
COX	6	8	0	2	2(2)	1	0	19
ULT (A/B)	8/9	5/6	0	2	0/1(2), 1(3)	0	1/3	16/22
PPM	19	16	0	2	1(3)	1	3	42
UNO2	9	8	0	3	2(2)	3	0	25
ZDL	6	6	0	1	1(2)	0	1	15
WENO	81	85	15	0	0	0	0	181

[a] Number in parenthesis is the number of terms in min or max.

boundary, a scheme reverts to one-point upstream. For the schemes we consider, the number of boundary cells which use one-point upstream ranges from none to two. The operation counts reflect an implementation intended for a scalar architecture. They assume that a loop through consecutive interior cells is made so that certain values can be reused from iteration to iteration (e.g., $\mathrm{sign}(\delta_{j+1})$ becomes $\mathrm{sign}(\delta_j)$).

The operation counts reflect the implementation of the candidate schemes as formulated in Section 3. Since the expense incurred by the limiting process in the monotonic region better reflects the true expense associated with these schemes, we do not include the non-monotonic region case in Table 3 for HARM and ULT. The latter does contain another switch, however, which significantly impacts the scheme's overall computational cost. If $|\Delta_j^n| \leqslant 0.6|\bar{\delta}_j|$ no other limiting is required. This is ULT (A) in Table 3, while the full limiting operation count is ULT (B). We include both the cases to illustrate the whole range of expense associated with ULT. For PPM, we present the maximum possible operations where all three constraints are evaluated and one is found true, since this should occur in smooth regions where the higher-order interpolation is used. The floating point operations required when a constraint is true are included as well, since this is an insignificant portion of the overall expense.

As mentioned above, the historical development of adaptive–stencil methods can be loosely tied to the pursuit of higher-order monotonic approximations. Higher-order methods might be expected, in turn, to increase complexity and computational expense. Table 3 shows a general trend connecting increased order of approximation with an increased operation count. There are, of course, exceptions such as ZDL. Higher formal order does not necessarily imply either higher quality results or greater computational expense, so we rely ultimately on actual numerical results for accuracy and computational expense. An initial look at Table 3 does lead to a few useful observations. We base these observations on the assumption that floating point multiplies and additions are measurably cheaper than logical operations, such as conditionals, sign, max or min calculations:

1. The linear upstream weighted schemes should be much cheaper than any of the adaptive methods.
2. The number of floating point operations required by PPM and WENO is much greater than the other schemes. PPM is much closer to the other methods than WENO but still contains three conditional statements.
3. The amount of work for ULT (A) should be about the same as that of ZDL, while the full limiting process is much more expensive.
4. The performance of FCT, COX, LIU, UNO2, and the traditional second-order TVD schemes should be fairly similar, with HARM occupying the low end of the range. The variation arising should be based on how well a compiler/platform can optimize the sign and max/min intrinsics and conditionals.

Given these conclusions, the issue of computational efficiency should be decided largely by the answer to the following questions:

1. Are any of the lower order TVD methods accurate enough to compete with the higher-order methods?
2. How do LIU, COX ULT, UNO2, and ZDL compare in terms of efficiency?
3. Is PPM sufficiently more accurate than the other methods of similar order to compensate for the additional computational effort needed to implement the method?
4. Can WENO be accurate enough to compensate, on any computational platform, for the fact that its computational burden is substantially greater than all the other methods considered?

4.2. Numerical experiments

Given our preliminary consideration of the candidate numerical methods and their algorithms, we performed a set of numerical experiments to investigate the efficiency of the various schemes for a range of computational conditions. The experiments were run on a Pentium III workstation (see Table 2) for a range of initial conditions and discretization patterns. For each of the problems, measures of accuracy and computational expense were taken. The candidate methods were then compared to find those which consistently required the least computational effort to achieve given levels of accuracy over the set of experiments.

4.2.1. Work and error measures

For each problem in our investigation, a numerical result from the PPM scheme was used to determine an accurate solution against which to measure error. The result was obtained on a dense grid with 10 000 cells with $\lambda = 0.5$. Error was measured as the norm of the difference between numerical and dense-grid solutions. We calculated error values in the discrete $L_{1,2,\infty}$ norms. The relative performance of the schemes was similar for the various norms used, and so we show results for the L_1 norm only.

Our basic measure of computational expense was the total elapsed CPU time required for the each simulation. For our initial set of runs we also recorded the total CPU time spent evaluating the advective fluxes. As before, the times were obtained using the `itime` intrinsic after a number of loops through the appropriate calculations. We then compared the two different timings to validate our methodology and use of total elapsed CPU time to compare the efficiency of the numerical schemes. The differences between the advective flux evaluation and total simulation times for the considered schemes

Table 4
Initial condition

Initial condition		$C^o(x)$
Semi-ellipse	A	$C^o(\mathrm{i}\Delta x) = \begin{cases} \sqrt{1-(i-i_c)^2/i_w^2} & \text{if } \lvert i-i_c\rvert < i_w, \\ 0 & \text{otherwise.} \end{cases}$ where $i_c = \frac{1}{2}(i_s + i_e)$, $i_w = i_e - i_c = i_c - i_s$, $i_s = N_x \cdot a$, $i_e = N_x \cdot b$, $a = 0.1, b = 0.3$
Semi-ellipse	B	Same as A except $b = 0.22$
Square	C	$\begin{cases} 1 & \text{if } x \in [0.1, 0.22], \\ 0 & \text{otherwise.} \end{cases}$
Triangle	D	$\begin{cases} 1 - 16.\bar{6}\lvert x - 0.16\rvert & \text{if } x \in [0.1, 0.22], \\ 0 & \text{otherwise.} \end{cases}$

were consistent for the initial runs. We also checked that the timings scaled appropriately. For example, when both the number of time steps and cells were doubled, we expected an increase of four in CPU time, since the advective flux evaluations and the Thomas algorithm used for the dispersion solve are of linear complexity. The average ratio (across schemes) determined from our initial runs was 3.96 with a coefficient of variation of 5.22×10^{-2}. We verified our implementations of the various schemes by comparing them with the results from the original sources. Unfortunately, the literature tends to show results for relatively few initial conditions and CFL numbers, but we compared our error norms with those presented where possible.

4.2.2. Problems considered

The one-dimensional model problems used in our investigation involved various initial conditions, grid sizes, CFL numbers, and *Pe*. The four traditional initial profiles are given in Table 4. The spatial and temporal domain was $(x,t) \in [0,1] \times [0,0.5]$. The number of cells, N_x, ranged from 50 to 1000. The choice of one-dimensional grids was weighted towards coarse discretizations, since computational resources still limit the vast majority of numerical simulations for three-dimensional applications to grids with not much more than a 100 cells per coordinate axis. Homogeneous Dirichlet and Neumann boundary conditions were applied at the left and right boundaries, respectively. The various numerical calculations were performed with $\lambda = 0.05$ or $\lambda = 0.5$. The details of the runs performed are given in Table 5.

4.2.3. Baseline comparisons

In order to eliminate poor schemes from further consideration and to refine the questions posed at the end of Section 4.1, we first performed an initial screening test (Runs 1a–1f in Table 5). We considered the model problem relatively easy, since the dispersion coefficient was held fixed at $D = 0.001$ so that the largest $Pe = 20$. The semi-ellipse initial condition was also spread over a relatively large number of cells (11) on the coarsest grid.

Table 5
Simulation summary

Run	$C^o(x)$	*Pe*	CFL
1a	A	20	0.5
1b	A	20	0.05
1c	A	10	0.5
1d	A	10	0.05
1e	A	5	0.5
1f	A	5	0.05
1g	A	1	0.05
2, 3, 4[a]	B, C, D	100	0.5
2, 3, 4[b]	B, C, D	100	0.05
2, 3, 4[c]	B, C, D	50	0.5
2, 3, 4[d]	B, C, D	50	0.05
2, 3, 4[e]	B, C, D	25	0.5
2, 3, 4[f]	B, C, D	25	0.05
2, 3, 4[g]	B, C, D	5	0.5
2, 3, 4[h]	B, C, D	5	0.05

This is still a narrower profile than is often presented in the literature, however [24,46,63]. The results from the screening run are summarized in Tables 6 and 7, in which we show global L_1 error and total CPU time (normalized by the number of times the calculations were repeated).

Low accuracy for Runs 1a–1f could generally be attributed to oscillation, over-compression, or excessive numerical diffusion. These are well known problems for advection schemes [49,50]. Too much numerical diffusion leads to fronts so smeared that the basic character of the true solution is lost. On the other hand, a given scheme may under-represent the amount of dispersion in an attempt to reduce numerical diffusion. This over-compression leads to squared, non-physical solutions. As expected, the fixed-stencil schemes were not competitive. 1-PT was too diffusive while 2-PT was oscillatory. For the traditional TVD schemes, SBEE was over-compressive, and MMOD was excessively diffusive. These phenomena can be easily seen in Fig. 1. FCT produced smeared profiles similar to MMOD. COX followed by MUSCL were the two most accurate schemes from the group MMOD, MUSCL, SBEE, HARM, LIU, COX, UNO2, and FCT (see Fig. 1), while

Table 6
Screening runs results for $\lambda = 0.5$

Scheme	1a		1c		1e	
	Error	CPU (s)	Error	CPU (s)	Error	CPU (s)
1-PT	6.40e−02	4.90e−04	3.85e−02	1.87e−03	2.18e−02	7.50e−03
2-PT	2.29e−02	5.20e−04	6.83e−03	2.00e−03	1.79e−03	7.50e−03
MMOD	1.95e−02	6.80e−04	6.23e−03	2.63e−03	1.76e−03	1.05e−02
MUSCL	5.79e−03	7.60e−04	1.82e−03	3.00e−03	6.01e−04	1.15e−02
SBEE	7.06e−03	7.90e−04	5.36e−03	3.13e−03	1.64e−03	1.25e−02
HARM	9.44e−03	6.60e−04	2.38e−03	2.50e−03	6.38e−04	1.00e−02
FCT	2.29e−02	8.30e−04	8.55e−03	3.25e−03	3.27e−03	1.30e−02
LIU	1.21e−02	9.70e−04	2.64e−03	3.75e−03	7.34e−04	1.50e−02
COX	5.59e−03	7.40e−04	1.83e−03	2.87e−03	6.00e−04	1.15e−02
ULT	5.59e−03	9.00e−04	1.83e−03	3.37e−03	6.00e−04	1.35e−02
PPM	5.18e−03	1.21e−03	1.82e−03	4.75e−03	5.98e−04	1.90e−02
UNO2	9.74e−03	9.30e−04	2.55e−03	3.63e−03	6.69e−04	1.50e−02
ZDL	8.27e−03	6.90e−04	2.74e−03	2.75e−03	8.65e−04	1.05e−02
WENO	8.42e−03	4.30e−03	2.03e−03	2.00e−02	6.03e−04	7.00e−02

Table 7
Screening runs results for $\lambda = 0.05$

Scheme	1b		1d		1f	
	Error	CPU (s)	Error	CPU (s)	Error	CPU (s)
1-PT	9.39e−02	4.80e−03	6.18e−02	1.84e−02	3.72e−02	7.40e−02
2-PT	4.57e−02	5.00e−03	1.50e−02	1.96e−02	4.48e−03	7.80e−02
MMOD	3.28e−02	6.60e−03	1.08e−02	2.58e−02	3.25e−03	1.04e−01
MUSCL	1.10e−02	7.20e−03	3.51e−03	2.88e−02	1.26e−03	1.14e−01
SBEE	6.54e−03	7.60e−03	7.01e−03	3.00e−02	2.98e−03	1.20e−01
HARM	1.63e−02	6.30e−03	4.04e−03	2.54e−02	1.30e−03	1.00e−01
FCT	3.60e−02	8.00e−03	9.78e−03	3.30e−02	4.11e−03	1.30e−01
LIU	9.02e−03	9.40e−03	1.99e−03	3.72e−02	6.74e−04	1.48e−01
COX	7.32e−03	7.20e−03	1.89e−03	2.88e−02	6.40e−04	1.16e−01
ULT	7.34e−03	8.80e−03	1.88e−03	3.44e−02	6.35e−04	1.36e−01
PPM	5.35e−03	1.18e−02	1.91e−03	4.78e−02	6.88e−04	1.90e−01
UNO2	1.43e−02	9.00e−03	3.97e−03	3.62e−02	1.30e−03	1.46e−01
ZDL	1.07e−02	6.80e−03	2.82e−03	2.64e−02	8.01e−04	1.06e−01
WENO	8.37e−03	4.15e−02	2.06e−03	1.70e−01	6.42e−04	7.00e−01

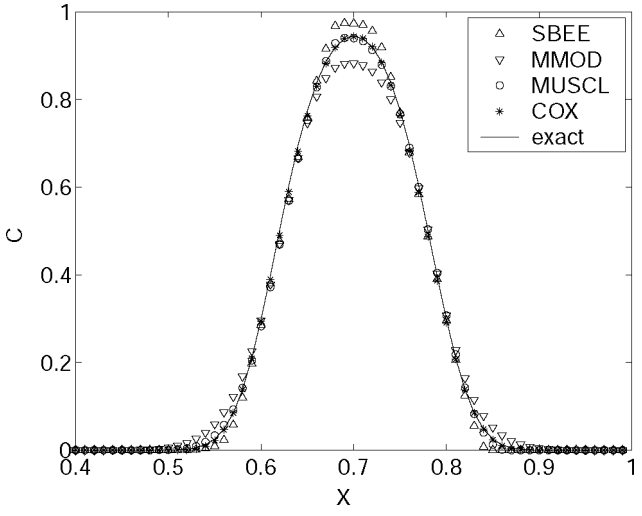

Fig. 1. Solution profiles for Run 1d.

HARM was the fastest. UNO2, HARM and LIU exhibited similar, adequate error values with UNO2 and HARM performing better for the high CFL number and LIU maintaining higher accuracy for the low CFL number. Across the runs, however, they were not as accurate as MUSCL, nor was HARM sufficiently cheap to be more efficient than MUSCL. ULT and COX produced almost identical error values and proved to be consistently accurate for each of the Runs 1a–1f. The higher-order formal accuracy for ZDL, PPM, and WENO also translated into more accurate numerical solutions compared in general to the low-order schemes. The low-error provided by PPM did in fact come at noticeably lower cost than WENO. PPM was between 1.60 and 1.67 times as slow as MUSCL for Runs 1a–1f while WENO was another 3.51 to 4.21 times slower than PPM.

4.2.4. Detailed comparisons

After the initial tests, we were able to narrow our focus and select a subset of the adaptive schemes for

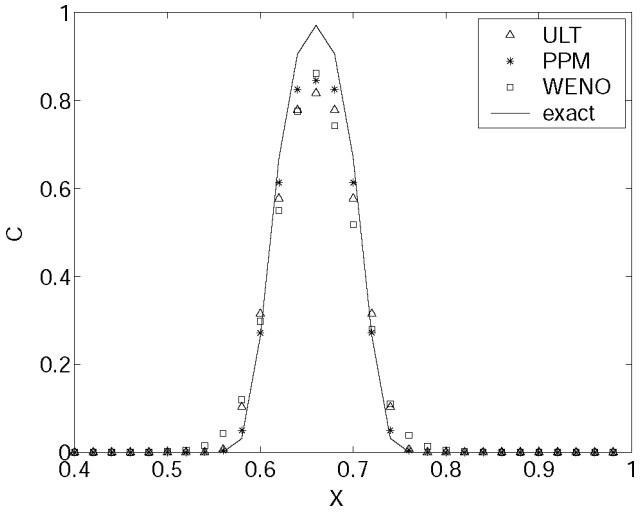

Fig. 2. Solution profiles Run 2a.

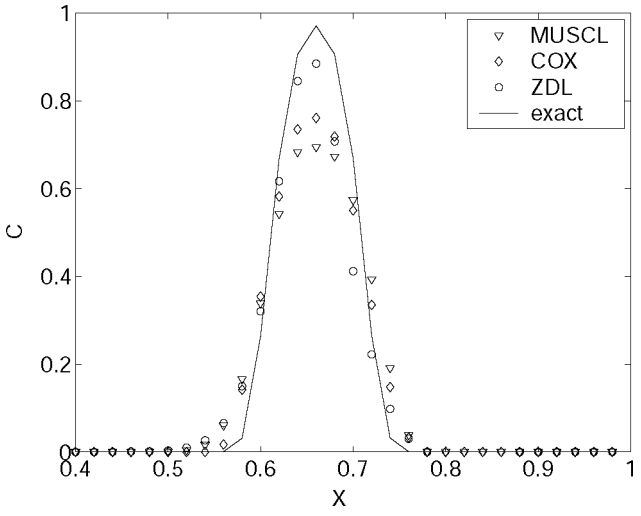

Fig. 3. Solution profiles Run 2b.

further comparison. Given the results of Runs 1a–1f and the analysis from Section 4.1, we chose the MUSCL, COX, ZDL, ULT, PPM, and WENO schemes for additional investigation.

In an effort to distinguish the remaining schemes, we increased the difficulty of the second set of tests. We decreased the dispersion coefficient $D = 0.0002$ and extended the range of Pe to 25–100. In addition to the semi-ellipse profile used in Runs 1a–1f, we considered square and triangular wave initial conditions, choosing narrower slugs that spanned only seven cells on the coarsest grid.

Comparison of the numerical solutions for the narrower semi-ellipse profile in Figs. 2 and 3 with those in Fig. 1 confirms the added severity of the second tests. Despite the variation across the initial conditions, we can discern several general trends from Figs. 2 and 3 and Tables 8 and 9:

1. The solutions provided by COX and ULT were essentially the same, although the COX solution was achieved with about 20% less computational effort.
2. ZDL tended to resolve peaks' value well but was also asymmetric, showing a strong upwind bias.
3. MUSCL was symmetric but more diffusive than ZDL, especially for $\lambda = 0.05$.
4. PPM and WENO were typically the most accurate schemes, and PPM was usually more accurate and always much less expensive than WENO.
5. The L_1 error values for the 1000 cell grids (Runs 2g–2h) were identical for ULT, COX, PPM, and WENO, suggesting that error was essentially due to the operator splitting and dispersion discretization. ZDL, however, lagged in L_1 error for both values of λ, while MUSCL was less accurate for $\lambda = 0.05$ only.

Tables 10 and 11 report the L_1 and CPU values for the square wave initial condition. As with Runs 2a–2h, COX and ULT were basically identical, while MUSCL was the most diffusive. The solutions from both ENO schemes, ZDL and WENO contained peak values slightly greater than the exact solution on the coarse grids. PPM's advantage in terms of L_1 error was diminished especially for the grids with more than 100 cells. In fact, the accuracy of all of the schemes considered was similar for the finest grid (Runs 3g–3h). Tables 12 and 13 contain the results for the triangle initial condition. This proved to be the most severe of the numerical experiments with clear variation among the schemes still evident for the 100 cell grid computations. In contrast to the other profiles, WENO was more accurate than PPM, especially for the low CFL computations while ZDL's upwind bias produced poorer results (see Fig. 4).

Table 8
Results for narrow semi-ellipse profile with $\lambda = 0.5$

Scheme	2a		2c		2e	
	Error	CPU (s)	Error	CPU (s)	Error	CPU (s)
MUSCL	2.10e−02	7.50e−04	4.23e−03	2.87e−03	1.19e−03	1.18e−02
COX	1.70e−02	7.40e−04	4.03e−03	2.87e−03	1.13e−03	1.24e−02
ULT	1.70e−02	9.30e−04	4.02e−03	3.75e−03	1.13e−03	1.44e−02
PPM	9.19e−03	1.21e−03	2.77e−03	4.87e−03	9.11e−04	1.92e−02
ZDL	1.90e−02	7.00e−04	6.08e−03	2.75e−03	2.23e−03	1.08e−02
WENO	2.02e−02	4.20e−03	5.87e−03	1.80e−02	1.21e−03	7.20e−02

Table 9
Results for narrow semi-ellipse profile with $\lambda = 0.05$

Scheme	2b		2d		2f	
	Error	CPU (s)	Error	CPU (s)	Error	CPU (s)
MUSCL	3.14e−02	7.20e−03	8.29e−03	2.88e−02	3.31e−03	1.19e−01
COX	2.42e−02	7.30e−03	5.69e−03	2.90e−02	1.64e−03	1.24e−01
ULT	2.42e−02	9.00e−03	5.69e−03	3.60e−02	1.64e−03	1.61e−01
PPM	1.48e−02	1.19e−02	3.02e−03	4.76e−02	1.15e−03	1.96e−01
ZDL	2.19e−02	6.80e−03	8.00e−03	2.68e−02	2.32e−03	1.07e−01
WENO	1.96e−02	4.20e−02	5.82e−03	1.72e−01	1.23e−03	7.04e−01

Table 10
Results for square profile with $\lambda = 0.5$

Scheme	3a		3c		3e	
	Error	CPU (s)	Error	CPU (s)	Error	CPU (s)
MUSCL	2.97e−02	7.50e−04	1.14e−02	2.87e−03	4.90e−03	1.16e−02
COX	2.66e−02	7.40e−04	1.08e−02	2.87e−03	4.94e−03	1.16e−02
ULT	2.66e−02	9.20e−04	1.08e−02	3.75e−03	4.94e−03	1.46e−02
PPM	2.17e−02	1.22e−03	9.90e−03	4.87e−03	4.90e−03	1.94e−02
ZDL	2.76e−02	7.00e−04	1.32e−02	2.75e−03	5.65e−03	1.08e−02
WENO	3.06e−02	4.20e−03	1.35e−02	2.00e−02	5.25e−03	7.00e−02

Table 11
Results for square profile with $\lambda = 0.05$

Scheme	3b		3d		3f	
	Error	CPU (s)	Error	CPU (s)	Error	CPU (s)
MUSCL	3.71e−02	7.20e−03	1.52e−02	2.98e−02	6.34e−03	1.19e−01
COX	3.12e−02	7.40e−03	1.27e−02	2.92e−02	5.07e−03	1.24e−01
ULT	3.12e−02	8.90e−03	1.27e−02	3.62e−02	5.07e−03	1.61e−01
PPM	2.49e−02	1.19e−02	1.01e−02	4.80e−02	4.93e−03	1.97e−01
ZDL	3.00e−02	6.70e−03	1.66e−02	2.66e−02	6.36e−03	1.07e−01
WENO	3.04e−02	4.20e−02	1.35e−02	1.80e−01	5.29e−03	7.00e−01

Table 12
Results for triangle profile with $\lambda = 0.5$

Scheme	4a		4c		4e	
	Error	CPU (s)	Error	CPU (s)	Error	CPU (s)
MUSCL	2.21e−02	7.40e−04	4.10e−03	3.00e−03	7.16e−04	1.16e−02
COX	1.94e−02	7.30e−04	3.13e−03	3.00e−03	5.06e−04	1.16e−02
ULT	1.94e−02	9.60e−04	3.13e−03	3.63e−03	5.06e−04	1.46e−02
PPM	1.72e−02	1.19e−03	2.01e−03	4.75e−03	3.21e−04	1.94e−02
ZDL	3.48e−02	6.80e−04	8.53e−03	2.63e−03	2.76e−03	1.08e−02
WENO	2.46e−02	4.10e−03	1.89e−03	1.70e−02	3.10e−04	7.10e−02

Table 13
Results for triangle profile with $\lambda = 0.05$

Scheme	4b		4d		4f	
	Error	CPU (s)	Error	CPU (s)	Error	CPU (s)
MUSCL	2.63e−02	7.30e−03	8.00e−03	2.88e−02	2.65e−03	1.19e−01
COX	2.14e−02	7.30e−03	4.94e−03	2.90e−02	8.27e−04	1.25e−01
ULT	2.14e−02	9.00e−03	4.94e−03	3.62e−02	8.28e−04	1.62e−01
PPM	1.66e−02	1.19e−02	3.75e−03	4.78e−02	7.47e−04	1.96e−01
ZDL	2.13e−02	6.70e−03	6.79e−03	2.68e−02	2.38e−03	1.07e−01
WENO	1.79e−02	4.10e−02	1.89e−03	1.72e−01	3.48e−04	7.04e−01

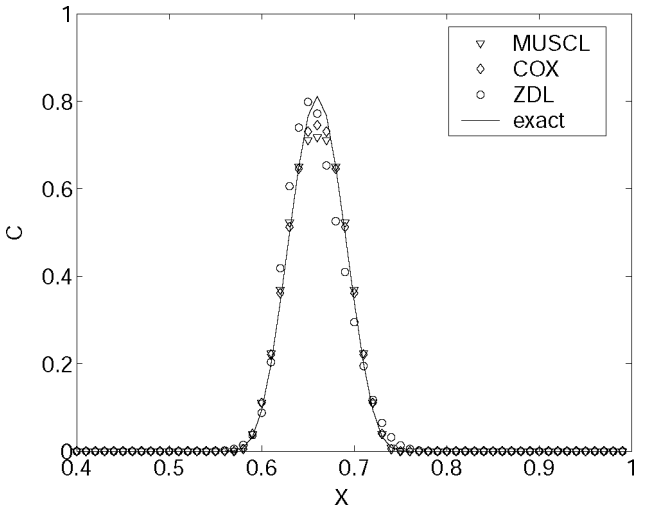

Fig. 4. Solution profiles Run 4c.

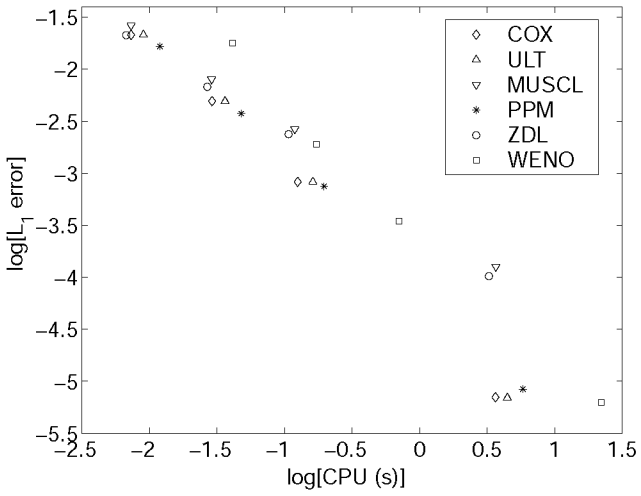

Fig. 6. Efficiency for triangle initial condition, $\lambda = 0.05$.

To examine efficiency, we recorded log error-CPU results from the second set of numerical tests. Fig. 5 is representative of the results from the semi-ellipse and square wave initial condition tests. In general, the trends from the initial experiments carry over. ZDL, MUSCL, and COX had comparable CPU times and were consistently the cheapest schemes. ULT was consistently slower. PPM was again slower than ULT but was sufficiently accurate, particularly on the coarser grids, to make it competitive in terms of efficiency. On the other hand, WENO was measurably slower than the other methods and no more accurate than PPM. The exception to this were computations with the triangle initial condition and $\lambda = 0.05$ (see Fig. 6). Here, WENO was more accurate than PPM and so more competitive for the 100 and 200 cell grids. While ZDL and MUSCL were the least accurate schemes in general, Fig. 6 also represents the test for which the deficiency in their accuracy was the most severe.

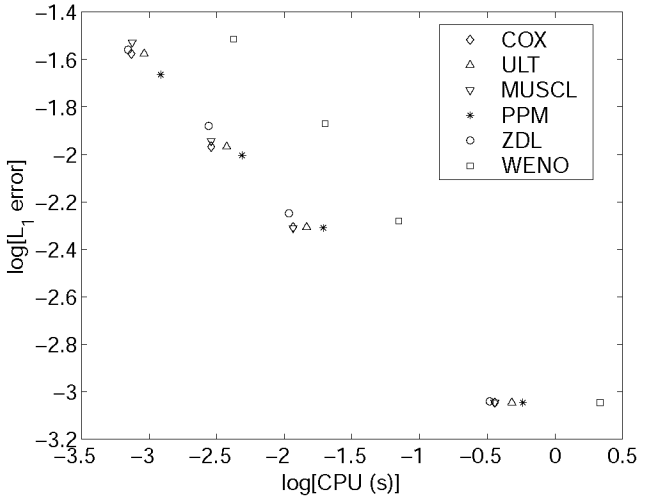

Fig. 5. Efficiency for square initial condition, $\lambda = 0.5$.

5. Discussion

We began our numerical experiments with the goal of identifying the most efficient and robust methods from those introduced in Section 3.5. We chose a set of model problems designed to test the basic properties necessary for the accurate solution ADE problems. The initial, baseline tests were run at low values of *Pe* number with sharp fronts resolved over a relatively large number of cells. These tests produced a significant subset of the candidate schemes that can be expected to perform well under less demanding conditions. The subsequent, more difficult numerical experiments allowed us to differentiate further among a narrower group of the most promising adaptive–stencil schemes. While the performance of each of the remaining methods: MUSCL, COX, ULT, PPM, ZDL, and WENO was generally good for the final tests, several distinctions can still be made.

The accuracy of COX and ULT was identical. The advantage of COX in terms of computational expense was about 20% which made it the most efficient scheme for the test problems considered. Thus for one-dimensional problems of the type considered in this work, COX is an attractive method. If, however, multidimensional problems with variable velocity fields are the target application, some caution is in order before choosing COX, since:

1. The coefficients used in this work depend upon λ and *Pe* and thus need to be recalculated at each computational point, which adds to the computational expense.

2. The third-order accuracy is lost for the case of variable velocity fields.
3. The form of the optimal parameter forms was determined for one-dimensional problems and would likely not be optimal for multidimensional cases [9].

It should also be cautioned that the relative efficiency of methods may vary from platform to platform, so the difference between COX and ULT may not be very significant.

WENO was markedly slower than the other methods, and its accuracy for these problems did not offset the added computational expense. If numerical efficiency is a concern, WENO would then be an unwise choice as an advection discretization. It has been shown, however, that there are problems in gas dynamics where the characteristics of the solution are well suited to WENO [14,24].

Both ZDL and MUSCL proved to be relatively simple and inexpensive. Each encountered conditions where its relative accuracy was severely degraded, however. The asymmetry of ZDL proved was a significant problem for the triangle initial condition, while MUSCL was much more diffusive than the other five methods for $\lambda = 0.05$. Both of these situations represent, in some sense, extreme cases but should be considered before using either ZDL or MUSCL.

Aside from our comments above about COX, the most efficient, reliable methods were ULT and PPM. ULT was consistently accurate with moderate expense. PPM was slower, but it proved to be the most accurate of the reviewed schemes. This reduced error, especially on the coarser grids, compensated for the increased CPU times.

While we have tried to make our methodology and conclusions as insensitive as possible to computational platform, there are questions that can arise based on platform-specific behavior. The WENO scheme, for example, was originally designed with vector architectures in mind. While the importance of vector computers has perhaps diminished in recent years, performance on these platforms is still relevant for numerical simulation in the water resources field. We thus evaluated the speed of WENO, PPM, and MUSCL on a Cray T-90 (see Table 2) for a simple model problem (Run 1g in Table 5). The WENO scheme vectorized extremely well, achieving 1139 megaflops. The original MUSCL implementation also performed well, however, achieving 667 megaflops and required about one-third of the total time used by WENO. On the other hand, the original PPM scheme did not vectorize well. This problem was alleviated by using the sign intrinsic to replace the conditionals used in the three constraints. This was done simply by replacing if (a), then $x = y$ with $x = x[1 + \text{sign}(a)]/2 + y[1 - \text{sign}(a)]/2$. The modified implementation of PPM achieved 1077 megaflops and required about two-thirds less total CPU time than WENO. This exercise indicates that WENO is still no more competitive than PPM on a vector platform for which it was originally designed. Furthermore, the same basic modifications applied to PPM are applicable for the other adaptive–stencil methods reviewed here.

There are a number of issues relevant to the solution of realistic ADE problems with adaptive-stencil methods that fell outside the scope of this work. As mentioned in Section 3, there are several ways to extend these methods to multiple space dimensions. These are often based on core operations that are one-dimensional [7,36,49], suggesting that the most promising one-dimensional schemes are a wise starting point for work in two and three dimensions.

A number of higher-order methods were not included in our comparisons. For example, higher-order ENO schemes are possible, and an eighth-order version of FCT is used in [63]. Leonard also combines the universal limiter with third-, seventh-, and ninth-order approximations via an adaptive-stencil expansion technique [27]. Our results suggest that one interested in implementing a higher-order method should compare the results with PPM or ULT to gauge its value.

Many processes of interest to the water resources field can be described by non-linear PDE's with sharp front solutions (e.g., the ADE with non-linear reactive terms, multiphase flow). Solution of these problems presents its own set of difficulties that are not directly addressed by our work. We note only that the solution of scalar non-linear hyperbolic problems as well as non-linear hyperbolic systems is much better established for the MUSCL, PPM, and WENO schemes than it is for ZDL, ULT, or COX.

Lastly, the time discretizations employed in our work were explicit. There are problems for which an implicit discretization of the ADE is preferable, however. The combination of implicit time discretizations with adaptive–stencil methods requires consideration of additional issues not considered in this work [51,58,60] and warrants further investigation.

6. Conclusions

Based upon our consideration of 14 different explicit-in-time, finite-volume discretization methods for solving the ADE drawn from several scientific fields, we make the following conclusions:

1. A variety of alternatives to standard discretization schemes used to solve the ADE in the water resources field can result in more efficient solutions.
2. The improvement in efficiency of a solution is obtained by using computational stencils that adapt depending upon the profile of the dependent variable, including the existence of local extrema.

3. Operation counts and local truncation analysis can provide some guidance for method selection, but extensive numerical experiments are important as well.
4. Stencil adaption and order of approximation varies from scheme to scheme and poses computational efficiency implementation challenges that depend upon the computational platform being used.
5. A method's efficiency varies with the initial condition and discretization pattern, making it necessary to consider a broad class of problems typical of those of interest in real applications.
6. ULT and PPM are two higher-order TVD schemes that performed well for the test problems considered and in general were more efficient than the best lower order TVD method (MUSCL) and the three UNO and ENO schemes evaluated.
7. Issues associated with the extension of this work to other problems in the field are considered.

Acknowledgements

The authors appreciate the valuable suggestions of Dr. Simon N. Gleyzer and Dr. John A. Trangenstein. This work supported in part by US Army Waterways Experiment Station Contract DACA39-95-K-0098, and National Institute of Environmental Health Sciences grant 5 P42 ES05948. MWF was supported by a Department of Energy Computational Science Fellowship. Computing activity was partially supported by allocations from the North Carolina Supercomputing Center.

References

[1] Barry DA, Miller CT, Culligan-Hensley PJ. Temporal discretization errors in non-iterative split-operator approaches to solving chemical reaction/groundwater transport models. J Contam Hydrol 1996;22(1/2):1–17.
[2] Bentley L, Pinder G. Eulerian–Lagrangian solution of the vertically averaged groundwater transport equation. Water Resour Res 1992;28(11):3011–20.
[3] Binning P, Celia M. A finite-volume Eulerian–Lagrangian localized adjoint method for solution of the contaminant transportequations in two-dimensional multiphase flow systems. Water Resour Res 1996;32(1):103–14.
[4] Book D, Boris J, Hain KH. Flux-corrected transport II, generalization of the method. J Comput Phys 1975;18:248–83.
[5] Boris J, Book D. Flux-corrected transport I, SHASTA, a fluid transport algorithm that works. J Comput Phys 1973;11:38–69.
[6] Brown LB, Pope GA, Abriola LM, Sepehrnoori K. Simulation of surfactant-enhanced aquifer remediation. Water Resour Res 1994;30(11):2959–77.
[7] Colella P. Multidensional upwind methods for hyperbolic conservation laws. J Comput Phys 1990;54:171–200.
[8] Colella P, Woodward P. The piecewise parabolic method (PPM) for gas dynamic simulations. J Comput Phys 1984;54:174–201.
[9] Cox R, Nishikawa T. A new total variation diminishing scheme for the solution of advective-dominant solute transport. Water Resour Res 1991;27(10):2645–54.
[10] Dahle HK, Espedal MS, Ewing RW. Characteristic Petrov–Galerkin subdomain methods for convection–diffusion problems. In: Wheeler MF, editor. Numerical simulation in oil recovery. New York: Springer; 1988. p. 77–87.
[11] Godlewski E, Raviart P-A. Numerical approximation of hyperbolic systems of conservation laws. New York: Springer; 1996.
[12] Godunov S. A finite-difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. Mat Sb 1959;47:428–41.
[13] Han T, Humphrey JAC, Launder BE. A comparison of hybrid and quadratic-upstream differencing in high Reynolds number elliptic flows. Comput Meth Appl Mech Eng 1981;29:81–95.
[14] Hannappel R, Hauser T, Friedrich R. A comparison of ENO and TVD schemes for the computation of shock turbulence interaction. J Comput Phys 1995;121:176–84.
[15] Harten A. High resolution schemes for hyperbolic conservation laws. J Comput Phys 1983;49:357–93.
[16] Harten A. ENO schemes with subcell resolution. J Comput Phys 1989;83:148–84.
[17] Harten A, Engquist B, Osher S, Chakravarthy S. Uniformly high-order accurate essentially non-oscillatory schemes III. J Comput Phys 1987;71:231–303.
[18] Harten A, Osher S. Uniformly high-order accurate nonoscillatory schemes. SIAM J Numer Anal 1987;24:279–309.
[19] Healy R, Russel T. A finite-volume Eulerian–Lagrangian localized adjoint method for solution of the advection–dispersion equation. Water Resour Res 1993;29(7):2399–413.
[20] Helmig R. Multiphase flow and transport processes in the subsurface. Berlin: Springer; 1997.
[21] Hirsch C. Numerical computation of internal and external flows. New York: Wiley; 1988.
[22] Huynh H. Second-order accurate nonoscillatory schemes for scalar conservation laws. In: Proceedings of the Sixth International Conference on Numerical Methods in Laminar and Turbulent Flows. UK: Pineridge; 1989, p. 25–38.
[23] Jiang G-S. Algorithm analysis and efficient computation of conservation law. Providence (RI): Brown University; 1995.
[24] Jiang G-S, Shu C-W. Efficient implementation of weighted ENO schemes. J Comput Phys 1996;126:202–28.
[25] Khan SA, Pope GA, Trangenstein JA. Micellar/polymer physical-property models for contaminant cleanup problems and enhanced oil recovery. Technical report, Department of Mathematics, Duke University, Durham, NC, November 1994.
[26] Leonard B. A stable and accurate convective modeling procedure based on quadratic upstream interpolation. Comput Meth Appl Mech Eng 1979;19:59–89.
[27] Leonard B. The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection. Comput Meth Appl Mech Eng 1991;88:17–74.
[28] LeVeque R. Numer Meth Conserv Laws. Boston: Birkhäuser; 1992.
[29] Lichtner PC, Steefel CI, Oelkers EH. Reactive transport in porous media: general principles and applications to geochemical processes. Washington (DC): Mineralogical Society of America; 1996.
[30] Liu J, Delshad M, Pope G, Sepehrnoori K. Application of higher order flux-limited methods in compositional simulation. Transp Porous Media 1994;16(1):1–29.
[31] Liu X-D, Osher S. Convex ENO high-order multi-dimensional schemes without field by field decomposition or staggered grids. J Comput Phys 1998;142(2).
[32] Liu X-D, Osher S, Chan T. Weighted essentially non-oscillatory schemes. J Comput Phys 1994;115:200–12.
[33] McDonald E, Ambrosiano J. High-order upwind flux correction methods for hyperbolic conservation laws. J Comput Phys 1984;56:448–60.
[34] Miller CT, Christakos G, Imhoff PT, McBride JF, Pedit JA, Trangenstein JA. Multiphase flow and transport modeling in

heterogeneous porous media: challenges and approaches. Adv Water Resour 1998;21(2):77–120.
[35] Miller CT, Gleyzer SN, Imhoff PT. Napl dissolution fingering in porous media. In: Selim M, Ma L, editors, Physical nonequilibrium in soils: modeling and application. Ann Arbor (MI): Ann Arbor Press (in review), 1997.
[36] Minion ML. On the stability of Godunov-projection methods for incompressible flow. J Comput Phys 1996;123:435–49.
[37] Morton KW. Numerical solution of convection–diffusion problems. London (UK): Chapman and Hall; 1996.
[38] Neuman S. A Eulerian–Lagrangian numerical scheme for the dispersion–convection equation using conjugate space–time grids. J Comput Phys 1981;41(2):270–94.
[39] Poulain CA, Finlayson BA. A comparison of numerical methods applied to non-linear adsorption columns. Inter J Meth Fluids 1993;17:839–59.
[40] Putti M, Yeh WWG, Mulder WA. A triangular finite-volume approach with high-resolution upwind terms for the solution of groundwater transport equations. Water Resour Res 1990;26(12):2865–80.
[41] Reitsma S, Kueper B. Non-equilibrium alcohol flooding model for immiscible phase remediation: 2. Model development and application. Adv Water Resour 1998;21(8):663–78.
[42] Roe P. Some contribution to the modeling of discontinuous flows. In: Enquist BE, Osher S, Somerville RJ, editors. The lectures in applied mathematics, part 2, vol. 22. Providence (RI): American Mathematical Society; 1985. p. 163–94.
[43] Rood RB. Numerical advection algorithms and their role in atmospheric transport and chemistry models. Rev Geophys 1987;25(1):71–100.
[44] Russel T. Eulerian–Lagrangian localized adjoint methods for advection dominated problems. In: Griffiths DF, Watson GA, editors. Numerical analysis 1989 Pitman res. notes math. ser., vol. 228. Harlow (UK): Longman; 1990, p. 206–28.
[45] Shu C, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes I. J Comput Phys 1998;77(1):439–71.
[46] Shu C, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes II. J Comput Phys 1989;83:32–78.
[47] Sleep BE, Sykes JF. Compositional simulation of groundwater contamination by organic compounds, 1. Model development and verification. Water Resour Res 1993;29(6):1697–708.
[48] Sweby P. High-resolution schemes using flux limiters for hyperbolic conservation laws. SIAM J Numer Anal 1984;21: 995–1011.
[49] Toro EF. Riemann solvers and numerical methods for fluid dynamics. London: Springer; 1998.
[50] Trangenstein J. Numerical solution of partial differential equations. Lecture notes.
[51] Unger A, Forsyth P, Sudicky E. Variable spatial and temporal weighting schemes for use in multi-phase compositional problems. Adv Water Resour 1996;19(1):1–27.
[52] Unger A, Forsyth V, Sudicky E. Nonlinear iteration methods for non-equilibrium multiphase subsurface flow. Adv Water Resour 1998;21:433–49.
[53] VanLeer B. Towards the ultimate conservative difference scheme a second-order sequel to Godunov's method. J Comput Phys 1979;32(1):101–36.
[54] Wang H, Ewing R, Russel T. Eulerian–Lagrangian localized adjoint methods for convection–diffusion equations. IMA J Numer Anal 1995;15(3):405–59.
[55] Yakirevich A, Borisov V, Sorek S. A quasi three-dimensional model for flow and transport in unsaturated and saturated zones: 1. Implementation of the quasi two-dimensional case. Adv Water Resour 1998;21(8):679–89.
[56] Yang H. An artificial compression method for ENO schemes: the slope modification method. J Comput Phys 1990;89:125–60.
[57] Yang H, Przekwas A. A comparative study of advanced shock-capturing schemes applied to Burgers' equation. J Comput Phys 1992;102:139–59.
[58] Yee H. Explicit and implicit multidimensional compact high-resolution shock-capturing methods. AIAA J 1989;27(3):299–307.
[59] Yee H, LeVeque R. A study of numerical-methods for hyperbolic conservation-laws with stiff source terms. J Comput Phys 1990;86(1):187–210.
[60] Yee H, Shinn J. A class of high-resolution explicit and implicit shock-capturing methods. AIAA J 1989;27(3):299–307.
[61] Yost A, Rao P. A non-oscillatory scheme for open channel flows. Adv Water Resour 1998;22(2):133–43.
[62] Zaidel Y, Levi B. Numerical schemes for two-phase three-component flows in porous media. Tchislen. Met. Mechan. Sploshn. Sredy Sb. nautch. trudov 1980;11(5):75–89 (in Russian).
[63] Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. J Comput Phys 1979;31:335–62.
[64] Zalesak ST. A preliminary comparison of modern shock-capturing schemes: linear advection. In: Vichnevetsky R, Spelman RS, editors, Advances in computer methods for partial differential equations, vol. VI. IMACS Rutgers University. 1987, p. 15–22.
[65] Zheng C. A Modular three-dimensional transport model for simulation of advection dispersion and chemical reactions of contaminants in groundwater systems. US EPA Robert S. Kerr Environmental Research Laboratory, Ada, OK, 1990.
[66] Zheng C, Bennett GD. Applied contaminant transport modeling. New York: Van Nostrand Reinhold; 1995.
[67] Zheng C, Wang P. MT3DMS: a modular three-dimensional multispecies transport model for simulation of advection dispersion and chemical reactions of contaminants in groundwater systems. Technical Report, Waterways Experiment Station, US Army Corps. of Engineers, Vicksburg, MS 39187, 1998.