One-sided post-processing for the discontinuous Galerkin method using ENO type stencil choosing and the local edge detection method

Rick Archibald[1], Anne Gelb[2], Sigal Gottlieb[3], and Jennifer Ryan[4]

ABSTRACT

In a previous paper by Ryan and Shu [20], a one-sided post-processing technique for the discontinuous Galerkin method was introduced for reconstructing solutions near computational boundaries and discontinuities in the boundaries, as well as for changes in mesh size. This technique requires prior knowledge of the discontinuity location in order to determine whether to use centered, partially one-sided, or one-sided post-processing. We now present two alternative stencil choosing schemes to automate the choice of post-processing stencil. The first is an ENO type stencil choosing procedure, which is designed to choose centered post-processing in smooth regions and one-sided or partially one-sided post-processing near a discontinuity, and the second method is based on the edge detection method designed by Archibald, Gelb, and Yoon [1, 2]. We compare these stencil choosing techniques and analyze their respective strengths and weaknesses. Finally, the automated stencil choices are applied in conjunction with the appropriate post-processing procedures and it is determine that the resulting numerical solutions are of the correct order.

[1] Center for System Science and Engineering Research, Arizona State University, Tempe, AZ 85287.

[2] Department of Mathematics, Arizona State University, Tempe, AZ 85287. Partial support provided by NIH grant EB O2533-01, and NSF grants CNS 0324957, DMS 0107428, and EAR 0222327.

[3] Department of Mathematics, University of Massachusetts-Dartmouth, Dartmouth, MA 02747. Supported by NSF grant DMS-0106743.

[4] Corresponding author. Computer Science and Mathematics Division, Oak Ridge National Lab. E-mail: ryanjk@ornl.gov, Phone: (865)576-5761, Fax: (865)574-0680. Research supported by Householder Fellowship in Scientific Computing sponsored by the DOE Applied Mathematical Sciences program. Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Oak Ridge National Lab, P.O. Box 2008, MS6367,Oak Ridge, TN 37831-6367.

Key Words: accuracy enhancement, post-processing, discontinuous Galerkin method, ENO, local edge detection

AMS(MOS) subject classification: 65M60

1 Introduction

In a previous paper by Ryan and Shu [20], a one-sided post-processing technique for the discontinuous Galerkin method was introduced for reconstructing solutions near computational boundaries and discontinuities in the solution, as well as for changes in mesh size. The technique requires apriori knowledge of shock locations, since no biasing should occur in smooth regions. Hence in this paper we propose two options to automate the stencil choices for the post-processor. The first technique borrows ENO (essentially non-oscillatory) constructions to find the smoothest stencil for post-processing. The second technique utilizes a local edge detection (LED) method developed by Archibald, Gelb, and Yoon [1, 2] to find the shock location, and determines the post-processing stencil based on the proximity of the shock. This study analyzes the behavior of these stencil choosing procedures for both stationary and moving shocks on a series of different uniform meshes and numerically evaluates their use on four scalar problems.

This paper is organized as follows. In §2 we review the discontinuous Galerkin method and the symmetric and one-sided post-processing technique utilized in this paper. The ENO and LED methods for choosing post-processing stencils are discussed in §3. Finally, in §4 we present numerical results demonstrating the benefits of using either of these techniques for choosing the appropriate post-processing stencil.

2 The discontinuous Galerkin method and post-processing

In this section we give a brief review of the discontinuous Galerkin method, developed by Cockburn and Shu et al. [5, 6, 9, 10, 11], for treating nonlinear problems, and the related post-processing technique, introduced by Cockburn, Luskin, Shu and Süli [4, 7, 8, 12]. The authors show that the post-processor nearly doubles the order of accuracy for time-dependent linear hyperbolic systems solved over a locally uniform mesh. In addition, numerical evidence strongly indicates that the method also improves the convergence for nonlinear problems as well. The applications of the post-processor were extended in [19] to include multi-domains with different mesh sizes as well as variable and discontinuous coefficient equations. The extension of this technique to one-sided post-processing appears in [20]. Detailed discussion of the post-processor, including implementation, can be found in [7, 8, 19, 20].

2.1 The discontinuous Galerkin method (DGM)

We begin by considering the discontinuous Galerkin method for the one-dimensional conservation law

$$u_t + f(u)_x = 0. \tag{2.1}$$

The mesh is defined by $\triangle x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ where $I_i = (x_i - \frac{\triangle x_i}{2}, x_i + \frac{\triangle x_i}{2})$, and x_i denotes the cell center, $i = 1, \cdots, N$. The approximation space consists of piecewise polynomials of degree less than or equal to k, where $k+1$ is the order of accuracy of the approximation, that is, $V_h = \{v | v \in \mathbb{P}^k \text{ for } x \in I_i\}$. The discontinuous Galerkin method is found by multiplying equation (2.1) by a test function $v \in V_h$ and integrating by parts to obtain the variational formulation:

Find $u_h(x,t) \in V_h$ such that

$$\int_{I_i} u_t v dx = \int_{I_i} f(u) v_x dx - f(u_{i+1/2}) v_{i+1/2} + f(u_{i-1/2}) v_{i-1/2} \quad \forall v \in V_h.$$

The numerical scheme is then given by:

$$\int_{I_i} (u_h)_t v dx = \int_{I_i} f(u_h) v_x dx - \hat{f}_{i+1/2} v^-_{i+1/2} + \hat{f}_{i-1/2} v^+_{i-1/2} \tag{2.2}$$

for all test functions $v \in V_h$. The numerical flux, $\hat{f}_{i+1/2} = \hat{f}(u^-_{i+1/2}, u^+_{i+1/2})$, is chosen to be an upwind monotone flux, i.e. it is a non-decreasing function of the first argument u^- and a non-increasing function of the second argument u^+. The test function v is taken from inside the cell. We note that numerical integration of (2.2) by the third-order SSP Runge-Kutta method (see e.g. [14, 15, 21]) guarantees the order of accuracy to agree with the spatial discretization error.

2.2 Post-processing for DGM

It was shown in [8] for linear hyperbolic equations that the negative-order norm error estimate, which provides information about the oscillatory nature of the error, is of higher order than the usual L_2 norm error estimate for the DGM. The post-processing technique in [20] both reduces oscillations as well as improves the L_2 norm accuracy. In fact it is possible to enhance the overall accuracy to be up to the order of the error estimate in the negative-order norm. In addition, the post-processor can be numerically simplified if a locally uniform mesh is assumed, yielding translation invariance and subsequently localizing the post-processor. This assumption will be used in our study, resulting in a post-processor of the form

$$u^*(x) = \frac{1}{h}\int_{-\infty}^{\infty} K^{2(k+1),k+1}\left(\frac{y-x}{h}\right) u_h(y)\, dy, \tag{2.3}$$

where $K^{2(k+1),k+1}$ is a linear combination of B-splines and u_h is the numerical solution. The symmetric form of the post-processing kernel can be written as

$$K^{2(k+1),k+1}(x) = \sum_{\gamma=-k}^{k} c_\gamma^{2(k+1),k+1}\psi^{(k+1)}(x-\gamma), \tag{2.4}$$

where $\psi^{(k+1)}(x)$ is obtained by convolving the characteristic function over the interval $(-\frac{1}{2},\frac{1}{2})$ with itself k times and $c_\gamma^{2(k+1),k+1} \in \mathbb{R}$. The calculation of $c_\gamma^{2(k+1),k+1}$ as well as the construction and implementation of (2.3) is described in [3, 19, 20, 22, 23].

We note that since we are using an uniform mesh, the kernel (2.4) has a particularly simple form as in [7, 8, 19, 20]. Furthermore, the post-processor as well as the B-spline coefficients are symmetric. Finally there is an additional advantage in the local behavior of the post-processor, specifically that the kernel needs information only from its nearest neighbors.

The symmetric version of the post-processed solution in (2.3) uses $2k+1$ B-splines, giving a total support of $2k'+1$ cells where $k' = \lceil\frac{3k+1}{2}\rceil$. A one-sided version of this post-processor is performed by simply moving the support bias to one-side. For example, a purely left-sided post-processed solution would be of the form

$$u^*(x) = \frac{1}{h}\int_{-\infty}^{\infty} \sum_{\gamma=-3k}^{-k} c_\gamma^{2(k+1),k+1}\psi^{(k+1)}\left(\frac{y-x}{h}-\gamma\right) u_h(y)\,dy \tag{2.5}$$

for $k = 1$ or $k = 2$. This purely one-sided post-processor has a form similar to the centered one, just with different bounds on the summation and new coefficients $c_\gamma^{2k+1,2k}$ [20]. Similarly, a partially left-sided post-processor is obtained by changing the bounds of the summation. In each case, the use of $2k+1$ B-splines remains consistent. The right-sided post-processed solution is a mirror image of the left.

3 Post-processing stencil options

As explained in [20], advanced knowledge of the shock locations is required to determine the post-processing stencil best suited for reconstruction in a shock vicinity. A stencil defines a collection of points $x_{j-l}, \cdots, x_{j+k}$ surrounding the point x_j where we wish to determine the smoothness of the solution. Because of the considerations outlined in previous sections, we are only considering stencils of equally spaced points. We also assume that the mesh is sufficiently well resolved so that the shocks are well separated and therefore there is a sufficient number of points between shocks.

There are $2k'+1$ candidate stencils for the post-processor in each considered region, such as the five point candidate post-processing stencils for piecewise linear polynomials shown in Figure 5.1. An admissible stencil for the post-processing procedure avoids points containing shock values. To this end, here we present two alternative techniques for determining suitable stencils for post-processing the computed numerical solution. The ENO approach chooses the post-processor stencil based on the pattern of the overall smoothness of qualifying stencils. Alternatively, the LED method finds the shock locations directly and then dictates the post-processing stencil to avoid any shocks.

3.1 ENO type stencil choosing

The essential non-oscillatory (ENO) finite difference method for hyperbolic equations with discontinuities approximates the values at each x_j by interpolating values only in identified smooth regions [16, 17, 18, 21]. The ENO stencil choosing technique for the DGM post-processor developed here similarly seeks to evaluate the numerical solution based on the identified smoothest stencil, and then chooses a one-sided, partially one-sided, or a centered stencil to post-process the solution. Unlike the standard ENO algorithm, here we impose a bias toward centered stencils for the DGM solution post-processor since the magnitude of the error is reduced more significantly than if we use a one-sided stencil. (Note that $2k+1$ order accuracy is still obtained throughout the domain, including regions where one-sided post-processing is implemented). Below is a description of the ENO type stencil construction process.

Let us first consider the case where $k = 1$ in which a third order post-processor is used. Given the data u_j at all of the cell faces, we employ the recursive formula for $r = 1, 2, 3$:

$$\begin{aligned} D_i^1 &= |u_{i+1}(x_{i-1/2}) - u_i(x_{i-1/2})| \\ D_i^{r+1} &= |D_{i+1}^r - D_i^r|, \end{aligned} \tag{3.1}$$

where $i = j-4, \cdots, j+4-r$. Clearly D_j^1 is the difference between the downwind points across each cell I_j. Based on this difference, we make the following stencil choice, $S(j)$, for reconstruction at each cell I_j:

$$S(j) = S_2(j) = \begin{cases} 1 & \text{if } NP_j^1 = 4, \\ -1 & \text{if } NP_j^1 = 5, \\ 0 & \text{else,} \end{cases} \tag{3.2}$$

where NP_j^r is defined as the value for which $j - 4 + NP_j^r$ maximizes the rth difference, i.e.

$$D_{j-4+NP_j^r}^{r-1} = \max_{j-4 \le J \le j+4-r} D_J^{r-1}.$$

Table 5.1 displays the possible stencil choices when $S(j) = S_2(j)$. Although employing the maximum undivided difference yields post-processing stencils without shocks, the stencil might still include cells in which steep gradients of the solution exist, reducing the the overall accuracy of the method. This can be alleviated by implementing a stencil that considers both the first and second undivided difference. Namely,

$$S(j) = S_{32}(j) = \begin{cases} S_2(j) & \text{if } NP_j^2 = 2 \text{ and } NP_j^1 = 3, \\ S_2(j) & \text{if } NP_j^2 = 3,\ NP_j^2 = 4,\ NP_j^2 = 5, \\ S_2(j) & \text{if } NP_j^2 = 6 \text{ and } NP_j^1 = 6, \\ 0 & \text{else.} \end{cases} \tag{3.3}$$

The $S_{32}(j)$ stencil examines the seven options of maximal 3-point stencils and compares them to the maximal two point stencil. To refine this further, we consider the first three differences:

$$S(j) = S_{432}(j) = \begin{cases} S_{32}(j) & \text{if } NP_j^3 = 1 \text{ and } 1 \leq NP_j^2 \leq 2, \\ S_{32}(j) & \text{if } NP_j^3 = 2 \text{ and } 2 \leq NP_j^2 \leq 3, \\ S_{32}(j) & \text{if } NP_j^3 = 3 \text{ and } 3 \leq NP_j^2 \leq 4, \\ S_{32}(j) & \text{if } NP_j^3 = 4 \text{ and } 4 \leq NP_j^2 \leq 5, \\ S_{32}(j) & \text{if } NP_j^3 = 5 \text{ and } 5 \leq NP_j^2 \leq 6, \\ S_{32}(j) & \text{if } NP_j^3 = 6 \text{ and } 6 \leq NP_j^2 \leq 7, \\ 0 & \text{else,} \end{cases} \tag{3.4}$$

or the first four differences:

$$S(j) = S_{5432}(j) = \begin{cases} S_{432}(j) & \text{if } NP_j^4 = 1 \text{ and } 1 \leq NP_j^3 \leq 4, \\ S_{432}(j) & \text{if } NP_j^4 = 2 \text{ and } 1 \leq NP_j^3 \leq 5, \\ S_{432}(j) & \text{if } NP_j^4 = 3, \\ S_{432}(j) & \text{if } NP_j^4 = 4 \text{ and } 2 \leq NP_j^3 \leq 6, \\ S_{432}(j) & \text{if } NP_j^4 = 5 \text{ and } 3 \leq NP_j^3 \leq 6, \\ 0 & \text{else.} \end{cases} \tag{3.5}$$

Implementing successive levels for $S(j)$ is necessary to provide consistency in determining a suitable post-processing stencil.

The $k = 1$ stencil choice essentially finds the discontinuity, and is thus sufficient for suggesting the correct post-processing stencil. Hence, the same stencil choosing method (for $k = 1$) described above can be performed with the information given by the $\mathbb{P}^k$-polynomial approximation, for $k \geq 2$. Once this skeleton stencil choice – the stencil choice that is based on 5-point stencils – is determined, the 9-point stencil choice easily follows. Specifically, if the 5-point stencil is biased then the 9 point stencil is biased in the same way to avoid the same discontinuous region. The ENO stencil choosing algorithm for $\mathbb{P}^k$, $k = 1, 2$ is the following:

```
If $S(j) \neq 0$ then do nothing.
Else if $S(j) = 0$ then
    If $S(j-1) = 1$ then $S(j) = 2$.
    If $S(j+1) = -1$ then $S(j) = -2$.
    If $k \geq 2$ then
        If $S(j-1) = 2$ then $S(j) = 3$.
        If $S(j-1) = 3$ then $S(j) = 4$.
        If $S(j+2) = -1$ then $S(j) = -3$.
        If $S(j+3) = -1$ then $S(j) = -4$.
    End If
End If
```

Table 5.2 displays the stencil options for $k = 2$. As is evident from the above algorithm, an advantage of using an ENO stencil choosing technique is that the complexity of the stencil choosing method increases minimally with increasing polynomial degree approximation.

3.2 A Local Edge Detection Method

The local edge detection (LED) method used in this study is based on the multivariate edge detection method for scattered data first described in [1]. Since here we are considering a uniform mesh, the analytical and computational aspects of the original method are considerably simplified, and essentially the method is reduced to constructing Newton divided difference formulations of various orders. Once the edges are located, the post-processing stencil is simply chosen to avoid the cells that contain jump discontinuities.

Let us denote $J = \{\xi : a \leq \xi \leq b\}$ as the jump discontinuities of $u(x)$ on $x \in [a, b]$. Each ξ is computed by the LED method to within a grid cell value. Subsequently, the size of the stencil $[x_l, x_r]$ used to post-process the solution at the value x^* is restricted by

$$x_{l-1} \leq \xi_\nu < x_l \leq x^* \leq x_r \leq \xi_{\nu+1} < x_{r+1},$$

for each $\xi_\nu \in J$. In this case, the size of the stencil guaranteeing a smooth region is $r - l + 1$, implying that for $u(x) \in C^k(x)$ in the region $[x_l, x_r]$, we can obtain $2k + 1$ accuracy as long as $2k' + 1 \leq r - l + 1$ (where $k' = \lceil \frac{3k+1}{2} \rceil$), which is in line with the resolution requirements for high order post-processing discussed in §2.2. Since the boundary points are defined as jump discontinuities, there is no formal reduction of accuracy as they are approached.

The main difference between the LED stencil construction procedure we describe in this section and the ENO stencil described in §3.1 is that while the ENO construction chooses candidate stencils based on a perceived smoothness of $u(x)$ in the region of reconstruction, the LED subscribes a particular stencil based directly on the location of the detected jump discontinuities. This procedure has been further extended to identify discontinuities in the derivative in [2], which will be described in §3.2.2. Hence the region of reconstruction can be chosen to avoid discontinuities in the derivative of the function as well.

As in the ENO stencil case, the LED method for one-dimensional data can be formulated using Newton divided differences. We also note that the LED method is completely independent of the order of the post-processing method since it is based solely on given grid point data. Hence there is no added complexity for higher order post-processing stencils. A brief synopsis is provided here.

3.2.1 Review of the one-dimensional mth order LED

Consider a piecewise smooth function $u(x)$ on $[a, b]$ known at the discrete grid point values $\mathcal{S} = \{x_1, \cdots, x_N\}$. We wish to determine whether or not there is a jump discontinuity at any point x in $[a, b]$. Following the development in [1], we construct an edge detector $L_m u(x)$ based on a stencil $\mathcal{S}_x$ containing $m + 1$ points around x to be an approximation of the jump function

$$[u](x) := u(x^+) - u(x^-), \tag{3.6}$$

where $u(x^+)$ and $u(x^-)$ are the right and left hand side limits of u at x. Typically in the uniform data case $\mathcal{S}_x$ is chosen so that there is an equal distribution of $m+1$ points around $x \in [x_{j-1}, x_j]$, i.e.

$$\mathcal{S}_x = \{x_{i-\frac{m}{2}}, \cdots, x_{i+\frac{m}{2}}\},$$

for each $x_i \in \mathcal{S}$. If x is near the boundary, the point distribution can be easily modified without affecting the properties of the LED method [1]. To ease notation we re-label the local set $\mathcal{S}_x$ for any point $x \in [a, b]$ as

$$\mathcal{S}_x = \{x_1, \cdots, x_{m+1}\}.$$

Recall that $J = \{\xi : a \leq \xi \leq b\}$ is the set of jump discontinuities in $[a, b]$ for $u(x)$. The edge detector $L_m u(x)$ is characterized by the asymptotic convergence property

$$L_m u(x) \longrightarrow \begin{cases} [u](\xi), & \text{if } x_i \leq \xi, x \leq x_{i+1} \text{ for } \xi \in J, \\ 0, & \text{if } I_x \cap J = \emptyset, \end{cases} \tag{3.7}$$

where I_x is the smallest closed interval such that $\mathcal{S}_x \subset I_x$.

The general form of the one-dimensional mth order edge detection method is given by

$$L_m u(x) = \frac{1}{q_m(x)} \sum_{x_j \in \mathcal{S}_x} c_{j,\mathcal{S}_x}(x) u(x_j), \quad j = 1, \cdots, m+1. \tag{3.8}$$

The coefficients $c_{j,\mathcal{S}_x}(x)$ are chosen to annihilate polynomials of degree m. Hence (3.8) can yield up to mth order accuracy in recovering smooth regions, i.e. where $I_x \cap J = \emptyset$, depending on the underlying smoothness of $u(x)$. Specifically, it was shown in [1] that for $\Delta x = \max\{|x_{i+1} - x_i|, i = 1, \cdots, N\}$, we have

$$L_m u(x) = \begin{cases} [u](\xi) + \mathcal{O}(\Delta x), & \text{if } x_i \leq \xi, x \leq x_{i+1} \text{ for } \xi \in J, \\ \mathcal{O}(\Delta x^{min(m,k)}), & \text{if } u \in C^k(I_x) \text{ for } k > 0. \end{cases} \tag{3.9}$$

We determine $c_{j,\mathcal{S}_x}(x)$ by solving the system

$$\sum_{x_j \in \mathcal{S}_x} c_{j,\mathcal{S}_x}(x) p_i(x_j) = p_i^{(m)}(x). \tag{3.10}$$

Here $p_i(x)$, $i = 1, \cdots, m$ is a basis of $\mathbb{P}^m$, where $\mathbb{P}^m$ denotes the space of all polynomials of degree $\leq m$. Note that $p_i^{(m)}(x)$ is constant for all m, and therefore all $c_{j,\mathcal{S}_x}(x)$ are also constant. Hence we write $c_{j,\mathcal{S}_x} := c_{j,\mathcal{S}_x}(x)$ for each $x_j \in \mathcal{S}_x$.

The normalization factor $q_m(x)$ is set as

$$q_m(x) := \sum_{x_j \in \mathcal{S}_x^+} c_{j,\mathcal{S}_x},$$

where we have defined the sets

$$\mathcal{S}_x^+ := \{x_j \in \mathcal{S}_x | x_j \geq x\} \text{ and } \mathcal{S}_x^- := \mathcal{S}_x \setminus \mathcal{S}_x^+. \tag{3.11}$$

(Note that $\sum_{x_j \in \mathcal{S}_x^+} c_{j,\mathcal{S}_x} = -\sum_{x_j \in \mathcal{S}_x^-} c_{j,\mathcal{S}_x}$.)

In the one-dimensional case, (3.8) can be reduced to a Newton divided difference formula [1]. Specifically, by defining

$$\omega_j(\mathcal{S}_x) := \omega_j(x_1, x_2, \ldots, x_{k+1}) = \prod_{\substack{i=1 \\ i \neq j}}^{m+1} (x_j - x_i), \tag{3.12}$$

we can directly compute

$$c_{j,\mathcal{S}_x} = \frac{m!}{\omega_j(\mathcal{S}_x)}, \quad j = 1, \cdots, m+1, \tag{3.13}$$

from (3.10) yielding

$$L_m u(x) = \frac{m!}{q_m(x)} u[\mathcal{S}_x]. \tag{3.14}$$

This is further simplified if the given set of points $\mathcal{S}$ is uniform on $[a, b]$:

$$\mathcal{S} := \{x_i := a + i\Delta x \mid i = 1, \cdots, N\}, \quad \Delta x = \frac{b-a}{N} > 0.$$

In this case we have

$$c_j = c_{j,\mathcal{S}_x} = \frac{m!}{\omega_j(\mathcal{S}_x)} = \frac{m!}{h \prod_{i=1, i \neq j}^{m+1} (j-i)}, \quad j = 1, \cdots, m+1,$$

and the corresponding edge detection method

$$L_m u(x) = \frac{1}{q_m} \sum_{x_j \in \mathcal{S}_x} c_j u(x_j), \tag{3.15}$$

for $q_m = \sum_{x_j \in \mathcal{S}_x^+} c_j$. It was shown in [1] that $q_m(x)$ is never zero.

3.2.2 Discontinuities in the derivative

Many partial differential equations, including some of the examples in §4, admit solutions with discontinuities in the derivative of the solution. Since high order reconstruction requires the function to be as smooth as possible, it is desirable to determine all of the jump discontinuities for $u^{(\gamma)}(x)$ where $\gamma = 0, \cdots, k$ and k is the order of the reconstruction. In [2] the LED method was extended to compute the jump discontinuities of $u^{(\gamma)}(x)$. The method is given for one dimension as

$$L^{\gamma}_{m,\mathcal{S}_x}u(x) = \frac{1}{q^{\gamma}_{m}(x)} \sum_{x_j \in \mathcal{S}_x} c_{j,\mathcal{S}_x}(x)u(x_j), \qquad j = 1, \cdots, m+1, \tag{3.16}$$

with

$$q^{\gamma}_{m}(x) := \sum_{x_j \in \mathcal{S}^{+}_{x}} c_{j,\mathcal{S}_x} \frac{(x_j - x)^{\gamma}}{\gamma!}. \tag{3.17}$$

When $\gamma = 0$ the construction is exactly the same as in (3.8). The convergence rate was shown in [2] to be

$$L^{\gamma}_{m}u(x) = \begin{cases} D^{\gamma}_{m}(x,\xi)[u^{\gamma}](\xi) + \mathcal{O}(\Delta x), & \text{if } x_i \le \xi, x \le x_{i+1} \text{ for } \xi \in J, \\ \mathcal{O}(\Delta x^{min(m,k)-\gamma}), & \text{if } u \in C^k(I_x) \text{ for } k > 0, \end{cases} \tag{3.18}$$

where

$$D^{\gamma}_{m}(x,\xi) = \frac{\sum_{x_j \in \mathcal{S}^{+}_{\xi}} c_{j,\mathcal{S}_x}(x_j - \xi)^{\gamma}}{\sum_{x_j \in \mathcal{S}^{+}_{x}} c_{j,\mathcal{S}_x}(x_j - x)^{\gamma}}.$$

Note that $D^{\gamma}_{m}(x,\xi) = 1$ when $\gamma = 0$. Clearly the approximation (3.16) of the jump $[u^{(\gamma)}(x)]$ improves as resolution is increased. Also note that $D^{\gamma}_{m}(x,\xi)$ is closer to 1 when γ is small. More details can be found in [2]. For our purposes here $D^{\gamma}_{m}(x,\xi)$ need not be computed. Although the edge detection method yields up to $m - \gamma$ order convergence away from the discontinuities of a piecewise smooth function, some difficulties arise. For small m, there is a risk of wrongly identifying steep gradients as jump discontinuities, especially in the case of limited resolution. On the other hand, as m increases, oscillations that occur in the neighborhood of a jump discontinuity can be wrongly identified as discontinuities. As noted in [2, 13], nonlinear enhancement techniques and outside thresholds can be utilized to pinpoint the jump discontinuities when $\gamma = 0$. Unfortunately for $\gamma > 0$ the oscillations near the jump discontinuities grow too large to distinguish them from the true jumps. This is particularly problematic if the jump discontinuities are located close together, i.e., when the resolution is poor. Hence a different approach was used in [2]. Specifically, rather than using outside thresholds to determine the jump discontinuities from the approximation (3.16), the set of jump discontinuities

$$J_\gamma = \left\{ \xi : f \in C^{\gamma-1}(\xi) \text{ and } [u^{(\gamma)}](\xi) \neq 0 \right\}, \tag{3.19}$$

must be determined explicitly for each γ. The procedure is given as follows:

Consider the midpoints $x_{i-\frac{1}{2}} = \frac{x_{i-1}+x_i}{2}$ for $i = 1, \ldots, N$ and construct the approximation (3.16) for the particular local sets

$$\mathcal{S}_{x_{i-\frac{1}{2}}} = \{x_{i-m}, \ldots, x_i\} \quad \text{and} \quad \tilde{\mathcal{S}}_{x_{i-\frac{1}{2}}} = \{x_{i-1}, \ldots, x_{i+\tilde{m}-1}\} \tag{3.20}$$

such that $i - m \geq 0$, $i + \tilde{m} - 1 \leq N$, and $m, \tilde{m} > \gamma$. By choosing an appropriate threshold, the regions where $u^{(\gamma)}$ is smooth can be eliminated, i.e. we eliminate all grid cells that contain values of x such that

$$L^{\gamma}_{m,\mathcal{S}_{x_{i-\frac{1}{2}}}} u(x) < (\Delta x)^{\gamma} \quad \text{or} \quad L^{\gamma}_{m,\tilde{\mathcal{S}}_{x_{i-\frac{1}{2}}}} u(x) < (\Delta x)^{\gamma}. \tag{3.21}$$

The discontinuities of $u^{(\gamma)}(x)$ are then obtained by looking at the remaining regions where we find the root of

$$L^{\gamma}_{m,\mathcal{S}_{x_{i-\frac{1}{2}}}} u(x) - L^{\gamma}_{m,\tilde{\mathcal{S}}_{x_{i-\frac{1}{2}}}} u(x) = 0. \tag{3.22}$$

The solution of (3.22) $x = \xi$ is added to the set J_γ. In this way, all of the jump discontinuity locations of $u^{(\gamma)}(x)$ for $\gamma \geq 0$ can be pinpointed exactly. Further details and analysis is provided in [2], as well as an algorithm for computing (3.22).

Recall that the data is given on uniform points and therefore c_j and $\tilde{c}_j$ are determined by (3.13). Since the local sets (3.20) are fixed, the only variability comes from the normalization factor (3.17). As shown in [2], this particular construction enforces the uniqueness of the root satisfying (3.22) to within the cell $[x_{i-1}, x_i]$. Furthermore, the jump discontinuity is found to within subcell resolution accuracy. Hence the LED method, in combination with the subcell resolution algorithm (3.22) can successfully detect discontinuities for any order derivative. We note that the jump discontinuities are required to be at a minimum distance of $(m + \tilde{m})\Delta x$ with $m, \tilde{m} > \gamma$. In the applications that follow, this resolution requirement is not that prohibitive. Furthermore, as the resolution is increased, the order of the edge detection method can be increased since the jump discontinuities have more points between them. In the examples that follow the method is used for $\gamma = 0, 1$ and $m = \tilde{m}$, typically taking values between 2 and 5, which is enough to effectively resolve the jump discontinuities while still maintaining the advantages of the LED in low resolution environments. The post-processing reconstruction will only occur in regions that are at least C^2, yielding at least second order accuracy.

4 Numerical results

We analyzed the ENO type and LED stencil choosing algorithms by testing them on four scalar test cases, each with a DG approximation of polynomial degree $k = 1$ and $k = 2$. We evaluated the performance of the ENO S_{32}, S_{432} and S_{5432} and the LED stencil choosing methods to demonstrate their respective strength and weaknesses. For each example, we compared the post-processed results from the automated stencil choosing using the ENO S_{5432} and the comparable LED stencil choosing method. We chose to work with the S_{5432} ENO type method as it was the most consistent among the ENO type methods. Three of the sample problems were linear, chosen because the theory only guarantees the order enhancement for the linear case. First the stencil choices were tested on a smooth linear problem in Example 1 to ensure that neither algorithm introduces any errors if used unnecessarily. The second example features two stationary shocks. The third example has two stationary shocks and two moving discontinuities in the derivative. Finally a nonlinear problem with a moving discontinuity is presented to demonstrate that the algorithms effectively capture a moving discontinuity.

We present error error tables for the numerical solution as well as the first derivative of the solution to demonstrate that the $2k+1$ order accuracy is indeed achieved throughout the domain as long as the discontinuity lies on a cell boundary. The L_2-errors presented are calculated at the cell centers unless otherwise indicated. Furthermore, we remark that the main advantage of using one-sided post-processing near a discontinuity is in obtaining the $2k+1$ order accuracy in the first derivative of the solution. As seen in (2.5), the post-processed solution is a polynomial of degree $2k+1$, which typically would allow us to obtain $2k+2$ order accuracy. In [7, 8] it was shown that $2k+1$ order accuracy can be obtained for the solution to time-dependent linear hyperbolic equations. This same order accuracy is also achieved in the first derivative for these same equations, as numerically demonstrated in [20].

Example 1: Linear scalar convection equation

$$u_t + u_x = 0, \quad 0 \le x \le 2\pi$$

with initial condition

$$u(x,0) = \sin(x)$$

is solved to time $T_f = 12.5$ and the error for the solution is calculated throughout the entire domain. For this problem, all the ENO stencil choosing methods as well as the LED method recognize the smoothness of the solution and thus choose centered post-processing consistently (Figure 5.2). The order of accuracy for the solution and the first derivative of the solution is increased from second order to above third order in the $k = 1$ case, and from third order to above fifth order in the $k = 2$ case (see Table 5.3). This example confirms that the post-processor with automated shock detection does not introduce errors.

Example 2: Linear discontinuous coefficient problem with two stationary shocks

$$u_t + (au)_x = 0$$

for $-1 \le x \le 1$ where

$$a(x) = \begin{cases} \frac{1}{2}, & \text{for } -\frac{1}{2} \le x \le \frac{1}{2} \\ 1 & \text{else,} \end{cases}$$

with initial conditions

$$u(x,0) = \begin{cases} -2\cos(4\pi x) & \text{for } -\frac{1}{2,} \le x \le \frac{1}{2} \\ \cos(2\pi x) & \text{else.} \end{cases}$$

The exact solution at time $T_f = 4$ is shown in Figure 5.5 (left). This solution features two stationary shocks lying on cell boundaries at $x = \pm\frac{1}{2}$. The numerical solution is computed on a sequence of meshes ranging in resolution from $N = 40$ to $N = 160$. The error in the solution and in the first derivative of the solution, before and after post-processing, is calculated at time $T_f = 4$ over the entire domain and is given in Table 5.4. For this example, the ENO type methods all bias unnecessarily for the low resolution meshes $N = 40$ and $N = 60$. However, the S_{5432} falsely biases less often, and thus gives more reliable stencil choices than the S_{432} and S_{32}. The ENO S_{5432} stencil choices for $N = 40, 80, 160$ are plotted in Figure 5.3. For $N = 40$ and $N = 60$ the errors from the ENO stencil choice is larger than the error from the LED stencil (Table 5.4), implying that the LED method handles low resolution environments better than the ENO type methods. Once the the grid is sufficiently refined, all four methods identically yield the correct stencil choices.

Since the discontinuities are located at the cell boundaries, the discontinuous Galerkin method can produce $2k+1$ order accuracy in the post-processed numerical solution and its first derivative throughout the domain, as long as the stencil choice is correct (Table 5.4). As in [20], the errors for the numerical solution are better than the expected $2k+1$ order accuracy.

Example 3: Linear discontinuous coefficient problem with two stationary shocks and two moving shocks

$$u_t + (au)_x = 0$$

for $-2 \leq x \leq 2$ where

$$a(x) = \begin{cases} \frac{1}{2} & \text{for } -1 \leq x \leq 1, \\ 1 & \text{else,} \end{cases}$$

with initial condition,

$$u(x,0) = \begin{cases} \frac{2}{3}\sin(\pi x) & \text{for } -1 \leq x \leq 1, \\ \cos(\frac{1}{2}\pi x) & \text{else.} \end{cases}$$

The solution at time $T_f = 1.0$, shown in Figure 5.5 (right) features two stationary and two moving shocks where the stationary shocks at $x = \pm 1$ are located at cell boundaries and one moving shock is located near the right boundary and the other moving shock is located near the stationary shock at $x = -1$. The solution is computed to time $T_f = 1.0$, which is before the shocks cross. The error in the numerical solution and in the first derivative of the solution is calculated to include the stationary shocks at $x = \pm 1$, but to exclude a radius of 0.1 around the moving shocks as in [19, 20].

For $k = 1$, all of the ENO type stencil choices (S_{32}, S_{432} and S_{5432}), as well as the LED method, detect the stationary shocks reliably even for low mesh resolution. However, the moving shock in the middle of the domain, which has a discontinuity only in the derivative, is not detected by the ENO stencil choices, but is detected by the LED method. For low resolution the two shocks (the stationary and moving shocks) are too close together for a stencil between them. Thus, we are only interested in the case of $N \geq 60$. In this case, the LED method is clearly superior to the ENO type stencil choices. However, as we are excluding a region of 0.1 around the shock, this advantage is not evident in the error tables. For $k = 2$ the ENO type stencil choices (S_{32}, S_{432}, and S_{5432}) and the LED method are all equivalent and all capture the correct shock location for all resolutions.

These stencil choices allow the post-processor to both decrease the magnitude of error and increase the order of accuracy. Table 5.5 shows that the order of accuracy of the post-processed solution was raised from three to above six. Again we note that the post-processor gives better than expected errors for the solution, similar to those shown in [20]. In this case, the ENO stencil choosing method may have trouble in the presence of a discontinuity in the derivative, as this method is not designed to determine the locations of discontinuities in the derivative. Remarkably, this method does detect the discontinuity in the solution for the $k = 2$ case. The extension of the ENO type stencil choosing methods to cases of discontinuities in the derivative will be addressed in future work.

Example 4: Burgers' equation

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0, \quad 0 \leq x \leq 2\pi,$$

with initial conditions

$$u(x,0) = \frac{1}{2} + \sin(x),$$

is solved to time $T_f = 2.0$, after a shock has formed as in [8]. This example features a moving discontinuity which all of the ENO stencil choices as well as the LED method capture for all choices of N (Figure 5.6).

5 Conclusions

We have presented two effective methods for finding the appropriate post-processing stencil. The ENO type stencil choosing is designed to pick centered post-processing in smooth regions and one-sided or partially one-sided post-processing in the neighborhood of a discontinuity while the LED method directly locates the shocks in the numerical solution to determine the correct post-processing stencil.

The ENO type stencil choosing is based on the smoothness of the candidate stencils and is determined by examining the first four undivided differences over the candidate stencils. Implementation of the successive levels is necessary to avoid one-sided post-processing in regions where steep gradients in the solution exist. This type of stencil choosing method requires minimal complexity as the polynomial degree used in the approximation is increased.

The LED method is particularly effective in identifying the shock locations in the case of low resolution, as well as for non-uniform spacing. The algorithm is effective for determining jump discontinuities as well. The subcell resolution technique (3.22) guarantees that the jump discontinuities are pinpointed so that the post-processing reconstruction is guaranteed only to occur in smooth regions (at least C^2). The method is completely independent of the post-processing approximation order.

We have demonstrated through numerical examples that both these methods are viable and efficient techniques for automated choice of the post-processing stencil by enabling the designed $(2k+1)$-th order accuracy of the post-processor throughout the entire domain. No other pre-determination of discontinuity locations is necessary.

In the case where the discontinuity is in the derivative, the ENO type stencil choosing method is unreliable, while the LED method works very well. However, we suspect that the ENO type algorithms can be readily adapted for this purpose and will be the subject of future investigations. Future work will include extending these methods to systems, multiple space dimensions, and non-uniform mesh size.

Acknowledgments: The authors are grateful to the reviewer for pointing out the work of Thomée [22].

References

[1] Archibald, R., Gelb, A., and Yoon, J. (2004). Polynomial Fitting for Edge Detection in Irregularly Sampled Signals and Images, to appear in SINUM.

[2] Archibald, R., Gelb, A., and Yoon, J. (2004). Determining the Locations of the Discontinuities in the Derivatives of Functions, submitted to Journal of Scientific Computing.

[3] Bramble, J.H., and Schatz, A.H. (1977). Higher order local accuracy by averaging in the finite element method, *Mathematics of Computation,* **31**, 94-111.

[4] Cockburn, B. (1999). Discontinuous Galerkin methods for convection-dominated problems.In Barth, T.J., and Deconinck, H. (ed.), *High-Order Methods for Computational Physics,Lecture Notes in Computational Science and Engineering,* **9,** Springer, 69-224.

[5] Cockburn, B., Hou, S. and Shu, C.-W. (1990). The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case,*Mathematics of Computation,* **54**, 545-581.

[6] Cockburn, B., Lin, S.-Y. and Shu, C.-W. (1989) TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems,*Journal of Computational Physics,* **84**, 90-113.

[7] Cockburn, B., Luskin, M., Shu, C.-W. and Süli, E. (1999). Post-processing of Galerkin methods for hyperbolic problems, Karniadakis, G., Cockburn, B. and Shu, C.-W. (ed.), *Proceedings of the International Symposium on Discontinuous Galerkin Methods, Lecture Notes in Computational Science and Engineering,* **11**, Springer, 291-300.

[8] Cockburn, B., Luskin, M. , Shu, C.-W., and Süli, E. (2003). Enhanced accuracy by post-processing for finite element methods for hyperbolic equations, *Mathematics of Computation,***72**, 577-606.

[9] Cockburn, B., and Shu, C.-W. (1989). TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, *Mathematics of Computation,* **52,** 411-435.

[10] Cockburn, B., and Shu, C.-W. (1991). The Runge-Kutta local projection $\mathbb{P}^1$-discontinuous-Galerkin finite element method for scalar conservation laws,*Mathematical Modeling and Numerical Analysis (M^2AN),* **25**, 337-361.

[11] Cockburn, B., and Shu, C.-W. (1998). The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems,*Journal of Computational Physics,* **141**, 199-224.

[12] Cockburn, B., and Shu, C.-W. (2001). Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems,*Journal of Scientific Computing,* **16**, 173-261.

[13] Gelb, A., and Tadmor, E. (2000). *Detection of Edges in Spectral Data II. Nonlinear Enhancement,* SIAM J. Numer. Anal., **38** 1389-1408.

[14] Gottlieb, S., and Shu, C.-W. (1998). *Total variation diminishing Runge-Kutta schemes*, Mathematics of Computation, **67**, 73-85.

[15] Gottlieb, S., Shu, C.-W. and Tadmor, E. (2001). *Strong Stability Preserving High-Order Time Discretization Methods*, SIAM Review, **43**, 89-112.

[16] Harten, A., Engquist, B., Osher, S. , and Chakravarthy, S. (1987). Uniformly High Order Essentially Non-Oscillatory Schemes I, *SIAM Journal on Numerical Analysis,* **24**, 279-309.

[17] Harten, A., Engquist, B., Osher, S. , and Chakravarthy, S. (1987). Uniformly High Order Essentially Non-Oscillatory Schemes, *Journal of Computational Physics,* **71**, 231.

[18] Liu, X.-D., Osher, S., and Chan, T. (1994) Weighted Essentially Non-Oscillatory Schemes, *Journal of Computational Physics,* **115**, 200-212.

[19] Ryan, J.K., Shu, C.-W., and Atkins, H.L. Extension of a post-processing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem, *SIAM Journal on Scientific Computing,* to appear.

[20] Ryan, J.K., and Shu, C.-W. (2003) On a one-sided post-processing technique for the discontinuous Galerkin method, *Methods and Applications of Analysis,* **10(2)**, 295-307.

[21] Shu, C.-W., and Osher, S. (1988). Efficient implementation of essentially non-oscillatory shock-capturing schemes,*Journal of Computational Physics,* **77**, 439-471.

[22] Thomée, V. (1977). High order local approximations to derivatives in the finite element method, *Mathematics of Computation,* **31**, 652-660.

[23] Wahlbin, L.B. (1995). *Superconvergence in Galerkin finite element methods, Lecture Notes in Mathematics,* **1605**, Springer Verlag.

Figure Captions

Figure 5.1 Five point candidate post-processing stencil choices using a piecewise $\mathbb{P}^1$- polynomial approximation.

Figure 5.2 Example 1 for $k = 1$ (top) and $k = 2$ (bottom). The stencil choice for the S_{5432} ENO type method is on the left and the LED method is on the right. Both methods correctly choose centered post-processing away from th e domain boundary. $N = 40 = \bullet$, $N = 80 = +$, $N = 160 = \Box$.

Figure 5.3 Example 2 with $k = 1$ (top) and $k = 2$ (bottom). The LED stencil choosing (right) is able to handle low resolution better than th e S_{5432} ENO type method (left). Both the S_{5432} and LED methods are able to detect the stationary shocks at $x = \pm\frac{1}{2}$. $N = 40 = \bullet$, $N = 80 = +$, $N = 160 = \Box$.

Figure 5.4 Example 3 with $k = 1$ (top) and $k = 2$ (bottom). Both the ENO-S_{5432} (left) and LED (right) methods detect the stationary and moving shocks for $k = 2$, but the ENO method has difficulty detecting the moving shocks for $k = 1$. $N = 40 = \bullet$, $N = 80 = +$, $N = 160 = \Box$.

Figure 5.5 Numerical solutions for Example 2 (left) and Example 3 (right).

Figure 5.6 Example 4 for $k = 1$ (top) and $k = 2$ (bottom). Neither the ENO-S_{5432} (left) nor LED (right) method has difficulty detecting the moving shock. $N = 40 = \bullet$, $N = 80 = +$, $N = 160 = \Box$.

Table 5.1: Stencils and stencil numbers for $k = 1$.

Stencils for $k = 1$		
S(j)	stencil	bias
-1	$\{u_{j-4}, u_{j-3}, u_{j-2}, u_{j-1}, u_j \}$	left
-2	$\{u_{j-3}, u_{j-2}, u_{j-1}, u_j, u_{j+1} \}$	left + 1
0	$\{u_{j-2}, u_{j-1}, u_j, u_{j+1}, u_{j+2} \}$	centered
2	$\{u_{j-1}, u_j, u_{j+1}, u_{j+2}, u_{j+3} \}$	right - 1
1	$\{u_j, u_{j+1}, u_{j+2}, u_{j+3}, u_{j+4} \}$	right

Table 5.2: Stencils and stencil numbers for $k = 2$.

Stencils for $k = 2$		
S(j)	stencil	bias
-1	$\{u_{j-8}, u_{j-7}, u_{j-6}, u_{j-5}, u_{j-4}, u_{j-3}, u_{j-2}, u_{j-1}, u_j \}$	left
-2	$\{u_{j-7}, u_{j-6}, u_{j-5}, u_{j-4}, u_{j-3}, u_{j-2}, u_{j-1}, u_j, u_{j+1} \}$	left + 1
-3	$\{u_{j-6}, u_{j-5}, u_{j-4}, u_{j-3}, u_{j-2}, u_{j-1}, u_j, u_{j+1}, u_{j+2} \}$	left + 2
-4	$\{u_{j-5}, u_{j-4}, u_{j-3}, u_{j-2}, u_{j-1}, u_j, u_{j+1}, u_{j+2}, u_{j+3} \}$	left + 3
0	$\{u_{j-4}, u_{j-3}, u_{j-2}, u_{j-1}, u_j, u_{j+1}, u_{j+2}, u_{j+3}, u_{j+4} \}$	centered
4	$\{u_{j-3}, u_{j-2}, u_{j-1}, u_j, u_{j+1}, u_{j+2}, u_{j+3}, u_{j+4}, u_{j+5} \}$	right - 3
3	$\{u_{j-2}, u_{j-1}, u_j, u_{j+1}, u_{j+2}, u_{j+3}, u_{j+4}, u_{j+5}, u_{j+6} \}$	right - 2
2	$\{u_{j-1}, u_j, u_{j+1}, u_{j+2}, u_{j+3}, u_{j+4}, u_{j+5}, u_{j+6}, u_{j+7} \}$	right - 1
1	$\{u_j, u_{j+1}, u_{j+2}, u_{j+3}, u_{j+4}, u_{j+5}, u_{j+6}, u_{j+7}, u_{j+8} \}$	right

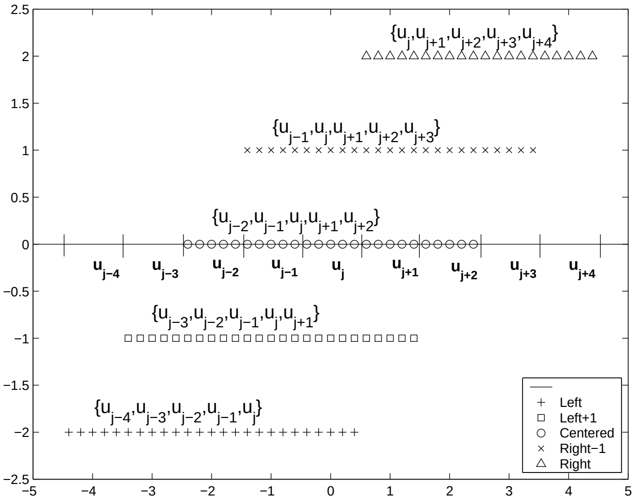

Figure 5.1:

Table 5.3: The L_2 error and order of accuracy for the solution of Example 1 calculated with $k = 1$ and $k = 2$. Results are shown before and after post-processing determined by stencil choices S_{5432} and LED.

	Before		S_{5432}		LED	
N	L_2 error	order	L_2 error	order	L_2 error	order
			Errors in Solution			
k=1						
40	1.1994E-03	—	8.4052E-04	—	8.4052E-04	—
60	4.6352E-04	2.34	2.2290E-04	3.27	2.2290E-04	3.27
80	2.4106E-04	2.27	8.7144E-05	3.26	8.7145E-05	3.26
100	1.4670E-04	2.23	4.2253E-05	3.24	4.2253E-05	3.24
120	9.8368E-05	2.19	2.3474E-05	3.22	2.3474E-05	3.22
140	7.0427E-05	2.17	1.4321E-05	3.21	1.4321E-05	3.21
160	5.2861E-05	2.15	9.3540E-06	3.19	9.3541E-06	3.19
k=2						
40	1.1404E-05	—	8.4145E-05	—	1.0557E-05	—
60	3.3814E-06	3.00	1.0195E-05	5.21	1.2234E-06	5.32
80	1.4269E-06	3.00	2.1801E-06	5.36	2.5790E-07	5.41
100	7.3066E-07	3.00	6.5063E-07	5.42	7.6474E-08	5.45
120	4.2287E-07	3.00	2.4103E-07	5.45	2.8234E-08	5.47
140	2.6631E-07	3.00	1.0385E-07	5.46	1.2141E-08	5.48
160	1.7841E-07	3.00	5.0013E-08	5.47	5.8398E-09	5.48

Table 5.4: The L_2 errors and order of accuracy for the solution (top) and first derivative of the solution (bottom) for Example 2, calculated with $k = 1$ and $k = 2$. Results are shown for before and after post-processing with stencil choices determined by the S_{5432} and the LED method.

	Before		S_{5432}		LED	
N	L_2 error	order	L_2 error	order	L_2 error	order
	Errors in Solution					
k=1						
40	8.0432E-02	—	2.2491E-01	—	8.4815E-02	—
60	2.7136E-02	2.68	2.3724E-02	5.55	2.2551E-02	3.27
80	1.2584E-02	2.67	8.9582E-03	3.39	8.9582E-03	3.21
100	6.9905E-03	2.63	4.4844E-03	3.10	4.4844E-03	3.10
120	4.3553E-03	2.60	2.5753E-03	3.04	2.5753E-03	3.04
140	2.9360E-03	2.56	1.6183E-03	3.01	1.6183E-03	3.01
160	2.0959E-03	2.52	1.0841E-03	3.00	1.0841E-03	3.00
k=2						
40	1.0663E-03	—	7.4252E-02	—	2.0510E-02	—
60	3.0828E-04	3.06	1.8714E-02	3.40	2.4536E-03	5.24
80	1.2955E-04	3.01	4.2724E-03	5.13	4.4396E-04	5.94
100	6.6271E-05	3.00	1.1202E-03	6.00	1.1186E-04	6.18
120	3.8343E-05	3.00	3.6331E-04	6.18	3.5549E-05	6.29
140	2.4145E-05	3.00	1.3816E-04	6.27	1.3361E-05	6.35
160	1.6176E-05	3.00	5.9332E-05	6.33	5.6958E-06	6.39
	Errors in first derivative of the solution					
k=1						
40	1.5086E+00	—	3.0423E+00	—	8.4586E-01	—
60	9.0952E-01	1.25	3.2012E-01	5.55	3.0056E-01	2.55
80	6.6871E-01	1.07	1.3100E-01	3.11	1.3100E-01	2.89
100	5.3212E-01	1.02	6.6682E-02	3.03	6.6682E-02	3.03
120	4.4264E-01	1.01	3.8018E-02	3.08	3.8018E-02	3.08
140	3.7914E-01	1.00	2.3561E-02	3.10	2.3561E-02	3.10
160	3.3165E-01	1.00	1.5552E-02	3.11	1.5552E-02	3.11
k=2						
40	1.2741E-01	—	3.5888E+00	—	1.0608E-01	—
60	5.5669E-02	2.04	2.3168E+00	1.08	8.3199E-03	6.28
80	3.1175E-02	2.02	1.9245E-02	16.65	3.7412E-03	2.78
100	1.9920E-02	2.01	9.4099E-03	3.21	1.4121E-03	4.37
120	1.3823E-02	2.00	4.2776E-03	4.32	5.8566E-04	4.83
140	1.0152E-02	2.00	2.0576E-03	4.75	2.6923E-04	5.04
160	7.7706E-03	2.00	1.0601E-03	4.97	1.3513E-04	5.16

Table 5.5: The L_2 errors and order of accuracy for the solution and first derivatives of the solution for Example 3 which features two moving and two stationary shocks, calculated with $k = 1$ and $k = 2$. The errors are calculated outside a radius of 0.1 of the moving shocks. Results are shown before post-processing and after post-processing determined by stencil choices ENO-S_{5432} and LED methods.

	Before		S_{5432}		LED	
N	L_2 error	order	L_2 error	order	L_2 error	order
	Errors in Solution					
k=1						
40	1.0969E-03	—	2.0388E-03	—	1.0107E-02	—
60	4.5949E-04	2.15	3.4209E-04	4.40	3.4209E-04	8.35
80	2.5043E-04	2.11	9.4821E-05	4.46	9.4821E-05	4.46
100	1.5715E-04	2.09	3.5083E-05	4.46	3.5083E-05	4.46
120	1.0768E-04	2.07	1.5693E-05	4.41	1.5693E-05	4.41
140	7.8344E-05	2.06	8.0425E-06	4.34	8.0425E-06	4.34
160	5.9541E-05	2.06	4.5692E-06	4.23	4.5692E-06	4.23
k=2						
40	3.1528E-05	—	1.8373E-03	—	1.9032E-04	—
60	9.4838E-06	2.96	1.5620E-04	6.07	1.5224E-05	6.23
80	4.0275E-06	2.98	2.5501E-05	6.30	2.4366E-06	6.37
100	2.0698E-06	2.98	6.1383E-06	6.38	5.8134E-07	6.42
120	1.2006E-06	2.99	1.9034E-06	6.42	1.7942E-07	6.45
140	7.5734E-07	2.99	7.0484E-07	6.44	6.6257E-08	6.46
160	5.0797E-07	2.99	2.9754E-07	6.46	2.7922E-08	6.47
	Errors in First Derivative					
k=1						
40	3.8324E-02	—	8.7819E-03	—	3.0425E+01	—
60	2.5115E-02	1.04	2.4476E-03	3.15	2.4476E-03	11.89
80	1.8657E-02	1.03	9.3777E-04	3.33	9.3777E-04	3.33
100	1.4836E-02	1.03	4.3900E-04	3.40	4.3900E-04	3.40
120	1.2312E-02	1.02	2.3478E-04	3.43	2.3478E-04	3.43
140	1.0521E-02	1.02	1.3794E-04	3.45	1.3794E-04	3.45
160	9.1844E-03	1.02	8.6904E-05	3.46	8.6904E-05	3.46
k=2						
40	1.9866E-03	—	2.0750E-03	—	3.9894E-04	—
60	8.7943E-04	2.01	4.6036E-04	3.71	6.2260E-05	4.58
80	4.9381E-04	2.01	1.1390E-04	4.86	1.4300E-05	5.11
100	3.1572E-04	2.00	3.6119E-05	5.15	4.4020E-06	5.28
120	2.1911E-04	2.00	1.3808E-05	5.27	1.6575E-06	5.36
140	1.6091E-04	2.00	6.0605E-06	5.34	7.2110E-07	5.40
160	1.2315E-04	2.00	2.9534E-06	5.38	3.4945E-07	5.43

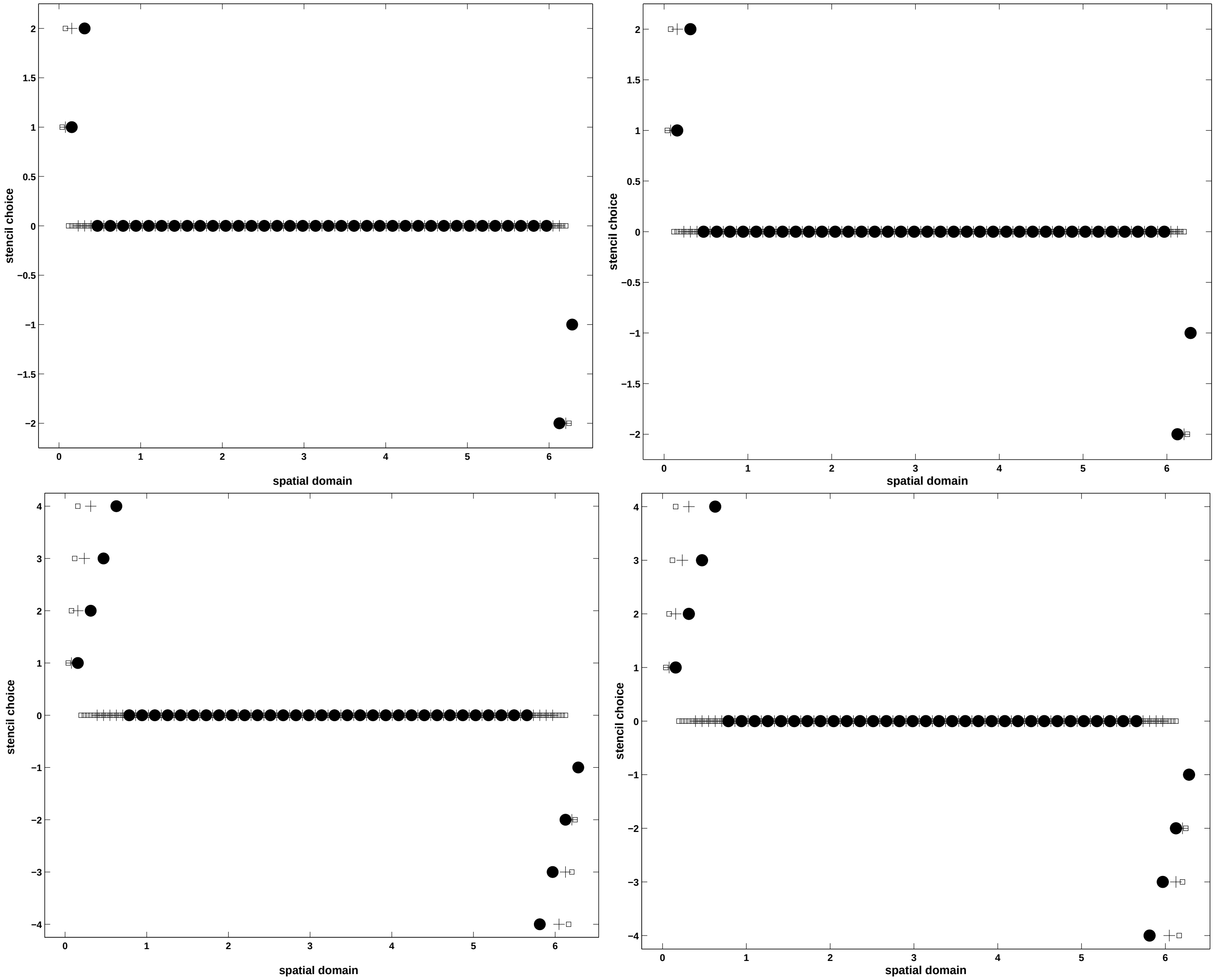
stencil choice
spatial domain
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
4
3
2
1
0
−1
−2
−3
−4
0
1
2
3
4
5
6

Figure 5.2:

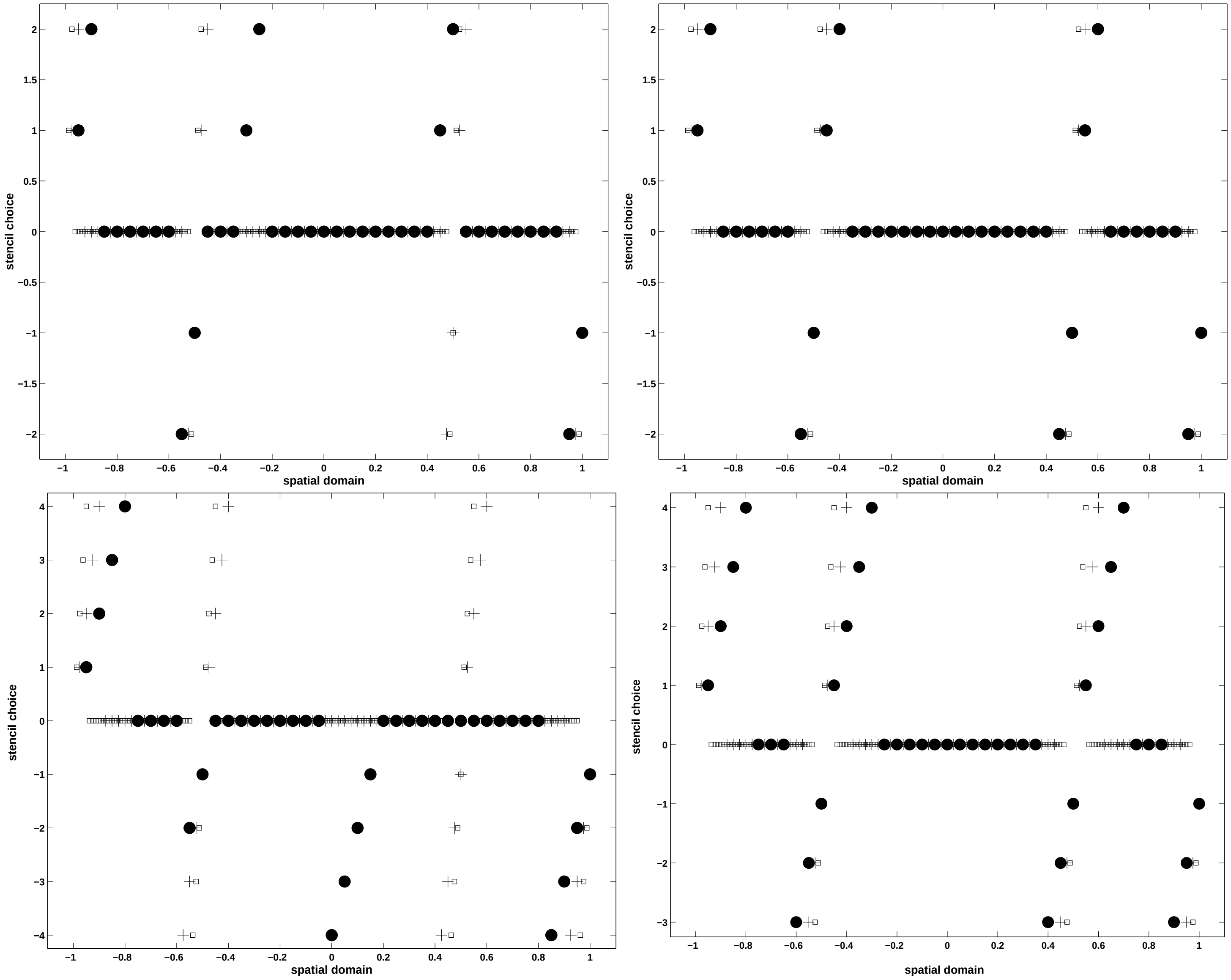
stencil choice
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
spatial domain
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
stencil choice
4
3
2
1
0
−1
−2
−3
−4
spatial domain

Figure 5.3:

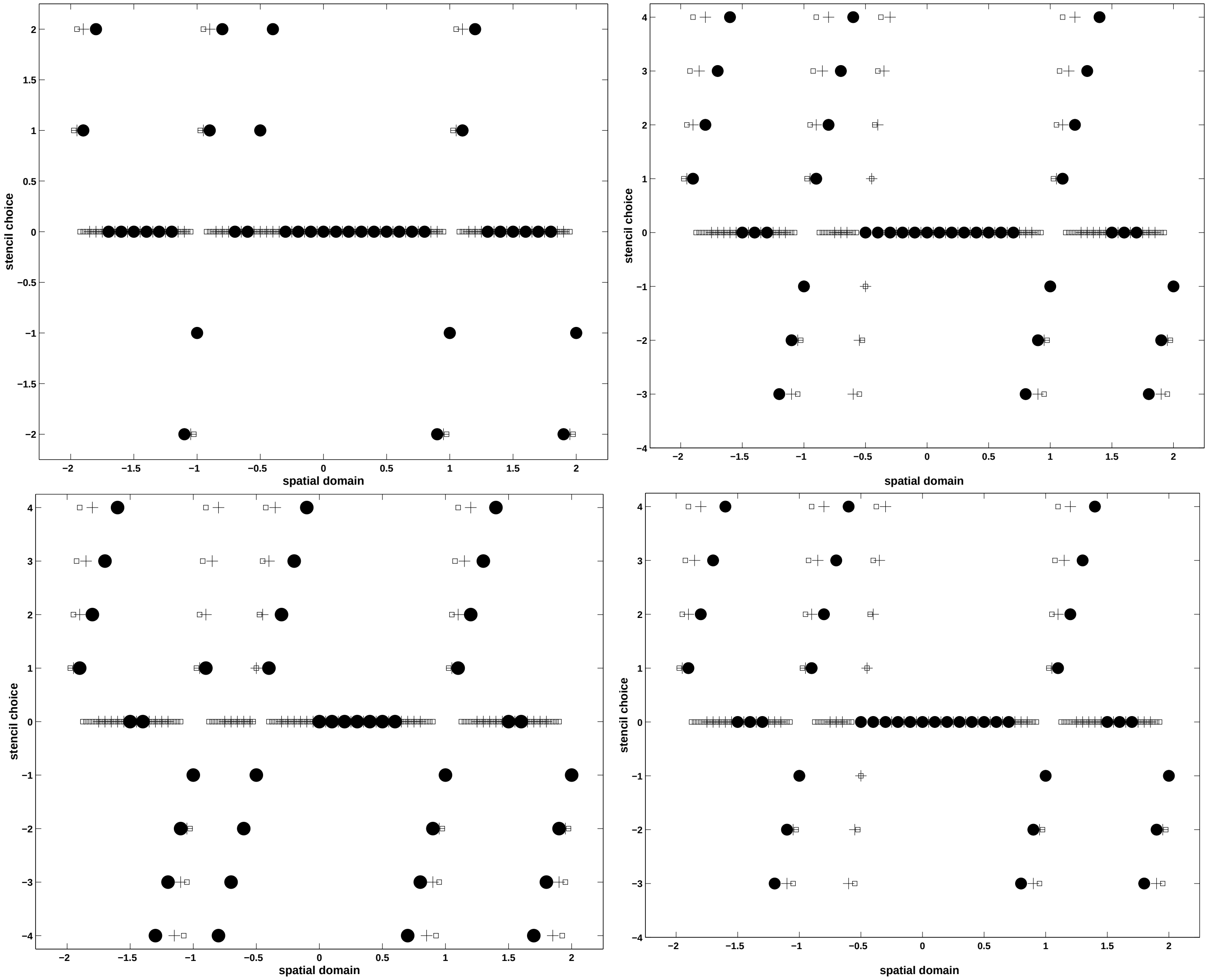

Figure 5.4:

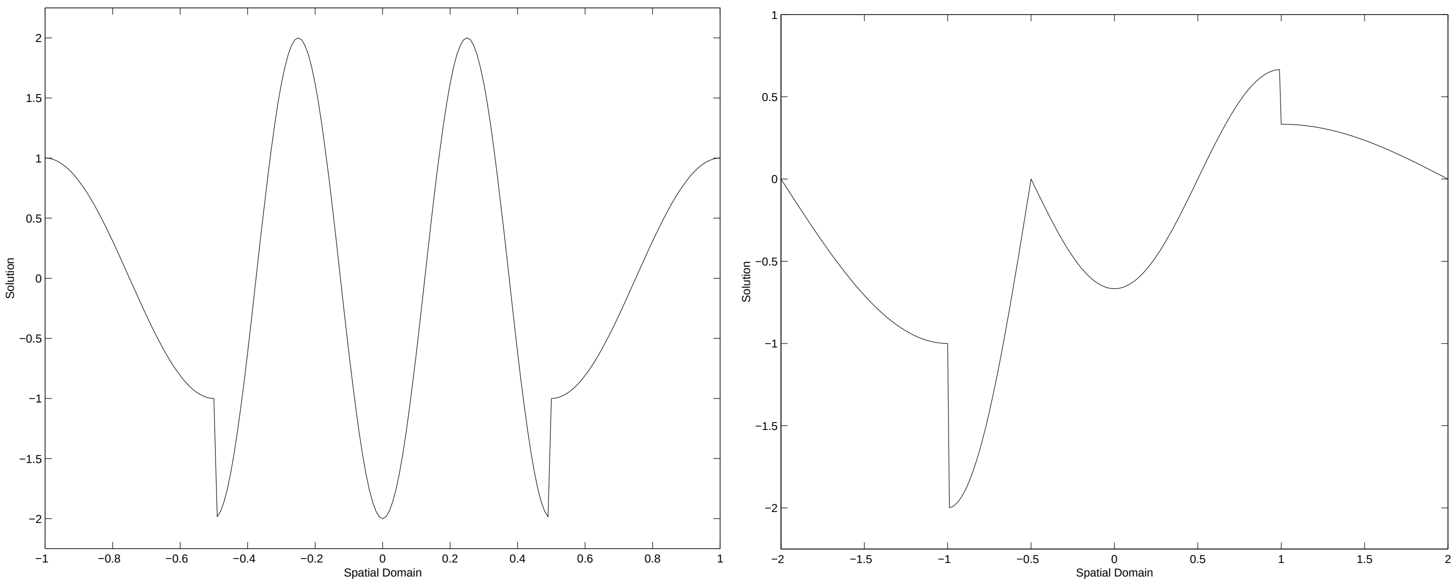
Solution
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Spatial Domain
Solution
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
Spatial Domain

Figure 5.5:

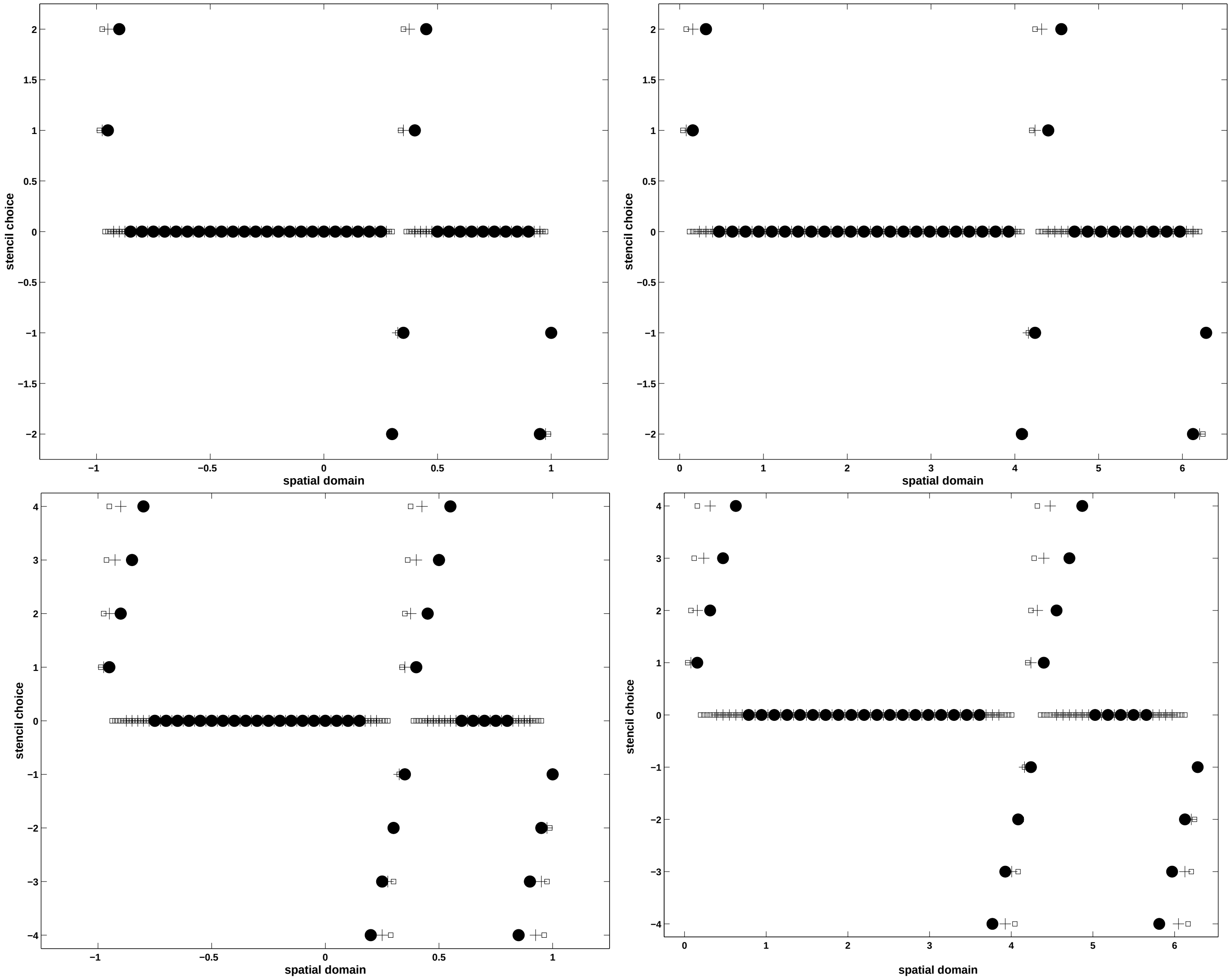
stencil choice
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−1
−0.5
0
0.5
1
spatial domain
stencil choice
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
1
2
3
4
5
6
spatial domain
stencil choice
4
3
2
1
0
−1
−2
−3
−4
−1
−0.5
0
0.5
1
spatial domain
stencil choice
4
3
2
1
0
−1
−2
−3
−4
0
1
2
3
4
5
6
spatial domain

Figure 5.6: