

Revisiting the Least-Squares Procedure for Gradient Reconstruction on Unstructured Meshes

Dimitri J. Mavriplis

National Institute of Aerospace
144 Research Drive
Hampton, VA 23666

The accuracy of the least-squares technique for gradient reconstruction on unstructured meshes is examined. While least-squares techniques produce accurate results on arbitrary isotropic unstructured meshes, serious difficulties exist for highly stretched meshes in the presence of surface curvature. In these situations, gradients are typically under-estimated by up to an order of magnitude. For vertex-based discretizations on triangular and quadrilateral meshes, and cell-centered discretizations on quadrilateral meshes, accuracy can be recovered using an inverse distance weighting in the least-squares construction. For cell-centered discretizations on triangles, both the unweighted and weighted least-squares constructions fail to provide suitable gradient estimates for highly stretched curved meshes. Good overall flow solution accuracy can be retained in spite of poor gradient estimates, due to the presence of flow alignment in exactly the same regions where the poor gradient accuracy is observed. However, the use of entropy fixes, or the discretization of physical viscous terms based on these gradients has the potential for generating large but subtle discretization errors, which vanish in regions with no appreciable surface curvature.

Introduction

Current-day unstructured mesh aerodynamic production codes rely almost exclusively on formally second-order accurate discretizations. The two main approaches for achieving second-order accuracy involve centrally-differenced convective terms with added artificial dissipation^{1,2} and projection-evolution schemes using linearly extrapolated values based on gradient reconstruction.^{3,4} Other approaches include fluctuation splitting schemes,⁵ and streamwise upwind Petrov-Galerkin schemes,⁶ although these approaches have not seen widespread use in computational aerodynamics. Although artificial dissipation schemes and projection evolution schemes have different origins, the final discretizations are closely related. Consider the evaluation of a flux at a control volume interface, as depicted in Figure 1. The projection evolution scheme requires the solution of an approximate Riemann-solver at the interface. For example, the often-used Roe Riemann-solver can be written as:⁷

$$F_{ik} = F(u_L, u_R) = F_L(u) + F_R(u) + T|\Lambda|T^{-1}(u_L - u_R) \quad (1)$$

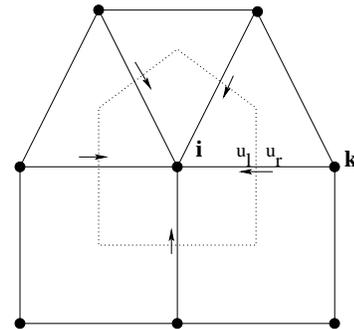


Fig. 1 Illustration of flux evaluation at control volume interface

where u_L and u_R represent the value of the flow variables at the left and right sides of the control volume interface. For a first-order scheme, these are simply taken as the values at the vertices corresponding to the control volume on either side of the face:

$$u_L = u_i \quad (2)$$

$$u_R = u_k \quad (3)$$

Copyright © 2003 by Dimitri J. Mavriplis. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

To obtain second-order accuracy, the left and right states must be obtained by extrapolating the control volume values based on a reconstructed gradient. Thus, the second-order accurate scheme is obtained using:

$$u_L = u_i + \nabla u_i \cdot \vec{r}_{if} \quad (4)$$

$$u_R = u_k + \nabla u_k \cdot \vec{r}_{kf} \quad (5)$$

where \vec{r}_{if} represents the position vector drawn from vertex i to the center point of the control volume interface. This formulation requires the evaluation of the gradients ∇u at the mesh vertices. These gradients may be evaluated using a Green-Gauss contour integration around the vertex-based control volumes, or by taking a least-squares approximation to the gradient at each vertex by constructing a tangent plane which best fits the surrounding neighboring data in some (weighted) least-squares sense.^{8,9} The Green-Gauss and least-squares constructions are outlined in the Appendix. The least-squares construction may include weights on the error terms, leading to different gradient approximations for non-linear functions. In all cases, the least-squares constructions represent a linear function exactly for vertex and cell-centered discretizations on arbitrary mesh types, while the Green-Gauss construction represents a linear function exactly only for a vertex-based discretization on simplicial elements (triangles or tetrahedra). Various construction techniques for the least-squares gradients have been proposed and discussed in the literature. In this work, a Gram-Schmidt construction,⁸ and a QR decomposition method^{9,10} have been implemented and tested. However, very little differences in the computed gradients has been observed between these two construction techniques, while much more important differences due to the choice of weights has been found, as will be shown in the paper.

Artificial dissipation schemes employ a central difference for the convective terms, and augment these quantities by a dissipative term which is required for stability. The flux at an interface for a first-order accurate artificial dissipation scheme can be written as:

$$F_{ik} = F_i(u) + F_k(u) + \alpha(u_i - u_k) \quad (6)$$

where α may be a scalar (scalar artificial dissipation) or a matrix (matrix artificial dissipation). In the case where α is a matrix, a natural choice for α , by analogy with equation (1) is:

$$\alpha = \kappa T|\Lambda|T^{-1} \quad (7)$$

where κ is a constant to be determined empirically. If κ is taken as unity, then the first-order accurate

matrix dissipation scheme becomes identical to the first-order accurate projection evolution scheme. On structured meshes, second-order accurate artificial dissipation schemes are obtained by replacing the first difference in equation (6) by a third difference.¹¹ On unstructured meshes, a second-order accurate artificial dissipation flux can be constructed as:

$$F_{ik} = F_i(u) + F_k(u) + \alpha (L_i(u) - L_k(u)) \quad (8)$$

where $L_i(u)$ represents an undivided Laplacian operator, taken as:

$$L_i(u) = \sum_{k=1}^{neighbors} (u_k - u_i) \quad (9)$$

resulting in an artificial dissipation term which is of the same order as a third difference. Thus, the second-order accurate matrix dissipation scheme can be obtained by replacing the difference of reconstructed states in the projection evolution scheme by a difference of undivided Laplacian operators. Although these quantities are of the same order, they are not directly proportional to each other, and therefore the parameter κ cannot be taken as unity in this case, but must be determined empirically. There are also discrepancies between the centrally differenced convective fluxes in both schemes, since these are evaluated at reconstructed states in the upwind scheme, rather than at vertex values as in the artificial dissipation scheme. However, numerical experiments reveal that these differences have virtually no effect on solution accuracy in the subsonic and transonic regimes.

The T matrices on the right hand side of equation (1) represent the eigenvectors associated with the linearization of the equations of inviscid compressible flow normal to the control volume face ik , while the $|\Lambda|$ matrix is a diagonal matrix containing the absolute values of the five eigenvalues associated with these equations. Of these five eigenvalues, three are repeated, leaving three distinct eigenvalues which are proportional to: u , $u+c$, $u-c$, where u is the velocity normal to the control volume face, and c is the speed of sound. When one of these eigenvalues vanishes, the dissipation for that component at that location also vanishes, which may lead to numerical instabilities. For this reason, it is common to limit the eigenvalues to a minimum fraction of the maximum eigenvalue, such as:

$$u = \text{sign}(u) * \max(|u|, \delta(|u| + c)) \quad (10)$$

$$u + c = \text{sign}(u + c) * \max(|u + c|, \delta(|u| + c)) \quad (11)$$

$$u - c = \text{sign}(u - c) * \max(|u - c|, \delta(|u| + c)) \quad (12)$$

where $|u| + c$ is the maximum eigenvalue, and δ is a parameter to be chosen empirically which varies between

0 and 1. When δ is taken as 0, no eigenvalue limiting is applied. When δ is taken as 1, the $|\Lambda|$ matrix reverts to a scaled identity matrix, since all eigenvalues are now taken as $|u| + c$, and the triple matrix product $T|\Lambda|T^{-1}$ reduces to a scalar quantity. For the artificial dissipation discretization, this constitutes the definition of the scalar artificial dissipation, i.e.

$$\alpha = \kappa * \max \text{ eigenvalue} \quad (13)$$

which can be computationally cheaper than requiring the evaluation of the full matrices. Small values of δ of the order of 0.1 are common in many production codes, and this process is often referred to as an *entropy fix*.

For flows with strong gradients, most notably in the vicinity of shock waves, the above second-order accurate formulations may lead to instabilities, and additional dissipative mechanisms are required. In the upwind scheme, these take the form of limiters applied to the computed gradients,³ while in the artificial dissipation schemes, the differences of undivided Laplacian operators is replaced by a blend of first differences and undivided Laplacian operators.^{1,12} In both cases, accuracy is reduced from first to second order locally in regions where this additional dissipation is required. For transonic flows with shocks of moderate strength, the use of limiters or additional dissipation is generally not required. For the purposes of the current study, we will confine ourselves to cases where no limiting or additional dissipation is employed.

Motivation

The motivation for the current study comes from the observed behavior of various unstructured mesh discretizations for a Reynolds-averaged Navier-Stokes solver on problems of aerodynamic interest. The viscous transonic flow over a wing-body configuration has been computed using the vertex-based unstructured mesh flow solver NSU3D¹³ using a matrix artificial dissipation discretization and an upwind scheme based on the unweighted least-squares gradient construction technique, and the sensitivity of the solution to the different discretizations has been investigated.¹⁴ The particular test case is taken from the 1st AIAA Drag prediction workshop.¹⁵ Figure 2 illustrates the mesh and a sample solution for a Mach number of 0.75, a Reynolds number of 3 million and $C_L = 0.6$. The mesh contains a total of 1.65 million vertices, and uses prismatic elements in the boundary layer regions and tetrahedral elements elsewhere. The solution is shown as computed surface pressure contours, illustrating the shock wave on the upper surface of the wing.

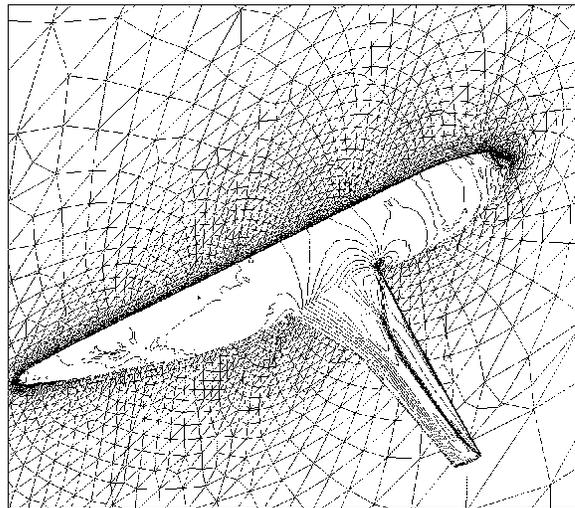


Fig. 2 Computed surface pressure contours for transonic flow over wing-body configuration. Mach=0.75, $C_L = 0.6$, Reynolds number = 3,000,000

Figure 3 depicts the computed lift and drag values using both the artificial dissipation discretization, with an entropy fix value of $\delta = 0.1$ and the least-squares upwind discretization scheme using the entropy fix value of $\delta = 0.0$ as well as $\delta = 0.1$.

The lift values produced by the upwind scheme with no entropy fix are slightly lower than those given by the matrix dissipation case. Comparing these differences with the increased lift values reported by the matrix dissipation on a finer grid of 13 million points, one can conclude that the least-squares based discretization is slightly more diffusive than the matrix dissipation discretization. Because the nominal value of the κ coefficient in the matrix dissipation scheme has been determined empirically, it is conceivable that a simple rescaling of the dissipation terms could be used to improve the accuracy in the upwind scheme as well. However, there are significant differences between these two discretizations which extend beyond the simple scaling of the final terms. When the entropy fix parameter for the artificial dissipation scheme is increased from $\delta = 0.0$ to $\delta = 0.1$, which is the level used in the baseline matrix dissipation settings, the results of the upwind scheme are now much different than in either baseline cases. The lift is reduced by over 20% and the drag values in the polar plot are substantially overpredicted. In essence, the accuracy of the upwind scheme has been completely compromised by this small value of the entropy fix. For the matrix dissipation scheme, previous studies¹⁴ have shown the solution accuracy to be insensitive to small values ($\delta = 0.1$ to 0.2) of the entropy fix parameter, while the scalar dissipation scheme ($\delta = 1.0$) achieves reasonable accuracy in lift with slight drag overprediction (≈ 25 counts) for the case shown above.¹⁴ The unexpected

sensitivity of the upwind scheme to small values of the entropy fix prompted the current investigation.

with a vanishing entropy fix agree closely, while the upwind scheme accuracy degrades severely when the value $\delta = 0.1$ for the entropy fix is used.

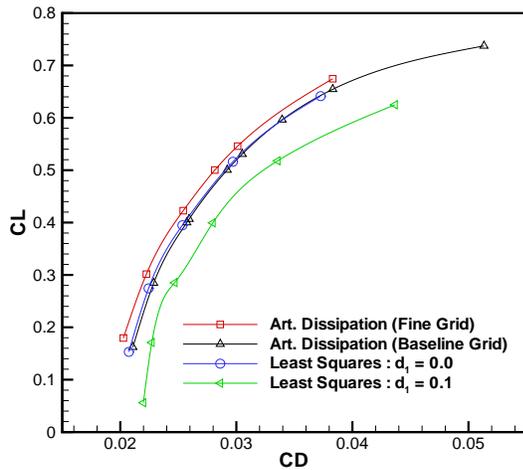
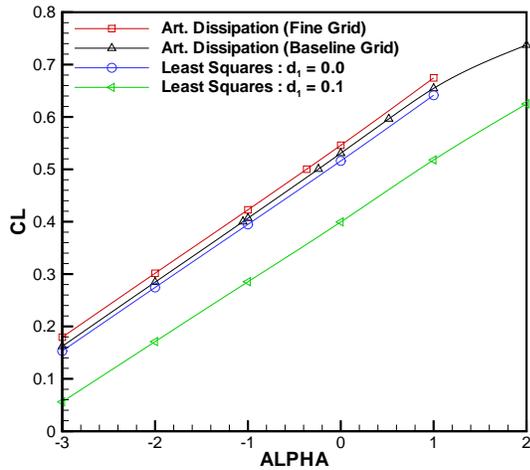


Fig. 3 Variations in computed lift and drag coefficients as a function of discretization for transonic flow over wing-body configuration

To better study this problem, we resort to a simpler two-dimensional example. Figure 4 illustrates the mesh for computations of viscous transonic flow about an RAE2822 airfoil at Mach=0.73, Reynolds=6.5 million, and an incidence of 2.31 degrees. The mesh contains a total of 16167 vertices, with the distance of the first point normal to the wall being 2.e-06 chords. Although the figure shows a fully triangular mesh, quadrilateral elements are employed by the solver in the boundary layer regions by removing the diagonal associated with pairs of stretched triangles. A flow solution behavior similar to that discussed for the three-dimensional example is observed in Figure 5. The solutions using the matrix artificial dissipation scheme and the least-squares based upwind scheme

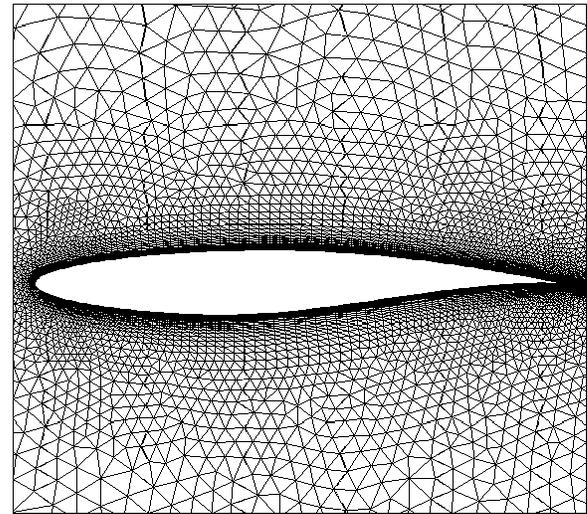


Fig. 4 Two-dimensional unstructured mesh for computation of transonic flow over RAE2822 airfoil

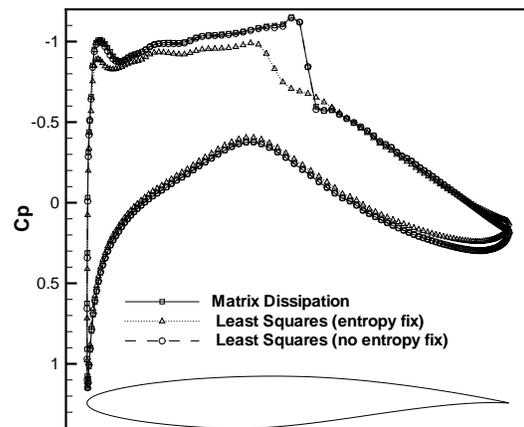


Fig. 5 Variations in computed surface pressure coefficients as a function of discretization for two-dimensional transonic airfoil problem. Mach=0.73, Incidence=2.31 degrees, Re = 6.5 million

Investigation

To investigate the cause of this accuracy degradation, we study the accuracy of the various gradient construction techniques. Clearly, all the employed gradient construction techniques produce the exact result for a linear function. Thus, we must investigate the accuracy of these constructions for non-linear functions,

and preferably functions which are representative of the types of gradients found in real flow simulations. We seek an analytic function for which an exact value of the gradient is available, against which the discrete gradient values can be compared. This is achieved using the distance function, which represents the distance from any given point in the plane to the nearest point on the airfoil surface. Contours of the distance function are shown in Figure 6. This represents a convenient choice, since the distance function has similar characteristics to boundary layer velocity gradients, (i.e. exhibits strong normal gradients, and vanishing streamwise gradients) and is readily available, since it is required in the formulation of the Spalart-Allmaras turbulence model.¹⁶ By definition, the gradient of the distance function in the direction normal to the surface is unity:

$$\nabla D \cdot \vec{n} = 1.0 \quad (14)$$

and the streamwise gradient vanishes. Thus, the norm of the derivative is given by:

$$\left(\frac{\partial D}{\partial x}\right)^2 + \left(\frac{\partial D}{\partial y}\right)^2 = 1.0 \quad (15)$$

Figure 7 illustrates the contours of the percent error between the unweighted least-squares computed gradient of the distance function and the exact value. For regions away from the leading and trailing edges, the maximum error is no more than 0.5%, illustrating the accuracy of this construction for nearly linear functions. Similar results are observed with the Green-Gauss gradient construction.

A more non-linear function is constructed by considering a quadratic variation in D:

$$F = (1 + \alpha D)^2 \quad (16)$$

At any point, the norm of the gradient in F is given by:

$$\|\nabla F\| = 2\alpha(1 + \alpha D) \left[\left(\frac{\partial D}{\partial x}\right)^2 + \left(\frac{\partial D}{\partial y}\right)^2 \right] \quad (17)$$

or

$$\|\nabla F\| = 2\alpha(1 + \alpha D) \quad (18)$$

following equation (15). For the purposes of this study, the value $\alpha = 200$ was used exclusively. This choice was motivated by the desire to minimize roundoff error problems for small values of D.

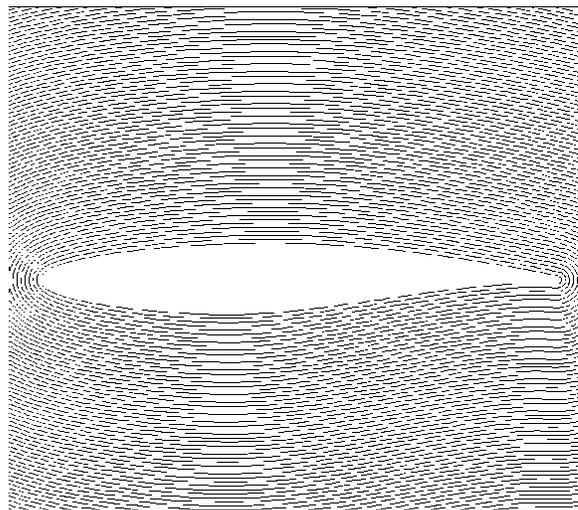


Fig. 6 Contours of distance function on grid of Figure 4

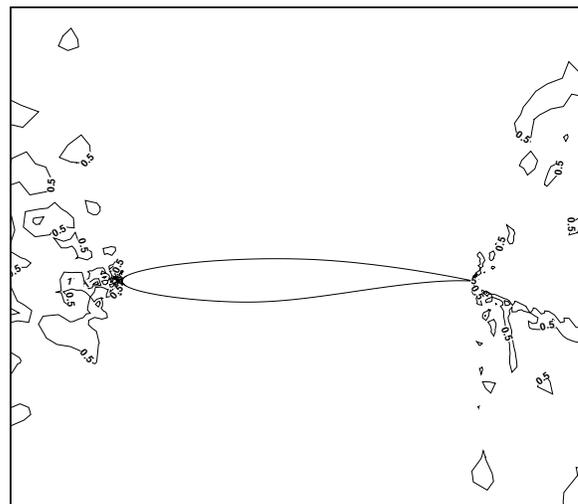


Fig. 7 Contours of percent error between computed and exact distance function gradient. Errors are below 0.5% except near the leading and trailing edges

In Figure 8, the ratio of the computed to the exact value of $\|\nabla F\|$ is plotted at the station $x=0.3$ on the airfoil, using the Green-Gauss gradient construction, and the unweighted least-squares gradient construction. Additionally, the values obtained from a weighted least-squares gradient construction (using inverse distance weighting) are shown, as well as the value of dF/dn obtained by finite difference along the normal grid line in the boundary layer region. The Green-Gauss and the finite-difference approach produce very accurate estimates of the gradient in all regions of the domain. However, the unweighted least-squares construction is seen to grossly underpredict

the gradient near the wall, and throughout a large inner portion of the boundary layer region. When inverse distance weighting is used in the least-squares approach, accuracy similar to that achieved by the other methods is recovered. Figure 9 shows the value of the gradient at the first grid point away from the airfoil surface (in the normal direction), plotted along the entire upper and lower surfaces of the airfoil. Once again, the gradient values are well predicted by all methods except the unweighted least-squares construction, which shows severe under-prediction and considerable scatter aft of the mid-chord location.

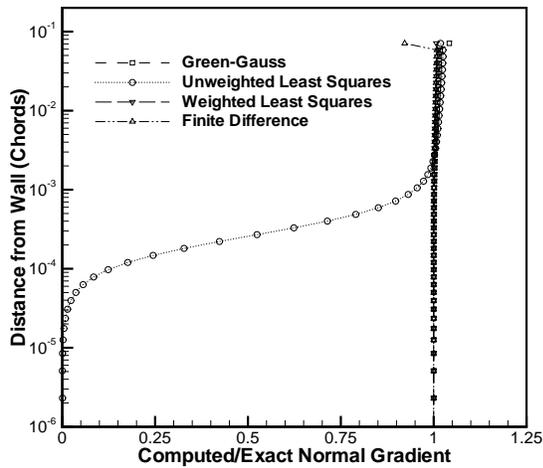


Fig. 8 Ratio of computed to exact gradient of function F using various methods along normal station at $x=0.3$ location on RAE2822 airfoil grid using quadrilateral elements in boundary layer region

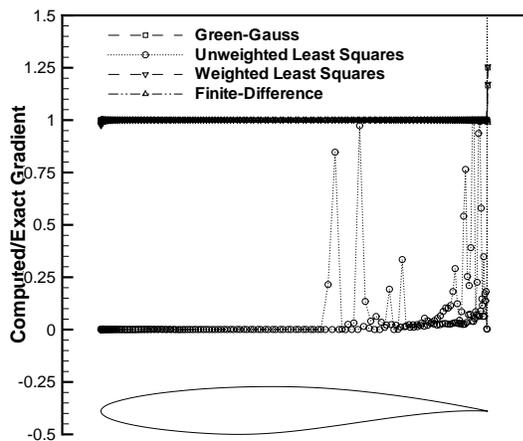


Fig. 9 Ratio of computed to exact gradient of function F at first interior grid point closest to wall along RAE2822 airfoil

To obtain insights into this behavior, we perform the same experiment on a flat plate geometry. The grid for this case is shown in Figure 10. The geometry consists of a flat plate with a rounded and tapered leading edge. Figure 11 illustrates the comparison between analytical and computed gradients at the first point off the wall as a function of streamwise coordinate. In this case, the unweighted least-squares gradients compare poorly with the exact values near the leading edge, but compare favorably over the downstream region of the plate. In fact, the sudden increase in accuracy for these gradients occurs precisely at the location where the plate surface becomes horizontal, or more importantly, where the surface curvature vanishes.

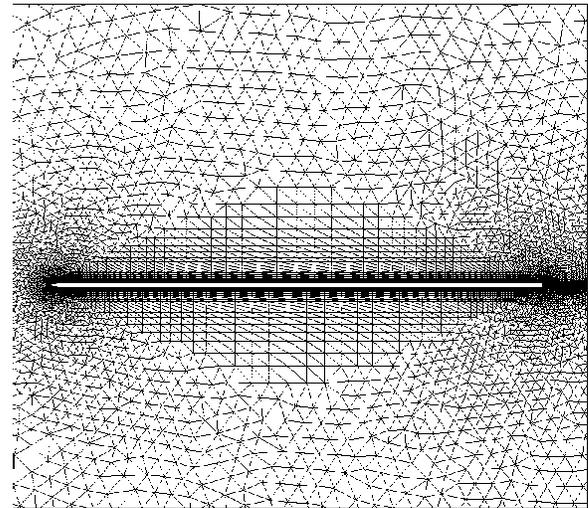


Fig. 10 Illustration of unstructured mesh employed for computation of viscous flow over flat plate with rounded and tapered leading edge

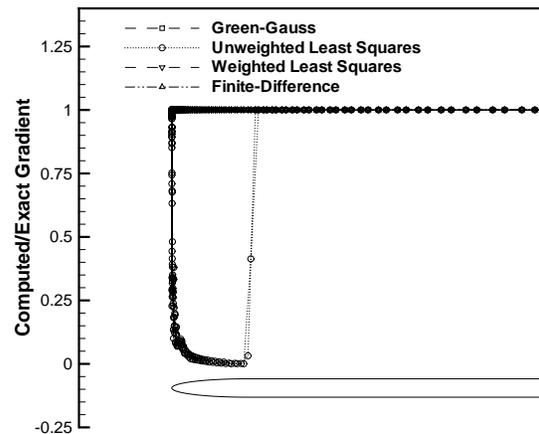


Fig. 11 Ratio of computed to exact gradient of function F at first interior grid point closest to wall along upstream portion of flat plate geometry

This provides an indication as to the mechanism subverting the accuracy of the unweighted least-squares gradient construction. This is illustrated in Figure 12 using the simplest possible configuration, i.e. a highly stretched quadrilateral mesh in the presence of surface curvature. Without loss of generality, we assume the surface normal at the station under consideration to be in the y-coordinate direction, and plot the topology using an expanded scale in the normal direction. Due to the surface curvature, the upstream and downstream neighbors are not aligned with the center point in the y-coordinate. While this y-direction variation is indeed very small (order 1.e-04 chords on the RAE 2822 mesh near the mid-chord location), it is nonetheless much larger than the small normal spacing used in the inner portion of the boundary layer region (2.e-06 chords). Therefore, for the unweighted least-squares gradient, these points exert a large influence on the determination of the normal gradient, in spite of the fact that they are much more distant from the point under consideration than the two neighboring values in the upper and lower y-direction. This is an unavoidable consequence of the use of an unweighted procedure, which treats all (neighboring) stencil points equally. Using the inverse distance weighting in the least-squares construction deemphasizes these distant upstream and downstream points, thus resulting in much more accurate gradients in such situations. Referring to Figure 12, the accuracy of the unweighted least-squares gradient is seen to break down when the normal grid spacing h becomes comparable to the distance H . Writing H as a function of the angle α yields the necessary condition for avoiding accuracy breakdown of the unweighted least-squares gradient:

$$h > R(1 - \cos\alpha) \quad (19)$$

or

$$h > \frac{s^2}{2R} \quad (20)$$

where s denotes the streamwise grid spacing, and the approximations $\cos\alpha \approx 1 + \frac{\alpha^2}{2} + \dots$ and $\alpha \approx \frac{s}{R}$ have been used. Considering a circle of unit radius, discretized with $O(10^2)$ circumferential points, equation (20) predicts poor gradient accuracy for normal grid spacing below $O(10^4)$. This rough estimate correlates well with the behavior observed in Figure 8 for the airfoil case.

The failure of the unweighted least-squares gradient construction is perhaps surprising because this method possesses many of the often sought-after properties for numerical schemes:

- It represents a linear function exactly, for arbitrary grid topologies.
- It has been shown to produce superior gradient estimates for highly irregular (but isotropic) meshes.

- It performs well for cases with no surface curvature, such as flat plate boundary layer cases, where most numerical investigations of viscous flow solvers are initiated.
- It has often been found to be more robust for viscous flows.

Although the combination of high mesh stretching with surface curvature may be considered pathological situations in some disciplines, such topologies are common-place for aerodynamic simulations and better gradient estimates are desirable.

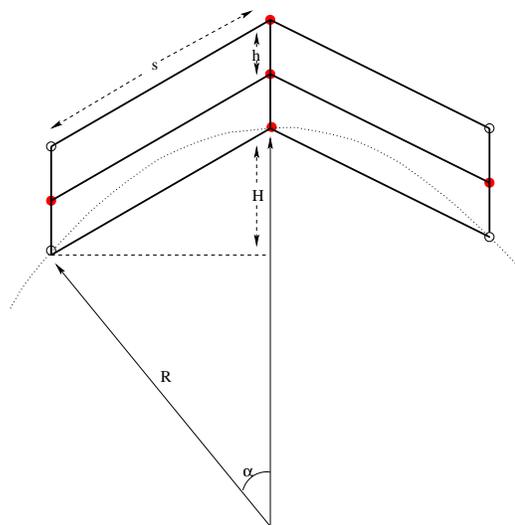


Fig. 12 Illustration of stencil for least-squares gradient calculation on quadrilateral mesh in the presence of stretching and surface curvature

Effects with Alternate Discretizations

The above discussion was confined to vertex-based schemes operating on prismatic (3D) or quadrilateral (2D) element meshes in the boundary layer region. In this section we examine the suitability of the various gradient construction methods for vertex-based discretizations on triangular boundary layer meshes, and for cell-centered discretizations using fully triangular or mixed triangular and quadrilateral meshes. Figure 13 illustrates the topology of the least-squares stencil for a vertex-based triangular mesh, and Figure 14 depicts the estimates of the gradient of the function F produced by the various methods. In this case, the stencil is augmented by two additional points joined by the triangle diagonals, which are at upstream and downstream locations from the point under consideration. These additional points are similar in character to the upstream and downstream points obtained from the quadrilateral stencil, and thus both the unweighted and weighted least-squares methods retain similar performance on triangular meshes as on quadrilateral meshes. Additionally, the Green-Gauss approach is

seen to yield similar results on triangular meshes as in the previous case on quadrilateral meshes. Note that the Green-Gauss construction is exact for linear functions on triangular meshes only, although this does not appear to have any appreciable effect on the accuracy of the results shown in Figure 14.

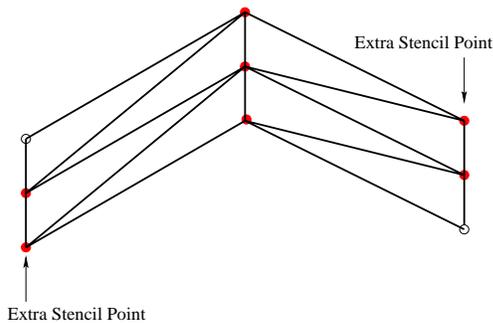


Fig. 13 Illustration of stencil for least-squares gradient calculation on triangular mesh in the presence of stretching and surface curvature

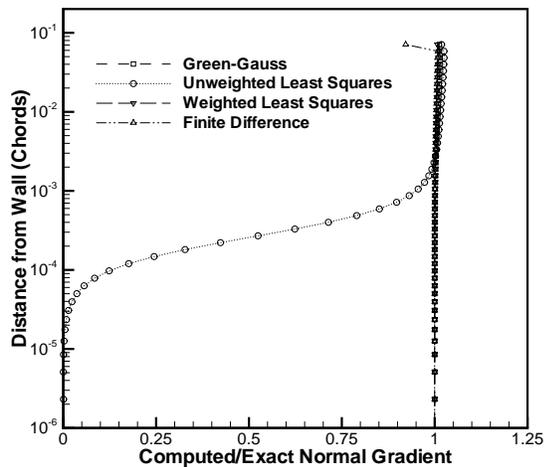


Fig. 14 Ratio of computed to exact gradient of function F using various methods along normal station at $x=0.3$ location on fully triangular RAE2822 airfoil grid

For cell-centered schemes operating on quadrilateral meshes, the stencil is topologically similar to that of a vertex scheme operating on a quadrilateral mesh, as shown in Figure 15. Thus similar behavior for the various gradient construction methods can be expected.

For a cell-centered method operating on a fully triangular mesh, the stencil topology is shown in Figure 16. In all cases, a stencil with only three neighbors is obtained, and none of these neighbors are located close (within the order of a cell normal height) to the cell center under consideration. Hence, it is not surprising that the unweighted least-squares gradient exhibits poor accuracy in the boundary layer region, similarly to the vertex discretization cases. However, inverse distance weighted least-squares construction also exhibits poor accuracy in these cases, as shown in Figures 17 and 18, since there are no close points to provide accurate normal derivative information. For cell-centered discretizations, the Green-Gauss gradient construction generally will not produce the exact value for a linear function. This is only achieved if the segments joining neighboring cell centroids bisect the mesh edges, which is generally not achieved.³ For the function F , the gradient values are either over-predicted or under predicted by roughly 10 % depending on the orientation of the triangle diagonal, as seen by the oscillatory behavior in the plot of Figure 18. (Only one branch of these two triangle types is plotted in Figure 17.) The average of these two triangle estimates closely approximates the exact gradient value, which is equivalent to performing the integral around the quadrilateral formed by the union of the two constituent triangles. In spite of this shortcoming on triangular meshes, the Green-Gauss gradients are seen to provide superior estimates of the gradients of F in the boundary layer regions to the least-squares methods. This illustrates the danger of relying on simple properties such as exact representation of linear functions for accuracy certification.

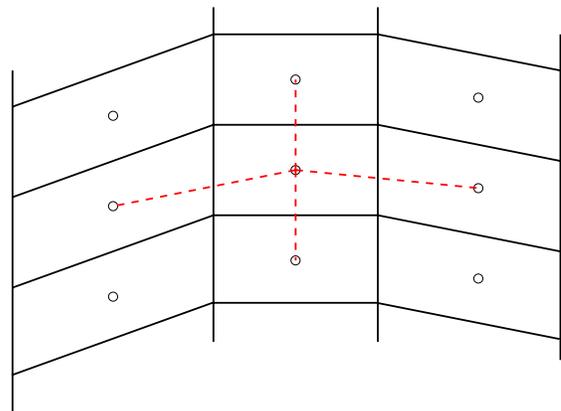


Fig. 15 Illustration of stencil for least-squares gradient calculation for cell-centered discretization on quadrilaterals

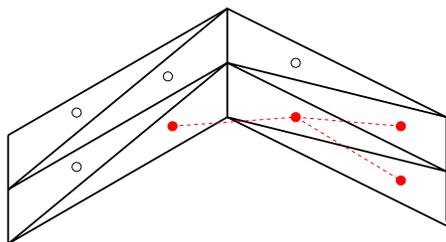


Fig. 16 Illustration of stencil for least-squares gradient calculation on cell-centered discretization on triangular mesh

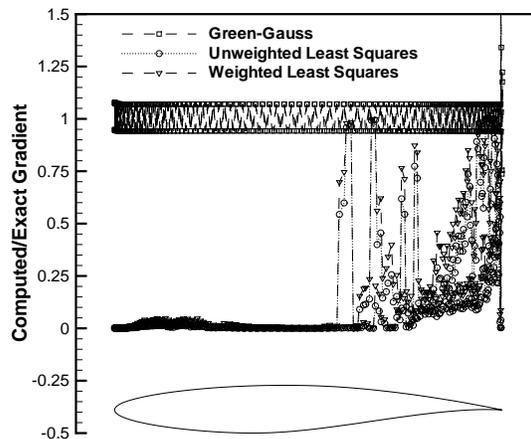


Fig. 18 Ratio of computed to exact gradient of function F at second layer of triangles off wall for cell-centered discretization on fully triangular RAE2822 airfoil grid

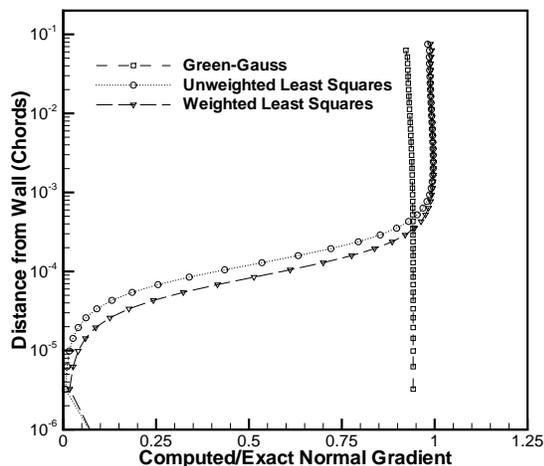


Fig. 17 Ratio of computed to exact gradient of function F along normal station at $x=0.3$ location for cell-centered discretization on fully triangular RAE2822 airfoil grid

Alternate techniques for constructing gradients on cell centered simplicial discretizations have been developed.¹⁷⁻²⁰ In one of these approaches,^{19,20} vertex values are obtained by averaging surrounding cell-centroidal values, often using a weighting factor, and the cell based gradient is then computed using a Green-Gauss contour integral using the constructed vertex values. The technique described in^{17,18} extrapolates gradients computed on neighboring triangles using a Green-Gauss contour integration. The performance of these strategies for highly stretched meshes in the presence of curvature has not been studied in this work, but warrants further investigation.

Effect on Solution Accuracy

While we have pointed out inadequacies in the unweighted least-squares gradient formulation, the fact remains that in the absence of any entropy fix, upwind discretizations based on this approach achieve good overall accuracy, as evidenced by the results in Figures 3 and 5. It may seem perplexing how one can obtain a viable solution with such poor gradient estimates in the inner boundary layer region, and why this solution is so sensitive to small values of the entropy fix parameter. The answer lies in the alignment of the grid with the flow direction in the boundary layer region. The use of a highly stretched mesh aligned with the wall direction in boundary layer regions, which is commonplace for high-Reynolds number flow simulations, results in near vanishing flow velocity normal to the control volume interfaces in this direction. This is shown in Figure 19, where the computed normal and streamwise flow velocities are plotted along the

normal station at $x=0.3$ for the transonic airfoil flow solution using matrix dissipation depicted in Figure 5. This plot indicates that the normal velocity is two to three orders of magnitude smaller than the streamwise velocity throughout the inner portion of the boundary layer. In Figure 20, the normal and streamwise convective eigenvalues are plotted at the same station. The convective eigenvalue is defined as the integrated velocity flux through the control volume face. Due to the high cell aspect ratios, the normal control volume face is much larger than the streamwise face, and the normal eigenvalue becomes substantially larger than the streamwise value in the inner boundary layer region. Thus, in spite of decoupling through flow alignment, the high grid cell aspect-ratio ensures strong coupling between neighboring normal points in the boundary layer, even for the convective modes. However, the overall dissipation terms are formed by the product of the eigenvalue with the jump in left and right flow variables across the control volume face. Assuming (as a worst case scenario) a first order variation in the flow variables, the normal (streamwise) dissipation terms scale as the product of the normal (streamwise) eigenvalue and the normal (streamwise) grid spacing, with this latter quantity being of the same order as the streamwise (normal) control volume face. Thus the overall scaling of the dissipation terms is closely approximated by Figure 19, which implies much lower diffusion in the normal direction as compared to the streamwise direction. The application of an entropy fix places a lower limit on the velocity values used in scaling the dissipation terms, which from Figure 19 can be seen to have a large effect on the normal velocity values.

It is interesting to note that, although the dissipation terms associated with the normal velocity are small, the two acoustic wave eigenvalues associated with $u+c$ and $u-c$ are not affected by flow alignment, and yet good accuracy is retained despite the use of inaccurate gradients for the dissipative terms associated with the acoustic waves. On the other hand, the use of more accurate gradient estimates resolves the loss of accuracy for small values of the entropy fix parameter. In Figure 21, the transonic airfoil flow case has been recomputed using the upwind scheme with an inverse-distance weighted least-squares gradient construction, using a vanishing entropy fix, as well as an entropy fix parameter value of $\delta = 0.1$. The computed surface pressures in both cases compare well with each other and agree closely with those produced by the matrix artificial dissipation scheme, illustrating the superior characteristics of the weighted versus the unweighted least-squares construction.

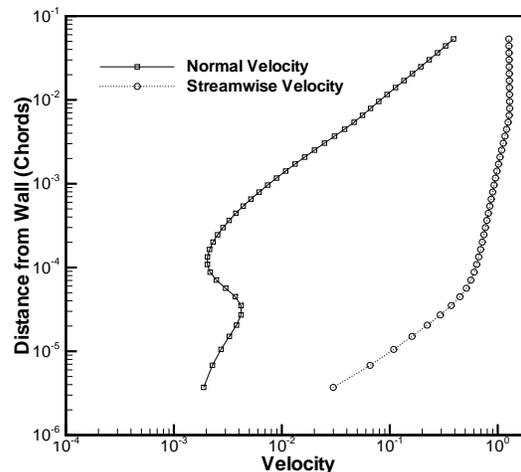


Fig. 19 Magnitudes of normal and streamwise velocities along normal station at $x=0.3$ for transonic airfoil flow solution

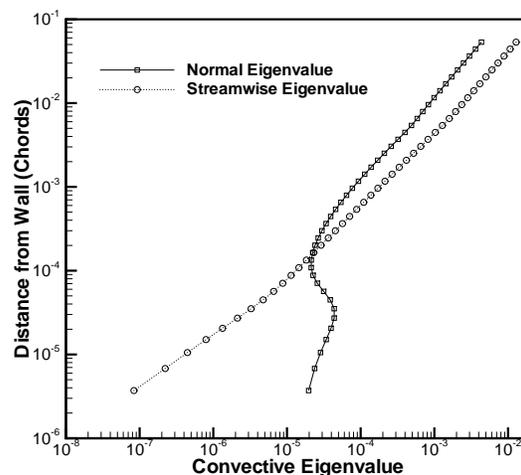


Fig. 20 Magnitudes of normal and streamwise convective eigenvalues along normal station at $x=0.3$ for transonic airfoil flow solution

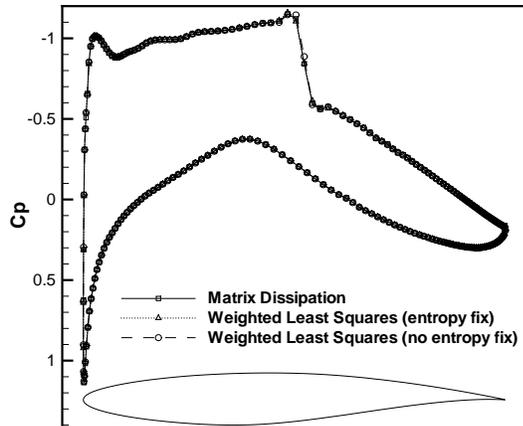


Fig. 21 Variations in computed surface pressure coefficients for weighted least-squares gradient construction as a function of discretization for two-dimensional transonic airfoil problem. Mach=0.73, Incidence=2.31 degrees, Re = 6.5 million

Implications

In the above discussion, we have demonstrated how and why the unweighted least-squares gradient construction severely under-estimates normal gradients for highly stretched meshes in the presence of surface curvature. Furthermore, it has been shown that this behavior can be expected for vertex based discretizations operating on triangular and quadrilateral meshes (tetrahedral and prismatic meshes in 3D) and for cell centered discretizations on either types of meshes as well. The use of inverse distance weighting in the least-squares construction can be used to recover good accuracy in these situations for vertex and cell centered discretizations on quadrilateral meshes, and for vertex discretizations on triangular meshes. However, this technique is not effective for cell centered discretizations on triangular meshes. The Green-Gauss construction technique produces adequate gradient estimates in all cases, even for cell centered discretizations where it may not represent linear functions exactly.

On the other hand, the failure of the un-weighted least-squares gradient is mitigated by the phenomenon of flow alignment in precisely the same locations, thus enabling adequate overall accuracy to be achieved. Therefore, the least-squares gradient construction can be used competitively for producing accurate solutions, but the user must be aware of the limitations of this approach: notably that no entropy fix be used, and that the mesh be well aligned with the viscous surfaces, and thus with the boundary layer flow direction. Alternately, mesh cell aspect ratio constraints based on equation (20) may be enforced.

Discretization of the physical viscous terms in the Navier-Stokes equations is often achieved in a two pass approach where the flow gradients are evaluated in the first pass, and then used in the second pass to build up these terms. Clearly, the use of unweighted least-squares gradients in the construction of these viscous terms has the potential to generate large discretization errors overall. However, the deficiency of this approach may be extremely subtle in that it will not at all manifest itself for flat plate boundary layer calculations, where most viscous flow solvers are initially validated, but only in the presence of bodies with non-negligible surface curvature.

Finally, a more prudent strategy would appear to be one which employs inverse distance weighted least-squares gradients or even Green-Gauss gradients in the discretization of convective and viscous terms. However, these approaches have proven to be substantially less robust than upwind schemes based on unweighted least-squares gradients, and often require gradient limiting to achieve stable solutions, which in turn, may have an adverse impact on accuracy. While it was initially argued that this was due to superior approximation properties of the unweighted least-squares approach especially for irregular meshes, it should now be evident that the main reason for the robustness of this approach can be attributed to the use of under-predicted gradients, effectively using limited gradients which correspond to a first order scheme in the inner part of the boundary layer.

An alternative to the inadequacies of the unweighted approach, and the poor robustness of the weighted approach, is to resort to different gradient formulations, such as those described in,¹⁷⁻²⁰ or to employ one-dimensional reconstruction and limiting in the normal direction in the boundary layer, analogous to structured mesh techniques, although this incurs obvious data-structure drawbacks for an unstructured mesh approach. Future work will investigate the accuracy and robustness of various such discretizations for both vertex-based and cell centered approaches.

Appendix

The Green-Gauss formulation constructs gradients by integrating around the boundary of a closed control-volume. From Green's theorem, the average gradients over a control volume can be written as:

$$(u_x)_i = \frac{1}{Vol_i} \int \int u_x dV = \frac{1}{Vol_i} \oint u dy \quad (21)$$

$$(u_y)_i = \frac{1}{Vol_i} \int \int u_y dV = \frac{-1}{Vol_i} \oint u dx \quad (22)$$

where Vol_i represents the area of the two-dimensional control volume. These contour integrals are then discretized as:

$$(u_x)_i = \frac{1}{Vol_i} \sum_{k=1}^N \frac{u_i + u_k}{2} \Delta y_{ik} \quad (23)$$

$$(u_y)_i = \frac{-1}{Vol_i} \sum_{k=1}^N \frac{u_i + u_k}{2} \Delta x_{ik} \quad (24)$$

where the x and y subscripts denote differentiation, and i and k identify the associated control volume. For vertex schemes, the median dual control volumes are employed, as depicted in Figure 1. In this case, i and k refer to the vertices on either side of the control volume face, and Δx_{ik} and Δy_{ik} denote the increments of x and y along the control volume face. For cell-centered schemes, the grid cells themselves form the control volumes. In this case, i and k refer to the cells on either side of a mesh edge, and Δx_{ik} and Δy_{ik} denote the increments of x and y along the mesh edge. The Green-Gauss formulation is exact for vertex discretizations of linear functions only on triangular elements. For cell-centered discretizations, this formulation is generally not exact for linear functions on quadrilaterals or triangles. In the special case where the segments joining neighboring triangle cell centers exactly bisect the shared mesh edge, the formulation becomes exact for linear functions on triangles.

The least-squares gradient construction is a technique which is unrelated to the mesh topology. This construction relies on a stencil which identifies relevant neighboring points for use in the gradient estimation. Although this stencil can be chosen arbitrarily, the most obvious construction for mesh-based data is to choose the stencil of nearest neighboring values. For a vertex-based discretization, the stencil is thus formed by the set of mesh edges incident on the considered vertex i . For a cell-centered discretization, the stencil is formed by the edges joining neighboring cell centroids, which corresponds to the dual graph of the mesh, as shown in Figures 15 and 16. The least-squares gradient construction is obtained by solving for the values of the gradients which minimize the sum of the squares of the differences between neighboring values $k = 1, N$ and values extrapolated from the point i under consideration to the neighboring locations. The objective to be minimized is given as

$$\sum_{k=1}^N w_{ik}^2 E_{ik}^2 \quad (25)$$

The error is given as

$$E_{ik}^2 = (-du_{ik} + (u_x)_i dx_{ik} + (u_y)_i dy_{ik})^2 \quad (26)$$

where du_{ik} represents the difference $u_k - u_i$, with similar expressions for dx_{ik} and dy_{ik} , and w_{ik} is a weighting factor. Dropping the i subscripts on the gradients for clarity, a system of two equations for the two gradients u_x and u_y is obtained by solving the minimization problem i.e.

$$\frac{\partial \sum_{k=1}^N w_{ik}^2 E_{ik}^2}{\partial u_x} = 0 \quad (27)$$

and

$$\frac{\partial \sum_{k=1}^N w_{ik}^2 E_{ik}^2}{\partial u_y} = 0 \quad (28)$$

Using straight-forward algebra, the following set of equations is obtained

$$a_i u_x + b_i u_y = d_i \quad (29)$$

$$b_i u_x + c_i u_y = e_i \quad (30)$$

where

$$a_i = \sum_{k=1}^N w_{ik}^2 dx_{ik}^2 \quad (31)$$

$$b_i = \sum_{k=1}^N w_{ik}^2 dx_{ik} dy_{ik} \quad (32)$$

$$c_i = \sum_{k=1}^N w_{ik}^2 dy_{ik}^2 \quad (33)$$

$$d_i = \sum_{k=1}^N w_{ik}^2 du_{ik} dx_{ik} \quad (34)$$

$$e_i = \sum_{k=1}^N w_{ik}^2 du_{ik} dy_{ik} \quad (35)$$

In practice, all the above terms can be precomputed and stored, since these are only a function of the grid metrics. The above system of equations for the gradients is then easily solved using Cramer's rule. Note that the determinant of this system is given by:

$$DET = ac - b^2 \quad (36)$$

For the unweighted case ($w_{ik} = 1$), the determinant corresponds to a difference in quantities of the order $O(dx^4)$, which may lead to ill-conditioned systems. This may be the motivation for investigations into alternate solution techniques for the least-squares construction, such as the QR factorization method advocated in.^{9,10} For the flows computed in this work, very little difference in the calculated gradients was observed between these methods. Note that when inverse distance weighting is used ($w_{ik} = \frac{1}{\sqrt{dx_{ik}^2 + dy_{ik}^2}}$), the determinant scales as $O(1)$, and the system is much better conditioned.

References

- ¹A. Jameson, T. J. Baker, and N. P. Weatherill. Calculation of inviscid transonic flow over a complete aircraft. AIAA Paper 86-0103, January 1986.
- ²D. J. Mavriplis. A three-dimensional multigrid Reynolds-averaged Navier-Stokes solver for unstructured meshes. *AIAA Journal*, 33(3):445-4531, March 1995.
- ³T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. AIAA Paper 89-0366, January 1989.
- ⁴W. K. Anderson. A grid generation and flow solution method for the Euler equations on unstructured grids. *Journal of Computational Physics*, 110(1):23-38, January 1994.
- ⁵H. Deconinck, H. Paillere, R. Struijs, and P. L. Roe. Multidimensional upwind schemes based on fluctuation-splitting of conservation laws. *Comp. Mechanics*, 11(5/6):323-340, 1993.
- ⁶T. J. R. Hughes. Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier-Stokes equations. *Intl. J. for Numer. Meth. in Fluids*, 7:1261-1275, 1987.

⁷P. L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comp. Phys.*, 43(2):357–372, 1981.

⁸T. J. Barth. Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes. AIAA Paper 91-0721, January 1991.

⁹A. Haselbacher and J. Blazek. On the accurate and efficient discretization of the Navier-Stokes equations on mixed grids. In *Proceedings of the 14th AIAA CFD Conference, Snowmass, CO*, pages 946–956, June 1999. AIAA Paper 99-3363-CP.

¹⁰W. K. Anderson and D. L. Bonhaus. An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Computers Fluids*, 23(1):1–21, 1994.

¹¹A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. AIAA Paper 81-1259, 1981.

¹²D. J. Mavriplis. *Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes*. PhD thesis, Princeton University, MAE Department, 1987.

¹³D. J. Mavriplis and D. W. Levy. Transonic drag prediction using an unstructured multigrid solver. AIAA-Paper 2002-838, January 2002.

¹⁴D. J. Mavriplis. Aerodynamic drag prediction using unstructured mesh solvers. In *Lecture notes from the VKI Lecture Series on CFD-Based Drag Prediction and Reduction*, February 2003.

¹⁵D. W. Levy, T. Zickuhr, J. Vassberg, S. Agrawal, R. A. Wahls, S. Pirzadeh, and M. J. Hensch. Summary of data from the first AIAA CFD drag prediction workshop. AIAA Paper 2002-0841, January 2002.

¹⁶P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *La Recherche Aéronautique*, 1:5–21, 1994.

¹⁷J. A. Desideri and A. Dervieux. Compressible flow solvers using unstructured grids. *VKI Lecture Series*, 1988-05:1–115, 1988.

¹⁸A. Jameson. Analysis and design of numerical schemes for gas dynamics I, artificial diffusion, upwind biasing, limiters and their effect on multigrid convergence. *Int. J. of Comp. Fluid Dyn.*, 4:171–218, 1995.

¹⁹N. T. Frink. Recent progress toward a three-dimensional unstructured flow solver. AIAA Paper 94-0061, January 1994.

²⁰R. D. Rausch, J. T. Batina, and H. T. Y. Yang. Spatial adaptation of unstructured meshes for unsteady aerodynamic flow computations. *AIAA J.*, 30(5):1243–1251, 1992.