OCCAM

OXFORD CENTRE FOR COLLABORATIVE APPLIED MATHEMATICS

Spatially Partitioned Embedded Runge-Kutta Methods

by

David I. Ketcheson
Colin B. Macdonald
Steven J. Ruuth

OCCAM Preprint Number 13/04

SPATIALLY PARTITIONED EMBEDDED RUNGE–KUTTA METHODS

DAVID I. KETCHESON*, COLIN B. MACDONALD†, AND STEVEN J. RUUTH‡

Abstract. We study spatially partitioned embedded Runge–Kutta (SPERK) schemes for partial differential equations (PDEs), in which each of the component schemes is applied over a different part of the spatial domain. Such methods may be convenient for problems in which the smoothness of the solution or the magnitudes of the PDE coefficients vary strongly in space. We focus on embedded partitioned methods as they offer greater efficiency and avoid the order reduction that may occur in non-embedded schemes. We demonstrate that the lack of conservation in partitioned schemes can lead to non-physical effects and propose conservative additive schemes based on partitioning the fluxes rather than the ordinary differential equations. A variety of SPERK schemes are presented, including an embedded pair suitable for the time evolution of fifth-order weighted non-oscillatory (WENO) spatial discretizations. Numerical experiments are provided to support the theory.

Key words. Embedded Runge–Kutta methods, spatially partitioned methods, conservation laws, method of lines.

AMS subject classifications. 35L65, 65L06, 65M20

1. Introduction.

The method-of-lines is a popular discretization technique for the numerical solution of time-dependent partial differential equations. In it, a spatial discretization is applied to the PDE, yielding an initial value problem consisting of a large system of ordinary differential equations (ODEs). These are evolved by some time-stepping method, for example, a Runge–Kutta method.

The choice of a suitable time-stepping scheme may depend on a variety of considerations. Typically, schemes are chosen to give good linear stability and accuracy, although in some applications it is also useful or necessary to preserve the monotonicity or other properties of the true PDE solution. Often, the key properties for determining a suitable time-stepping scheme vary locally in space. Examples of such properties include the grid spacing, the coefficients of the PDE, the smoothness of the solution and the geometry of the domain. As a consequence, it is possible that a scheme that is effective in one portion of the domain is unsuitable or inefficient in another. It is therefore natural to consider the development and analysis of spatially partitioned time-stepping methods, in which different step sizes or different methods are used over subdomains.

A class of methods that often benefit from spatially partitioned time-stepping are PDE discretizations with grid adaptivity. In regions where the solution is nonsmooth or exhibits rapid variation, fine grids are needed; other regions may be more efficiently discretized with coarser meshes. If all components of an ODE system are evolved using some explicit time-stepping scheme and a single time step-size Δt, the evolution of all components will be restricted by the stiffest components of the system.

*4700 King Abdullah University of Science & Technology, Thuwal 23955, Saudi Arabia (david.ketcheson@kaust.edu.sa). The work of this author was supported by funding from King Abdullah University of Science and Technology (KAUST).

†Mathematical Institute, University of Oxford, OX1 3LB, UK (macdonald@maths.ox.ac.uk). The work of this author was supported by NSERC Canada and Award No KUK-C1-013-04 made by King Abdullah University of Science and Technology (KAUST)

‡Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada (sruuth@sfu.ca). The work of this author was partially supported by a grant from NSERC Canada and by award KUK-C1-013-04 made by King Abdullah University of Science and Technology (KAUST).

Improved efficiency is often possible by considering time-stepping methods that vary their time step-size according to the local mesh spacing. Time-stepping schemes of this type are called *multirate* schemes. The first multirate schemes for one-dimensional conservation laws were developed by Osher and Sanders in [14]. Their approach carries out forward Euler time-stepping with step-sizes that vary locally. More recently higher-order methods have been considered; for example, Tang and Warnecke [22] develop second-order multirate schemes based on Runge–Kutta or Lax-Wendroff type schemes. Other recent work includes that by Constantinescu and Sandu [2] where a simple construction algorithm is given to form a second-order accurate multirate scheme based on an arbitrary strong-stability-preserving (SSP) scheme of order two or higher. Notably, their approach preserves a variety of monotonicity properties, such as positivity and the maximum principle. As explained by Hundsdorfer, Mozartova and Savcenco [7] multirate schemes for conservation laws are either locally inconsistent (e.g., [14]) or lack mass conservation (e.g., [22]). Fortunately, order reduction due to such local inconsistency may be less severe than expected, due to cancellation and damping effects; see [7] for details.

Whereas multirate methods use different step sizes, in the present work we focus on using different methods on different spatial subdomains. Specifically, we investigate spatially partitioned embedded Runge–Kutta (SPERK) schemes applied to one-dimensional conservation laws

$$u_t + f(u)_x = 0. \tag{1.1}$$

Here the term *embedded* refers to methods having the same coefficient matrix A (but different weights b), which avoids the unnecessary duplication of computations that can occur when combining two unrelated time-stepping schemes. Two classes of SPERK schemes are considered: *equation-based partitioning* and *flux-based partitioning*. We shall find that equation-based partitioning maintains the overall accuracy of the schemes composing the embedded pair. This approach is, however, not conservative and can lead to wrong shock propagation speeds when applied to hyperbolic PDEs. Flux-based partitioning is conservative, but experiences a theoretical order reduction in the local consistency. In practice, however, we shall find that the overall accuracy of the combined scheme is unaffected.

We will see that the methods we study can be viewed in the general framework of partitioned or additive Runge–Kutta methods. However, such methods are usually applied in a way that applies different numerical treatment to *different physical processes*, whereas the emphasis here is on different numerical treatments for *different spatial domains*. These approaches are based on different motivations, and some approaches that work well for the former fail for the latter (see Example 2.3).

As an example, consider a convection-diffusion problem in which the Reynolds number varies spatially, such that the system is dominated by convection on one subdomain and dominated by diffusion elsewhere. The third-order, three-stage SSP Runge–Kutta scheme of Shu and Osher (SSPRK(3,3)) [20] might be desired for the convection-dominated regions, while a second-order Runge–Kutta–Chebyshev method, e.g., [24], might be preferred in diffusion-dominated regions. Unfortunately, neither scheme is particularly attractive on its own since SSPRK(3,3) requires small time steps when applied to diffusive problems, while the Runge–Kutta–Chebyshev method is unstable when applied to convective problems. On the other hand, a combination that applies each time-stepping scheme where it is best suited may provide better linear stability than either scheme on its own. An example of this kind is explored in Section 4.

There are other situations where spatially partitioned time-stepping schemes show a strong potential. For example, consider the evolution of a conservation law involving both shocks and smooth regions. The preservation of monotonicity properties may be the most crucial property near the shock, whereas high-order accuracy may be of primary interest in smooth regions. Use of a spatially partitioned time-stepping scheme opens the possibility of simultaneously obtaining good accuracy and monotonicity in such problems. An example of this kind is explored in Section 5, where SPERK schemes are applied to fifth-order weighted essentially non-oscillatory (WENO) spatial discretizations [19]. The motivation here is that fifth-order SSPRK methods are complicated by their use of the downwind-biased operator [20, 18, 21, 16] while monotonicity and the corresponding SSP property is likely only useful in the vicinity of non-smooth features. In these regions WENO discretizations are formally third-order accurate [19]. Thus, in our approach, a fifth-order linearly stable Runge–Kutta scheme is used in smooth regions while an embedded third-order SSP Runge–Kutta scheme is used near shocks or other discontinuities.

The paper unfolds as follows. Section 2 introduces equation-based partitioning and examines its conservation properties. This is followed by the introduction and analysis of flux-based partitioning. Errors and positivity properties for both classes of schemes are considered in this section. A variety of generalizations are given in Section 3. Section 4 gives some examples of SPERK schemes, and considers applications of the methods to a spatially discretized advection-diffusion equation. Section 5 considers SPERK schemes in the context of WENO spatial discretizations and a nonlinear partitioning step. In our approach, the WENO weights are used to select between an SSP Runge–Kutta scheme, and a high-order scheme chosen for its linear stability properties. Finally, Section 6 concludes with a discussion of some other application areas and some of our current research directions.

2. Spatially partitioned embedded Runge–Kutta methods. We begin by introducing equation- and flux-based SPERK schemes. An analysis of the accuracy, conservation and positivity properties of such schemes is also provided in this section.

2.1. Equation-based partitioning. Consider a system of N ordinary differential equations

$$U'(t) = G(U), \tag{2.1}$$

typically arising as the spatial discretization of a PDE where each component in the solution approximates, for example, point values of the PDE solution $u_i \approx u(x_i)$ at discrete points $x_i, 1 \le i \le N$. To apply an s-stage Runge–Kutta method, we first compute the stage values

$$Y^{(j)} = U^n + \Delta t \sum_{k=1}^{s} a_{jk} G(Y^{(k)}), \quad j = 1, \ldots, s. \tag{2.2a}$$

A standard Runge–Kutta method would then advance by one step using the formula $U^{n+1} = U^n + \Delta t \sum_{j=1}^{s} b_j G(Y^{(j)})$. Instead, let a different set of weights be applied at each point x_i, $1 \le i \le N$, choosing between coefficients b or $\hat{b}$. This results in

$$u_i^{n+1} = u_i^n + \Delta t \left[\chi_i^n \sum_{j=1}^{s} b_j g_i(Y^{(j)}) + (1 - \chi_i^n) \sum_{j=1}^{s} \hat{b}_j g_i(Y^{(j)}) \right] \tag{2.2b}$$

where g_i is the ith component of G and

$$\chi_i^n = \begin{cases} 1, & \text{if weights } b \text{ are used to compute } u_i^{n+1}, \\ 0, & \text{if weights } \hat{b} \text{ are used to compute } u_i^{n+1}. \end{cases} \tag{2.2c}$$

We write the coefficients of this method A, b, $\hat{b}$ using the same tableau notation that is employed for embedded Runge–Kutta methods:

$$\begin{array}{c|c} c & A \\ \hline & b^T \\ & \hat{b}^T \end{array}. \tag{2.3}$$

We refer to the embedded method (2.2) as a *spatially partitioned* time-stepping method because the "mask" χ selects which scheme to propagate at each point in space, at each time step. In some cases, χ depends on the solution values, although we do not explicitly represent this for notational clarity.

2.1.1. Connection to partitioned Runge–Kutta methods. Methods of the form (2.2) form a special subclass of partitioned Runge–Kutta methods [5]. Generally, methods in a partitioned RK pair may have different coefficient matrices A as well as different weights b. In our embedded approach, methods have fewer degrees of freedom available for their design, but they possess the advantage of automatically satisfying the "extra" order conditions for partitioned RK methods. That is, if each of the component methods (A, b) and $(A, \hat{b})$ is accurate to order p, the SPERK method is also accurate to order p in time (see also Proposition 3.1 below).

Method (2.2) partitions (2.1) by equation; we refer to this type of partitioning as *equation-based partitioning*. Because the ith equation corresponds to grid node x_i, $1 \le i \le N$, our approach also gives a spatial partitioning of a semi-discretized PDE. It is worth noting, however, that equation-based partitioning is a very general technique that does not require any correspondence to grid locations.

2.1.2. Conservation. Many important physical phenomena are modeled by conservation laws, which in one dimension have the form

$$u_t + f(u)_x = 0. \tag{2.4}$$

In the numerical solution of (2.4), one should use a conservative scheme in order to ensure that shocks propagate at the correct speed. Typically, (2.4) is semi-discretized using a flux-differencing method:

$$u_i'(t) = -\frac{1}{\Delta x}\left(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}\right) \tag{2.5}$$

where $f_{i\pm\frac{1}{2}}$ are numerical approximations to the flux at $x_{i\pm\frac{1}{2}}, 1 \le i \le N$. Integrating with a Runge–Kutta method gives

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(\sum_{j=1}^{s} b_j f_{i+\frac{1}{2}}(Y^{(j)}) - \sum_{j=1}^{s} b_j f_{i-\frac{1}{2}}(Y^{(j)})\right)$$

where the stage values $Y^{(j)}, 1 \le j \le s$, are defined in equation (2.2a) above. This method is conservative since corresponding fluxes cancel out (except at the boundaries) if we sum over the spatial index i, $1 \le i \le N$.

Applying method (2.2) gives instead

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(\sum_{j=1}^{s}(\chi_i^n b_j + (1-\chi_i^n)\hat{b}_j) f_{i+\frac{1}{2}}(Y^{(j)}) - \sum_{j=1}^{s}(\chi_i^n b_j + (1-\chi_i^n)\hat{b}_j) f_{i-\frac{1}{2}}(Y^{(j)})\right).$$

This method is not conservative, since the flux terms at $x_{i-\frac{1}{2}}$ will not cancel if $\chi_{i-1}^n \neq \chi_i^n$. This can lead to solutions in which discontinuities move at incorrect speeds. Here we give two examples.

EXAMPLE 2.1. *Consider the advection equation $u_t + u_x = 0$ discretized in space by upwind differencing:*

$$u_i'(t) = g_i(U) \tag{2.6}$$

where $g_i(U) = -\frac{1}{\Delta x}(u_i - u_{i-1})$. Let $G = [g_i(U)]$ and discretize in time with equation-based partitioning:

$$\begin{aligned} Y^{(1)} &= U^n, \\ Y^{(2)} &= U^n + \Delta t G(Y^{(1)}), \\ u_i^{n+1} &= u_i^n + \Delta t\left(\chi_i^n \cdot g_i(Y^{(1)}) + (1-\chi_i^n)\cdot g_i(Y^{(2)})\right). \end{aligned} \tag{2.7}$$

The scheme (2.7) is a SPERK scheme and can be represented in the tableau form (2.3) as

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1 & 0 \\ & 0 & 1 \end{array}.$$

Consider the Cauchy problem on the real line with a step function initial condition:

$$u(x, t=0) = \begin{cases} 1 & x \leq 0, \\ 0 & x > 0. \end{cases}$$

Suppose $x_i = i$, $-\infty < i < \infty$, $t_n = n$, $n = 0, 1, \ldots$ and for the spatial partitioning take

$$\chi_i^n = \begin{cases} 1 & i + 2n \leq 1, \\ 0 & i + 2n > 1. \end{cases}$$

Direct computation shows that the numerical solution is

$$u_i^n = \begin{cases} 1 & i \leq 2n, \\ 0 & i > 2n. \end{cases}$$

This is not conservative since the discontinuity moves 2 grid cells per time step, whereas in the true solution it moves 1 grid cell per time step. Note that this example employs a scheme that is unstable for the given CFL number; the stability of

the solution results from the judicious choice of combination of methods, the initial condition, and the spatial partitioning. □

EXAMPLE 2.2. *Next we consider Burgers' equation,* $u_t + \left(\frac{1}{2}u^2\right)_x = 0$*, with the step function initial condition*

$$u(x, t = 0) = \begin{cases} 2 & x \le 0, \\ 0 & x > 0. \end{cases}$$

We discretize in space with a finite difference flux-differencing scheme using fifth-order WENO interpolation. In time, we discretize using the SPERK scheme in Table 5.1 and equation-based partitioning. We use a time step-size of $\Delta t = 0.6\Delta x$*, corresponding to a CFL number of 1.2. We take*

$$\chi(x, t) = \begin{cases} 0 & 0.01 < u(x, t) < 1.99, \\ 1 & \textit{otherwise.} \end{cases}$$

This simple choice of χ *ensures that the SSP method is used near the shock while the non-SSP method is used elsewhere. It also (unfortunately) ensures that the jumps in* χ *are near the shock, maximizing the effect of conservation errors. The true shock velocity is* 1*, but the numerical shock velocity converges to approximately* 0.925*. If we replace* χ *above by* $1 - \chi$*, the shock moves instead too rapidly.* □

2.2. Flux-based partitioning. In (2.2) we applied one of the RK schemes to each equation in the ODE system. In order to obtain a conservative method, we partition by the fluxes $f_{i+\frac{1}{2}}, 0 \le i \le N$, rather than by the equations.

Suppose we are given two Runge–Kutta methods with identical coefficient matrices A, but different weights b and $\hat{b}$. Similar to equation-based partitioning, we first compute stage values according to (2.2a), which when applied to the flux-differencing semi-discretization (2.5) is

$$y_i^{(j)} = u_i^n - \frac{\Delta t}{\Delta x} \sum_{k=1}^{s} a_{jk} \left(f_{i+\frac{1}{2}}(Y^{(j)}) - f_{i-\frac{1}{2}}(Y^{(j)}) \right), \quad j = 1, \dots, s. \tag{2.8a}$$

However, instead of varying the time-stepping method by equation, we vary the method by flux. An application of this partitioning to (2.5) yields

$$\begin{aligned} u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x} \Bigg(& \chi_{i+\frac{1}{2}}^n \sum_{j=1}^{s} b_j f_{i+\frac{1}{2}}(Y^{(j)}) + (1 - \chi_{i+\frac{1}{2}}^n) \sum_{j=1}^{s} \hat{b}_j f_{i+\frac{1}{2}}(Y^{(j)}) \\ & - \chi_{i-\frac{1}{2}}^n \sum_{j=1}^{s} b_j f_{i-\frac{1}{2}}(Y^{(j)}) - (1 - \chi_{i-\frac{1}{2}}^n) \sum_{j=1}^{s} \hat{b}_j f_{i-\frac{1}{2}}(Y^{(j)}) \Bigg), \end{aligned} \tag{2.8b}$$

where we have partitioned using the characteristic functions corresponding to *cell edges* $x_{i+\frac{1}{2}}$ (rather than grid points x_i)

$$\chi_{i+\frac{1}{2}}^n = \begin{cases} 1, & \text{if weights } b \text{ are used for fluxes at } x_{i+\frac{1}{2}}, \\ 0, & \text{if weights } \hat{b} \text{ are used fluxes at } x_{i+\frac{1}{2}}. \end{cases} \tag{2.8c}$$

Figure 2.1 compares a pseudo-code implementation of the equation-based (2.2) and flux-based partitioning (2.8) for a three-stage SPERK scheme.

As we show next, the flux-based method corresponds to an additive Runge–Kutta method instead of a partitioned Runge–Kutta method.

2.2.1. Connection to additive Runge–Kutta schemes. Consider again the flux-differencing formula (2.5). Let $\Phi(U)$ denote the vector of fluxes, with components $\phi_i(U) = f_{i+\frac{1}{2}}(U), 0 \le i \le N$. We can write the flux-differencing method (2.5) in vector form:

$$U'(t) = -\frac{1}{\Delta x} D\Phi, \tag{2.9}$$

where D is the $N \times (N+1)$ differencing matrix with 1 on the superdiagonal and -1 on the main diagonal.

We split the flux vector based on the vector $\chi \equiv [\chi_{\frac{1}{2}}, \chi_{\frac{3}{2}}, \dots, \chi_{N+\frac{1}{2}}]^T$:

$$\begin{aligned} U'(t) = -\frac{1}{\Delta x} D\Phi(U) &= -\frac{1}{\Delta x} D\big[\operatorname{diag}(\chi)\Phi(U) + (I - \operatorname{diag}(\chi))\Phi(U)\big], \\ &= \underbrace{-\frac{1}{\Delta x} D \operatorname{diag}(\chi)\Phi(U)}_{G_\chi(U)} \underbrace{-\frac{1}{\Delta x} D(I - \operatorname{diag}(\chi))\Phi(U)}_{G_{1-\chi}(U)}, \\ &= G_\chi(U) + G_{1-\chi}(U), \end{aligned} \tag{2.10}$$

where $G_\chi(U) = -\frac{1}{\Delta x} D \operatorname{diag}(\chi)\Phi(U)$, $G_{1-\chi}(U) = -\frac{1}{\Delta x} D(I - \operatorname{diag}(\chi))\Phi(U)$ and $\operatorname{diag}(\chi)$ is the $(N+1)\times(N+1)$ diagonal matrix which has χ as its main diagonal. Flux-based partitioning applies different Runge–Kutta methods to G_χ and $G_{1-\chi}$. Because we have chosen embedded pairs with identical coefficient matrices A, the stage values $Y^{(j)}, 1 \le j \le s$, are computed according to (2.2a) in the standard fashion. Different weights for G_χ and $G_{1-\chi}$ are applied, however, according to the formula

$$U^{n+1} = U^n + \Delta t \left[\sum_{j=1}^{s} b_j G_\chi(Y^{(j)}) + \sum_{j=1}^{s} \hat{b}_j G_{1-\chi}(Y^{(j)}) \right]. \tag{2.11}$$

This approach (i.e., using (2.11)) may be viewed as applying an additive Runge–Kutta method [9] to (2.10). We emphasize that the methods we consider have identical A matrices and therefore form a special *embedded* subclass of the additive RK methods. The effect of the splitting introduced in (2.10) is crucial for understanding the order of accuracy of flux-based SPERK schemes. Theoretically, we might expect a reduction of order by one, a result we establish in Theorem 2.4. In practice, however, we observe that the order of accuracy of SPERK schemes is the minimum order of the two schemes for both equation- and flux-based partitioning. Section 5 gives some numerical experiments illustrating this property.

2.3. Accuracy. In general, additive and partitioned Runge–Kutta methods may exhibit lower order convergence rates than either of their component methods. In order for the method to be fully accurate, additional order conditions relating the coefficients of the different component methods must be satisfied. Even then, in our context of spatial splitting, such methods can exhibit order reduction when the coefficient matrices A differ.

EXAMPLE 2.3. *Flux-based partitioning based on a characteristic function χ gives the ODE system (2.10). The classical Godunov operator splitting is a first-order method which may be applied to this system. Applying this splitting yields*

$$\begin{aligned} U^* &= U^n + \Delta t G_\chi(U^n), \\ U^{n+1} &= U^* + \Delta t G_{1-\chi}(U^*). \end{aligned}$$

```
# Program 1: Equation-based SPERK implementation
# First compute the standard stage values
Y^(1) = U
for i in range(1, N):
    y_i^(2) = u_i - dt/dx(a21(y_i^(1)-y_{i-1}^(1)))
for i in range(1, N):
    y_i^(3) = u_i - dt/dx(a31(y_i^(1)-y_{i-1}^(1)) + a32(y_i^(2)-y_{i-1}^(2)))
# Advance in time from u to unew using:
for i in range(1, N):
    unew_i = u_i - dt/dx( χ_i (b1(y_i^(1)-y_{i-1}^(1)) + b2(y_i^(2)-y_{i-1}^(2)) + b3(y_i^(3)-y_{i-1}^(3)))
                        + (1-χ_i)(b̂1(y_i^(1)-y_{i-1}^(1)) + b̂2(y_i^(2)-y_{i-1}^(2)) + b̂3(y_i^(3)-y_{i-1}^(3))))
```

```
# Program 2: Flux-based SPERK implementation
# First compute the standard stage values as in lines 3-7 above
...
# Then advance in time from u to unew using:
for i in range(1, N):
    unew_i = u_i - dt/dx(
          χ_{i+1/2} (b1 y_i^(1) + b2 y_i^(2) + b3 y_i^(3))   -   χ_{i-1/2} (b1 y_{i-1}^(1) + b2 y_{i-1}^(2) + b3 y_{i-1}^(3))
     + (1-χ_{i+1/2})(b̂1 y_i^(1) + b̂2 y_i^(2) + b̂3 y_i^(3)) - (1-χ_{i-1/2})(b̂1 y_{i-1}^(1) + b̂2 y_{i-1}^(2) + b̂3 y_{i-1}^(3)))
```

FIG. 2.1. *Pseudocode implementation of the equation-based and flux-based SPERK algorithms. Here we apply one step of an explicit three-stage method to the first-order upwinding spatial discretization of the advection PDE* $u_t + u_x = 0$.

This can be written as an additive Runge–Kutta method with coefficient arrays

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1 & 0 \end{array}, \qquad \begin{array}{c|cc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline & 0 & 1 \end{array}.$$

Unlike all other pairs considered in this paper, this pair uses different coefficients a_{ij}*. We apply them to the upwind differencing semidiscretization* (2.6) *of the linear advection equation with* $\Delta t = \Delta x$ *and constant initial data,* $u_i^0 = 1, -\infty < i < \infty$*. We partition the domain into two regions by choosing an integer* J *with* $\chi_{i-\frac{1}{2}} = 1$ *for* $i \le J$ *and* $\chi_{i-\frac{1}{2}} = 0$ *for* $i > J$*. After one step with this method, the computed solution is*

$$u_i^1 = \begin{cases} 1 & \text{for } i < J, \\ 0 & \text{for } i = J, \\ 2 & \text{for } i = J+1, \\ 1 & \text{for } i > J+1. \end{cases}$$

Despite the constant initial data, the local error is $\mathcal{O}(1)$ *in the maximum norm.* □

The difficulty encountered in the example above is typical of methods in which $A \neq \hat{A}$. In the next two theorems, we establish the accuracy of equation- and flux-based SPERK methods.

THEOREM 2.1. *Suppose that the partitioned Runge–Kutta method* (2.2) *with coefficients* $(A, b, \hat{b})$ *is applied to the semi-discretization* (2.1)*. Then the fully discretized system has a local order of accuracy equal to* $\min(p, \hat{p})$ *where* p *(respectively,* $\hat{p}$*) is the*

order of accuracy of the full discretization obtained by using the Runge–Kutta method with coefficients (A, b) *(respectively,* $(A, \hat{b})$*).*

Proof. The full discretizations are one-step methods of the form

$$w_i^{n+1} = Q_i(W^n),$$
$$\widehat{w}_i^{n+1} = \widehat{Q}_i(\widehat{W}^n).$$

The exact solution satisfies

$$u(x_i, t_{n+1}) = Q_i(u_h(t_n)) + \mathcal{O}(\Delta t^{p+1}), \tag{2.12a}$$
$$u(x_i, t_{n+1}) = \widehat{Q}_i(u_h(t_n)) + \mathcal{O}(\Delta t^{\hat{p}+1}). \tag{2.12b}$$

where $u_h(t_n) = [u(x_1, t_n), u(x_2, t_n), \ldots, u(x_N, t_n)]^T$ is a vector of true solution values at the grid nodes at time t_n. Here we have assumed that Δx is given by some prescribed relationship in terms of Δt, so that the error can be characterized in terms of Δt only.

We now determine the local truncation error of the discretization: starting from the exact solution at time t_n, the solution computed by the partitioned method is

$$u_i^{n+1} = \chi_i^n Q_i(u_h(t_n)) + (1 - \chi_i^n)\widehat{Q}_i(u_h(t_n)).$$

Applying (2.12), we find

$$\begin{aligned} u_i^{n+1} &= \chi_i^n \cdot u(x_i, t_{n+1}) + \mathcal{O}(\Delta t^{p+1}) + (1 - \chi_i^n) \cdot u(x_i, t_{n+1}) + \mathcal{O}(\Delta t^{\hat{p}+1}), \\ &= u(x_i, t_{n+1}) + \mathcal{O}(\Delta t^{\min(p,\hat{p})+1}). \end{aligned}$$

□

In order to prove accuracy of the flux-based decomposition approach, we need to know how accurately a Runge–Kutta method approximates the fluxes.

LEMMA 2.2. *Suppose we are given an initial value problem (2.1) for* $U \in \mathbb{R}^n$*, a Runge–Kutta method* (A, b) *of order* p*, and a smooth function* $V : \mathbb{R}^n \to \mathbb{R}^n$*. Let* $W(t) = \int_0^t V(U(s))ds$*,* $W^n = W(t_n)$ *and compute* W^{n+1} *using (2.2a) and*

$$W^{n+1} = W^n + \Delta t \sum_{j=1}^{s} b_j V(Y^{(j)}). \tag{2.13}$$

This scheme approximates $W(t)$ *to order* p*, i.e.* $W^{n+1} = W(t_{n+1}) + \mathcal{O}(\Delta t^{p+1})$*.*

Proof. Suppose an $\mathcal{O}(\Delta t^p)$ Runge–Kutta method is applied to the system

$$U' = G(U),$$
$$W' = V(U).$$

Stage values are given by

$$Y^{(j)} = U^n + \Delta t \sum_{k=1}^{s} a_{jk} G(Y^{(k)}), Z^{(j)} = W^n + \Delta t \sum_{k=1}^{s} a_{jk} V(Y^{(k)}), j = 1, \ldots, s.$$

The $Z^{(j)}, j = 1, \ldots, s$, are not used and would not be computed in practice. We advance by one step via

$$U^{n+1} = U^n + \Delta t \sum_{j=1}^{s} b_j G(Y^{(j)}),$$
$$W^{n+1} = W^n + \Delta t \sum_{j=1}^{s} b_j V(Y^{(j)}).$$

Recall that the Runge–Kutta method gives an error that is $\mathcal{O}(\Delta t^p)$. Applying this fact to the second equation gives the desired result. ∎

From Lemma 2.2 we obtain

LEMMA 2.3. *Let $V : \mathbb{R}^n \to \mathbb{R}^n$ be a smooth function, and let $(A, b), (A, \hat{b})$ be an embedded RK pair with order $p, \hat{p}$. Given an initial value problem (2.1) in U, let $Y^{(j)}$ denote the stage values in (2.2a); then*

$$\sum_{j=1}^{s} b_j V(Y^{(j)}) = \sum_{j=1}^{s} \hat{b}_j V(Y^{(j)}) + \mathcal{O}(\Delta t^{\min(p,\hat{p})}). \tag{2.14}$$

Proof. Applying Lemma 2.2 to both schemes gives

$$W(t) + \Delta t \sum_{j=1}^{s} b_j V(Y^{(j)}) = W(t + \Delta t) + \mathcal{O}(\Delta t^{p+1}),$$
$$W(t) + \Delta t \sum_{j=1}^{s} \hat{b}_j V(Y^{(j)}) = W(t + \Delta t) + \mathcal{O}(\Delta t^{\hat{p}+1}).$$

Combining these gives the stated result. ∎

One must take care in applying Lemma 2.3 to a PDE semi-discretization, since the error constant appearing in (2.14) might involve a factor like Δx^{-r}, $r > 0$, which would grow as the spatial grid is refined. A detailed analysis of the propagation of the spatial discretization error within a Runge–Kutta step is beyond the scope of this work; we refer the interested reader to [17, 1]. In the following theorem we simply assume that the error constant in (2.14) is bounded as $\Delta x \to 0$. Note that this is an assumption on each of the component RK schemes separately. Theorem 2.4 indicates that if each of the component schemes gives a solution free from order reduction (in the sense just described), then the embedded pair loses at most one order of accuracy.

THEOREM 2.4. *Suppose that the flux-based spatially partitioned Runge–Kutta method (2.8) with coefficients $(A, b, \hat{b})$ is applied to the semi-discretization (2.5) with $\Delta t = \mathcal{O}(\Delta x)$. Furthermore, suppose that the error constant appearing in (2.14) is bounded as $\Delta x \to 0$ when one takes $V = f_{i\pm\frac{1}{2}}$. Then the full discretization of (2.5) has order of accuracy equal to $\min(p, \hat{p}) - 1$ where p (respectively, $\hat{p}$) is the order of accuracy of the full discretization obtained by using the RK method with coefficients (A, b) (respectively, $(A, \hat{b})$).*

Proof. The full discretizations are

$$w_i^{n+1} = w_i^n - \frac{\Delta t}{\Delta x}\left(\sum_{j=1}^{s} b_j f_{i+\frac{1}{2}}(Y^{(j)}) - \sum_{j=1}^{s} b_j f_{i-\frac{1}{2}}(Y^{(j)})\right),$$
$$\hat{w}_i^{n+1} = \hat{w}_i^n - \frac{\Delta t}{\Delta x}\left(\sum_{j=1}^{s} \hat{b}_j f_{i+\frac{1}{2}}(Y^{(j)}) - \sum_{j=1}^{s} \hat{b}_j f_{i-\frac{1}{2}}(Y^{(j)})\right).$$

Using Lemma 2.3 with $V(u) = f_{i\pm\frac{1}{2}}(u)$ gives

$$\sum_{j=1}^{s} b_j f_{i\pm\frac{1}{2}}(Y^{(j)}) = \sum_{j=1}^{s} \hat{b}_j f_{i\pm\frac{1}{2}}(Y^{(j)}) + \mathcal{O}(\Delta t^{\min(p,\hat{p})}). \tag{2.15}$$

Flux-based partitioning gives

$$\begin{aligned} u_i^{n+1} &= u_i^n - \frac{\Delta t}{\Delta x}\Bigg(\chi^n_{i+\frac{1}{2}}\sum_{j=1}^{s} b_j f_{i+\frac{1}{2}}(Y^{(j)}) + (1-\chi^n_{i+\frac{1}{2}})\sum_{j=1}^{s}\hat{b}_j f_{i+\frac{1}{2}}(Y^{(j)}) \\ &\qquad - \chi^n_{i-\frac{1}{2}}\sum_{j=1}^{s} b_j f_{i-\frac{1}{2}}(Y^{(j)}) - (1-\chi^n_{i-\frac{1}{2}})\sum_{j=1}^{s}\hat{b}_j f_{i-\frac{1}{2}}(Y^{(j)})\Bigg), \\ &= u_i^n - \frac{\Delta t}{\Delta x}\left(\sum_{j=1}^{s} b_j f_{i+\frac{1}{2}}(Y^{(j)}) - \sum_{j=1}^{s} b_j f_{i-\frac{1}{2}}(Y^{(j)})\right) + \mathcal{O}(\Delta t^{\min(p,\hat{p})+1}/\Delta x), \end{aligned}$$

where we have used (2.15). The first two terms on the right are just the solution computed by the scheme with weights b_j, which is accurate to order p by assumption. Thus, inserting the exact solution in the last equation above and using the assumption that $\Delta t = \mathcal{O}(\Delta x)$ gives the desired result. □

REMARK 2.1. *More generally, for a semi-discretization of an evolution PDE that is order q in space, one takes (for explicit methods) $\Delta t = \mathcal{O}(\Delta x^q)$ and a factor of Δx^{-q} appears in the spatial discretization. The net effect is the same in that it may reduce the accuracy by a factor of Δt.*

REMARK 2.2. *It might be tempting to try to apply the theorem to the general additive Runge–Kutta method*

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \qquad \textit{with} \qquad \begin{array}{c|c} \hat{c} & \hat{A} \\ \hline & \hat{b}^T \end{array}, \tag{2.16}$$

by first making a larger additive Runge–Kutta method that does *share abscissae and stage weights, namely*

$$\begin{array}{c|cc} c & A & 0 \\ \hat{c} & 0 & \hat{A} \\ \hline & b^T & 0 \end{array} \qquad \textit{with} \qquad \begin{array}{c|cc} c & A & 0 \\ \hat{c} & 0 & \hat{A} \\ \hline & 0 & \hat{b}^T \end{array}, \tag{2.17}$$

and then concluding that Theorem 2.1 applied to the latter implies the results on the former. Indeed the theorem does apply to (2.17); however, the implication does not follow because (2.16) and (2.17) are not the same method. *To see this, note that (2.16) computes stages combining function values from the two schemes (according to χ) whereas (2.17) computes each stage value using only one scheme.*

2.4. Positivity. The nonlinear stability properties of SPERK schemes are of interest as well. In this section we consider the positivity of flux-based SPERK schemes which are comprised of two explicit SSP Runge–Kutta schemes.

We begin by reviewing some results. Of particular relevance to our derivations is that SSP Runge–Kutta schemes may be re-written in optimal Shu–Osher form [18] via the transformation provided in [3]. For the ODE system $U' = G(U)$ this yields schemes of the form

$$Y^{(j)} = \left(1 - \sum_{k=1}^{j-1}\alpha_{jk}\right)U^n + \sum_{k=1}^{j-1}(\alpha_{jk}Y^{(k)} + \Delta t\beta_{jk}G(Y^{(k)})), \tag{2.18a}$$

$$U^{n+1} = Y^{(s+1)} \tag{2.18b}$$

where all $\alpha_{jk} \geq 0$, $\sum_{k=1}^{j-1} \alpha_{jk} \leq 1$ and $j = 1, 2, \ldots, s+1$.

If both sets of coefficients α_{jk}, β_{jk} are nonnegative, and forward Euler is positivity-preserving, then it may be shown that the Runge–Kutta scheme preserves positivity under a suitable time step restriction [20, 4]:

LEMMA 2.5. *If the forward Euler method is positivity-preserving under the restriction $0 \leq \Delta t \leq \Delta t_{FE}$, then the Runge–Kutta method (2.18) with $\beta_{jk} \geq 0$ is positivity-preserving provided*

$$\Delta t \leq \mathcal{C} \Delta t_{FE},$$

where $\mathcal{C}$ is the SSP coefficient

$$\mathcal{C} \equiv \min \{c_{jk} : 1 \leq k < j \leq s+1\} \text{ where } c_{jk} = \begin{cases} \frac{\alpha_{jk}}{\beta_{jk}} & \text{if } \beta_{jk} \neq 0, \\ \infty & \text{otherwise.} \end{cases}$$

We now show the corresponding result for flux-based SPERK schemes.

THEOREM 2.6. *Suppose that the flux-differencing semi-discretization (2.5) of the one-dimensional conservation law (2.4) satisfies the positivity properties*

$$w_i - \frac{\delta t_1}{\Delta x} f_{i+\frac{1}{2}}(W) \geq 0, \text{for all } \delta t_1 \leq \Delta t_1, \tag{2.19}$$

$$w_i + \frac{\delta t_2}{\Delta x} f_{i-\frac{1}{2}}(W) \geq 0, \text{for all } \delta t_2 \leq \Delta t_2 \tag{2.20}$$

for all $w_i \geq 0, 1 \leq i \leq N$. Further, suppose that a flux-based SPERK scheme is applied to (2.5), and that the schemes composing the embedded pair have SSP coefficients $\mathcal{C}$ and $\hat{\mathcal{C}}$. Then the full discretization is positivity-preserving for time steps satisfying

$$0 \leq \Delta t \leq \min(\mathcal{C}, \hat{\mathcal{C}}) \Delta t_* \tag{2.21}$$

where

$$\Delta t_* = \begin{cases} \Delta t_2 & \text{if } \Delta t_1 = \infty, \\ \Delta t_1 & \text{if } \Delta t_2 = \infty, \\ \frac{\Delta t_1 \Delta t_2}{\Delta t_1 + \Delta t_2} & \text{otherwise.} \end{cases}$$

Proof. We give a proof for the case where Δt_1 and Δt_2 are finite; the proof for the two remaining cases is straightforward and follows in a similar fashion.

Let $\chi^n_{i+\frac{1}{2}}$ and $1 - \chi^n_{i+\frac{1}{2}}$ specify the (non-negative) weightings of the two schemes at the cell boundaries at time step n. The update for U is given by

$$\begin{aligned} u_i^{n+1} &= u_i^n + \lambda \sum_{j=1}^{s} b_i \left(-\chi^n_{i+\frac{1}{2}} f^{(j)}_{i+\frac{1}{2}} + \chi^n_{i-\frac{1}{2}} f^{(j)}_{i-\frac{1}{2}} \right) + \lambda \sum_{j=1}^{s} \hat{b}_i \left(-(1 - \chi^n_{i+\frac{1}{2}}) f^{(j)}_{i+\frac{1}{2}} + (1 - \chi^n_{i-\frac{1}{2}}) f^{(j)}_{i-\frac{1}{2}} \right) \\ &= \chi^n_{i+\frac{1}{2}} \left(\gamma_p u_i^n + \lambda \sum_{j=1}^{s} -b_i f^{(j)}_{i+\frac{1}{2}} \right) + (1 - \chi^n_{i+\frac{1}{2}}) \left(\gamma_p u_i^n + \lambda \sum_{j=1}^{s} -\hat{b}_i f^{(j)}_{i+\frac{1}{2}} \right) \\ &+ \chi^n_{i-\frac{1}{2}} \left(\gamma_m u_i^n + \lambda \sum_{j=1}^{s} b_i f^{(j)}_{i-\frac{1}{2}} \right) + (1 - \chi^n_{i-\frac{1}{2}}) \left(\gamma_m u_i^n + \lambda \sum_{j=1}^{s} \hat{b}_i f^{(j)}_{i-\frac{1}{2}} \right) \end{aligned} \tag{2.22}$$

where $\lambda = \frac{\Delta t}{\Delta x}$, $\gamma_p = \frac{\Delta t_2}{\Delta t_1 + \Delta t_2}$, and $\gamma_m = \frac{\Delta t_1}{\Delta t_1 + \Delta t_2}$. We denote the fluxes associated with stage j by $f^{(j)}_{i\pm\frac{1}{2}} \equiv f_{i\pm\frac{1}{2}}(Y^{(j)})$ where $Y^{(j)}$ is the approximation of $u_h(t_n)$ at the jth stage. The first term will be non-negative if

$$\gamma_p u_i^n + \lambda \sum_{j=1}^{s} -b_i f^{(j)}_{i+\frac{1}{2}} \tag{2.23}$$

is non-negative. Transforming (2.23) to optimal Shu–Osher form yields

$$\gamma_p \left(1 - \sum_{j=1}^{s} \alpha_{s+1,j}\right) u_i^n + \gamma_p \left(\sum_{j=1}^{s} \alpha_{s+1,j} y_i^{(j)} - \frac{\lambda}{\gamma_p} \beta_{s+1,j} f^{(j)}_{i+\frac{1}{2}}\right).$$

We observe that forward Euler is positivity-preserving when applied to the semi-discretization (2.5) provided $\Delta t \le \Delta t_*$. An application of Lemma 2.5 gives $Y^{(j)} \ge 0, 1 \le j \le s$, provided the step-size restriction (2.21) is satisfied and $U^n \ge 0$, where vector inequalities are to be interpreted as being taken componentwise. Combining this result with $\Delta t \le \mathcal{C}\Delta t_*$ and hypothesis (2.19) yields that the first term in (2.22) is non-negative. The proof follows by applying a similar analysis to the second, third, and fourth terms in (2.22). □

The theorem may be applied to a variety of common discretizations. For example, consider an upwind discretization of the linear advection equation

$$u_t + a u_x = 0, \quad a > 0,$$

on a grid with uniform spacing Δx. In this example $f_{i+\frac{1}{2}} = a u_i$ from which we observe that $\Delta t_2 = \infty$ and $\Delta t_1 = \frac{1}{a}\Delta x$. Assuming that the schemes forming the embedded pair have SSP coefficients $\mathcal{C}$ and $\hat{\mathcal{C}}$ we find that the flux-based SPERK scheme is positivity-preserving provided

$$\Delta t \le \frac{1}{a} \min(\mathcal{C}, \hat{\mathcal{C}}) \Delta x. \tag{2.24}$$

We can also apply the theorem to a discretization of the diffusion equation. Consider

$$u_t - \nu u_{xx} = 0, \nu > 0,$$

on a grid with uniform spacing Δx. Taking $f_{i+\frac{1}{2}} = \frac{\nu}{\Delta x}(-u_{i+1} + u_i)$ gives $\Delta t_1 = \Delta t_2 = \frac{1}{\nu}\Delta x^2$. Assuming that the schemes forming the embedded pair have SSP coefficients $\mathcal{C}$ and $\hat{\mathcal{C}}$, we find that the flux-based SPERK scheme is positivity-preserving under the time step-size restriction

$$\Delta t \le \frac{1}{2\nu} \min(\mathcal{C}, \hat{\mathcal{C}}) \Delta x^2. \tag{2.25}$$

The proof for equation-based partitioning follows in a similar manner. For completeness we state the result below.

THEOREM 2.7. *Suppose that the semi-discretization (2.1) of the one-dimensional conservation law (2.4) satisfies the positivity property*

$$W + \Delta t G(W) \ge 0$$

for all $W \geq 0$ and $\Delta t \leq \Delta t_{FE}$ where the inequalities are taken component-wise. Further, suppose that an equation-based SPERK scheme is applied to (2.1), and that the schemes composing the embedded pair have SSP coefficients $\mathcal{C}$ and $\hat{\mathcal{C}}$. Then the full discretization is positivity-preserving for time steps satisfying

$$0 \leq \Delta t \leq \min(\mathcal{C}, \hat{\mathcal{C}}) \Delta t_{FE}.$$

3. Generalizations. In this section we provide some useful generalizations of our previous spatial partitioning methods. For notational simplicity we state these results for partitions that are invariant in time; however, it is also possible to vary the partitioning function χ at each time level n.

3.1. More than two methods. Both equation- and flux-based partitioning can be generalized to more than two methods. Accuracy and positivity results essentially identical to those in Sections 2.3 and 2.4 can be shown for both cases.

3.1.1. Equation-based partitioning. Let $(\mathcal{I}_1, \mathcal{I}_2, \ldots \mathcal{I}_r)$ be a partitioning of the integers from 1 to N (i.e., the partitioning satisfies $\cup_{k=1}^{r} \mathcal{I}_k = \{1, 2, \ldots, N\}$, and $\mathcal{I}_j \cap \mathcal{I}_k = \emptyset$ if $j \neq k$). The generalization to r embedded methods defined by the coefficients $A, b^{(1)}, \ldots, b^{(r)}$ is defined as

$$\begin{aligned} Y^{(j)} &= U^n + \Delta t \sum_{k=1}^{s} a_{jk} G(Y^{(k)}), \quad 1 \leq j \leq s, \\ U^{n+1} &= U^n + \Delta t \sum_{k=1}^{r} \sum_{j=1}^{s} b_j^{(k)} \operatorname{diag}(\chi^{(k)}) G(Y^{(j)}), \end{aligned}$$

where

$$\chi_i^{(k)} = \begin{cases} 1 & \text{if } i \in \mathcal{I}_k, \\ 0 & \text{otherwise.} \end{cases}$$

3.1.2. Flux-based partitioning. Flux-based partitioning can also be written more generally in terms of $b^{(1)}, \ldots, b^{(r)}$. In this case we take

$$\begin{aligned} Y^{(j)} &= U^n - \frac{\Delta t}{\Delta x} \sum_{k=1}^{s} a_{jk} D\Phi(Y^{(k)}), 1 \leq j \leq s, \\ U^{n+1} &= U^n - \frac{\Delta t}{\Delta x} \sum_{k=1}^{r} \sum_{j=1}^{s} b_j^{(k)} D \operatorname{diag}(\chi^{(k)}) \Phi(Y^{(j)}), \end{aligned}$$

to combine the r different embedded schemes.

3.2. Blending and accuracy. Rather than shifting discontinuously between methods from one spatial point to the next, an appealing approach is to have a transition region in which a weighted average of the two methods is used, with the weight shifting from one method to the other over the transition region. The following proposition shows that this approach is still accurate in time, which is a necessary condition for the full SPERK discretization to be accurate.

PROPOSITION 3.1. *Let $(A, b^{(1)}), (A, b^{(2)}), \ldots, (A, b^{(r)})$ denote a set of embedded RK methods, and let p_k denote the order of accuracy of method $(A, b^{(k)})$. Then the*

RK method

$$\left(A, \sum_{k=1}^{r} \alpha_k b^{(k)}\right) \quad \textit{where} \quad \sum_{k=1}^{r} \alpha_k = 1$$

has order of accuracy at least $p = \min_k p_k$.

Proof. The order conditions involve only expressions that are linear in the weights. Since all of the component method coefficients satisfy the conditions up to order $p = \min_k p_k$, the averaged method does as well. □

REMARK 3.1. *The proposition applies only to embedded methods. If the component methods had different coefficient matrices A, their average would in general be only first-order accurate, even if they share common abscissas c.*

Theorems 2.1 and 2.4 for equation- and flux-based SPERK methods can be extended to the blended case in this way. In practice, we observe the full accuracy when blending using either approach.

4. Example: Spatially partitioned time-stepping for advection-diffusion. The semi-discretization of parabolic PDEs leads to systems of ODEs with eigenvalues near the negative real axis, whereas the semi-discretization of hyperbolic PDEs leads to systems with eigenvalues near the imaginary axis. Appropriate time integrators must include the corresponding portion of the complex plane in their absolute stability region. In problems with both hyperbolic and parabolic terms where the relative importance of these terms varies in space, e.g., in problems with strongly varying coefficients or mesh spacing, these criteria may be in conflict.

Consider, for example, the linear, variable coefficient advection-diffusion equation

$$u_t + (b(x)u)_x = (a(x)u_x)_x, \tag{4.1}$$

with periodic boundary conditions on $[0, 1]$ and initial conditions $u(x, 0) = \sin^3(2\pi x)$. Discretizing the spatial derivatives in (4.1) with three-point centered differences yields the flux-differencing method (2.5) with

$$f_{i+\frac{1}{2}} = b(x_{i+\frac{1}{2}}) \left(\frac{u_{i+1} + u_i}{2}\right) - a(x_{i+\frac{1}{2}}) \frac{1}{\Delta x} (u_{i+1} - u_i)$$

and a constant mesh spacing Δx. In our experiments, we take $a(x) > 0$ and $b(x)$ to be functions that are periodically defined by

$$\begin{aligned} a(x) &= \tfrac{1}{5}\left(\tfrac{1}{4} + \tfrac{1}{\pi^2} \arctan(1000(x - 0.1)) \cdot \arctan(1000(0.4 - x))\right), \quad -\tfrac{1}{4} \le x \le \tfrac{3}{4}, \\ b(x) &= 1 + 38\left(\tfrac{1}{4} + \tfrac{1}{\pi^2} \arctan(1000(x - 0.6)) \cdot \arctan(1000(0.9 - x))\right), \quad \tfrac{1}{4} \le x \le \tfrac{5}{4}. \end{aligned}$$

As shown in Figure 4.1, there is a region centered at $x = 0.25$ where diffusion dominates and negative real axis inclusion is critical for the design of the time-stepping scheme. Similarly, in the region around $x = 0.75$, convection dominates and imaginary axis inclusion is the most crucial design feature. The strong variation in the dominant term makes this a natural test problem for SPERK schemes.

4.1. Second-order embedded pairs. We now design three- and four-stage embedded explicit Runge–Kutta pairs suitable for this problem. In each embedded pair, one method has a stability polynomial that maximizes (or nearly maximizes) the real axis interval of absolute stability, while the other method has a stability polynomial that maximizes (or nearly maximizes) the imaginary axis interval of absolute

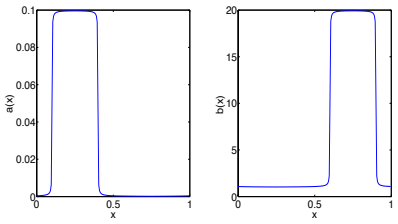

FIG. 4.1. *The variable coefficients $a(x)$ (left) and $b(x)$ (right). We observe a spatial variation in the relative importance of advection and diffusion.*

stability. Given a coefficient matrix A, the stability polynomial of a Runge–Kutta method (and hence its linear stability properties) can be chosen by solving a linear system of equations for the weights b_j. We do not consider two-stage pairs since all second-order two-stage schemes have the same stability polynomial.

The stability polynomial of a three-stage, second-order RK method has the form $R(z) = 1+z+z^2/2+\alpha_3 z^3$ where $\alpha_3 = b^T Ac$. For real-axis stability, we take the 3-stage Runge–Kutta–Chebyshev method of the family presented in [24], which we refer to as RKC(3,2), as the second scheme of our embedded pair. The stability polynomial of this method is a Bakker–Chebyshev polynomial and contains nearly the largest possible portion of the negative real axis over the class of three-stage second-order schemes. The first scheme is obtained by setting $\alpha_3 = 1/4$, which approximately maximizes imaginary axis inclusion, and selecting weights $\hat{b}_j$ corresponding to the resulting polynomial. This gives the following embedded pair:

$$\begin{array}{c|ccc} 0 & & & \\ 3/8 & 3/8 & & \\ 3/8 & 3/16 & 3/16 & \\ \hline \hat{b}^T & -1/3 & -20/9 & 32/9 \\ b^T & -1/3 & 4/9 & 8/9 \end{array} \tag{4.2}$$

The absolute stability regions are shown in Figure 4.2 (left).

The stability polynomial of a four-stage, second-order RK method has the form $R(z) = 1 + z + z^2/2 + \alpha_3 z^3 + \alpha_4 z^4$ where $\alpha_3 = b^T Ac$ and $\alpha_4 = b^T A^2 c$. The optimal imaginary axis inclusion occurs for the choice $\alpha_3 = 1/6$, $\alpha_4 = 1/24$, which gives the stability polynomial of the classical fourth-order Runge–Kutta method. Hence we take the classical method, which we shall denote by RK4, as the first scheme of our pair. Sufficient degrees of freedom remain to obtain a scheme with the same linear stability as the optimal four-stage, second-order scheme RKC(4,2). This gives the following embedded pair (with absolute stability regions as shown in Figure 4.2 right)

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline \hat{b}^T & 1/6 & 1/3 & 1/3 & 1/6 \\ b^T & 2/125 & 17/25 & 36/125 & 2/125 \end{array} \tag{4.3}$$

4.2. Numerical results. We now give the results of numerical experiments carried out using our three- and four- stage SPERK schemes. The partitioning parameter χ is set equal to 1 wherever $a(x)$ is more than 1/10th of its maximum value; elsewhere it is zero. In all cases, we vary the time step-size and compute max-norm absolute

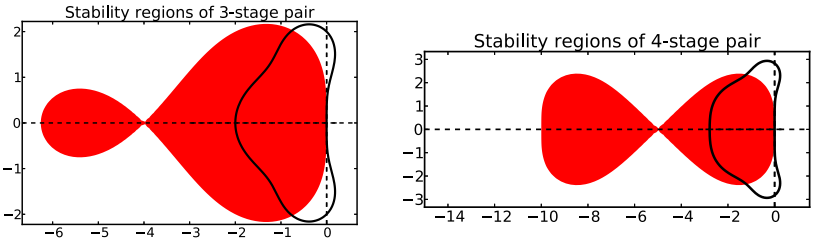

FIG. 4.2. *Absolute stability regions for the three-stage embedded pair (4.2) (left) and the four-stage embedded pair (4.3) (right). The shaded red region corresponds to the method optimized for real axis inclusion and the black line corresponds to the method optimized for imaginary axis inclusion.*

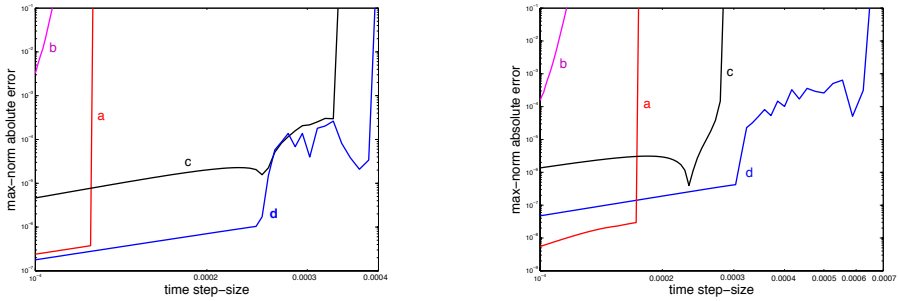

FIG. 4.3. *Maximum norm absolute errors for SPERK schemes applied to the variable coefficient advection-diffusion problem. The left plot gives results for three-stage schemes: (a) embedded scheme, (b) RKC(3,2), (c) equation-based partitioning, (d) flux-based partitioning. The right plot gives results for four-stage schemes: (a) RK4, (b) our variant of RKC(4,2), (c) equation-based partitioning, (d) flux-based partitioning.*

errors by comparing to a highly accurate RK4 approximation of the spatially discretized system at the final time $t = 1$. All computations use a constant mesh spacing of $\Delta x = 1/200$.

The results for three-stage methods are given in the left-hand plot of Fig. 4.3. We find that RKC(3,2) is unstable over nearly the entire interval. The second method composing the embedded pair gives good results for $\Delta t \leq 0.000125$, but stability is lost for larger time step-sizes. The SPERK schemes derived from the combination are stable for values of Δt which are two or more times greater. Flux-based partitioning gives the smallest error of the three-stage schemes considered, and also allows for the largest stable time steps.

The right-hand plot of Fig. 4.3 gives the results for four-stage methods. Similar to the three-stage case, our variant of RKC(4,2) is unstable for nearly the entire interval. The classical fourth-order Runge–Kutta method RK4 has the smallest error for $\Delta t \leq 0.00017$ but becomes unstable for larger time step-sizes. Moving to SPERK schemes gives improved stability and second-order accurate results. Once again, we observe that flux-based partitioning has a smaller error and allows for larger stable time steps than equation-based partitioning. A review of the numerical results underlying the plot indicates that flux-based partitioning gives accurate results for $\Delta t < 0.00065$ and a regular, second-order error for $\Delta t \leq 0.00031$.

5. Example: Spatially partitioned time-stepping for WENO. Weighted essentially non-oscillatory (WENO) spatial discretizations involve an adaptive, data-dependent combination of several candidate stencils to compute fluxes [19]. The most commonly used scheme provides fifth-order accuracy in smooth regions of the solution and formally third-order spatial discretizations near shocks or other discontinuities. Assuming a positive flux function, the finite difference WENO scheme computes the positive flux differences in (2.5) with

$$f_{j+\frac{1}{2}} = \omega_0 \left(\tfrac{2}{6} f_{j-2} - \tfrac{7}{6} f_{j-1} + \tfrac{11}{6} f_j\right) + \omega_1 \left(-\tfrac{1}{6} f_{j-1} + \tfrac{5}{6} f_j + \tfrac{2}{6} f_{j+1}\right) + \omega_2 \left(\tfrac{2}{6} f_j + \tfrac{5}{6} f_{j+1} - \tfrac{1}{6} f_{j+2}\right), \tag{5.1}$$

where the weights ω_0, ω_1, and ω_2 are chosen by computing data-dependent smoothness indicators [19]. We will use the WENO weights $\omega_{0,1,2}$ to select the mask χ in a WENO-based SPERK scheme.

5.1. An embedded pair for WENO. We start with the SSPRK(5,3) scheme [21]. This will be the lower-order scheme which is used in spatial regions where the solution is not smooth. If we embed this method in a larger 7-stage, 5th-order method, we have the following Butcher tableau

0							
0.3773	0.3773						
0.7545	0.3773	0.3773					
0.7290	0.2430	0.2430	0.2430				
0.6992	0.1536	0.1536	0.1536	0.2385			
c_6	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}		
c_7	a_{71}	a_{72}	a_{73}	a_{74}	a_{75}	a_{76}	
$\hat{b}^T$	0.2067	0.2067	0.1171	0.1818	0.2876	0	0
b^T	b_1	b_2	b_3	b_4	b_5	b_6	b_7

(5.2)

where we display only a few digits of the SSPRK(5,3) coefficients. The unknown coefficients will determine an RK(7,5) scheme which will optimized for linear stability and use in spatial regions where the solution is smooth.

A technique for determining these coefficients by satisfying the order conditions is explored in [11] following a strategy designed in [23]. We attempt to maximize the linear stability properties of the RK(7,5) scheme; these are determined by the stability polynomial which in this case is parameterized by the coefficients of the z^6 and z^7 terms [6], polynomial expressions in the coefficients (5.2) which we denote by α_6 and α_7 respectively. We find that specifying the SSPRK(5,3) coefficients restricts the possible solutions to a straight line through the α_6–α_7 space. By examining the linear stability properties along this line (see Figure 5.1), we can choose one of the degrees of freedom (the "homogeneous polynomial" I_{65}, see [11] for details) to maximize the linear stability properties of the resulting scheme. There are still six (nonlinear) order conditions to satisfy and six coefficients to be determined. Solving these remaining equations with a computer algebra system results in a single discrete solution and a one-parameter family of solutions. The one-parameter family was discarded because each member had either a negative node c_6 or unreasonably large coefficients. The lone solution did not have these deficiencies and we use it for our numerical experiments. The coefficients are shown to 15 digits in Table 5.1.

5.2. Using the WENO weights to select a temporal scheme. We present some preliminary numerical results for the new RK(7,5) / SSPRK(5,3) embedded pair. The WENO weights $\omega_{0,1,2}$ are used to determine which scheme is propagated for each grid point; if the weights are within some small threshold of their theoretical

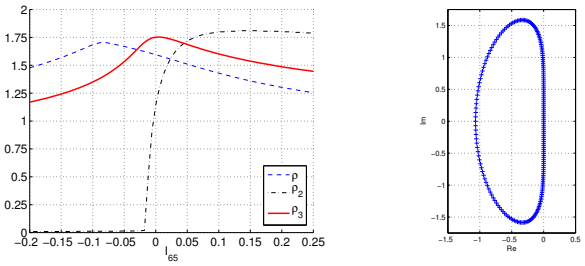

FIG. 5.1. *Choosing the linear stability properties for RK(7,5) scheme. Left: various measurements of the linear stability region versus a parameter* I_{65}*. Here* ρ *is the radius of the largest inscribed disc and* ρ_2 *is the radius of the largest interval of the imaginary axis included. Right: the "WENO bean", the spectrum of the linearized WENO operator in smooth regions [8, 25, 13]:* ρ_3 *measures the largest scaling of this bean that will fit in the linear stability region. We choose* $I_{65} = 0.0037$ *and find* $\rho_3 \approx 1.76$*,* $\rho \approx 1.58$*, and* $\rho_2 \approx 1.2$*.*

$a_{21} = 0.377268915331368,$		
$a_{31} = 0.377268915331368,$	$a_{32} = 0.377268915331368,$	
$a_{41} = 0.242995220537396,$	$a_{42} = 0.242995220537396,$	$a_{43} = 0.242995220537396,$
$a_{51} = 0.153589067695126,$	$a_{52} = 0.153589067695126,$	$a_{53} = 0.153589067695126,$
$a_{54} = 0.23845893284629,$	$\hat{b}_1 = 0.206734020864804,$	$\hat{b}_2 = 0.206734020864804,$
$\hat{b}_3 = 0.117097251841844,$	$\hat{b}_4 = 0.18180256012014,$	$\hat{b}_5 = 0.287632146308408,$
$a_{61} = 0.113015751552667,$	$a_{62} = 1.49947221487533,$	$a_{63} = 0.134753400626063,$
$a_{64} = -1.06421259296782,$	$a_{65} = 0.205145170072233,$	$a_{71} = -0.512110930783855,$
$a_{72} = 3.91735780781337,$	$a_{73} = -0.0470520461913835,$	$a_{74} = -0.218621292015928,$
$a_{75} = -1.64543995945252,$	$a_{76} = -0.494133579369683,$	$b_1 = 0.122097569374901,$
$b_2 = 0.492898173466563,$	$b_3 = -0.232023614650883,$	$b_4 = -1.98394581022939,$
$b_5 = 1.85394392181784,$	$b_6 = 0.965538124667539,$	$b_7 = -0.21850836444657.$

TABLE 5.1
Coefficients of the embedded RK(7,5)/SSPRK(5,3) method.

smooth-region values of $\frac{1}{10}$, $\frac{6}{10}$ and $\frac{3}{10}$, then we propagate the higher-order linearly stable RK(7,5) solution. On the other hand, if the WENO weights do not indicate smoothness, then we propagate the lower-order SSPRK(5,3). The mask is widened slightly by four grid cells to increase the usage of the SSP scheme. We use a threshold of 0.06 and compute the mask based on U^n (the mask is fixed over each time-step).

5.3. Numerical results. We perform convergence studies which demonstrate that the SPERK methods achieve the predicted orders of accuracy. Our tests use the inviscid Burgers' equation $u_t + (u^2)_x = 0$ on the periodic domain $[-1, 1]$ with smooth initial conditions $u(x, 0) = \frac{1}{2} - \frac{1}{2}\cos\left(\pi\left(x - \frac{\sin 2\pi x}{4\pi}\right)\right)$. Table 5.2 shows that the methods achieve (at least) their expected orders of accuracy. Namely, five when the fifth-order scheme is used ($\chi(x) = 1$ for all x) and three when using the third-order scheme at any points. As noted in Section 3.2, χ need not be a binary choice: we see that the method maintains third-order even when random values of χ (in the range $[0, 1]$) are selected at each point and at each time-step. Note also that the flux-partitioned scheme exhibits an order of accuracy $\min(p, \hat{p}) = 3$, *without* the loss of one order as predicted by our Theorem 2.4; this is typical of what we observed in all our numerical tests.

In addition to these finite difference convergence studies, we also tested the equation-based SPERK scheme with a finite-volume WENO code [12] and observed similar results.

		error for different $\chi(x)$ functions			
partitioning	Δx	1	0	Heaviside(x)	random(0,1)
equation-	1/320	1.29×10^{-7}	5.59×10^{-7}	1.29×10^{-7}	3.33×10^{-7}
based	1/640	4.09×10^{-9}	7.00×10^{-8}	4.44×10^{-9}	3.76×10^{-8}
	1/1280	1.29×10^{-10}	8.64×10^{-9}	5.48×10^{-10}	4.39×10^{-9}
est. order		4.99	3.02	3.45	3.08
flux-based	1/320	2.01×10^{-8}	1.08×10^{-7}	2.44×10^{-8}	6.63×10^{-8}
	1/640	6.38×10^{-10}	1.34×10^{-8}	1.87×10^{-9}	8.06×10^{-9}
	1/1280	2.00×10^{-11}	1.65×10^{-9}	2.18×10^{-10}	1.05×10^{-9}
est. order		4.99	3.02	3.44	2.99

TABLE 5.2

Convergence studies for various choices of the mask function $\chi(x)$, performed on Burgers' equation with smooth initial conditions discretized using WENO and the SPERK 5th-order/3rd-order pair. We use a CFL number of 1.2 *and the error is measured in an approximate L_2 norm.*

method	time	error	error ($\Delta x/2$)	est. order	TV increase
RK(7,5)	$t = 0.25$	2.01e-8	6.38e-10	4.98	0
	$t = 1.25$	8.72e-3	8.79e-3	—	0.264
SSPRK(5,3)	$t = 0.25$	1.08e-7	1.34e-8	3.01	0
	$t = 1.25$	1.35e-3	9.53e-4	0.50	0
SPERK	$t = 0.25$	2.01e-8	6.38e-10	4.98	0
	$t = 1.25$	1.35e-3	9.53e-4	0.50	0

TABLE 5.3

The flux-partitioned SPERK scheme combined with WENO has the best features of each method. Here the initial smooth curve sharpens into a shock. At small times $t = 0.25$ before shock formation, both of the underlying schemes exhibit their design orders of 5 and 3 respectively. At a later time $t = 1.25$, a shock has formed; all schemes exhibit larger errors as the solution is no longer smooth. However, RK(7,5) now exhibits spurious oscillations, indicated by an increase in the total variation seminorm. We use a CFL number of 1.2, $\Delta x = \frac{1}{320}$, *and error measured in an approx. L_2 norm.*

5.3.1. Total variation tests. We consider the discrete total variation seminorm of the solution on a test problem consisting of the inviscid Burgers' equation with a square wave initial condition in $[0, 1]$. We determined the largest CFL σ, $\Delta t = \sigma \Delta x$ such that the resulting solution experiences no increase in the TV seminorm. For the flux-partitioned scheme we found that the SSPRK(5,3) scheme (i.e., $\chi = 0$ everywhere) is no longer TVD for $\sigma > 1.3$. The RK(7,5) solution is no longer TVD for $\sigma > 0.5$. With the SPERK embedded pair, the loss of the TVD property occurs for $\sigma > 1.3$ (that is, the same as the SSPRK(5,3) scheme). The results using equation-based partitioning are about the same.

The example in Table 5.3 demonstrates that the RK(7,5)/SSPRK(5,3) SPERK scheme offers high-order accuracy when the solution is smooth and remains TVD when the solution is non-smooth; the best of both worlds!

6. Discussion and Future Work. This paper considers spatially partitioned embedded Runge–Kutta (SPERK) schemes. Such methods give an efficient means to apply two or more Runge–Kutta methods to a spatially discretized PDE. In this paper, SPERK schemes are applied in two ways. The first of these partitions the ODEs, and is therefore referred to as equation-based partitioning. In these methods the order of accuracy is shown to equal the minimum of the order of accuracy of the schemes composing the embedded pair. Equation-based partitioning is not conservative when applied to a conservation law, however. The second partitioning

method partitions by fluxes. This flux-based partitioning has the advantage of being conservative when applied to a conservation law. Theoretically, it may lead to a loss of one order of accuracy when compared to the underlying schemes; however, this loss of accuracy is not observed in practice. We show that both equation- and flux-based partitioning are positivity-preserving under a suitable time step restriction when the underlying schemes are strong-stability-preserving (SSP). Numerical experiments on a spatially discretized variable coefficient advection-diffusion equation show that SPERK schemes can have superior linear stability characteristics than either scheme individually. SPERK schemes may be applied to weighted non-oscillatory (WENO) spatial discretizations of conservation laws. Our partitioning is carried out using the weights introduced in the WENO spatial discretization. Specifically, a fifth-order Runge–Kutta method is used in smooth regions where WENO chooses a fifth-order stencil, and a third-order SSP Runge–Kutta method is used in non-smooth regions where WENO chooses a formally third-order stencil. We find that this combination avoids oscillations in problems with shocks, and gives fifth-order accuracy in smooth flows.

As part of this paper, several explicit SPERK schemes are designed and numerically validated. Future work will carry out a more systematic development of SPERK schemes. Of particular interest for us are methods that have good monotonicity where the solution is nonsmooth, and good accuracy elsewhere. Such methods are particularly attractive for approximating conservation laws with isolated shocks and other nonsmooth features.

Schemes with improved local absolute stability properties (like those designed in Section 4) seem promising for problems with strong spatial variation in the coefficients or grid size. The stability polynomial optimization algorithm developed in [10] could be used more generally to design appropriate methods for such problems.

Implicit SPERK schemes are also of interest. For example, in regions with nonsmooth features, one may wish to use backward Euler since it is unconditionally monotone, and therefore avoids spurious oscillations. However, in regions characterized by smooth solutions, a higher-order method may be preferred since backward Euler is only first-order accurate and is highly dissipative. The corresponding SPERK combination may give improved accuracy and less smearing than backward Euler, while simultaneously offering improved monotonicity over second- or higher- order schemes. A detailed examination of implicit, equation-based partitioning based on combinations of the θ-scheme appears in the report of Van Slingerland, Borsboom and Vuik [15]. They find optimal local θ-values to minimize the numerical diffusion, while ensuring that the scheme is stable, positivity-preserving, and non-oscillatory. It is natural to consider extensions of their work and our own. For example, the development and analysis of higher-order implicit SPERK schemes as well as the study of implicit methods based on flux partitioning would be particularly interesting.

Theoretically, the accuracy of flux-based partitioning is expected to be less than the accuracy of the schemes composing the embedded pair but this loss of accuracy is not seen in our tests. This effect may be similar to what is observed in multirate schemes. For example, the Osher–Sanders scheme [14] is locally inconsistent, but has been found to give first-order convergence in practice. An explanation for this is given in [7] where it is shown that local inconsistencies may not show up in the global errors due to cancellation and damping effects. We hope to transfer this analysis to the case of SPERK schemes to obtain a better understanding of the unexpectedly good performance of flux-based partitioning.

REFERENCES

[1] M. Carpenter, D. Gottlieb, S. Abarbanel, and W. Don, *The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a study of the boundary error*, SIAM J. Sci. Comput., 16 (1995), pp. 1241–1252.

[2] E. Constantinescu and A. Sandu, *Multirate timestepping methods for hyperbolic conservation laws*, J. Sci. Comput., 33 (2007), pp. 239–278.

[3] L. Ferracina and M. N. Spijker, *An extension and analysis of the Shu–Osher representation of Runge–Kutta methods*, Math. Comp., 74 (2005), pp. 201–219.

[4] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong-stability-preserving high-order time discretization methods*, SIAM Rev., 43 (2001), pp. 89–112.

[5] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations I: Nonstiff problems*, Springer-Verlag, second ed., 1993.

[6] E. Hairer and G. Wanner, *Solving ordinary differential equations. II: Stiff and differential-algebraic problems*, Springer-Verlag, second ed., 1996.

[7] W. Hundsdorfer, A. Mozartova, and V. Savcenco, *Monotonicity conditions for multirate and partitioned explicit Runge-Kutta schemes*, Recent Developments in the Numerics of Nonlinear Hyperbolic Conservation Laws, (2012), pp. 177–195.

[8] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.

[9] C. A. Kennedy and M. H. Carpenter, *Additive Runge–Kutta schemes for convection-diffusion-reaction equations*, Appl. Numer. Math., 44 (2003), pp. 139–181.

[10] D. I. Ketcheson and A. J. Ahmadia, *Optimal stability polynomials for numerical integration of initial value problems*, Commun. Appl. Math. Comput. Sci., (2012). In press. Currently available at http://arxiv.org/abs/1201.3035.

[11] C. B. Macdonald, *Constructing high-order Runge–Kutta methods with embedded strong-stability-preserving pairs*, Master's thesis, Simon Fraser University, August 2003.

[12] K. T. Mandli, D. I. Ketcheson, et al., *Pyclaw software*, 2011.

[13] M. Motamed, C. B. Macdonald, and S. J. Ruuth, *On the linear stability of the fifth-order WENO discretization*, J. Sci. Comput., 47 (2010), pp. 127–149.

[14] S. Osher and R. Sanders, *Numerical approximations to nonlinear conservation laws with locally varying time and space grids*, Math. Comp., 41 (1983), pp. 321–336.

[15] M. B. P. Van Slingerland and C. Vuik, *A local theta scheme for advection problems with strongly varying meshes and velocity profiles*, tech. rep., Delft University of Technology, 2008.

[16] S. J. Ruuth and R. J. Spiteri, *High-order strong-stability-preserving Runge–Kutta methods with downwind-biased spatial discretizations*, SIAM J. Numer. Anal., 42 (2004), pp. 974–996.

[17] J. Sanz-Serna, J. Verwer, and W. Hundsdorfer, *Convergence and order reduction of Runge–Kutta schemes applied to evolutionary problems in partial differential equations*, Numer. Math., 50 (1986), pp. 405–418.

[18] C.-W. Shu, *Total-variation-diminishing time discretizations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 1073–1084.

[19] C.-W. Shu, *High order weighted essentially non-oscillatory schemes for convection dominated problems*, SIAM Rev., 51 (2009), pp. 82–126.

[20] C.-W. Shu and S. Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.

[21] R. J. Spiteri and S. J. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM J. Numer. Anal., 40 (2002), pp. 469–491.

[22] H. Tang and G. Warnecke, *High resolution schemes for conservation laws and convection-diffusion equations with varying time and space grids*, J. Comput. Math., 24 (2006), p. 121.

[23] J. H. Verner, *Explicit Runge–Kutta methods with estimates of the local truncation error*, SIAM J. Numer. Anal., 15 (1978), pp. 772–790.

[24] J. G. Verwer, *Explicit Runge–Kutta methods for parabolic partial differential equations*, Appl. Numer. Math., 22 (1996), pp. 359–379.

[25] R. Wang and R. J. Spiteri, *Linear instability of the fifth-order WENO method*, SIAM J. Numer. Anal., 45 (2007), pp. 1871–1901.

RECENT REPORTS

12/96	Mathematical Biomedicine and Modeling Avascular Tumor Growth	Byrne
12/97	Inference of the genetic network regulating lateral root initiation in Arabidopsis thaliana	Muraro Voß Wilson Bennett Byrne De Smet Hodgman King
12/98	Axisymmetric bifurcations of thick spherical shells under inflation and compression	deBotton Bustamante Dorfmann
12/99	Calculus from the past: Multiple Delay Systems arising in Cancer Cell Modelling	Wake Byrne
12/100	Nonlocal models of electrical propagation in cardiac tissue: electrotonic effects and the modulated dispersion of repolarization	Bueno-Orovio Kay Grau Rodriguez Burrage
12/101	Microfluidic Immunomagnetic Multi-Target Sorting A Model for Controlling Deflection of Paramagnetic Beads	Tsai Griffiths Stone
12/102	A New Lattice Boltzmann Equation to Simulate Density-Driven Convection of Carbon Dioxide	Allen Reis Sun
12/103	Control and optimization of solute transport in a porous tube	Griffiths Howell Shipley
12/104	Air-cushioning in impact problems	Moore Ockendon Oliver
12/105	Strain controlled biaxial stretch: An experimental characterization of natural rubber	Pancheri Dorfmann
12/106	Non-linear modeling of active biohybrid materials	Paetsch Dorfmann
12/107	Coalescence of Liquid Drops: Different Models Versus Experiment	Sprittles Shikhmurzaev
12/108	Adjoint Based A Posteriori Analysis of Multiscale Mortar Discretizations with Multinumerics	Tavener Wildey
12/109	Dynamics of mechanically induced fiber reorientation in the material reinforced by two families of fibers	Melnik Goriely

12/110	Multiscale stochastic reaction-diffusion modelling: application to actin dynamics in filopodia	Erban Flegg Papoian
12/111	Exploiting the Synergy Between Carboplatin and ABT-737 in the Treatment of Ovarian Carcinomas	Jain Richardson Meyer-Hermann Byrne
12/112	The integration of hormonal signaling networks and mobile microRNAs is required for vascular patterning in Arabidopsis roots	Muraro Pound Help Lucas Chopard Byrne Godin Hodgman King Pridmore Helariutta Bennett Bishopp
12/113	Fast solution of Cahn-Hilliard Variational Inequalities using Implicit Time Discretization and Finite Elements	Bosch Stoll Benner
12/114	An Embedding Technique for the Solution of Reaction-Diffusion Equations on Algebraic Surfaces with Isolated Singularities	Rockstroh März Ruuth
12/115	Mathematicians at the Movies: Sherlock Holmes vs. Professor Moriarty	Moulton Goriely
13/01	Rotation, inversion, and perversion in anisotropic elastic cylindrical tubes and membranes	Goriely Tabor
13/02	Drop spreading and penetration into pre-wetted powders	Marston Sprittles Zhu Li Vakarelski Thoroddsen
13/03	On the mechanics of thin films and growing surfaces	Holland Kosmata Goriely Kuhl

Copies of these, and any other OCCAM reports can be obtained from:

Oxford Centre for Collaborative Applied Mathematics
Mathematical Institute
24 - 29 St Giles'
Oxford
OX1 3LB
England
www.maths.ox.ac.uk/occam