44th AIAA Aerospace Sciences Meeting and Exhibit
9 - 12 January 2006, Reno, Nevada

AIAA 2006-304

A Spectral Difference Method for the Euler and Navier-Stokes Equations on Unstructured Meshes

Georg May* and Antony Jameson†

This work focuses on the extension of the recently introduced Spectral Difference Method to viscous flow. The spectral difference method is a conservative pseudo-spectral scheme based on a local collocation on unstructured elements. Recently results for scalar transport equations and the Euler equations have been presented. For the extension to viscous flow several techniques are investigated, such as a central discretization and a split upwind/downwind discretization, akin to the procedure used in the LDG method.

I. Introduction

Computations of compressible fluid flow on unstructured meshes has been dominated by schemes which are restricted to second order accuracy. While such low-order schemes have been very successful due to their robustness and relative efficiency, many fields of research require highly accurate numerical methods, and would certainly benefit from the applicability of high-order schemes to complex computational domains. This motivates the formulation of such schemes for unstructured meshes. Recently the Spectral Difference method, which has been proposed by Liu at al.[1] , has been extended to the Euler equations by Wang et al.[2] and the present authors[3] , and has proved to be a viable alternative to such schemes as the Discontinuous Galerkin Method[4,5] or the Spectral Volume Method[6] . The Spectral Difference method combines elements from finite-volume and finite-difference techniques, and is particularly attractive because of its simple formulation and implementation. It is conservative, and generically of arbitrary order of accuracy, using a local pseudo-spectral reconstruction on unstructured mesh elements.

The extension to viscous flow is an important problem. For the Discontinuous Galerkin method several discretization techniques for elliptic operators and schemes for the full Navier-Stokes equations have been proposed[7,8] , along with some analysis, see[9] and references therein. The Spectral Difference method uses exact differentiation of reconstructed polynomials to compute the divergence or curl of flux functions, which can be readily extended to the differentiation of primitive variables needed for the rate-of-strain tensor and heat flux vector. However, much like the DG methods, a stable discretization is strongly coupled to the treatment of element boundaries, where the reconstructed solution is discontinuous. Ideally the viscous discretization on the element boundary should involve only the two points which represent the logical left and right state. It is known that simple averaging may lead to instability, and more elaborate viscous discretization techniques should be considered. We present a technique that uses a split upwind/downwind discretization for gradients and viscous fluxes, akin to the procedure used in the Local Discontinuous Galerkin (LDG) method[8] .

The paper is organized as follows: Basic concepts related to the Spectral Difference Method are recalled in section II. A brief overview of important characteristics related to conservation and stability is given in section III. Details of the practical implementation of the scheme, and the proposed extension to viscous flow are discussed in sections IV and V.

*PhD Candidate, Department of Aeronautics and Astronautics, Stanford University, AIAA Member.
†Thomas V. Jones Professor of Engineering, Department of Aeronautics and Astronautics, Stanford University, AIAA Member.

II. The Spectral Difference Method

The spectral difference method has quite recently been proposed by Liu et al.[1] and further developed by Wang et al.[2] and by the present authors[3] . In this work we restrict ourselves to the solution of equations of the form

$$\frac{\partial u}{\partial t} + \nabla \cdot F = 0 \ , \qquad (x,t) \in \Omega \times [0,T] \ , \tag{1}$$

where $\Omega \subset \mathbb{R}^d$, and $u = (u_1, \ldots, u_p)$. We consider one and two dimensional problems ($d = 1$ and $d = 2$) subject to suitable initial and boundary conditions. Suppose we are given a triangulation of $\mathbb{R}^d$, which is assumed in this work, for the sake of simplicity, to consist of simplexes. The Spectral Difference method uses a pseudo-spectral collocation-based reconstruction for both the dependent variables $u(x)$ and the flux function $F(u)$ inside a mesh element, T_i, say. The reconstruction for the dependent variables can be written

$$u_i(x) = \mathcal{I}^m(u)(x) = \sum_{j=0}^{N(m)} L_j(x) u_{ij} \ , \qquad x \in T_i \ , \tag{2}$$

where $u_{ij} = u(x_{ij})$. Throughout the paper the double index notation refers to a node (second index) inside a cell (first index). Here x_{ij} is the j^{th} solution collocation node in the i^{th} mesh element. Henceforth we shall frequently omit the cell index i, whenever it is clear that the quantity in question is defined in a particular element. The interpolation operator $\mathcal{I}^m$ denotes a collocation using polynomials of total degree m. The $L_j(x)$ are the cardinal basis functions for the chosen set of collocation nodes x_j, where $j = 0, \ldots, N(m)$, and $N(m) = m$ for $d = 1$, whereas for $d = 2$ we have

$$N(m) = \frac{(m+1)(m+2)}{2} - 1 \ . \tag{3}$$

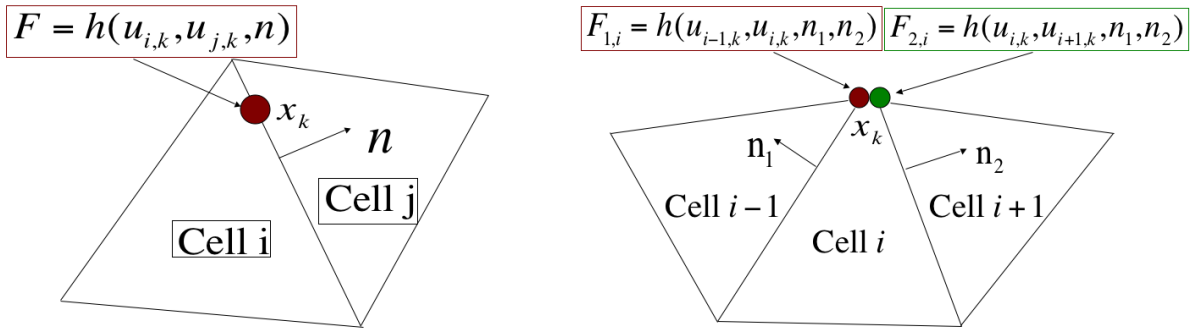
This leads to an asymptotic order of accuracy of $n = m + 1$ for the collocation. If the discrete solution u_h is defined as the union of all the interpolated functions on the mesh elements, $u_h = \cup_i u_i$, it will be discontinuous across the element boundaries. The treatment of the solution variables at the boundaries is one of the key ingredients of the spectral difference method and is discussed below. The reconstruction of the flux function in T_i reads

$$F_i(u(x)) = \mathcal{I}^{m+1}(F(u))(x) = \sum_{k=0}^{N(m+1)} M_k(x) F_k \ , \qquad x \in T_i \ , \tag{4}$$

where the M_k are the cardinal basis functions corresponding to the collocation nodes x_k , $k = 0 \ldots N(m+1)$, and $F_k = F(u(x_k))$. If the solution is reconstructed to order n, the flux nodes are interpolated to order $n + 1$, because of the differentiation operation in eq. (1). Again the interpolation will be discontinuous at element boundaries. We define the numerical flux function F_h on the triangle T_i as

$$F_h = \begin{cases} F_i & , \quad x \in T_i \\ G(u_i(x), u^{e,1}, \ldots, u^{e,k} \ldots) & , \quad x \in \partial T_i \end{cases} \ , \tag{5}$$

where the $u^{e,1}, \ldots, u^{e,k} \ldots$ are "external" solutions on triangles T_j with $j(k) \neq i$ such that $x \in \partial T_j$, and the value is given by $u^{e,k} = \lim_{y \to x} u_j(y)$. For nodes on edges of triangles there is only one external solution u^e. It is necessary for discrete conservation that the normal flux component be continuous across the edge, which suggests using numerical flux functions, standard in finite-volume formulations, such that the normal flux component $F_n = F_i \cdot n$, for the node x_k, say, is replaced by the numerical flux $h(u_i(x_k), u^e, n)$, where n is the edge normal, and h is the numerical flux function approximating F_n (see section III for a brief discussion on the properties of the numerical flux). A schematic illustration is shown in figure 1(a). For flux nodes on corners there is more than one external solution and the optimal treatment is still an open problem. One



(a) Flux computation for points on edges

(b) Flux computation for points on corners

Figure 1. Illustration of flux computation for nodes on element boundaries.

may compute the corner fluxes from normal fluxes on the two incident edges of the triangles by imposing

$$\begin{aligned} F_h \cdot n_1 &= h_{n_1} \\ F_h \cdot n_2 &= h_{n_2} \ , \end{aligned} \tag{6}$$

where n_1 and n_2 are the normals on the incident edges, and h_{n_1}, h_{n_2} are the associated numerical fluxes. This treatment was suggested by Wang et al.[2] and makes the flux unique, while allowing for conservation. The linear system, eq. (6), can be solved analytically to give modified numerical fluxes on corners that can be split into two parts, which are associated with the two incident edges. This is shown in figure 1(b).

A simple average with scalar dissipation may be used as numerical flux function for convective terms on element boundaries:

$$h(u_i(x_k), u^e, n) = \frac{1}{2} \left\{ (F(u_i(x_k)) + F(u^e)) \cdot n - \alpha_n \left(u^e - u_i(x_k) \right) \right\} \ , \tag{7}$$

where α_n is proportional to the spectral radius of the local flux Jacobian, $\alpha_n \propto |u_n| + c$, where u_n is the velocity normal to the edge, and c is the local speed of sound. We have also used the CUSP construction of artificial diffusion[10] instead of the simple dissipation of eq. (7) with good results. The CUSP flux is an attractive alternative, because it is significantly less dissipative, yet relatively inexpensive. This is beneficial for lower order computations, where one wishes to avoid excessive dissipation. For higher order reconstruction the difference becomes smaller, as the dissipation introduced is proportional to the discontinuity in the solution, which diminishes with increasing order or accuracy, at least in smooth regions. The treatment for nonsmooth flow is discussed in section IV. Tangential flux components can either be evaluated in each cell and left unchanged or averaged across cell interfaces using an arithmetic average to produce a flux function F_h which is single-valued everywhere. In practice the treatment of the tangential components seems to be of minor importance.

The baseline scheme is now readily defined in ODE form as $u_h = L_h(u_h)$, where the degrees of freedom are given by the values of the solution at the collocation nodes, $u_{ij} = u_h(x_{ij})$, where again x_{ij} is the j^{th} solution collocation node in the i^{th} mesh element, and the right-hand side solution operator is given by the exact differentiation of the reconstructed flux function:

$$\frac{du_{h,ij}}{dt} = \left(\nabla \cdot F_h \right) (x_{ij}) \tag{8}$$

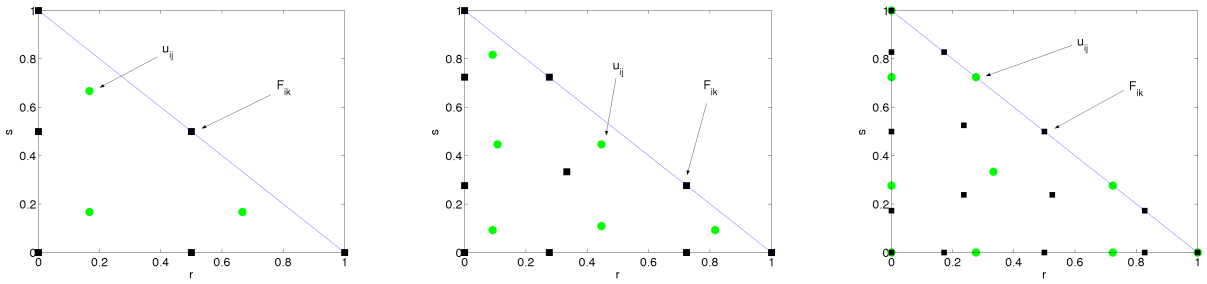
In most implementations this ODE is solved with a Runge-Kutta time integration scheme.

The SD method is closely related to staggered grid multidomain spectral methods, proposed by Kopriva[11] . Here, in a sense, each simplex is a subdomain, and instead of tensor product forms of one-dimensional basis functions, we use two-dimensional collocation methods.

Any combination of collocation nodes may be used, provided that the nodes for u support a quadrature of the order of the interpolation n, and the restriction of the flux nodes to the boundaries supports a $d-1$-dimensional quadrature of order $n+1$. This ensures discrete conservation in the sense that

$$\frac{d}{dt}\int_{T_i} u_h \, dx = -\int_{\partial T_i} \mathbf{F}_h \cdot \mathbf{n} \, dA \ , \tag{9}$$

is satisfied exactly for the solution and reconstructed flux function[3] . For the solution nodes one can choose Gauss quadrature points. Hesthaven proposed nodes based on the solution of an electrostatics problem for simplexes[12] , which support both a volume and a surface integration to the required degree of accuracy. These nodes can be used for both flux and solution collocation. Figure 2 shows examples of nodes for elements of various orders.



(a) Linear Triangle (b) Quadratic Triangle (c) Cubic Triangle

Figure 2. Schematic depiction of collocation nodes for triangles.

III. Properties of the Baseline Scheme

Conservation is a very important property in the context of numerical schemes for hyperbolic PDE. Consider for the sake of simplicity the one-dimensional equation on a uniform mesh. A scheme for the equation

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

is said to be conservative if it can be written for the time instance t^n with corresponding solution $u_h(t^n)$:

$$\bar{u}_i^{n+1} = \bar{u}_i^n - \lambda \left(h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}} \right) \tag{11}$$

where $\overline{u}_i$ denotes the volume average of u_h in cell i, $\lambda = \Delta t^n / \Delta x$, and h stands for the numerical flux function at the element interfaces. Here we consider only three-point consistent, Lipschitz-continuous flux functions, i.e. $h = h(u_1, u_2)$, and $h(u,u) = f$. Conservation requires $h(u_1, u_2) = -h(u_2, u_1)$. These properties are important in the theory of total-variation-diminishing (TVD) stability theory briefly discussed below.

The n^{th}-order accurate one-dimensional Spectral Difference Scheme uses polynomials of degree $m = n-1$ and $m+1$ for the solution and flux function, respectively. Let these collocations be defined by the local

solution nodes x_j , $j = 0 \dots m$ and flux nodes x_k , $k = 0, \dots m+1$. With a forward-Euler time discretization the scheme reads

$$u_{ij}^{n+1} = u_{ij}^{n} - \lambda \sum_{k=0}^{m+1} m_{jk} f_{ik} \tag{12}$$

where $\frac{1}{\Delta x} \sum m_{jk} f_k = \frac{d}{dx} \mathcal{I}^{m+1}(f_h)(x_j)$. Conservation is ensured by construction, which can be seen by noting that the equation for the volume averages can be written

$$\sum_{j=0}^{m} w_j u_{ij}^{n+1} = \bar{u}_i^{n+1} = \bar{u}_i^{n} - \lambda \sum_{j=0}^{m} w_j \sum_{k=0}^{m+1} m_{jk} f_{ik} = \sum_{k=0}^{m+1} \widetilde{w}_k f_{ik} \tag{13}$$

where the w_j are quadrature weights corresponding to the chosen set of solution nodes, which must support exact quadrature of polynomials at least up to degree m. Also,

$$f_k = \begin{cases} h_{i-\frac{1}{2}} & k = 0 \\ h_{i+\frac{1}{2}} & k = m+1 \\ f(u_k) & \text{otherwise} \end{cases} . \tag{14}$$

Hence the modified quadrature weights $\widetilde{w}_k$ are given by

$$\widetilde{w}_k = \sum_{j=0}^{m} w_j m_{jk} . \tag{15}$$

This, however, can be simplified by noting that $m_{jk} = \Delta x M_k'(x_j)$, where $M(x)$ is the cardinal basis function for the flux collocation, whose derivative is a polynomial of degree m by construction, and is hence integrated exactly, which means that

$$\sum_{j=0}^{m} w_j m_{jk} = \int M_k'(x)\, dx = M_k(x_{i+\frac{1}{2}}) - M_k(x_{i-\frac{1}{2}}) = \begin{cases} -1 & k = 0 \\ 1 & k = m+1 \\ 0 & \text{otherwise} \end{cases} , \tag{16}$$

using the fact that $M_k(x)$ is an interpolation polynomial for which $M_k(x_l) = \delta_{kl}$ holds. Note that the Interval endpoints must be flux collocation nodes for the scheme to be well defined, since numerical flux functions must be used there to provide the inter-element coupling. Upon substitution of this result into equation (13), the conservation form for the volume averages, equation (11), is recovered.

For our purposes we restrict ourselves to Lipschitz-continuous fluxes, which together with the conservation property allows us to inherit useful results from the theory of total-variation stability for the volume averages, as long as the total variation can be bounded. This is usually accomplished by using a suitable local limiting procedure (or "local projection" in the parlance of DG methods). An example is given by the minmod function, which reads

$$m(a,b,c) = \begin{cases} sign(a) \min(|a|, |b|, |c|) & sign(a) = sign(b) = sign(c) \\ 0 & \text{otherwise} \end{cases} , \tag{17}$$

where, when used during the reconstruction at the interface $x_{i+\frac{1}{2}} = x_{i,m+1} = x_{i+1,0}$, the numerical fluxes at the interface $x_{i+\frac{1}{2}}$ are evaluated with $u_{i+\frac{1}{2}} = \overline{u}_i + m(a,b,c)$ we have

$$a = \delta u_{i+\frac{1}{2}} = \mathcal{I}^m(u_h)(x_{m+1}) - \overline{u}_i \tag{18}$$

$$b = \Delta u_{i+\frac{1}{2}} = \overline{u}_{i+1} - \overline{u}_i \tag{19}$$

$$c = \Delta u_{i-\frac{1}{2}} = \overline{u}_i - \overline{u}_{i-1} . \tag{20}$$

We discuss the limiting procedure for practical implementations in greater detail in section IV.

While a discussion on TVD stability is outside the scope of this paper (basic concepts are documented for example in[13]), the line of thought is that the local projection bounds the variation at time instance n, which carries over to the numerical fluxes via the Lipschitz property. Since only the numerical fluxes at the interfaces affect the update of the volume average for a conservative scheme, it also can be bounded under some CFL-like condition for an explicit scheme. This allows for estimates following exactly along the lines of those for Discontinuous Galerkin and other methods. Roughly speaking, the method of reconstruction is immaterial, as long as the scheme is conservative, the limiting procedure ensures the verification of a local maximum principle, and standard Lipschitz continuous numerical fluxes are used.

IV. Implementation Details

A. Solution-Adaptive Meshing

To replace mesh resolution by increased order of approximation is perhaps the main rationale behind using high-order numerical methods. However, for hyperbolic equations, one has to expect to encounter shock waves. A significant challenge is thus introduced: If a coarser mesh is used to reap the benefits of a high-order numerical approximation, what is to become of the shock capturing capabilities? Putting all questions of robustness and stability aside, it is evident that shock resolution must necessarily deteriorate, as the shock thickness will be determined by the local characteristic mesh length. Adaptive meshing is thus particularly attractive in conjunction with high-order approximations when shock waves have to be expected. One may use this technique to locally refine the mesh near a discontinuity in order to improve shock capturing capabilities.

Furthermore, local limiting procedures, necessary for monotonicity in the presence of discontinuities, are rather intrusive for high-order methods, where a local high-order reconstruction is usually replaced by a carefully limited linear one (see section B). For large mesh elements this will be an extremely poor approximation and at the same time a very massive change in the solution representation, likely to lead to limit cycles and impede convergence. For smaller elements near a discontinuity, on the other hand, the linear approximation will be more fitting.

Although not used here, adaptive coarsening may also prove useful in conjunction with a locally increased order of approximation in smooth regions to reduce the computational cost.

The solution-adaptive mesh refinement used in the present work has been patterned after the methodology documented by Kim et al.[14] , and we refer the interested reader to their paper. The criterion for adaptation are gradients in the solution. We have used both gradients in the pressure and the entropy.

B. An improved Limiting Procedure

Limiting of the reconstruction is essential to shock capturing. For one-dimensional schemes strict estimates regarding TVD stability can be obtained depending on standard limiting procedures. The challenge is a successful translation to an implementation on unstructured meshes in higher dimensions. In[15] the authors present such an extension for the Runge-Kutta DG scheme and multidimensional scalar equations. Stability and convergence estimates have been obtained, albeit for special classes of triangulations. The procedure has subsequently been extended to the two-dimensional Euler equations in a semi-heuristic fashion[16] .

Let us first consider a scalar equation of the form

$$\frac{\partial u}{\partial t} + \nabla \cdot f(u) = 0 \ . \tag{21}$$

The most basic steps of the limiting procedure, to be carried out at each time instance in each cell of the triangulation, can be identified in generic terms as

1. compute a number of local reference states

2. compare the current reconstruction to the local reference states

3. if necessary, modify the reconstruction to comply with a local maximum/minimum principle established by a predefined criterion which limits the admissible range of the reconstructed solution as a function of the reference state

As an example consider again the one-dimensional procedure outlined in section III. Step one is represented by computing the volume averages in adjacent cells. The minmod function combines steps 2 and 3 by comparing the slopes of the reconstruction and reference states and returning the original reconstruction in smooth monotone regions and zero otherwise.

The main task in devising the multidimensional limiting procedure is to select the right method for each of these steps. For the local reference state the cell-averaged solutions in a local neighborhood of a cell, T_i say, may be used. We restrict the discussion in this section to the two-dimensional case. The extension to three dimensions is straight forward. Consider the quantities

$$u_{i,max} = \max(\overline{u}_i, \max_{k\in\mathcal{N}_i}(\overline{u}_k)) \tag{22}$$

$$u_{i,min} = \min(\overline{u}_i, \min_{k\in\mathcal{N}_i}(\overline{u}_k)) \tag{23}$$

where we define the index set $\mathcal{N}_i = \{k : T_k \cap T_i \in E\}$, and E is the set of edges of the triangulation. Alternatively a nodal neighborhood can be defined, for a node x_i, say: $\mathcal{N}_i = \{k : x_i \text{ is a node of } T_k\}$. The reference state becomes

$$u_{i,max} = \max_{k\in\mathcal{N}_i}(\overline{u}_k) \tag{24}$$

$$u_{i,min} = \min_{k\in\mathcal{N}_i}(\overline{u}_k) \tag{25}$$

Step 1 is thus complete. Step 2 depends to some extend on how the reference state has been chosen and on what part of the reconstruction is to be compared to the reference state. In[16] the authors suggest to use the linear portion of the reconstruction under the assumption that limiting is only necessary if it is necessary for the linear restriction. In any case, however, we may define the quantity Δu at new collocation nodes x_r, $r = 1, 2, 3$ inside the triangle.

$$\Delta u(x_r) = (\mathcal{P}(u_i)(x_r) - \overline{u}_i) \ , \tag{26}$$

where the projection $\mathcal{P}$ may be the projection onto the space of linear function or the identity if the full reconstruction is used in the comparison. If a cellwise neighborhood has been used in step 1 the nodes x_r may be chosen at edge midpoints, whereas for the nodal neighborhood the nodes of the triangle are used.

If one insists that the reconstructed solution must not exceed the bounds established by the maximum and minimum volume averages of the chosen local neighborhood, the limiter will be activated whenever the magnitude of Δu exceeds the magnitude of the allowed variation

$$\Delta u^{\text{ref}} = \max(\overline{u}_i - u_{i,\min}, u_{i,\max} - \overline{u}_i) \tag{27}$$

at any of the nodes x_r. The quantity $\phi(x_r)$ may be defined as

$$\phi(x_r) = \begin{cases} 1 & |\Delta u(x_r)| \le \Delta u^{\text{ref}} \\ \frac{\Delta u^{\text{ref}}}{|\Delta u(x_r)|} & \text{otherwise} \end{cases} \ . \tag{28}$$

Whenever $\phi < 1$ the local reconstruction may be modified according to

$$\widetilde{u}_i = \overline{u}_i + \mathcal{I}^1(\widetilde{\Delta u})(x) \ , \tag{29}$$

where $\mathcal{I}^1$ is the linear interpolation operator defined by the nodes x_r, and

$$\widetilde{\Delta u(x_r)} = \phi(x_r)\,(\mathcal{P}(u_i)(x_r) - \overline{u}_i) \ , \tag{30}$$

A modification may be introduced by rejecting any limiting whenever $\Delta u < M(\Delta x)^2$, where Δx is a characteristic mesh length. This will prevent a modification of the solution near smooth extrema and is related to the theory of total-variation boundedness (TVB)[4] . In practice we usually choose values of M between 0 and 40.

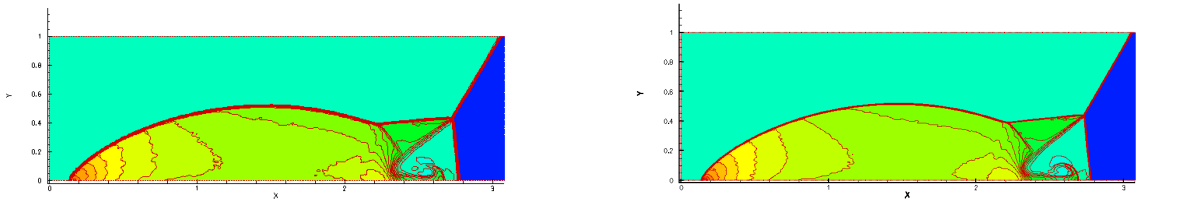
The extension to systems is straight forward. In principle a characteristic decomposition could be carried out, and the limiting could be applied to the characteristic variables, which has the advantage that the 1D stability estimates for the scalar equation carry over to 1D systems. For higher dimensions, however, these strict estimates do not hold, and the additional expense of a characteristic decomposition seems unwarranted. We prefer to apply the limiting procedure componentwise, which means that each conservative variable is tested for monotonicity and assigned its own limiter variable ϕ_l, where l stands for the l^{th} conserved variable.

The procedure is sufficient to ensure that the reconstruction stays within the prescribed bounds, but it is not conservative in the sense that in general

$$\int_{T_i} \widetilde{u}_i \, dx \neq \int_{T_i} u_i \, dx \ . \tag{31}$$

A further modification of the linear solution is proposed in[16] , which ensures conservation in a way that prevents an increase of the slope. We adopt this strategy for the present work, omitting the technical details for the sake of brevity, and refer instead to[16] . Our general procedure becomes identical to the one described in[16] if a cell-wise neighborhood is used to compute the reference states, and the linear component of the solution is used for comparison. In the present work we use the nodal neighborhood, and the full reconstruction is evaluated at the nodes of the triangles and compared to the reference state.

We have tested this limiting procedure for both unsteady and steady state problems. The well-known Mach reflection case, studied by Woodward and Colella[17] , and subsequently many other researchers, has been chosen for unsteady validation. Figure 3 shows results for the third-order SD scheme at time $t = 0.2$



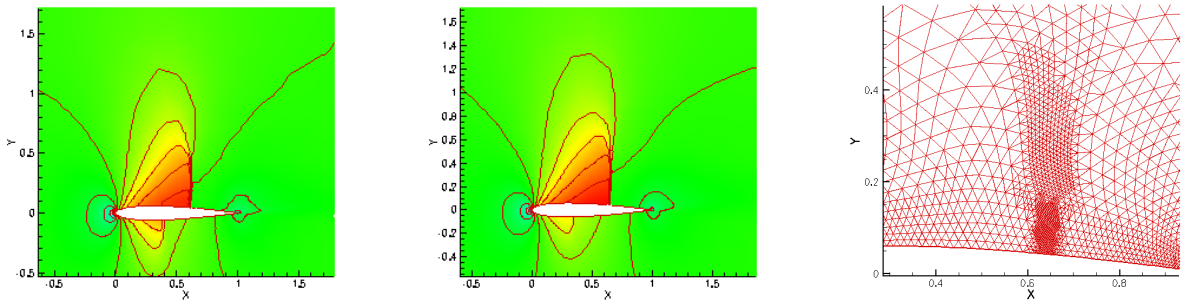
(a) contour lines of the density. Mesh: 133,480 Triangles

(b) contour lines of the density. Mesh: 533,920 Triangles

Figure 3. The Mach reflection testcase using the third order Spectral Difference scheme.

in terms of density contours for two different triangular meshes. While no mesh refinement has been used for this testcase, it can bee seen that shocks are nevertheless very well captured. To illustrate the operation of the limiting procedure, consider fig. 4, which corresponds to the numerical solution in fig. 3(a). The plot shows contour lines of the values of the limiting variable ϕ, clipped above $\min_l \phi_l = 0.99$ where the l stands for the l^{th} conserved variable. This effectively shows the regions of active limiters, which can be seen to be confined to the vicinity of the shock and the contact discontinuity in the solution. In smooth regions there is virtually no limiting, which means that the full resolution of the quadratic polynomials is available.

Steady-state test cases involving shocks are extremely challenging for high-order schemes. We have carried out preliminary test for transonic flow over a NACA 0012 airfoil. A rather popular testcase for inviscid aerodynamics is given by the flow conditions $M = 0.8$ at an angle of attack $\alpha = 1.25^o$. Fig. 3 shows contour lines of the Mach number for this test case for a finite-volume scheme and the SD scheme using quadratic polynomials. Both solutions have been computed on the same mesh, which can be seen in



(a) Contour lines of the Mach number for the third-order SD scheme

(b) Contour lines of the Mach number for the second order Finite-Volume scheme

(c) Close-up view of refined area on the suction side

Figure 5. The NACA 0012 airfoil in transonic flow. $M = 0.8$, $\alpha = 1.25$

fig. 5(c). Adaptive refinement has been used on the suction side to aid in shock resolution. While these are the first steady-state solutions for transonic flow computed with the high-order SD method, it must be pointed out that convergence of the residuals to machine zero was not attained, because the activity of the limiters causes the scheme to be caught in limit cycles, which locally stalls the reduction of the residuals, thus impeding convergence. The resolution of this problem will be subject of further research.

C. Surface Parametrization

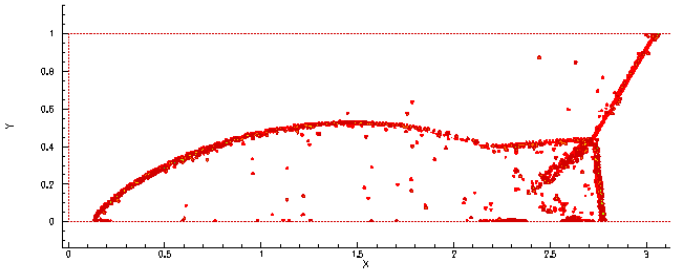
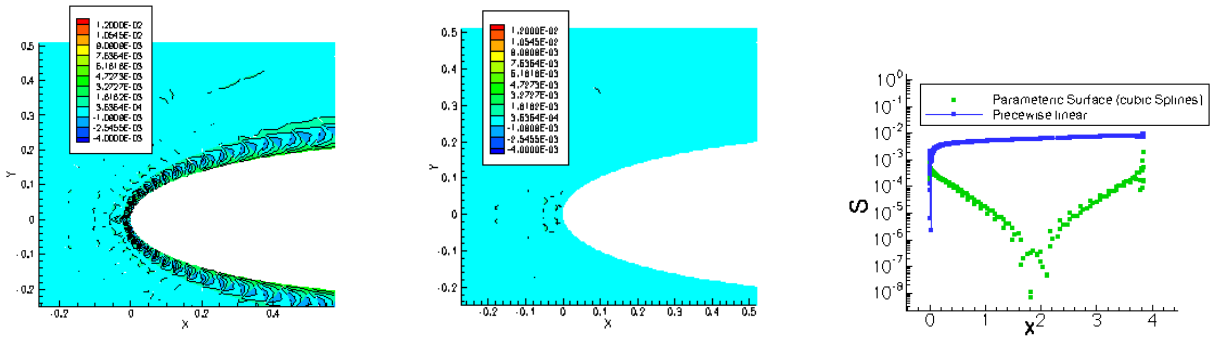


Figure 4. cells with active limiters

A piecewise linear approximation of surfaces is standard in second order accurate numerical schemes, which is sufficient, because such a scheme cannot tell the difference between the actual surface and a piecewise linear approximation. For higher-order schemes, however, such a simplified surface representation can have disastrous effects. This is demonstrated in fig. 6, which analyzes spurious entropy production for inviscid subsonic flow ($M_\infty = 0.3$) around the NACA 0012 airfoil at zero angle of attack. The relative entropy error is defined here as $s = (p/p_\infty)(\rho_\infty/\rho)^\gamma - 1$, where ρ is the density, p is the static pressure, and γ is the ratio of specific heats. The subscript ∞ refers to free stream conditions. In 6(a) contour lines of the entropy error are shown for a piecewise linear surface approximation computed with the 3^{rd} order SD scheme, while in 6(b) the same contour plot is shown for a parametric surface representation using cubic splines and 80 knots (i.e. the original surface points given as input parameters). The spurious entropy production for the piecewise linear approximation is clearly visible and is even more obvious in fig. 6(c) where the entropy error is plotted along the airfoil on a logarithmic scale. For the parametric surface treatment there is significant entropy production only at the stagnation point at the leading edge and at the sharp trailing edge. In between the entropy error is practically negligible. For the piecewise



(a) Contour lines of the entropy error near the airfoil leading edge for the third-order SD scheme and piecewise linear surface representation

(b) Contour lines of the entropy error near the airfoil leading edge for the third order SD scheme and parametric surface representation using cubic splines

(c) Comparison of the relative entropy error long the airfoil surface for a piecewise linear surface and parametric surface representation

Figure 6.

linear surface representation, however there is significant entropy production along the entire airfoil. The reason for this behavior is that if a piecewise linear airfoil is specified, a high order numerical scheme will compute precisely that, a flow over a piecewise linear surface, because it is able to discern the difference. This can dramatically deteriorate the quality of the solution. We use a parametric surface representation for all curved surfaces. This means that some cells will have curved boundaries, which in turn means that all reconstruction coefficients will no longer be universal for these elements. For linear elements, i.e. elements with straight boundaries theses coefficients are universal up to simple geometric scaling factors, which can be expressed in terms of the face areas and volumes. For curved boundaries these coefficients have to be computed individually and stored for each element, which is a one-time implementation effort (the memory penalty is mild since only few elements usually have a curved boundary).

V. Viscous Terms

The hyperbolic character of the Euler equations is a major influence in the formulation of the Spectral Difference Method. However, many physical phenomena, even if they are dominated by wave propagation, have dissipative features as well, which often cannot be neglected even if they are small. Fluid flow is a notable example: For one thing, the boundary conditions change (from slip to no-slip at walls). Furthermore, although the viscous effects are often restricted to a small boundary layer, for general flow situations viscous effects can cause phenomena, such as separation, that affect the entire flow. Therefore, a suitable way of treating dissipative terms must be found, while we would like to keep the treatment of the convective terms intact.

For the compressible Navier-Stokes equations in differential form the viscous fluxes can be modeled as

$$F_v = \nabla \cdot \tau \ , \tag{32}$$

where the tensor τ is given by

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \delta_{ij} \frac{\partial u_l}{\partial x_l} \right) \ , \tag{33}$$

where summation over repeated indices is understood. For incompressible flow the viscous terms reduce to a Laplacian operator, which is why a simplified equation including a Laplacian operator is often taken as

a model equation to test viscous discretization techniques. Let us first consider the one-dimensional linear advection-diffusion equation, where the laplacian operator is simply the second derivative:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{34}$$

One way of solving eq. (34) is to rewrite it as a system:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \nu\frac{\partial q}{\partial x} \tag{35}$$

$$\frac{\partial u}{\partial x} - q = 0 \ . \tag{36}$$

This general approach has been taken by many researchers for viscous discretization in finite element methods,[7,8,18] such as the Discontinuous Galerkin Method.

If the discretization of the variable q is "local", i.e. it depends only on the latest solution of u, the system can be solved in a segregated manner. This means that the approximation for the gradient q can be computed first, using the latest estimate of u and a suitable discretization of eq. (36). Subsequently this approximation is used in eq. (35) to advance the solution u. Consider eq. (35) of the new system in semi-ODE form and Spectral Difference discretization, where again x_j and x_k are the nodes defining the collocation for the solution and the flux function, respectively:

$$\Delta t\frac{du_{ij}}{dt} + \lambda\sum_{j=0}^{m+1} m_{jk} f_{h,ik} \ , \tag{37}$$

where $f_h = f_{h,c} - f_{h,v}$. The fluxes $f_{h,c}$ and $f_{h,v}$ discretize the convective and dissipative terms, $f_c = cu$ and $f_v = \nu q$, respectively. For the convective part one may choose an upwind discretization:

$$f_{h,c} = cv_k \ , \qquad \text{where} \quad v_k = \begin{cases} p_{ik} & k \neq 0 \\ p_{i-1,m+1} & k = 0 \end{cases} \ , \tag{38}$$

and $p_k = p(x_k) = \mathcal{I}^m(u_h)(x_k)$ is the reconstructed solution at the flux nodes. The concepts of wave propagation that guide the discretization of the convective terms do not apply to dissipation, which is not associated with a preferred direction of propagation. In principle a central discretization of dissipative terms is therefore appropriate, and has (in one form or another) often been applied to both scalar equations and systems such as the Navier-Stokes equations. However, stability concerns have to be addressed. We define a more general weighted discretization of the form

$$f_{v,k} = \frac{\nu}{\Delta x}\widetilde{q}_k \ , \qquad \text{where} \quad \widetilde{q}_k = \Delta x \cdot \begin{cases} \frac{1}{2}\left(q_{i-1,m+1} + q_{i,0}\right) - \sigma\left(q_{i-1,m+1} - q_{i,0}\right), & k = 0 \\ q_{ik}, & k \neq 0, m+1 \\ \frac{1}{2}\left(q_{i,m+1} + q_{i+1,0}\right) - \sigma\left(q_{i,m+1} - q_{i+1,0}\right), & k = m+1 \end{cases} \tag{39}$$

Using the definition of the CFL number $cfl = \lambda c$ and introducing the cell Peclet number $Pe = c\Delta x/\nu$ the scheme becomes

$$\Delta t\frac{du_{ij}}{dt} + \mathrm{c}fl\sum_{j=0}^{m+1} m_{jk}\left(v_{ik} - \frac{\widetilde{q}_{ik}}{Pe}\right) \tag{40}$$

The remaining task of discretizing the second equation of the new system, eq. (36), is equivalent to precomputing an estimate of the gradient if the scheme is local. Consistent with the above rationale we write a weighted discretization of the form

$$q_k = \frac{1}{\Delta x}\sum_{l=0}^{m+1} \mu_{kl}\widetilde{p}_l \tag{41}$$

where μ_{kl} are elements of a differentiation matrix, evaluating the derivative of the quantity $\widetilde{p}$ at the flux nodes x_k, i.e. $\frac{1}{\Delta x}\sum_{l=0}^{m+1}\mu_{kl}\widetilde{p}_l = \frac{d}{dx}\widetilde{p}(x)(x_k)$ in each cell i, and $\widetilde{p}$ is given by

$$\widetilde{p}_{il} = \begin{cases} \frac{1}{2}\left(p_{i-1,m+1} + p_{i,0}\right) + \sigma\left(p_{i-1,m+1} - p_{i,0}\right) & l = 0 \\ p_{il} & l \neq 0, m+1 \\ \frac{1}{2}\left(p_{i,m+1} + p_{i+1,0}\right) + \sigma\left(p_{i,m+1} - p_{i+1,0}\right) & l = m+1 \end{cases} \tag{42}$$

where p_l is the reconstructed solution at the l^{th} flux collocation node. Note the change of sign between eq. (39) and eq. (42) in the term augmenting the central average. We may characterize this discretization as having a complementary upwind/downwind bias. It can be easily verified that in the case of a first order scheme, i.e. $m = 0$, the above discretization collapses to the standard three-point stencil for the second derivative if $\sigma = 0.5$ or $\sigma = -0.5$. The central discretization is recovered for $\sigma = 0$, which leads to a decoupled five-point stencil for the first order scheme (to avoid such a decoupled stencil is the primary motivation behind introducing this weighted form of discretization). The method is closely related to the *LDG* method[8] , which utilizes a very similar upwind/downwind splitting for the viscous terms, and reduces to the same first order scheme (that is to say, for the first order scheme the present SD method and LDG scheme become identical if the same numerical flux function for the convective terms are used).

The asymptotic convergence in p refinement must be expected to be suboptimal in general, in the sense that convergence no better than order n can be guaranteed if polynomials of order n are used (as opposed to the inviscid case, where we expect order $n+1$). This is due to the extra derivative for the dissipative terms. This property is also shared by the LDG method. In practice, however, optimal order of convergence is often achieved. We will continue to refer to the design accuracy of $n = m+1$ if polynomials of degree m are used, keeping in mind that optimal convergence for viscous problems may not alway be achieved.

Consider the numerical solution of eq. (34) using the scheme (40). Several tests for varying order of accuracy, cell Peclet number, and discretization parameter σ have been carried out. The optimal order of convergence has always been achieved for this linear problem. We show a few illustrative examples (tables 1 through 4 on the following page). The accuracy is maintained until machine zero is approached. A four stage

# Elements	L_∞(Error)	Order	L_2(Error)	Order
10	6.547057e-02		3.888390e-02	
20	1.749077e-02	1.904253	1.002766e-02	1.955188
30	7.804592e-03	1.990211	4.479220e-03	1.987590
40	4.398009e-03	1.993730	2.526923e-03	1.989858
50	2.828825e-03	1.977608	1.621178e-03	1.989075
75	1.266920e-03	1.981114	7.239641e-04	1.988251
100	7.168179e-04	1.979693	4.086552e-04	1.987854

Table 1. Second Order scheme, $\sigma = -0.5$, $Pe = 0.01$

Runge-Kutta scheme has been used to advance the solution. However, time accuracy plays no role in this test, since the time step has been chosen small so that the error is dominated by the spatial discretization. The equation has been integrated to $T = 1$ for $Pe = 0.01$. For $Pe = 0.1$ we have chosen $T = 0.1$ as the solution will decay much more quickly. Fig. 7 on page 14 summarizes the convergence study for $\sigma = -0.5$ and $Pe = 0.1$. In fig. 7(a) the error is measured as the mesh is refined for the 2^{nd} through 5^{th} order scheme, while fig. 7(b) demonstrates spectral convergence as the order of accuracy is increased at constant number of mesh elements (40 in this case).

For the extension to higher dimensions we need to translate the symmetric upwind/downwind bias of the one-dimensional discretization to unstructured multidimensional mesh elements. For the Navier-Stokes

# Elements	L_∞(Error)	Order	L_2(Error)	Order
10	2.111335e-03		1.188727e-03	
20	2.386526e-04	3.145172	1.460845e-04	3.024542
30	7.050337e-05	3.007283	4.264994e-05	3.036413
40	2.974449e-05	2.999897	1.789916e-05	3.018165
50	1.515007e-05	3.023340	9.170556e-06	2.996976
75	4.611315e-06	2.933648	2.746238e-06	2.973785
100	1.985667e-06	2.928782	1.172301e-06	2.959043

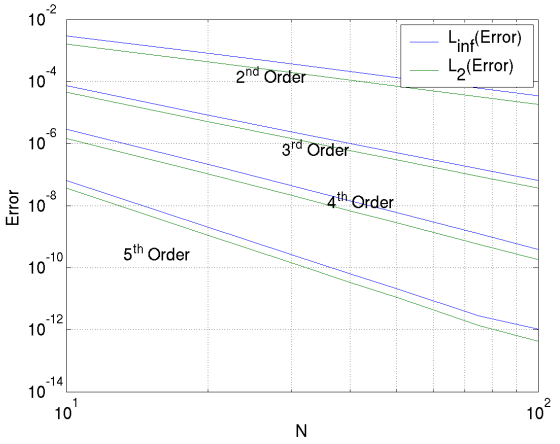
Table 2. Third Order scheme, $\sigma = 0$, $Pe = 0.01$

# Elements	L_∞(Error)	Order	L_2(Error)	Order
10	2.896343e-06		1.477824e-06	
20	2.137082e-07	3.760518	1.036278e-07	3.833991
30	4.458241e-08	3.865369	2.124086e-08	3.908792
40	1.452898e-08	3.897337	6.844647e-09	3.936498
50	6.049254e-09	3.926667	2.833991e-09	3.951631
75	1.219807e-09	3.949149	5.680424e-10	3.963954
100	3.897074e-10	3.966364	1.808876e-10	3.977725

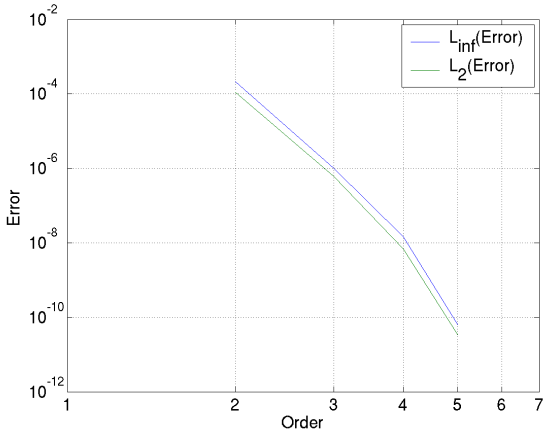
Table 3. Fourth Order scheme, $\sigma = -0.5$, $Pe = 0.1$

# Elements	L_∞(Error)	Order	L_2(Error)	Order
10	6.517415e-08		3.716928e-08	
20	2.042238e-09	4.996077	1.109798e-09	5.065742
30	2.675080e-10	5.013137	1.439591e-10	5.037188
40	6.346705e-11	5.000759	3.396990e-11	5.019619
50	2.088936e-11	4.980117	1.112197e-11	5.003740
75	2.818483e-12	4.940107	1.375349e-12	5.155104
100	1.060079e-12	3.399083	4.199459e-13	4.123778

Table 4. Fifth Order scheme, $\sigma = -0.5$, $Pe = 0.1$



(a) Accuracy in h refinement for the SD scheme, 2^{nd} through 5^{th} order scheme.



(b) Accuracy in p refinement for the linear advection diffusion equation and the spectral difference scheme with $\sigma = -0.5$ and $Pe = 0.1$ at $N = 40$

Figure 7.

equations we can write the viscous fluxes as

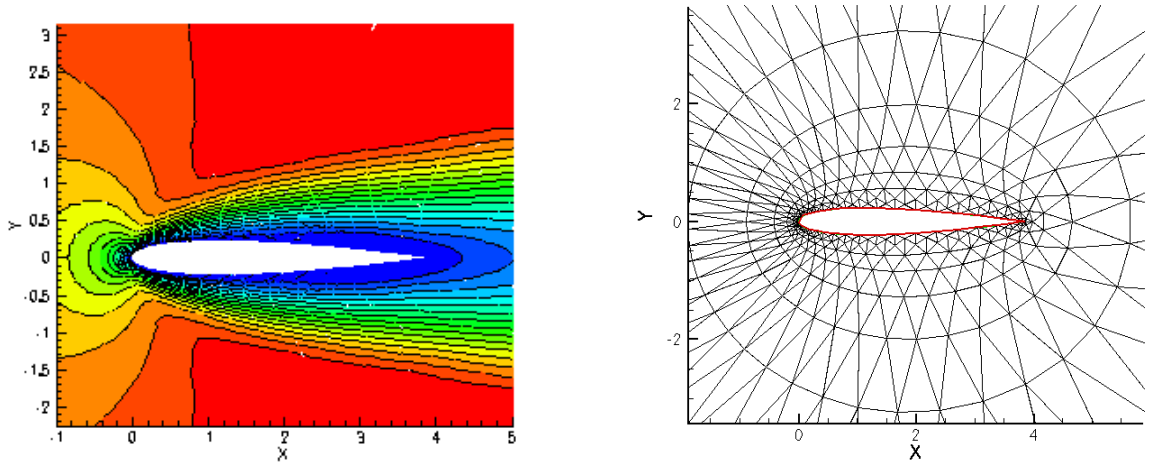
$$F_v = \nabla \cdot \mathbf{g}(u, v, q_1, q_2, q_3) \tag{43}$$
$$q_1 = \nabla u \tag{44}$$
$$q_2 = \nabla v \tag{45}$$
$$q_3 = \nabla T \tag{46}$$

where the velocity components are denoted by u and v, and T is the temperature. We can write for g

$$\mathbf{g} = (g, h) \qquad \text{where} \qquad g = \begin{pmatrix} \tau_{11} \\ \tau_{21} \\ u\tau_{11} + v\tau_{12} - \Xi_1 \end{pmatrix} \qquad h = \begin{pmatrix} \tau_{12} \\ \tau_{22} \\ u\tau_{21} + v\tau_{22} - \Xi_2 \end{pmatrix}, \tag{47}$$

where $\tau(q_1, q_2)$ is the rate-of strain tensor, eq. (33), and $\Xi = -\kappa q_3$ is the heat flux vector (κ is the thermal conductivity). Both are linear functions of the new variables q_i, and hence so is the the function $\mathbf{g}$. The ideas behind the discretization discussed above thus carry over to the Navier-Stokes equations. We introduce the weighted discretization first in the computation of the gradients q_i in eqns. (44) to (46) that form the rate of strain tensor and the heat flux vector, and subsequently a complementary discretization is used when discretizing the flux function $\mathbf{g}$ in a manner analogous to eq. (39) and eq. (42). Several techniques can be conceived to introduce the splitting, such as switching the sign of the coefficient σ based on the local normal velocity across a face, i.e. $q_n = v \cdot n$, where v is the velocity vector and n is the edge normal. This would in fact mimic the standard concept of upwinding. However, for the discretization of the viscous fluxes this splitting is not necessarily more meaningful than other heuristic techniques. The important property is the complementary character of the upwind/downwind bias, and the optimal treatment is very much an open problem. In the present work we use the free stream direction to split the discretization: Whenever $U_\infty \cdot n > 0$, where U_∞ is the free stream velocity vector and n is the edge normal pointing from the logical left cell to the right cell, upwind bias is introduced for the computation of the gradients, i.e. more weight is given to the left cell, while the opposite is true for the computation of the fluxes.

A subsonic low Reynolds number flow around the NACA 0012 profile serves as validation for the Navier Stokes equations. We have chosen the flow conditions $M = 0.3$ and $Re = 100$ at zero angle of attack. Fig. 8(a) shows a contour plot of the Mach number distribution computed with the 3^{rd} order SD scheme



(a) Contour lines of the Mach number for the third Order SD scheme

(b) Numerical grid with 960 triangles

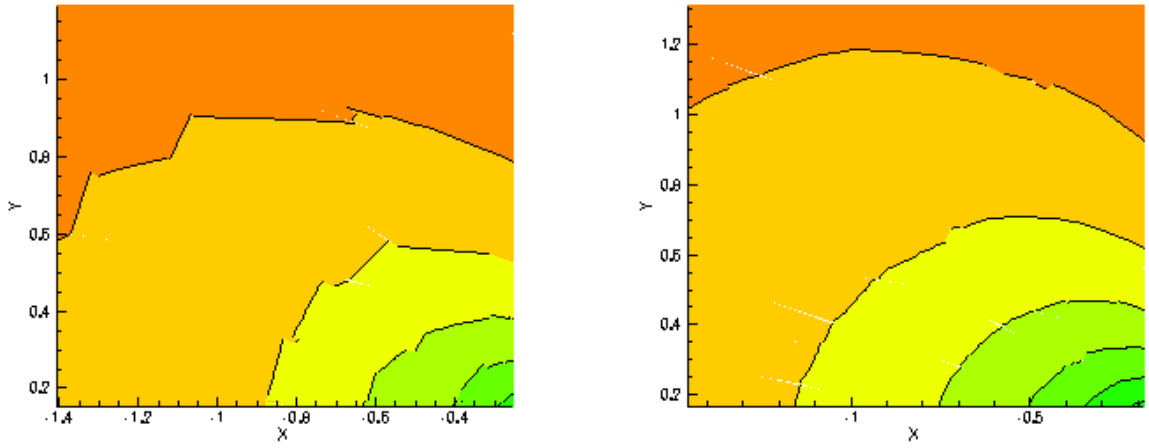
Figure 8. The NACA 0012 profile at $M = 0.3$, $Re = 100$, $\alpha = 0^o$

on a coarse triangular mesh with 960 elements. For the 3^{rd} order scheme this results in 5760 degrees of freedom (6 nodes to each triangle). The weighting parameter $\sigma = -0.5$ has been used, which results in a full splitting. The computational mesh is shown in figure 8(b).

We use discontinuous rendering of the solution for all high-order computations in this section, such as the one shown in fig. 8(a), which helps assess the quality of the solution. Each triangle is rendered individually which makes the contour lines discontinuous across elements, as is the solution. Fig. 9 on the next page illustrates this using a close-up view of a region upstream of the leading edge for the 2^{nd} and 3^{rd} order SD scheme. The highly curved streamlines can be seen to be discontinuous across the triangles for the 2^{nd} order scheme, but become almost continuous for 3^{rd} order computations. which suggests adequate resolution, even on this relatively coarse mesh (fig. 8(b)), for the 3^{rd} order scheme. For a 2^{nd} order finite-volume scheme this is not true, which is confirmed by fig. 10, where the same testcase is shown, computed on the same mesh (fig. 8(b)) with a standard cell-centered finite volume scheme using a linear reconstruction of the solution variables and a central discretization of the viscous terms, as documented in previous work[19,20] . It is clear that the mesh resolution is not adequate for a second order scheme. To compute a comparison for the 3^{rd} order SD solution the finite-volume scheme requires a finer mesh. A solution on a mesh with 10240 triangles is shown in figure 11 where a contour plot of the Mach number distribution is shown along with the computational mesh. Note that the finite volume scheme uses roughly twice as many degrees of freedom, compared to the 3^{rd} order SD scheme.

VI. Conclusion and Future Work

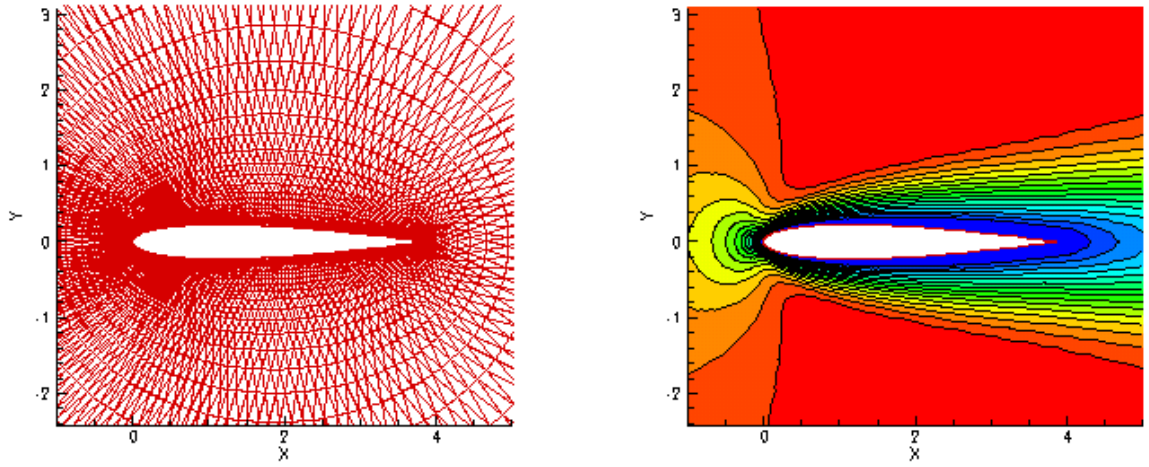
A high order methodology for the Euler and Navier-Stokes equations has been presented with emphasis on practical implementation, treatment of shock waves and viscous terms. The method has been tested



(a) Second Order SD scheme

(b) Third Order SD scheme

Figure 9. The NACA 0012 profile at $M = 0.3$, $Re = 100$, $\alpha = 0^o$. Contour lines of the Mach number. Close-up view of the solution near the leading edge



(a) Numerical grid with 10240 triangles

(b) Contour lines of the Mach number for the second order finite-volume scheme

Figure 11. The NACA 0012 profile at $M = 0.3$, $Re = 100$, $\alpha = 0^o$

on various testcases ranging from the linear advection-diffusion equation to the Euler and Navier-Stokes equations.

Good shock capturing capabilities have been demonstrated for both time-dependent and steady-state problems. For steady-state problems involving discontinuous solutions results are extremely rare in the literature. We are able to compute transonic steady flow on unstructured meshes using the SD scheme, for which adaptive mesh refinement is a valuable aid in confining the discontinuities to narrow regions in the flow field, where they can be captured using limited linear reconstruction. Convergence of the residuals to machine zero is the remaining goal for such testcases.

For viscous flow, superior resolution of the high-order SD scheme has been demonstrated, compared to standard finite-volume methods. Regarding the viscous discretization, further investigation of the optimal splitting and the influence of the weighting parameter σ is needed. This will be subject of further research.

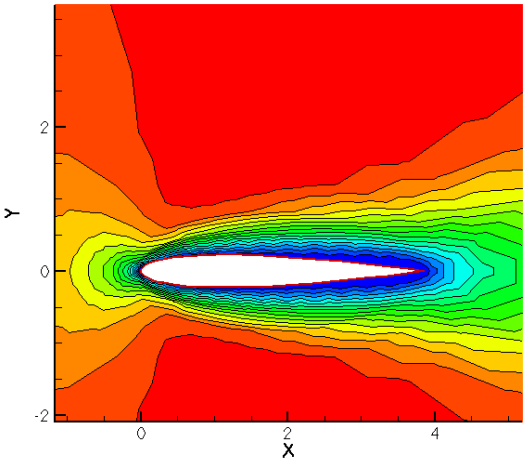


Figure 10. Cell-Centered Finite-Volume Scheme with linear reconstruction

In general, convergence acceleration is needed to make high-order methods competitive for steady-state problems. So far this has not been an objective of this work. While research in this area is taking place, for example multigrid techniques for high-order methods, the field is not very mature, in particular for transonic and viscous flow, and an increased research effort is necessary to address this problem.

VII. Acknowledgments

This research has been supported in part by the US Department of Energy as part of the Advanced Simulation and Computing (ASC) Program, and by Stanford University through a Graduate Student Fellowship.

References

[1]Liu, Y., Vinokur, M., and Wang, Z. J., "Discontinuous spectral difference method for conservation laws on unstructured grids," *Proceedings of the 3rd International Conference on Computational Fluid Dynamics, July 12-16, 2004, Toronto, Canada*, Springer, 2004.

[2]Wang, Z. J. and Liu., Y., "The Spectral Difference Method for the 2D Euler Equations on Unstructured Grids," Aiaa paper 2005-5112, 2005.

[3]May, G. and Jameson, A., "High-Order Accurate Methods for High-Speed Flow," Aiaa paper 2005-5251, 2005.

[4]Cockburn, B. and Shu, C. W., "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II: General Framework," *Math. Comp.*, Vol. 52, No. 186, 1988, pp. 411–435.

[5]Cockburn, B. and Lin, S. Y., "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws III: One Dimensional Systems," *J. Comp. Phys.*, Vol. 84, 1989, pp. 90–113.

[6]Z. J. Wang, L. Z. and Liu., Y., "High-Order Spectral Volume Method for 2D Euler Equations." Aiaa paper 2003-3534, 2003.

[7]Bassi, F. and Rebay, S., "A High-Order Accurate Discontinuous Finite-Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations," *J. Comp. Phys.*, Vol. 131, 1997, pp. 267–279.

[8]Cockburn, B. and Shu, S. W., "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," *SIAM. J. Numer. Anal.*, Vol. 35, No. 6, 1998, pp. 2440–2463.

[9]Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., "Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems," Report no. am233, Pennsylvania State University, http://www.math.psu.edu/ccma/reports.html, 2001.

[10]Jameson, A., "Analysis and design of numerical schemes for gas dynamcics 2: Artificial diffusion and discrete shock structure," *Int. J. Comp. Fluid. Dyn.*, Vol. 5, 1995, pp. 1–38.

[11]Kopriva, D. A., "A Conservative Staggered-Gri Chebyshev Multidomain Method for Compressible Flows II: Semi-Structured Method," *J. Comp. Phys.*., Vol. 128, 1996, pp. 475.

[12]Hesthaven, J. S., "From Electrostatics to Almost Optimal Nodal Sets for Polynomial Interpolation in a Simplex," *SIAM J. Num. Anal.*, Vol. 35, No. 2, 1998, pp. 655–676.

[13]LeVeque, R., *Finite-Voume Methods for Hyperbolic Problems*, Cambridge University Press, 2002.

[14]C. A. Kim, A. J., "A Robust and Accurate LED-BGK Solver on Unstructured Adaptive Meshes," *J. Comp. Phys.*., Vol. 143, 1998, pp. 598–627.

[15]Cockburn, B., Hou, S., and Shu, S. W., "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws IV: The Multidimensional Case," *Math. Comp.*, Vol. 54, No. 190, 1990, pp. 545–581.

[16]Cockburn, B. and Shu, S. W., "The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems," *Math. Comp.*, Vol. 141, 1998, pp. 199–224.

[17]Woodward, P. and Colella, P., "The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks," *J. Comp. Phys.*, Vol. 54, 1984, pp. 115–173.

[18]Lomtev, I. and Karniadakis, G. E., "A Discontinuous Galerkin Method for the Navier-Stokes Equations," *Int. J. Numer. Meth. Fluids*, Vol. 29, 1999, pp. 587–603.

[19]May, G., Srinivasan, B., and Jameson, A., "Three-Dimensional Flow Calculations on Arbitrary Meshes Using a Gas-Kinetic BGK Finite-Volume Method," Aiaa paper 2005-1397, 2005.

[20]May, G. and Jameson, A., "Improved Gaskinetic Multigrid Method for Three-Dimensional Computation of Viscous Flow," Aiaa paper 2005-5106, 2005.