An Implicit Space-Time Spectral Difference Method for Discontinuity Capturing Using Adaptive Polynomials

P. G. Huang*
Department of Mechanical Engineering, University of Kentucky, Lexington, KY 40506.

Z. J. Wang†
Department of Aerospace Engineering, Iowa State University, Ames, IA 50011.

Y. Liu‡
NASA Ames Research Center, Moffett Field, CA 94035.

A new implicit high-order space-time spectral difference method for capturing discontinuities is developed. The proposed method has the following new improvements over the conventional spectral difference method: (1) The method treats time and space in the same fashion and hence allows conservation laws to be satisfied in the coupled space and time coordinates. (2) The method is fully implicit and can be any order of accuracy and therefore it allows a much larger time step to be taken without losing accuracy. (3) The method can be implemented to obtain the solution by using adaptive polynomial refinements. An approach employing a self-adaptive polynomial refinement will be discussed. (4) A new approach for discontinuity-capturing will be discussed. This approach does not require a limiter and unlike the conventional shock capturing schemes, the proposed method resolves discontinuities by polynomial refinements.

I. Introduction

An efficient high-order, conservative method named the spectral difference (SD) method has been recently developed by Liu, Vinokur and Wang[1,2] for conservation laws on unstructured grids. The SD method can be viewed to be the extension of the multi-domain spectral (MDS) method[3,4] to unstructured grids. In the present study, the SD or the MDS method is further extended to uniformly treat the space and time coordinates. The primary motivation for developing another numerical method is to seek a simpler to implement and more efficient method than the current state of the art - the discontinuous Galerkin (DG) method[5-7], and the spectral volume (SV) method[8-12], to name just a few high-order methods for conservation laws on unstructured grids.

In order to improve the accuracy of numerical methods, mesh size refinement (h-refinement) is probably the most common way. This is generally done by refining the mesh using a fixed-order (usually low-order) piecewise polynomial interpolation over the mesh. On the other hand, polynomial refinement (p-refinement) is known to achieve a faster rate of convergence to an accurate solution with a fixed mesh size[13-15]. The high order schemes generally give rise to numerical instability and spurious numerical oscillations, which may result from either interpolation errors caused by constructing a polynomials of high degree on a uniform set of nodes (Runge phenomena[16]) or the interpolation errors caused by using a continuously differentiable function to approximate a jump discontinuity (Gibbs phenomena[13]).

There are two types of polynomial refinements: the finite difference methods[17] keep the number of unknowns the same but extend the stencil size to neighboring cells in order to construct high order polynomials and the finite element methods[14-15] enable high order construction of the polynomial within a cell by increasing the number of unknowns in the cell or element. When performing high-order interpolations, we should be careful to avoid the occurrence of Runge phenomenon. If the positions of the unknowns within a cell in the finite element refinement can be predefined to satisfy the properties of orthogonal polynomials and quadratures, the Runge phenomenon can

* Professor and Director of Graduate Studies of Mechanical Engineering, 151 RGAN Bld., ghuang@engr.uky.edu, Senior member of AIAA
† Associate Professor of Aerospace Engineering, 2271 Howe Hall, zjw@iastate.edu, Associate Fellow of AIAA
‡ Research Scientist, Yen.Liu@nasa.gov, Mail Stop T27B-1, AIAA Member.

be avoided. To illustrate this, Table 1 lists a comparison of the two methods in approximating the function $f(x) = 1/\left[2 + \cos(5x)\right]$, for $0 \le x \le \pi$: one with uniform mesh and the other with Gauss-Lobbato mesh. As can be seen from the table, a polynomial based on the Gauss-Lobbato mesh is capable of generating an oscillation free solution for a polynomial up to the order of more than 500 (where the error has dropped below machine zero) while the polynomials generated using a uniform mesh give rise to the Runge phenomenon when the order of polynomials is increased. It is for this reason that we shall use the finite element polynomial refinement as the basis of the polynomial refinement in the rest of the paper.

k (number of unknowns)	Uniform Mesh	Gauss-Lobbato Mesh
2	9.32E-02	9.32E-02
4	9.57E-02	5.10E-02
8	5.32E-02	1.69E-02
16	7.63E+00	1.31E-03
32	2.89E+05	3.93E-06
64	6.73E+16	7.45E-11
128	5.09E+39	2.74E-20
256	5.01E+85	3.03E-39
512	9.29E+228	1.94E-65

Table 1. The comparison of the polynomial refinement of $f(x) = 1/\left[2 + \cos(5x)\right]$, where $0 \le x \le \pi$, using uniform and Gauss-Lobbato's meshes. The order of the polynomial is k-1 and the resultant polynomial is used to extract the values of *f*'s at the Legendre points. The *L2* normal is calculated by integrating the difference of the values obtained by the polynomial and the exact functional values: $L2 = \int_o^{\pi} \left(u_{calculated} - u_{exact}\right)^2 dx/\pi$, using the Legendre integration scheme.

A recent development in the CFD community is to make use of adaptive mesh refinement in regions where large gradients of dependent variables are encountered. However, this method not only requires a complicated grid generation strategy to refine the mesh, but the approach sometimes requires a very fine mesh and small time step in regions of high gradients to enable a feasible solution to be obtained. Similarly, one can anticipate an adaptive finite element polynomial refinement under a fixed mesh to provide an accurate solution with a minimum number of unknowns. The adaptive polynomial not only is efficient in generating an accurate solution with relatively few unknowns, but also does not require the regeneration of the base mesh during the refinement process. Hence, in this paper an approach for the adaptive polynomial refinement will be developed.

One major issue in the development of adaptive polynomial refinement is the occurrence of spurious solutions in regions of large gradients or discontinuity, or the Gibbs phenomena. Traditionally, limiters, such as TVD, TVB and FCT, are applied to restrict the maximums and minimums to be bounded. While the limiter approach has proven to be very effective in reducing the Gibbs phenomena, it comes with a heavy price – the method reduces to a low order scheme in regions of discontinuity and it applies indistinguishably to a physical oscillation as well as a spurious solution. As a result, for an adaptive polynomial refinement to be successful, the concept of limiters must be abandoned. The paper discusses a new adaptive method that will allow an accurate solution to be obtained with successive polynomial refinements even in regions of discontinuity.

Finally, it should be noted that the current method is formulated in an implicit manner – both in space and time. In other words, the treatment for the time terms is formulated in exactly the same manner as that for the space terms. The treatment enables a fully coupled conservative solution in space and time simultaneously.

II. Basic formulation

For simplicity, the following derivation is performed in one dimension. Since the polynomial interpolation can be expanded in each direction independently, the extension of the method to multi-dimension is straightforward.

Consider the solution of the 1-D conservation law as follows:

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0 \tag{1a}$$

on domain Ω with the following initial condition:

$$u(x,0) = u_o(x) \tag{1b}$$

and appropriate boundary conditions on $\partial\Omega$ in (1). In (1), x is the space coordinate and $x \in \Omega$, t denotes time and $t \in [0,T]$, u is a state variable, and F is flux in the x direction. Domain Ω is discretized into n non-overlapping cells called spectral element, which are divided by the dashed lines as shown in Figure 1. Here, we depict the value of u with two subscripts, $u_{i,j}$, where i indicates the nodal point index in the time direction, $i = 1,2,\cdots,\ell-1,\ell$; and j denotes the nodal point index in the space direction, $j = 1,2,\cdots,k-1,k$. If the polynomial of approximation is of the order *k-1* (or $\ell-1$), the degree of freedom of the polynomial is k (or ℓ) and hence we need to provide k (or ℓ) independent data to determine the polynomial uniquely. One can choose k distinct points in space (including two end points), as shown in Figure 1. The points are denoted as the nodal points, located at x_j, $j = 1,\cdots,k$. The cardinal function for the algebraic interpolation is:

$$M_j(x) = \prod_{m=1,m\neq j}^{k} (x - x_m) \Big/ \prod_{m=1,m\neq j}^{k} (x_j - x_m) \tag{2a}$$

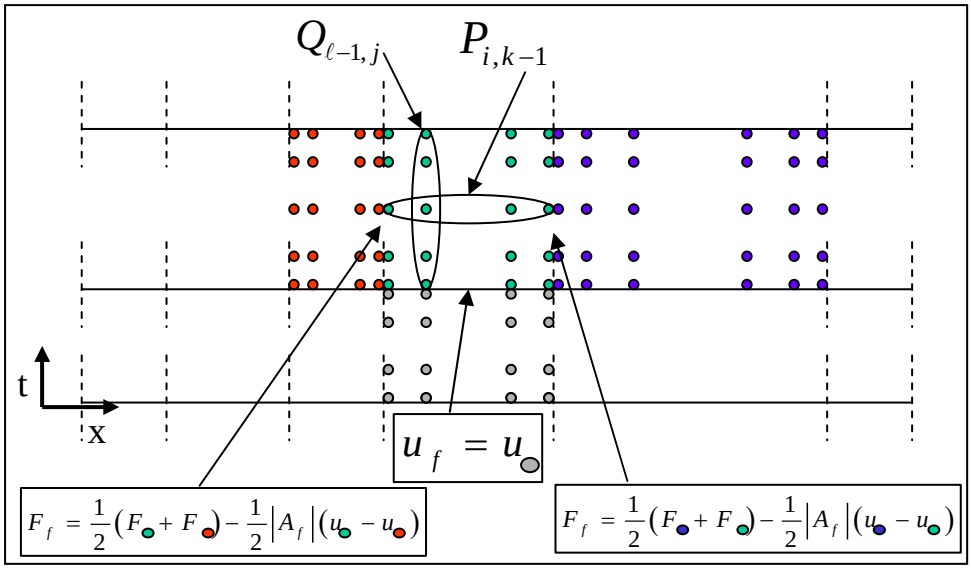

Figure 1. The implicit spectral difference methods.

where $M_j(x_m) = \delta_{jm}$. For given values of fluxes at a fixed time level, i, $F_{i,1}, F_{i,2}, \cdots, F_{i,k}$ at $x_1, x_2, \cdots, x_k$, the polynomial at the fixed time level, i, becomes:

$$P_{i,k-1}(x) = \sum_{j=1}^{k} M_j(x) F_{i,j} \tag{2b}$$

To avoid the Runge phenomenon, we select Gauss-Lobbato points as the nodal points.

The evaluations of $F_{i,j}$ for $j = 2,\cdots,k-1$, is straightforward since the values of interior $u_{i,j}$'s are uniquely defined. At the faces of two adjacent cell elements in which $j = 1$ and k, F contains two values, one from the left, F_L, and the other from the right, F_R. At the interfaces, we must replace the flux F by a Riemann flux. We suggest using the Roe's approximate Riemann solver[18] to uniquely define the interfacial fluxes,

$$F_f = \frac{1}{2}\left(F_R + F_L\right) - \frac{1}{2}\left|A_f\right|\left(u_R - u_L\right) \tag{3}$$

where $A_f = (\partial F / \partial u)_f$ is the Jacobian matrix at the interface, *f*. Once we establish the *(k-1)*th-order polynomial to describe the distribution of *F* in terms of *x* for a given time, t_i, the flux gradient at the nodal point *p* and the given t_i, $(\partial F / \partial x)_{i,p}$, where $p = 1, \cdots, k$, can be approximated by

$$\begin{aligned}\left(\frac{\partial F}{\partial x}\right)_{i,p} &= \left(\frac{\partial P_{i,k-1}}{\partial x}\right)_{i,p} = \sum_{j=1}^{k}\left\{\left(\frac{\partial M_j}{\partial x}\right)_p F_{i,j}\right\} \\ &= \sum_{j=1}^{k}\left\{\frac{\sum_{q=1,q\neq j}^{k}\left[\prod_{m=1,m\neq j,m\neq q}^{k}(x_p - x_m)\right]}{\prod_{m=1,m\neq j}^{k}(x_j - x_m)} F_{i,j}\right\}.\end{aligned} \tag{4}$$

The time gradient term of equation (1a) can be viewed as a time derivative of a flux, *u*, with a Jacobian of unity. Similar to the treatment in space, for a given time step, ΔT, one can construct a $(\ell - 1)$ th-order polynomial by subdividing the time step into ℓ nodal points, located at t_i, $i = 1, \cdots, \ell$, shown in Figure 1. The cardinal function for the algebraic interpolation is:

$$N_i(t) = \prod_{m=1,m\neq i}^{\ell}(t - t_m) \Big/ \prod_{m=1,m\neq i}^{\ell}(t_i - t_m) \tag{5a}$$

where $N_i(t_m) = \delta_{im}$. For given values of u at a fixed space level, i, $u_{1,j}, u_{2,j}, \cdots, u_{\ell,j}$ at $t_1, t_2, \cdots, t_\ell$, the polynomial at the fixed space level, j, becomes:

$$Q_{\ell-1,j}(t) = \sum_{i=1}^{\ell} N_i(t) u_{i,j} \tag{5b}$$

Again we choose to use Gauss-Lobbato points as the nodal points in time to avoid Runge phenomena.

The values of $u_{i,j}$ for $i = 2, \cdots, \ell - 1$ are uniquely defined in the interior points. At the faces of two adjacent time-step elements, *u* contains two values, one from the ℓ nodal point of the previous time step and the other from the 1st nodal point of the current time step. Since the time flux is treated as a flux with a positive eigenvalue (of unity), *u* at the time interface is taken from the ℓ nodal point of the previous time step, as shown in Figure 1. Similar to the treatment of the space term, the time gradient term in term of t_p at a given position x_j, $(\partial u / \partial t)_{p,j}$, where $p = 1, \cdots, \ell$, can be approximated by

$$\begin{aligned}\left(\frac{\partial u}{\partial t}\right)_{p,j} &= \left(\frac{\partial Q_{\ell-1,j}}{\partial t}\right)_p = \sum_{i=1}^{\ell}\left\{\left(\frac{\partial N_i}{\partial t}\right)_p u_{i,j}\right\} \\ &= \sum_{i=1}^{\ell}\left\{\frac{\sum_{q=1,q\neq i}^{\ell}\left[\prod_{m=1,m\neq i,m\neq q}^{\ell}(t_p - t_m)\right]}{\prod_{m=1,m\neq i}^{\ell}(t_i - t_m)} u_{i,j}\right\}.\end{aligned} \tag{6}$$

It should be noted that proposed method treats the time derivative in an exactly the same manner as space derivatives. This enables a strong coupling of the state variables in both space and time domains.

III. Implicit Time Integration

In the implicit treatment, the governing equation is solved by a successive iterative method by reformulating equation (1a) in a Δ-form:

$$\frac{\partial \Delta u}{\partial t} + \frac{\partial A \Delta u}{\partial x} = -\frac{\partial u^*}{\partial t} - \frac{\partial F(u^*)}{\partial x} \tag{7}$$

where u^* is the intermediate value during iteration, Δu is the correction to be made to u^* during iterations and the RHS is evaluated explicitly using equations (4) and (6). It should be noted that the RHS is the real physics and the LHS provides a numerical operator serving to drive the RHS to zero. Hence, we use the first order method for the LHS:

$$\begin{aligned}
\left(\frac{\partial \Delta u}{\partial t}\right)_{i,P} &\approx \frac{1}{t_i - t_{i-1}} \Delta u_{i,P}^n - \frac{1}{t_i - t_{i-1}} \Delta u_{i,P}^{n-1} \\
\left(\frac{\partial A \Delta u}{\partial x}\right)_{i,P} &\approx \frac{1}{2}\left(A_{i,P} + \left|A_{i,P}\right|\right) \frac{\Delta u_{i,P}^n - \Delta u_{i,W}^n}{x_P - x_W} + \frac{1}{2}\left(A_{i,P} - \left|A_{i,P}\right|\right) \frac{\Delta u_{i,E}^n - \Delta u_{i,P}^n}{x_E - x_P}
\end{aligned} \tag{8}$$

where E, P and W are the three adjacent nodal points in space; i and $i-1$ are the two adjacent time levels; and n and $n-1$ are the two adjacent pseudo-time levels. Equation (7) can thus be written in the following algebraic form:

$$a_{i,P} \Delta u_{i,P}^n = a_{i,E} \Delta u_{i,E}^n + a_{i,W} \Delta u_{i,W}^n + a_{i,T} \Delta u_{i,P}^{n-1} + RHS_{i,P}^* \tag{9a}$$

where

$$\begin{aligned}
a_{i,E} &= -\frac{1}{2}\left(A_{i,P} - \left|A_{i,P}\right|\right) \frac{1}{x_E - x_P} \\
a_{i,W} &= \frac{1}{2}\left(A_{i,P} + \left|A_{i,P}\right|\right) \frac{1}{x_P - x_W} \\
a_{i,T} &= \frac{1}{t_i - t_{i-1}} \\
a_{i,P} &= \sum_{nb=E,W \text{ and } T} a_{i,nb} \\
RHS_{i,P}^* &= -\left(\frac{\partial u^*}{\partial t}\right)_{i,P} - \left[\frac{\partial F(u^*)}{\partial x}\right]_{i,P}
\end{aligned} \tag{9b}$$

Equations (9a) and (9b) ensure that the coefficients are positive and also diagonally dominant and hence they are unconditional stable to solve iteratively. The solution of Δu is used to update the value of u^* and iterations are performed until the RHS approaches to the machine zero.

To demonstrate the usefulness of the current implicit method, the method is applied to solve the following scalar equation:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \tag{10}$$

with a sine wave initial condition

$$u(x,0) = \sin(\pi x) \tag{11}$$

where $-1 \le x \le 1$ and a periodic boundary condition is applied on both ends of the domain. The calculations were performed with $\Delta t = 0.01$ and $\ell = 11$, implying the 10^{th} order Gauss-Lobbato polynomial was placed along the time direction. A series of calculations were performed to vary the number of element cells, *n*, and the number of Gauss-Lobbato points within each element cell, *k*. The total time for the solution to advance is $T = 2$ and hence at the end of the calculation, the *u*–profile completes a full cycle and returns to its initial location. The L2 error norm is defined by

$$L2 = \int_{-1}^{1} \left(u_c - u_e\right)^2 dx / 2 \approx \sum_{i=1}^{k} w_i \left(u_{c,i} - u_{e,i}\right)^2 \ /2 \tag{12}$$

where u_c and u_e are the calculated and exact values, respectively. The integration was performed by the Gauss-Lobatto integration scheme with predefined weights for integration, w_i.

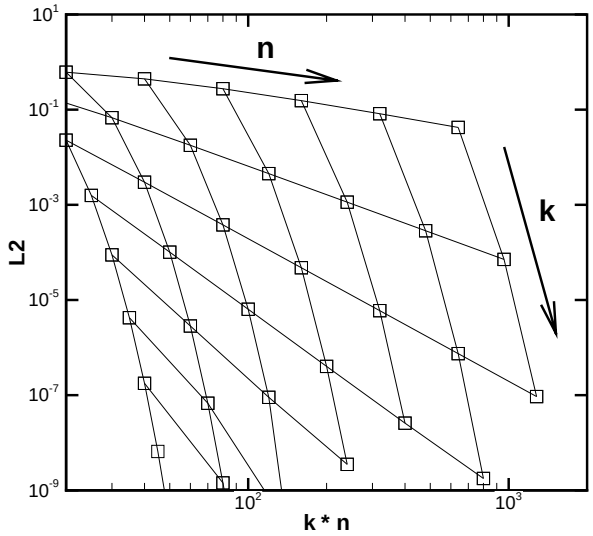

Figure 2.The prediction of a moving sine wave for the cell element number of *n* = 5, 10, 20, 40, 80, 160 and 320; and the Gauss-Lobbato points per element of *k* = 2, 3, 4, 5, 6, 7 and 8.

Figure 2 shows the L2 norm errors for a range of *n* (number of cell elements) and *k* (Gauss-Lobbato points per cell element) combinations. The vertical lines connect the predicted errors as *k* is varied for a given, fixed *n*. The horizontal lines connect the errors as *n* is varied and *k* is fixed. The *n* values tested are *n* = 5, 10, 20, 40, 80, 160 and 320 and the *k* value are *k* = 2, 3, 4, 5, 6, 7 and 8. The horizontal axis is the total number of nodal points used in the calculations, $n \times k$. Figure 2 demonstrates that the current implicit method can cover a range of *n* and *k* combinations and it also shows that the polynomial refinement is a much more effective way to achieve an accurate solution than the mesh refinement.

The implementation of the polynomial refinement does not require regenerating new cell elements and it is a much more effective way to achieve an accurate solution. This combination makes the polynomial refinement a better candidate for solution adaptive refinement than mesh refinement.

IV. Adaptive Polynomial Refinement

To perform an adaptive polynomial refinement, one has to a start with a known initial profile of *u*, which can be the solution at the previous time step or can be a solution obtained by a fixed set of *n* and *k* (generally low order) if steady state cases are involved. The initial values of *u* can be used to obtain the 2^{nd} derivative, which is used as an indicator for solution refinements. For a 1-D case, as discussed in section 2, the second derivative of *u* at x_p, where $p = 1, \cdots, k$, can be obtained by:

$$\left(\frac{\partial u}{\partial x}\right)_{i,r} = \sum_{j=1}^{k}\left\{\frac{\sum_{q=1,q\neq j}^{k}\left[\prod_{m=1,m\neq j,m\neq q}^{k}(x_r - x_m)\right]}{\prod_{m=1,m\neq j}^{k}(x_j - x_m)}u_{i,j}\right\},\ \text{for r} = 1,\cdots,k,$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{i,p} = \sum_{j=1}^{k}\left\{\frac{\sum_{q=1,q\neq j}^{k}\left[\prod_{m=1,m\neq j,m\neq q}^{k}(x_p - x_i)\right]}{\prod_{m=1,m\neq j}^{k}(x_j - x_i)}\left(\frac{\partial u}{\partial x}\right)_{i,j}\right\}. \tag{13}$$

Based on the given profile, the order of polynomial in each element can be successively refined until the order of polynomial, *k*, in the respective element satisfies the following relationship:

$$\left|1 - \frac{D_{k+2}}{D_k}\right| < \varepsilon \tag{14a}$$

where

$$D_k = \int_0^{\Delta x}\left|\frac{\partial^2 u}{\partial x^2}\right| dx \approx \sum_{j=1}^{k}\left|\frac{\partial^2 u}{\partial x^2}\right|_j w_j\ , \tag{14b}$$

the value of ε is chosen 0.001 and Δx is the size of the cell element. It should be noted that the increment of *k* is 2 in equation (14a) rather than 1 in order to allow a monotonic variation of *D* during the search. Once the values of *k* in each element are defined from the given profile, we can advance the solution to the next time step, $t = \Delta T$. With the new solution at $t = \Delta T$ being obtained, a search for a new set of *k*'s is repeated until the new solution equation satisfies equation (14). A correction of *k* is then made in each cell element to select the larger number of the two *k*'s (one at $t = 0$ and the other at $t = \Delta T$) and the solution at the $t = \Delta T$ is recalculated. This process is iterated for a maximum of 5 times. The values of *k*'s in each element and the solution at $t = \Delta T$ are then defined.

In figure 3, equation (10) is solved using the adaptive polynomial technique described above for three variants of initial conditions: (1) sine wave: $\sin(\pi x)$, (2) inverse cosine wave: $1/\left[2+\cos(5\pi x)\right]$ and (3) composite wave: $\sin(5\pi x)+\sin(4\pi x)$. There are 20 cell elements, the time step is fixed at 0.01, and the Gauss Lobbato points in the time direction are fixed at $\ell = 8$. Figures 3(a) and 3 (b) are the sine-wave solutions at $t = 1$ and $t = 2$, respectively; figures 3(c) and 3(b) are the inverse-cosine-wave solutions at $t = 1$ and $t = 2$, respectively; figures 3(e) and 3(f) are the composite wave solutions at $t = 1$ and $t = 2$, respectively. In these figures, the nodal points within the same cell element are depicted by the same colors; the solid line is the exact solution and the yellow bars indicate the final number of Gauss-Lobbato points in each element, *k*. As can be seen from the figures, the adaptive scheme seems to follow the shape gradients very well leading to the L2 error norm of the solutions after one revolution to be less than 10^{-7}.

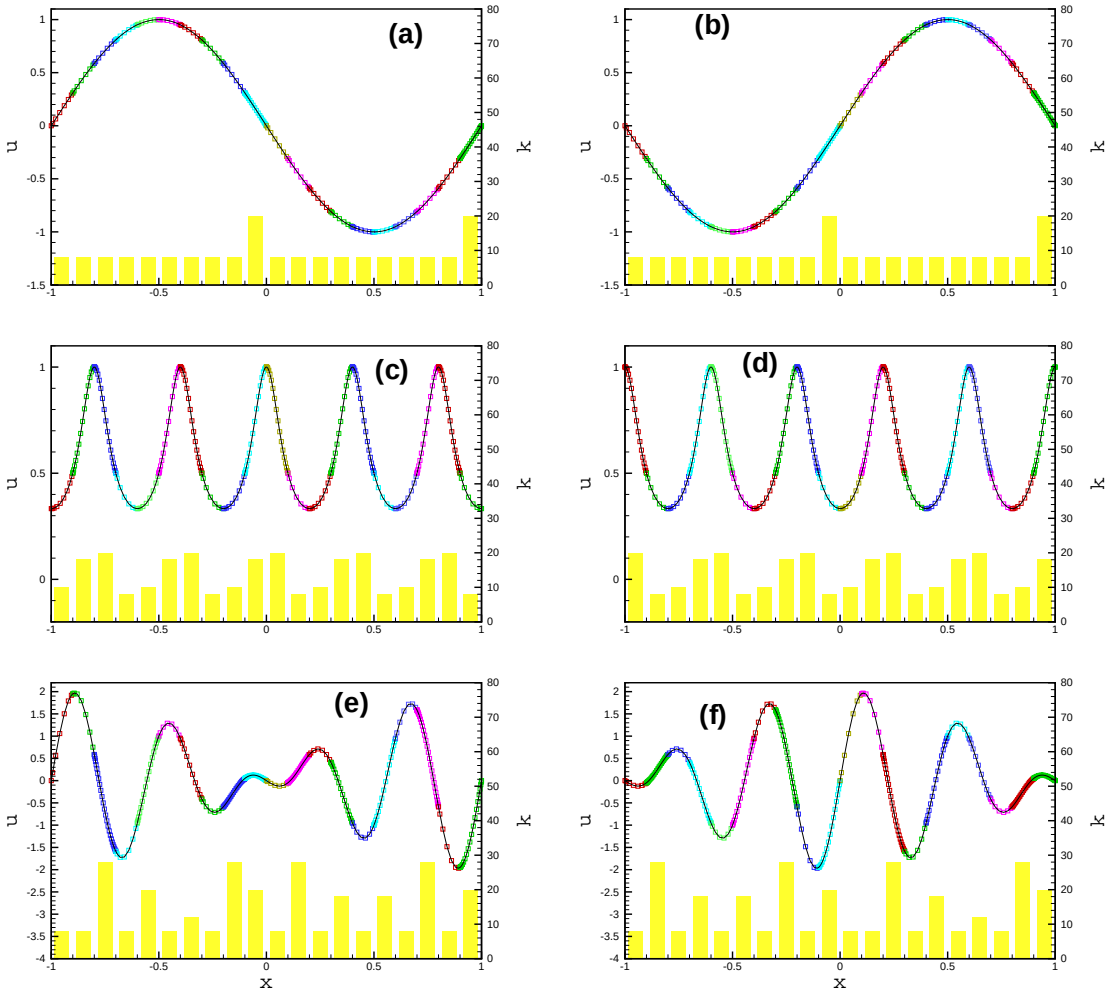

Figure 3. Adaptive polynomial refinement for (a) and (b) being a sine wave: $\sin(\pi x)$ **; (c) and (d) being a inverse cosine wave:** $1/\left[2+\cos(5\pi x)\right]$; **and (e) and (f) being a composite wave:** $\sin(5\pi x)+\sin(4\pi x)$ **. Figures (a), (c) and (e) are for** $t=1$; **(b),(d) and (f) are for** $t=2$. **Solid lines are exact solutions; colored circles (points within the same cell element are depicted with the same color) are calculations; and yellow bars are the number of nodal points in each cell elements.**

V. Discontinuity Capturing

The Gibbs Phenomenon occurs when high-order schemes are used to approximate a discontinuity. There are a number of approaches to eliminate Gibbs Phenomena in the presence of discontinuity. Most approaches rely on a judgment that the local solution must be bounded by the maximum or minimum in its neighboring vicinity. For example, if any nodal point within a cell element has a local extreme, one may choose to align the value of all nodal points within the element into a straight line staggered between the values of two boundary points, W and P, see figure 4.

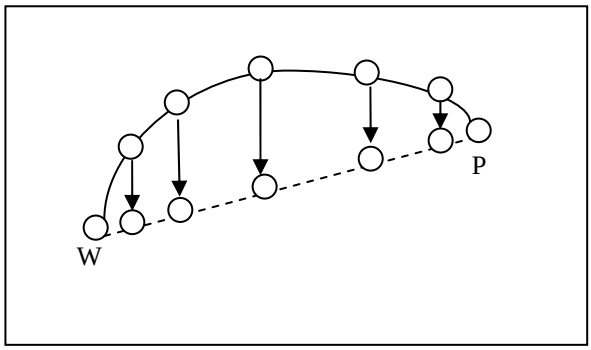

Figure 4. The limiter approach for fixing the Gibbs phenomena.

The limiter treatment has two drawbacks. First, it can not distinguish whether a local maximum (or minimum) is caused by Gibbs phenomena or is a part of the real physics. As a result, a clipping of the peak values can occur even in the simulations of a smooth wave. Second, the linear approximation within the element reduces the overall order of accuracy to the 2nd order and this makes the polynomial refinement difficult to achieve.

To abandon the use of the limiter treatment, we must first view the above mentioned linear-alignment limiter to be equivalent to the solution of the convection-diffusion equation with a conductivity K approaching infinity:

$$\frac{\partial u}{\partial t}+\frac{\partial F(u)}{\partial x}=K\frac{\partial^2 u}{\partial x^2}. \tag{15}$$

Because equation (15) with large K leads to a solution with a linear profile of u bounded by two imposed cell boundary values, the linear alignment limiter can be achieved implicitly. To demonstrate that the solution of (15) with a infinitely large value of K within the element is independent of the order of the scheme used and that the solution is exactly identical to that of equation (1a) solved by using the 1st order polynomial ($k=2$), we have repeated the sine wave problem with two different approaches – one with 8 Gauss-Lobbato points ($k=8$) and the other with the 1st order polynomial ($k=2$) within all cell elements. For the high order scheme, the artificial diffusive treatment is used in the interior points of all cell elements, while no diffusion is applied to the cell face points (the P and W points) in order to ensure that the conservation law of the elements is satisfied. Figure 5 shows the comparison of the two approaches after one complete cycle ($t=2$) and it can be seen that the two approaches give exactly identical solution. Obviously, the solution is not accurate because when K approaches infinity the truncation errors reduce to the 2nd order, irrespective of the number of Gauss-Lobbato points being used in the elements.

Now that we view the limiter treatment as a subset of convection-diffusion equation, we can increase the order of accuracy by reducing the value of the conductivity used in the simulation. If we define:

$$K=c\left|\lambda\right|_{\max}\Delta x_{\max} \tag{16}$$

where $\left|\lambda\right|_{\max}$ is the maximum of the absolute values of eigenvalues within the element (for a scalar problem, the eigenvalue is the convective velocity, V); $\Delta x_{\max}$ is the maximum value of the distances between the two adjacent nodal points within the cell element; c is the reciprocal of the Peclet number: $Pe=V\Delta x/K$. Since a Peclet number being less than 2 is a sufficient condition for a central differencing scheme to satisfy TVD properties[19] (it guarantees the coefficients of the discretized equations being positive), we have chosen $c=0.5$ (or $Pe=2$) in the current paper. Although the high order schemes cannot be proven to satisfy TVD properties when $Pe=2$, the numerical experiments presented in this paper did not show any spurious oscillation.

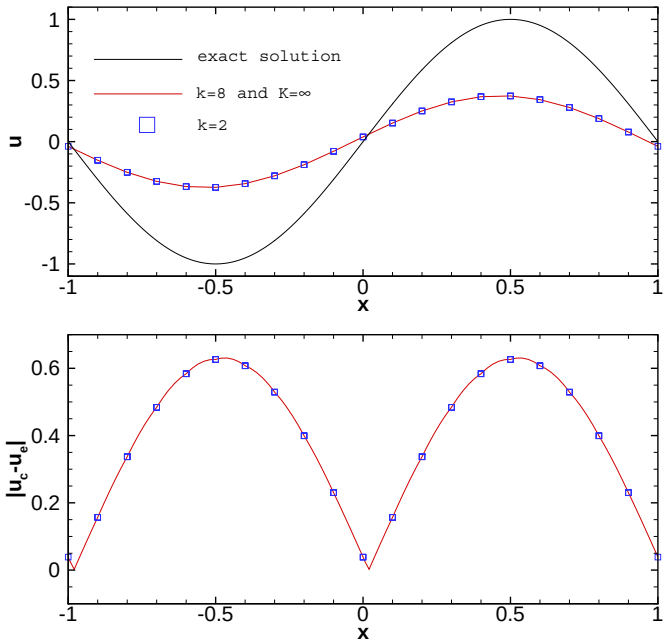

Figure 5. Compare the solutions of moving sine wave obtained by the 1st order polynomial ($k = 2$) and the 7th order polynomial ($k = 8$) with a dominant diffusion inside the cell elements, $K \to \infty$. The solutions are obtained with $\Delta t = 0.01$ and 11 Gauss-Lobbato points in the time direction. The results are presented for the solutions with one complete cycle ($t = 2$). It can be seen that the 7th order polynomial with $K \to \infty$ (applied to only the interior points) gives rise to exactly the same solution as that obtained by the 1st order polynomial without diffusion.

It should be noted that the diffusive treatment is used only in the interior points (excluding P and W points) and the conductivity, *K*, within the element must have the same value so that in the extreme cases when diffusion is dominant, a linear profile of *u* can be recovered. By reducing the value of *K*, we can now allow the solution to have maximum and minimum inside the cell element.

Figure 6 shows the solutions obtained after one complete cycle ($t = 2$) of a scalar equation, (10), with the following initial condition:

$$u(x,0) = \begin{cases} 1/\left[(x-1)^2 - 1.8(x-1) + 0.85\right], & x \geq 0 \\ 1/\left[(x+1)^2 - 1.8(x+1) + 0.85\right], & x<0 \end{cases} \tag{17}$$

The time step used is $\Delta t = 0.001$ and the Gauss-Lobbato points employed in the time direction is $\ell = 5$.

While figure 6(a) shows that the extent of Gibbs phenomena has propagated to the downstream of the discontinuity ($x \approx 0.8$) after one cycle rotation, the parabolic shape before the discontinuity is well preserved. Figure 6(b) shows that the linear-alignment limiter, although is capable of eliminating Gibbs phenomena, leads to too much numerical dissipation near discontinuity and also results in a clipping of the parabolic profile shape before the discontinuity. Figure 6(c) is the results obtained by using $c = 0.5$ in equation (16) but the diffusion is still added to cell elements having local extreme. As can be seen from the figure, although the reduction of *K* can give a sharper resolution of the discontinuity, the physical parabolic shape before the discontinuity is still clipped. To eliminate the unnecessary peak clipping, we have chosen to use the change of first order gradients as a criterion to add the diffusion term, shown in Figure 6(d):

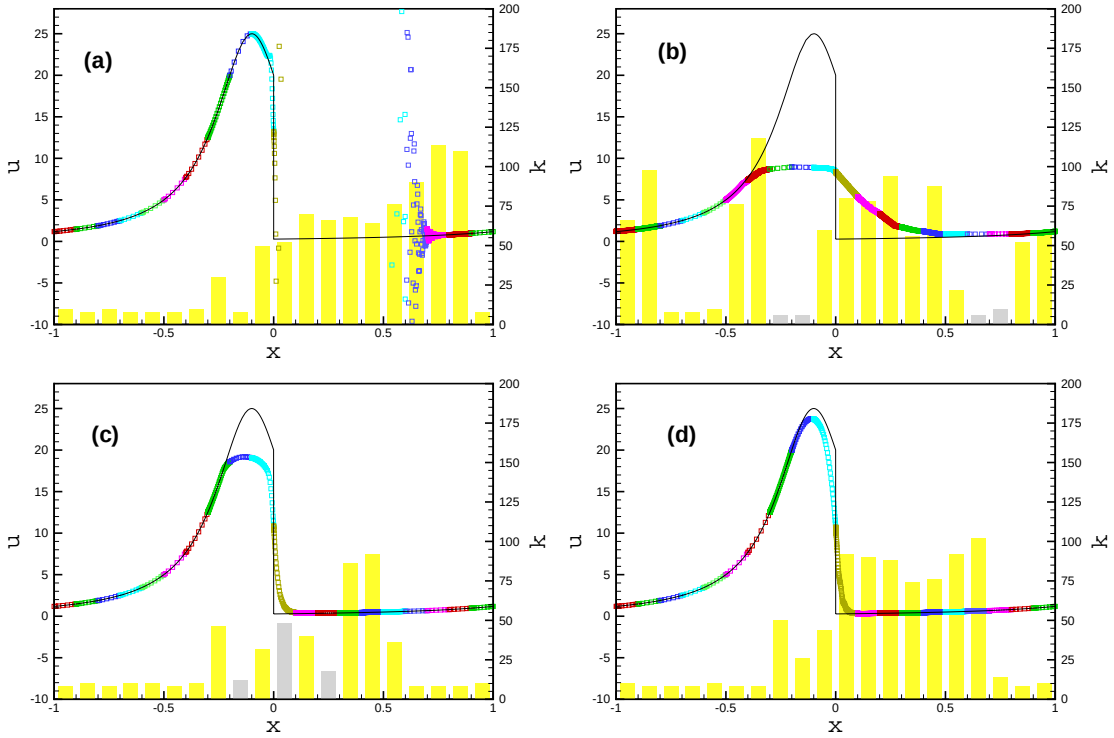

Figure 6. The solution of equation (10) in the presence of discontinuity: with initial condition (17). (a) no limiter: Gibbs phenomena is propagated to the downstream of the discontinuity; (b) the limiter is applied to the cells having local extreme and $K \to \infty$; (c) the limiter is applied to the cells with local extreme and $c = 0.5$; (d) the limiter is applied to the cells with the ratio of 1st order gradient of the two nodal points larger than 20, equation (18), and $c = 0.5$. The solid line is the exact solution; symbols are the results of the calculations (points depicted with the same color share the same polynomial); and bars are the number of Gauss-Lobbato points used in a cell element (bars with grey color are elements with diffusion being activated).

$$\left\{ \max\left[\left(\frac{\partial u}{\partial x}\right)_e, \left(\frac{\partial u}{\partial x}\right)_w \right] - \min\left[\left(\frac{\partial u}{\partial x}\right)_e, \left(\frac{\partial u}{\partial x}\right)_w \right] \right\} \Big/ \min\left[\left|\frac{\partial u}{\partial x}\right|_e, \left|\frac{\partial u}{\partial x}\right|_w \right] > 20 \tag{18a}$$

where

$$\left(\frac{\partial u}{\partial x}\right)_e = \frac{u_{j+1} - u_j}{x_{j+1} - x_j} \text{ and } \left(\frac{\partial u}{\partial x}\right)_w = \frac{u_j - u_{j-1}}{x_j - x_{j-1}}. \tag{18b}$$

If any of two adjacent nodal interfaces within a cell element, *e* and *w*, satisfies equation (18a), the diffusion is added to all the interior nodal points in the corresponding cell element. Hence, equation (18) enables the diffusion to be applied only when there is a rapid change of the profile within the element, monitored by the change of the first order gradients. Since the local extreme is no longer used as a criterion to apply the diffusion, the clipping of the parabolic shape before the discontinuity is reduced. Unfortunately, although the result is improved, the general profile shape is still somewhat blurry.

The above mentioned treatments can eliminate Gibbs phenomena, but they can not reproduce the exact solution even though the polynomial adaptation is in active, as can be seen from Figure 6. Therefore, we have to resort to a different approach. This approach requires a predefined maximum number of Gauss-Lobbato points allowed within each cell element, $k_{\max}$. The polynomial refinement technique as illustrated in section 4 is performed for each cell

element and the number of Gauss-Lobbato points selected is defined as k. The approach will activate the diffusion term in equation (15) when k reaches the value of k_{max}. In the presence of a discontinuity, k will approach to infinity implying that no polynomial can fit a discontinuity. The activation of the diffusion will smear the discontinuity to a smoother but still sharp profile that can be captured by the polynomial refinement. Once the profile is smeared to a shape that can be captured by the high order polynomial (hence $k < k_{max}$), the diffusion is no longer needed. As a result, the larger value of k_{max} is used, the shaper resolution of the discontinuity will be resolved.

The advantages of the new approach can be realized as follows. First, if k reaches the value of k_{max}, it implies that the polynomial with the maximum order allowed is still not able to resolve the profile shape. Since this inadequacy of the polynomial approximation may be the source of the numerical errors, the application of the numerical diffusion in this region may be the least damaging one, in contrast to adding the diffusion in regions where the profile shape is adequately captured by the polynomial, such as cell elements with local extremes. Second, the value of the diffusivity, K, is proportional to the space distance, Δx, as shown in equation (16) and hence if a larger value of k_{max} is picked (Δx_{max} is smaller), a smaller dose of artificial diffusion is necessary. A large value of k_{max} not only reduces the amount of artificial diffusion to be added, but also increase the order of polynomial to approximate the sharp profile shape. Hence, the resolution of the profile across the discontinuity can be refined with increasing value of k_{max}.

Figure 7 shows the solution of equations (10) with initial condition (17) using this new approach. The value of k_{max} is fixed at 160 and $c = 0.5$. Again, the time step used is $\Delta t = 0.001$ and the Gauss-Lobbato points employed in the time direction is $\ell = 5$. As can be seen from the result the solution is excellent and no Gibbs phenomenon is observed in the solution. Figure 8 provides the comparison of the solutions using $k_{max} = 20$, 40, 80 and 160; under the condition $c = 0.5$. Figure 8 shows that as the maximum order of polynomial, k_{max}, is increased,

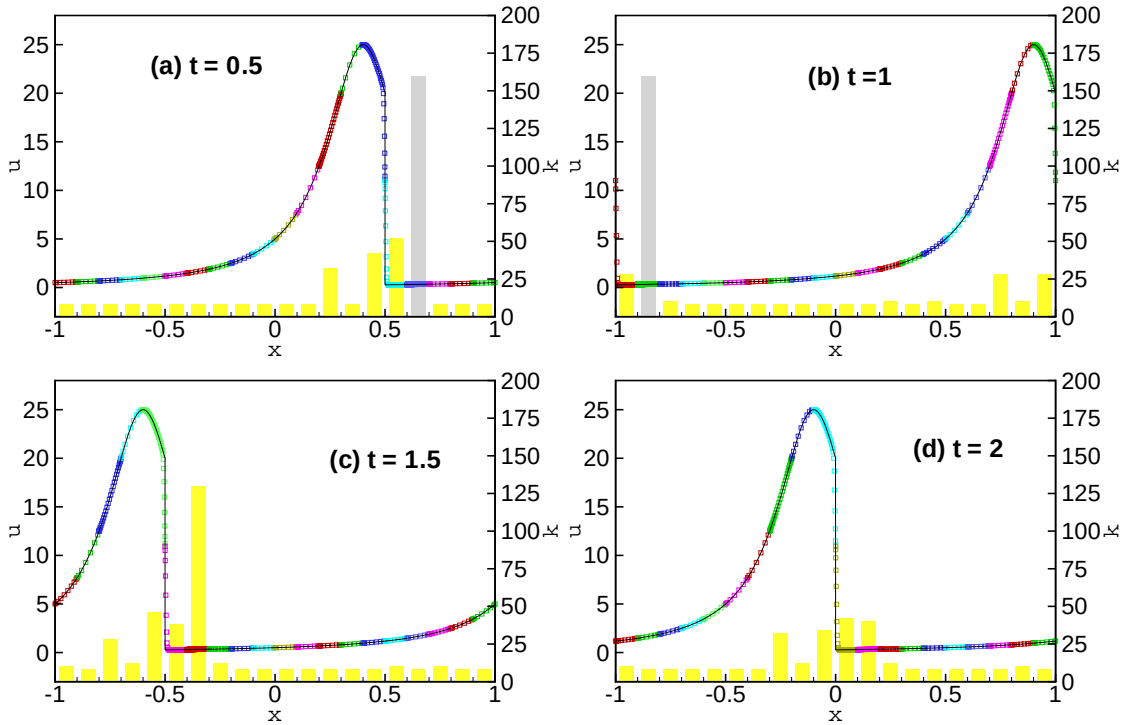

Figure 7. The adaptive polynomial refinement in the presence of discontinuity, with $k_{max} = 160$ and $c = 0.5$. The solid lines are exact values and the symbols are the solutions. The colors of the symbols represents that the Gauss-Lobbato points within the same cell element. The yellow bars depict the number of Gauss-Lobbato point, k, used in each element. The grey bars are the cell elements where diffusion is activated as a result of $k = k_{max}$.

the sharper profile shape is observed. The reduction of errors as k_{max} increases has demonstrated that with polynomial refinement it is possible to achieve an accurate solution in the presence of discontinuities.

It should be noted that in the simulation of equation (10) with initial condition (17), the diffusion is first added to the cell elements containing the discontinuity in the few initial steps, and after the discontinuity is smeared to a sharp profile that can be captured by the high order polynomial (the larger k_{max} is, the sharper the profile becomes), the diffusion is then occasionally added in the region downstream of the discontinuity to prevent the instability caused by the sharp gradient from magnifying. This mechanism is very different from the limiter treatments mentioned above.

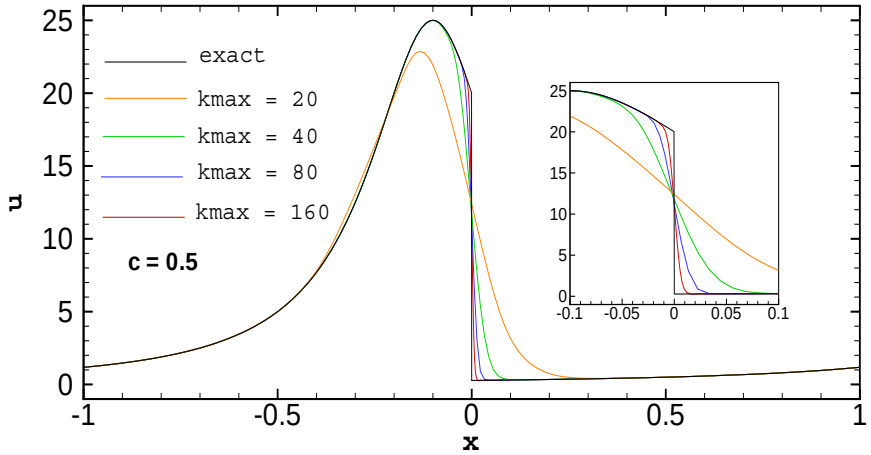

Figure 8. Adaptive polynomial refinement in the presence of discontinuity after one complete cycle– effects of k_{max} .

Figure 9 shows the comparison of calculation using $c = 0.5$, 0.25 and 0.1 under the conditions that $k_{max} = 80$ and 20. As can be seen that a slight reduction of c can increase the sharpness of the resolution without encountering Gibbs oscillations and this becomes most effective when a small value of k_{max} is used.

In figures 10 and 11, two further initial profiles were selected to validate the current method for capturing discontinuities. Figure 10 uses a step function:

$$f(x,0) = \begin{cases} 0, & |x| \geq 0.3 \\ 1, & |x| < 0.3 \end{cases}; \tag{19}$$

and figure 11 employs a triangular function:

$$f(x,0) = \begin{cases} \frac{3}{7}(1-|x|), & x \geq 0.3 \\ 1-\frac{7}{3}|x|, & x < 0.3 \end{cases}. \tag{20}$$

The calculation were performed with 20 cell elements, $k_{max} = 80$ and $c = 0.5$. The time step is $\Delta t = 0.001$ with Gauss-Lobbato point in time, $\ell = 5$. Both calculations show excellent agreements with the exact solutions indicating that the current adaptive-polynomial-refinement approach is capable of capturing discontinuities.

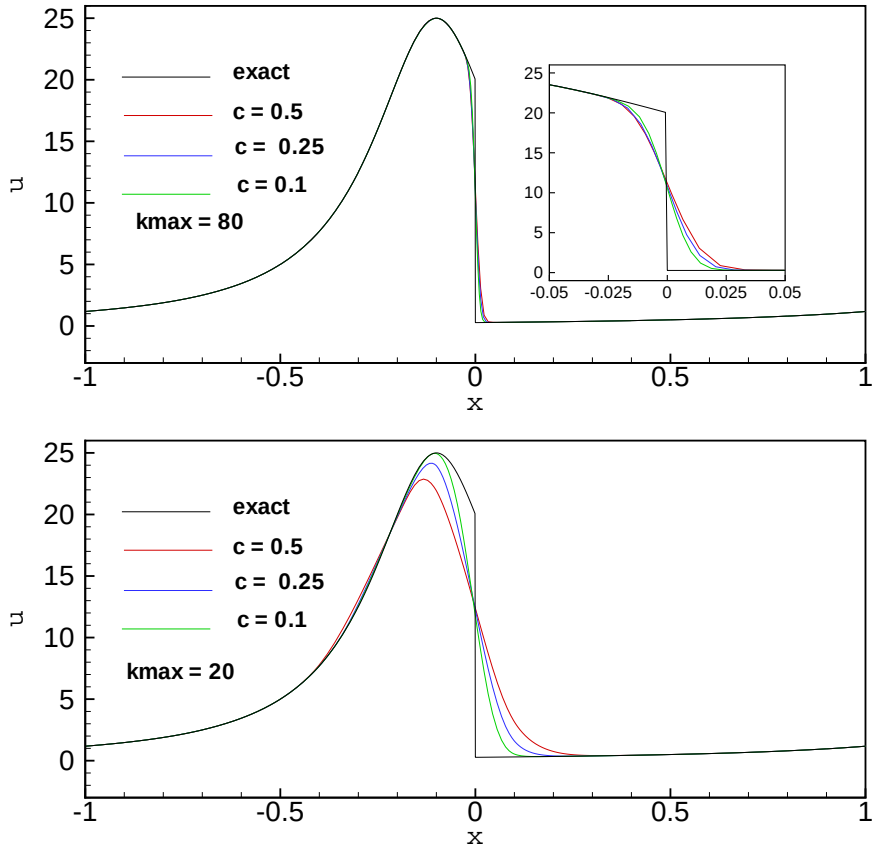

Figure 9. Adaptive polynomial refinement in the presence of discontinuity after one complete cycle – effects of *c*.

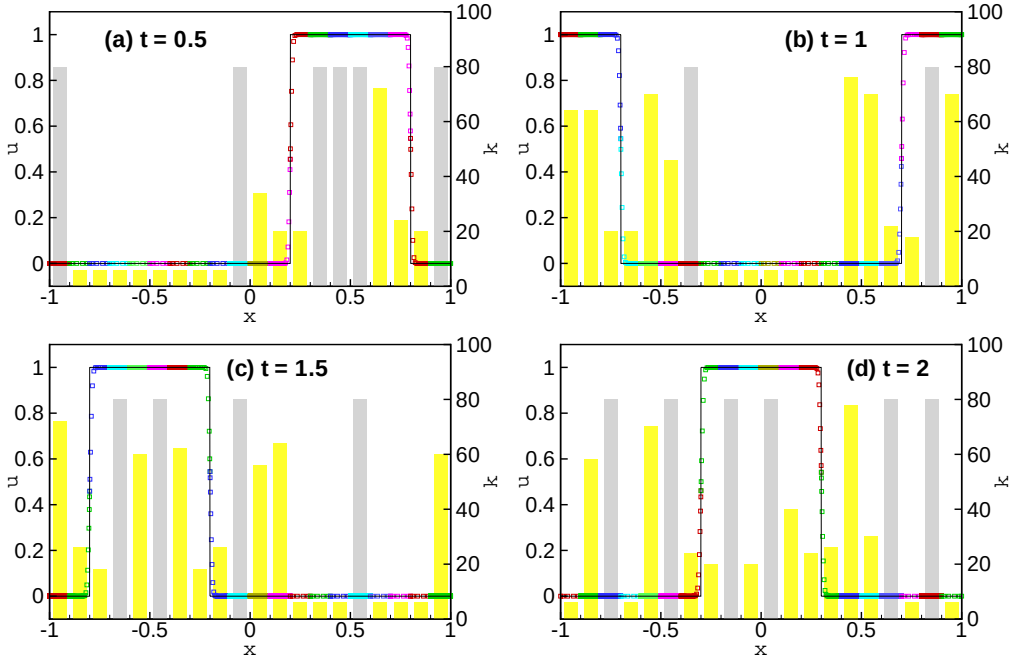

Figure 10. The prediction of the transport of a step function, equation (19). The solid lines are exact solutions, symbols are calculations (nodal points having the same colors are within the same cell element), and bars indicates the number of Gauss-Lobbato points used in each cell elements (grey color depicts the cell element with diffusion being activated as a results of $k = k_{\max}$).

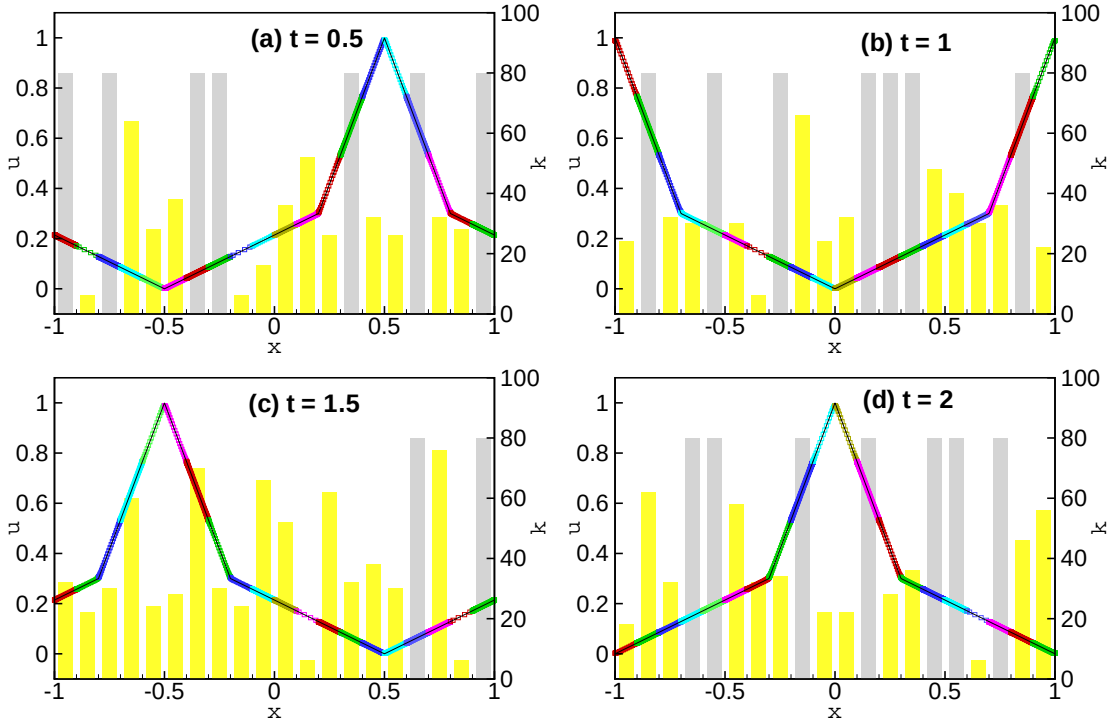

Figure 11. The prediction of the transport of a trangular function, equation (20). The solid lines are exact solutions, symbols are calculations (nodal points having the same colors are within the same cell element), and bars indicates the number of Gauss-Lobbato points used in each cell elements (grey color depicts the cell element with diffusion being activated as a results of $k = k_{\max}$).

VI. The solution of Burger's equation

The current adaptive polynomial refinement scheme is tested in a non-linear problem:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{21a}$$

with the following initial condition:

$$f(x,0) = 1 + 0.5\sin(\pi x). \tag{21b}$$

The calculation domain covers $-1 \le x \le 1$ and the periodic boundary conditions are used at the two boundaries.

Twenty cell elements are used with $k_{\max}$ =80 and $c = 0.5$ in the calculation. The time step used is $\Delta t = 0.001$ and 5 Gauss-Lobbato points are placed in the time direction. Figure 12 shows the solution obtained by the proposed adaptive polynomial refinement. The results are free from oscillation and the comparison with the exact solution is excellent. This shows that the current proposed method is also adequate to solve non-linear problems.

VII. Concluding Remarks

We have proposed a new implicit adaptive polynomial refinement to solve equations governing conservation laws. The scheme is implicit in both space and time and accurate solutions can be obtained by successive polynomial refinements, even across a discontinuity. The method has been tested to solve a scalar transport equation with a number of complicated initial profiles with excellent agreements. The method has also been applied to solve a Burger equation with impressively sharp resolution of the shock wave. Following the success we observed in this study, we plan to extend the method to multiple dimensions in the future.

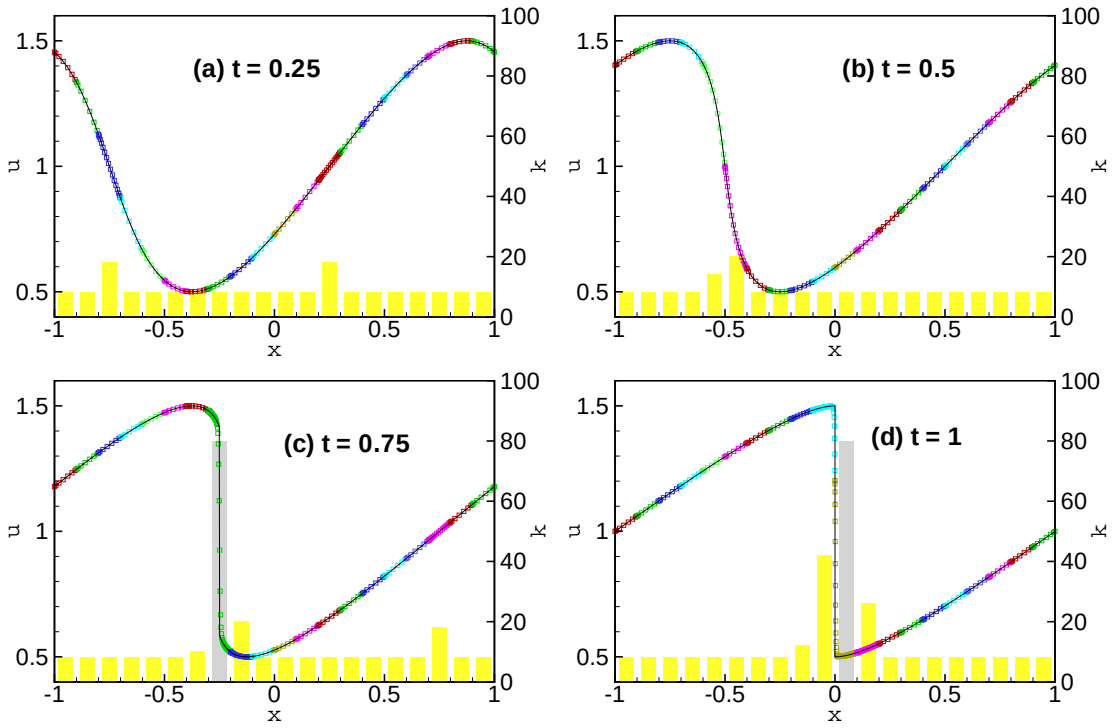

Figure 12. The prediction of the Burger's equation, (21). The solid lines are exact solutions, symbols are calculations (nodal points having the same colors are within the same cell element), and bars indicates the number of Gauss-Lobbato points used in each cell elements (grey color depicts the cell element with diffusion being activated as a results of $k = k_{\max}$).

References

[1]Y. Liu, M. Vinokur, Z.J. Wang, "Discontinuous Spectral Difference Method for Conservation Laws on Unstructured Grids," in Proceedings of the 3rd International Conference in CFD, Toronto, Canada July 2004.

[2]Y. Liu, M. Vinokur, Z.J. Wang, "Spectral Difference Method for Unstructured Grids I: Basic Formulation," submitted.

[3]D.A. Kopriva, A conservative staggered-grid Chebyshev multidomain method for compressible flows. II semi-structured method, J. Comput. Phys. 128, 475 (1996).

[4]D.A. Kopriva, A staggered-grid multidomain spectral method for the compressible Navier–Stokes equations, J. Comput. Phys., 143, No. 1, 125 (1998).

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

[6]Cockburn and C.-W. Shu, The Runge-Kutta discontinuous Garlerkin method for conservation laws V: multidimensional systems, J. Comput. Phys., 141, 199 - 224, (1998).

[7]F. Bassi and S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys. 138, 251-285 (1997).

[8]Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: I. Basic formulation, J. Comput. Phys. 178, 210 (2002).

[9]Z.J. Wang and Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation, J. Comput. Phys. 179, 665-697 (2002).

[10]Z.J. Wang and Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization, J. Scientific Comput. 20, 137-157 (2004).

[11]Z.J. Wang, L. Zhang, and Y. Liu and, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional systems, J. Comput. Phys. 194, 716-741 (2004).

[12]Y. Liu, M. Vinokur, and Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids V: extension to three-dimensional systems, submitted to J. Comput. Phys. 2005.

[13]D. Gottlieb and S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications (Society for Industrial and Applied Mathematics, Philadelphia, 1977).

[14]T.J.R. Hughes, The finite element method, linear static and dynamic finite element analysis, Prentice-Hall, Inc., 1987.

[15]G.E. Karniadakis and S.J. Sherwin, Spectral/hp element Methods for CFD, Oxford University Press, 1999.

[16]C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, Spectral Methods in Fluid Dynamics (Springer-Verlag, New York, 1987).
[17]J.L. Steger and R.F. Warming, Flux vector splitting of the inviscid gasdynamics equations with applications to finite difference methods, J. Comput. Phys. 40, 263 (1981).
[18]P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 357-372 (1981).
[19] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, McGraw-Hill, 1981.