Quasi-ENO Schemes for Unstructured Meshes Based on Unlimited Data-Dependent Least-Squares Reconstruction

Carl F. Ollivier-Gooch
Mathematics and Computer Science Division
Argonne National Laboratory

Abstract

A crucial step in obtaining high-order accurate steady-state solutions to the Euler and Navier-Stokes equations is the high-order accurate reconstruction of the solution from cell-averaged values. Only after this reconstruction has been completed can the flux integral around a control volume be accurately assessed.

In this work, a new reconstruction scheme is presented that is conservative, uniformly accurate with no overshoots, easy to implement on arbitrary meshes, has good convergence properties, and is computationally efficient. The new scheme, called DD-L_2, uses a data-dependent weighted least-squares reconstruction with a fixed stencil. The weights are chosen to strongly emphasize smooth data in the reconstruction. Because DD-L_2 is designed in the framework of k-exact reconstruction, existing techniques for implementing such reconstructions on arbitrary meshes can be used. The new scheme satisfies a relaxed version of the ENO criteria. Local accuracy of the reconstruction is optimal except in the case on functions that are continuous but have discontinuous low-order derivatives. The total variation of the reconstruction is bounded by the total variation of the function to within $\mathcal{O}(\Delta x)$.

The asymptotic behavior of the scheme in reconstructing smooth and piecewise smooth functions is demonstrated. DD-L_2 produces uniformly high-order accurate reconstructions, even in the presence of discontinuities. Two-dimensional flow solutions obtained using DD-L_2 reconstruction are compared with solutions using limited least-squares reconstruction. The solutions are virtually identical. The absence of a limiter reduces the CPU time required for DD-L_2 solutions by 15-20% as compared to limited reconstruction, even though the DD-L_2 gradient computation is slightly more expensive than ordinary least-squares reconstruction.

1 Introduction

A crucial step in obtaining high-order accurate steady-state solutions to the Euler and Navier-Stokes equations is the high-order accurate reconstruction of the solution from cell-averaged values. Only after this reconstruction has been completed can the flux integral around a control volume be accurately assessed. Ideally, the reconstruction should conserve the mean value of the function in each cell, be uniformly accurate with no overshoots, be easy to implement on arbitrary meshes, have good convergence properties, and be computationally efficient.

Both limited k-exact reconstruction and essentially non-oscillatory (ENO) reconstruction have many of these properties. For example, design criteria for k-exact reconstruction given by Barth and Frederickson [1] ensure conservation of the mean, good accuracy for smooth functions, and computational efficiency. A fixed stencil is used, allowing precomputation of certain purely geometric quantities used in the reconstruction, with a corresponding improvement in efficiency. Accuracy near discontinuities is poor, however, and limiting is required to prevent overshoots of $\mathcal{O}(1)$. Limiters that retain good convergence properties (e.g., [2]) are often computationally expensive.

By design, the ENO reconstruction schemes of Harten et al. [3, 4, 5] conserve the mean, are uniformly accurate at all points for which a smooth neighborhood exists, and guarantee that overshoots will be no larger than the order of the truncation error of the reconstruction. Uniform high-order accuracy is obtained by using reconstruction stencils that vary in both space and time. Unfortunately, such stencils can hamper convergence to steady state. Also, ENO schemes are not easily implemented on unstructured meshes [6, 7].

In this work, a new reconstruction scheme is presented that has the best properties of both limited k-exact reconstruction schemes and ENO schemes. The new scheme, called DD-L_2, uses a data-dependent least-squares weighted reconstruction with a fixed stencil. The weights are chosen to strongly emphasize smooth data in the reconstruction. Because DD-L_2 is designed in the framework of k-exact reconstruction, existing techniques for implementing such reconstructions on arbitrary meshes [1, 8] can be used with only slight modification. The new scheme satisfies a relaxed version of the ENO criteria. Local accuracy of the reconstruction is optimal except in the case of functions that are continuous but have discontinuous low-order derivatives. The total variation of the reconstruction is bounded by the total variation of the function to within $\mathcal{O}(\Delta x)$.

Sections 2 and 3 describe least-squares reconstruction and reconstruction-by-primitive (RP) ENO schemes, respectively. Section 4 presents the new data-dependent least-squares reconstruction and discuss its properties. Section 5 shows results for reconstruction of smooth and nonsmooth functions, providing numerical evidence for the properties of the DD-L_2 reconstruction. In Section 6, some sample flow solutions are shown, with attention paid to accuracy, steady-state convergence behavior, and computational expense. Section 7 revisits the data-dependent reconstruction and describes a weighting scheme that dramatically improves the performance of DD-L_2 reconstruction for functions that are $\mathcal{C}^0$ but not $\mathcal{C}^1$. Finally, Section 8 presents some conclusions and directions for future work.

2 k-exact Least-Squares Reconstruction

The fundamental feature of k-exact reconstruction schemes is the exact reconstruction of polynomials of degree no more than k [1]. Smooth functions that cannot be described by k^{th}-order polynomials are reconstructed with a local error of $\mathcal{O}\left(\Delta x^{k+1}\right)$. For example, if $k = 1$, the reconstruction will exactly reproduce linear functions and make no local errors larger than $\mathcal{O}\left(\Delta x^2\right)$ for smooth functions. Such reconstructions can be designed to have compact support, conserve the mean, and be implemented efficiently on arbitrary meshes.

Mathematically, if the reconstruction is exact, then for all vertices i in the support stencil for a vertex j,

$$\begin{aligned} \Delta u_i = \Delta x_i u_x + \Delta y_i u_y + \frac{(\Delta x_i)^2}{2} u_{xx} \\ + \cdots + \frac{(\Delta y_i)^k}{k!} \frac{\partial^k u}{\partial y^k} \end{aligned} \tag{1}$$

where $\Delta u_i \equiv u_i - u_j$, $\Delta x_i \equiv x_i - x_j$, and $\Delta y_i \equiv y_i - y_j$ and all derivatives are evaluated at j. Since the reconstruction will not be exact for functions that cannot be described by polynomials of degree k or lower, an optimization problem must be solved to find the best choice for the derivatives of u to fit the data. This optimization problem often is solved in a weighted least-squares sense, giving a data-independent least-squares reconstruction (DI-L_2). The following set of equations must be solved approximately:

$$\begin{bmatrix} \Delta_w x_1 & \Delta_w y_1 & \frac{(\Delta_w x_1)^2}{2} & \cdots & \frac{(\Delta_w y_1)^k}{k!} \\ \Delta_w x_2 & \Delta_w y_2 & \frac{(\Delta_w x_2)^2}{2} & \cdots & \frac{(\Delta_w y_2)^k}{k!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Delta_w x_n & \Delta_w y_n & \frac{(\Delta_w x_n)^2}{2} & \cdots & \frac{(\Delta_w y_n)^k}{k!} \end{bmatrix} \cdot \begin{pmatrix} u_x \\ u_y \\ u_{xx} \\ \vdots \\ \frac{\partial^k u}{\partial y^k} \end{pmatrix} = \begin{pmatrix} \Delta_w u_1 \\ \Delta_w u_2 \\ \vdots \\ \Delta_w u_n \end{pmatrix}$$

or equivalently

$$\left[\vec{L}_1 \ \vec{L}_2 \ \vec{L}_3 \ ... \ \vec{L}_{\frac{k(k+3)}{2}}\right] \begin{pmatrix} u_x \\ u_y \\ u_{xx} \\ \vdots \\ \frac{\partial^k u}{\partial y^k} \end{pmatrix} = \left(\vec{f}\right) \tag{2}$$

The weighted differences are defined as $\Delta_w(\cdot) \equiv w_i \Delta(\cdot)$. The w_i often are chosen to be $1/|\vec{x}_i - \vec{x}_0|^t$, with $t = 0, 1, 2$ [9]. While the choice of t does not affect the order of accuracy of the reconstruction, higher values of t typically give lower coefficients for the error terms by weighting data from more distant vertices less heavily.

If the system in Equation 2 is well conditioned, it can be solved by using normal equations. This approach involves multiplying from the left by $\left[\vec{L}_1\ \vec{L}_2\ \vec{L}_3\ ...\ \vec{L}_{\frac{k(k+3)}{2}}\right]^T$, resulting in the following square system of equations.

$$[\mathcal{L}] \cdot \begin{pmatrix} u_x \\ u_y \\ u_{xx} \\ \vdots \\ \frac{\partial^k u}{\partial y^k} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{\frac{k(k+3)}{2}} \end{pmatrix} \tag{3}$$

where $\mathcal{L}_{i,j} \equiv \vec{L}_i \cdot \vec{L}_j$ and $f_i \equiv \vec{f} \cdot \vec{L}_i$. When only first neighbors are involved—as for piecewise linear reconstruction—the use of normal equations requires only a single loop over edges in an unstructured mesh to accumulate the coefficients and right-hand side for the system.

For the case of piecewise quadratic reconstruction, Barth reports [8] that the normal equations are ill-conditioned or high-aspect ratio cells. A more robust solution technique for the least-squares problem, such as singular-value decomposition, should be used for such cases.

The sums required to solve Equation 2 by using normal equations for the piecewise-linear ($k = 1$) reconstruction of a function u on an unstructured mesh can be computed in the following single loop over the edges of the mesh.

Algorithm 1
Least-Squares Linear Reconstruction

```
! loop over all edges
do i_edge = 1, num_edges
  ! vertex at edge origin
  vert1 = edge_to_vert(1, i_edge)
  ! vertex at edge destination
  vert2 = edge_to_vert(2, i_edge)

  dx = x(vert2) - x(vert1)
  dy = y(vert2) - y(vert1)
  du = u(vert2) - u(vert1)

  ! geometric weight
  weight = 1 / (dx*dx + dy*dy)**(t/2)

  wdx = weight * dx
  wdy = weight * dy
  wdu = weight * du

  L11(vert1) = L11(vert1) + wdx * wdx
  L12(vert1) = L12(vert1) + wdx * wdy
  L22(vert1) = L22(vert1) + wdy * wdy
```

```
  L1f(vert1) = L1f(vert1) + wdx * wdu
  L2f(vert1) = L2f(vert1) + wdy * wdu

  L11(vert2) = L11(vert2) + wdx * wdx
  L12(vert2) = L12(vert2) + wdx * wdy
  L22(vert2) = L22(vert2) + wdy * wdy
  L1f(vert2) = L1f(vert2) + wdx * wdu
  L2f(vert2) = L2f(vert2) + wdy * wdu
enddo
```

The purely geometric parts of this loop—the summations for `L11, L12, L22`—can be precomputed as a one-time preprocessing step and stored. Once the sums `L11, L12, L22, L1f,` and `L2f` have been computed, gradients are found by solving Equation 3. The reconstructed solution in the control volume surrounding a vertex j is

$$u_j(x,y) = \bar{u}_j + \nabla u\big|_j \cdot (\vec{x} - \vec{x}_j) \tag{4}$$

3 ENO Reconstruction

The criteria set forth by Harten et al. [5] for essentially non-oscillatory reconstruction place a greater emphasis on accuracy than do the k-exact criteria, in that ENO schemes obtain high-order accurate reconstructions for all vertices having a smooth neighborhood. Also, ENO reconstruction conserves the mean and guarantees that the total variation of the reconstruction will not exceed that of the original function by more than $\mathcal{O}\left(\Delta x^{k+1}\right)$.

One approach to ENO reconstruction is reconstruction via a primitive function (RP-ENO) [5]. In one dimension, this approach integrates cell-averaged data $\bar{u}_j$ to get the primitive function $U_{j+\frac{1}{2}}$.

$$U_{j+\frac{1}{2}} = \sum_0^j \bar{u}_j \Delta x_j \tag{5}$$

The primitive function is interpolated by using a continuous, piecewise polynomial interpolation with polynomials of degree $k+1$. The support for interpolation in cell j always includes $j-\frac{1}{2}$ and $j+\frac{1}{2}$. The stencil is built by adding one vertex at a time, with the choice of which direction to extend the stencil being made to minimize the highest-order divided difference that can be formed on the new stencil. That is, the piecewise quadratic interpolation is chosen to have minimum curvature, and so on. Once the interpolation is complete, the reconstruction of u in cell j is found by differentiating the interpolating function U in the range $(j-\frac{1}{2}, j+\frac{1}{2})$.

As described above, RP-ENO is applicable to nonuniform meshes, and the approach can be extended to multiple dimensions and curvilinear meshes [10]. Extension of ENO schemes to unstructured meshes is more difficult, especially the task of finding smooth stencils for the reconstruction [6, 7].

4 Data-Dependent Least-Squares Reconstruction

Least-squares and ENO reconstruction schemes in one dimension can be analyzed in a uniform framework by writing the reconstructed derivative in cell j as

$$\left.\frac{du}{dx}\right|_j = \frac{1+s_j}{2}\frac{\Delta^+ u_j}{\Delta^+ x_j} + \frac{1-s_j}{2}\frac{\Delta^- u_j}{\Delta^- x_j} \tag{6}$$

With the definition

$$\phi_j \equiv \frac{\Delta^- u_j - \Delta^+ u_j}{\Delta^- u_j + \Delta^+ u_j} \tag{7}$$

a step discontinuity between $j-1$ and j gives $\phi_j = -1$, and a step discontinuity between j and $j+1$ gives $\phi_j = 1$. Smooth functions have $\phi_j = \mathcal{O}(\Delta x)$.

In this context, data-independent least-squares reconstruction corresponds to choosing s_j to be a constant depending only on the local geometry; uniform meshes give $s_j = 0$. This approach can lead to unphysical slopes near discontinuities.

ENO reconstruction results from setting $s_j = sign(\phi_j)$, because ENO reconstruction always chooses a one-sided derivative with the lowest possible magnitude. For evolution of smooth solutions, such a reconstruction has a distinct disadvantage, in that s_j can arbitrarily flip from 1 to -1; this behavior inhibits full convergence to steady state unless steps are taken to prevent binary switching for small changes s_j.

The approach taken here is to design weighting schemes for a data-dependent least-squares reconstruction scheme that will give the behavior of data-independent least-squares reconstruction for smooth functions and that of ENO reconstruction for discontinuous functions. That is, for smooth functions, the weightings should be asymptotically the same as some reasonable data-independent weights, and for discontinuous functions, the weights for nonsmooth data within the stencil should go to zero fast enough to ensure that reconstruction errors for a k-exact reconstruction will be uniformly of $\mathcal{O}\left(\Delta x^{k+1}\right)$. This amounts to requiring

$$\lim_{\phi_j \to \pm 1} s_j \to \pm 1 \tag{8}$$

More will be said later about functions that are continuous but have discontinuous derivatives.

4.1 Second-Order Accurate Data-Dependent Least-Squares Reconstruction

The appropriate behavior in the smooth and discontinuous limits can be obtained by using a simple weighting that takes advantage of the behavior of second differences. If a cell j is separated from one or more of its neighbors by a discontinuity of $\mathcal{O}(1)$, the second difference will go as $\mathcal{O}(\Delta x^{-2})$. In computing the gradient at vertex j, data from neighbors where the difference in function value between j and the neighbor is of $\mathcal{O}(\Delta x)$ should be given a weight of $\mathcal{O}(1)$, whereas data differing from data at j by $\mathcal{O}(1)$ should have a weight

of no more than $\mathcal{O}(\Delta x^2)$ to ensure a second-order accurate reconstruction. The following simple weighting function has this behavior.

$$w_{j,i} = \frac{1}{1 + c\left|D_j^2(u)\right|(u_j - u_i)^2} w^0 \tag{9}$$

Here, w^0 is the geometric weighting factor used previously, and $D^2(u)$ is a numerical second derivative. In multiple dimensions, any reasonable norm of the Hessian matrix is acceptable because all such norms will be of $\mathcal{O}(\Delta x^{-2})$ if there is a discontinuity in the neighborhood of j.* For a one-dimensional, constant-spacing case, the equivalent value of s_j is compared with those for DI-L_2 and ENO schemes in Figure 1.

By design, the implementation of this scheme for unstructured meshes requires only minor modifications to Algorithm 1 given in Section 2.

Algorithm 2
Data-dependent least-squares reconstruction

```
! loop over all edges
do i_edge = 1, num_edges
  ! vertex at edge origin
  vert1 = edge_to_vert(1, i_edge)
  ! vertex at edge destination
  vert2 = edge_to_vert(2, i_edge)

  dx = x(vert2) - x(vert1)
  dy = y(vert2) - y(vert1)
  ! geometric weight
  weight = 1 / (dx*dx + dy*dy)**(t/2)

  du = u(vert2) - u(vert1)
  ! data-dependent weight
  weight1 = weight /
   (1 + c * abs(LaplU(vert1)) * du**2)
  ! data-dependent weight
  weight2 = weight /
   (1 + c * abs(LaplU(vert2)) * du**2)

  wdx = weight1 * dx
  wdy = weight1 * dy
  wdu = weight1 * du

  L11(vert1) = L11(vert1) + wdx * wdx
  L12(vert1) = L12(vert1) + wdx * wdy
  L22(vert1) = L22(vert1) + wdy * wdy
```

*Note that the denominator of Equation 9 must be nondimensionalized for applications other than pure function reconstruction; this will be discussed further in Section 6.

```
  L1f(vert1) = L1f(vert1) + wdx * wdu
  L2f(vert1) = L2f(vert1) + wdy * wdu

  wdx = weight2 * dx
  wdy = weight2 * dy
  wdu = weight2 * du

  L11(vert2) = L11(vert2) + wdx * wdx
  L12(vert2) = L12(vert2) + wdx * wdy
  L22(vert2) = L22(vert2) + wdy * wdy
  L1f(vert2) = L1f(vert2) + wdx * wdu
  L2f(vert2) = L2f(vert2) + wdy * wdu
enddo
```

Here, `LaplU` is a numerical Laplacian. Note particularly that the weights used at opposite ends of the same edge are not the same. This approach has no effect on the conservation of the mean for the scheme, which is ensured by the form of the reconstruction, just as in the data-independent k-exact case.

4.2 Third-Order Accurate Data-Dependent Least-Squares Reconstruction

A third-order accurate ($k = 2$) data-dependent reconstruction can be formulated by analogy to the second-order case just described. The weighting function uses numerical third derivatives to detect nonsmoothness.

$$w_{j,i} = \frac{1}{1 + c\left|D_j^3(u)(u_j - u_i)^3\right|} w^0 \tag{10}$$

Here $D_j^3(u)$ is some appropriate norm of the numerical third derivatives of u in the neighborhood of j. As for second order, the choice of norm is not critical. Smooth data is used in the reconstruction (weight of $\mathcal{O}(1)$), while discontinuous data is ignored (weight of $\mathcal{O}(\Delta x^{-3})$).

The stencil used for third-order reconstruction includes second neighbors.[†] Although for smooth functions it is possible to compute a piecewise-quadratic reconstruction by using only first neighbors, there are two reasons why this approach is inadvisable in the present context. First, the third derivatives needed to compute the weight require a larger stencil in one dimension. Second, near discontinuities, small weights will eliminate data that is not smooth. The number of first neighbors in a two-dimensional unstructured mesh averages six. At least five neighbors are required for quadratic reconstruction. The presence of a discontinuity is almost certain to result in fewer than five vertices with high weights, leading to a seriously degraded reconstruction. The number of first and second neighbors in two dimensions is eighteen on average, so the presence of a discontinuity is unlikely to be

[†]The following assumes that no mid-side nodes have been added to the cells; that is, to use finite element terminology, linear elements are assumed.

problematic. The presence of nearby boundaries will, of course, severely restrict the stencil, and the combination of boundaries and discontinuities may reduce the stencil sufficiently that the quality of the reconstruction suffers. Variable-stencil ENO schemes can avoid this degradation by extending the stencil to include more distant neighbors.

5 Function Reconstruction

Demonstration of the properties of the DD-L_2 scheme begins by comparing the behavior of DD-L_2 with that of DI-L_2 and RP-ENO reconstruction on nonuniform meshes in one dimension. The locations of the mesh vertices are randomly perturbed by as much as 10% of the nominal mesh spacing. Results will be given for each of the following three classes of functions:

Smooth functions, shown in the upper part of Figure 2. For smooth functions, all derivatives will be $\mathcal{O}(1)$ while Δu always will be $\mathcal{O}(\Delta x)$. An example of such a function is $\exp(-5x^2)$, shown in the top half of Figure 3. The bottom half of the figure shows the maximum error for second- and third-order accurate reconstructions using both DD-L_2 and RP-ENO techniques. The difference between DI-L_2 and DD-L_2 for smooth functions is insignificant. The table included with the figure demonstrates that DD-L_2 and RP-ENO reconstruction attain their nominal order of accuracy in all norms for smooth functions.

Piecewise smooth functions with step discontinuities of $\mathcal{O}(1)$, shown in the middle of Figure 2. For cell j, adjacent to the discontinuity, data-independent least-squares will give a gradient that is $\mathcal{O}(\Delta x^{-1})$. This gradient error implies an $\mathcal{O}(1)$ error in the reconstructed function, which can be reduced to $\mathcal{O}(\Delta x)$ through use of a limiter. ENO reconstruction will give a one-sided gradient based on data from cells j and $j-1$; this is a first-order accurate approximation of the derivative. For the data-dependent reconstruction, the weights for cells $j+1$ and $j-1$ will be $\mathcal{O}(\Delta x^{-2})$ and $\mathcal{O}(1)$, respectively. The gradient will be a one-sided gradient plus a first-order error term; the gradient will be first-order accurate. Because DD-L_2 and RP-ENO give first-order accurate gradients, the solution is reconstructed to second-order accuracy. For higher-order reconstructions, DI-L_2 remains first-order accurate, while DD-L_2 and RP-ENO are expected to retain their nominal order of accuracy.

The function $sign(x)\exp(-5x^2)$, a member of this class, is shown in Figure 4, along with the maximum error in the reconstruction for DD-L_2 and RP-ENO. The numerical experiments indicate that both schemes maintain their nominal accuracy in all norms, as summarized in the table accompanying Figure 4.

Piecewise smooth $\mathcal{C}^0$ functions that are not $\mathcal{C}^1$, shown in the bottom part of Figure 2. The function in this figure is actually piecewise linear; the slopes to the left and right of the corner are a and b, respectively. For cell j, a data-independent reconstruction gives a slope of $\frac{3a+b}{4}$, which is a zero-order accurate slope. If $|b| < |a|$, an RP-ENO scheme constructs its slope by using cell $j+1$ rather than $j-1$, giving a zero-order accurate slope of $\frac{a+b}{2}$. Finally, a data-dependent reconstruction gives weights of $1+\mathcal{O}(\Delta x)$ to both cells $j+1$ and $j-1$; the slope computed is $\frac{3a+b}{4}+\mathcal{O}(\Delta x)$ and is zero-order accurate. For this case, each approach gives, in general, a zero-order accurate slope and therefore a first-order accurate reconstruction near discontinuous changes in slope. This result holds for higher-order reconstruction schemes as well.

The difference in these schemes lies not in their order of accuracy but in the change in the total variation of the reconstructed solution relative to that of the underlying function. The RP-ENO scheme is known to have total variation that is the same as the original function to within $\mathcal{O}\left(\Delta x^{k+1}\right)$. The least-squares schemes, on the other hand, allow an increase in the total variation for this class of functions that is of $\mathcal{O}\left(\Delta x\right)$, regardless of the order of the reconstruction.

As an example, consider a function $u = x(2 + sign\left(x\right))$, shown in Figure 5. Here, the bottom half of the figure shows the increase in total variation for both DI-L_2 and DD-L_2 schemes of second- and third-order nominal accuracy. In each case, the total variation increases by an amount of $\mathcal{O}\left(\Delta x\right)$.

6 Flow Solutions

To demonstrate the solution accuracy and convergence behavior of the new scheme for flows with shocks, two inviscid flow results will be shown. For both cases, the flow solver is a multigrid scheme using three coarse meshes. Multistage time advance is used, along with local preconditioning [11, 12]. The CFL number is 0.8, and the multistage coefficients are $\{0.5321,\ 1.3711,\ 2.7744\}$.

For function reconstruction, no normalization was done in the data-dependent weights for least-squares reconstruction. In reconstruction of physical quantities, normalization becomes more important. For piecewise-linear reconstruction, the weight is modified to be

$$w_{i,j} = \frac{1}{1 + C\frac{L^2}{u_\infty^3} D_i^2(u)(u_j - u_i)^2} w^0 \tag{11}$$

where L is a characteristic length scale of the physical problem, and u_∞ is a characteristic value for the physical variable being reconstructed; the freestream value is used in the work. Conserved variables are reconstructed in each of the following examples, and each variable is reconstructed independently.

The first test case is a high angle of attack flow around a NACA 0012 airfoil, at $M = 0.302$ and $\alpha = 9.86^\circ$. Figure 6 shows the fine mesh used for this case, which contains 3323 vertices. The solution was computed by using unlimited DI-L_2, limited DI-L_2 (Venkatakrishnan's limiter [2]), and unlimited DD-L_2. The solutions are virtually identical except near the suction peak on the upper surface. A detail of the surface pressure coefficient in this region is shown in Figure 7; the figure also includes a solution from INS2D to which the Karman-Tsien pressure correction has been applied. The two unlimited reconstruction schemes give identical results to plotting accuracy; the limited scheme comes slightly closer to capturing the suction peak. The limited resolution near the leading edge prevents any of the three solutions from matching the suction peak computed by INS2D on a fine mesh.

Figure 8 compares the convergence histories for the three solutions. CPU time has been normalized so that the limited DI-L_2 solution reaches a residual of 10^{-13} in 100 time units. As expected, the unlimited DI-L_2 solution is the fastest, at 71.5 time units. Not far behind is the unlimited DD-L_2 solution, at 78.7 time units. The DD-L_2 solution is 20% faster than the limited DI-L_2 solution for two reasons. First, the number of multigrid work units is smaller for the DD-L_2 solution, which required a time equivalent to 1190 fine mesh residual

evaluations as compared with 1290 for the DI-L_2 solution. Second, the time per residual evaluation is about 15% less for the DD-L_2 reconstruction. This difference occurs because the computational cost of the Laplacian calculation required by DD-L_2 reconstruction is much less than that of the limiter.

AGARD test case 1 [13] was used to test the new scheme for flows with shock waves. This case computes flow around a NACA 0012 airfoil at a Mach number of 0.8 and an angle of attack of 1.25°. A mesh with 4156 vertices was used (see Figure 9).

The solution was computed by using both limited DI-L_2 reconstruction and DD-L_2 reconstruction. The surface pressure coefficients for these two solutions and for the accepted AGARD solution [13] are given in Figure 10. An inset shows a close-up of the upper-surface shock, which is shifted by one mesh vertex between the two solutions. The quality of the solutions using limited DI-L_2 reconstruction and unlimited DD-L_2 reconstruction are comparable.

Figure 11 compares the convergence rates for the two methods. The CPU time has again been scaled so that the limited reconstruction method converged to a maximum residual of 10^{-13} in 100 time units, requiring 150 multigrid cycles. The DD-L_2 scheme reached the same level of convergence in 85 time units and 137 cycles, an improvement of 15% in CPU time. Again, the extra cost of computing gradients using DD-L_2 instead of DI-L_2 is offset by the need for fewer multigrid cycles and the reduced computational cost for a complete cycle, which is lower for DD-L_2 because no limiting of gradients is required.

7 Discussion

The new family of DD-L_2 reconstruction schemes has been shown to perform well for function reconstruction and for solution of the Euler equations. These schemes are uniformly high-order accurate for smooth functions and limit the magnitude of overshoots near discontinuities to the order of the truncation error. The major drawback of these schemes in comparison with ENO schemes — and this perhaps only from a mathematical viewpoint — is the weaker bound placed on the increase in total variation for functions with discontinuous low-order derivatives. This section outlines an approach to improving the total variation behavior of DD-L_2 reconstruction.

The accuracy of DD-L_2 reconstruction away from discontinuous changes in derivatives of the function ensures that the reconstructed and original functions differ by no more than $\mathcal{O}\left(\Delta x^{k+1}\right)$ at cell interfaces. There are no guarantees on the sign of the jump in the reconstructed function at cell faces, however, implying that the total variation can increase by as much as $\mathcal{O}\left(\Delta x^{k}\right)$ in these regions. While weaker than that provided by ENO schemes, this bound on total variation would still be quite strong.

The difficulty occurs near discontinuities in low-order derivatives; this problem can be readily overcome for second-order accurate reconstruction. Consider a function with a discontinuous change in slope in the neighborhood of vertex j. Because the change in the function is of the same order on both sides of the discontinuity, the weighting function in Equation 9 cannot distinguish between the "right" and "wrong" data, asymptotically. The only distinction between the two is that the "right" data has a slope that is lower in magnitude than that of the slope determined by data-independent least-squares reconstruction,

while the opposite is true for the "wrong" data. Therefore,

$$f(u_i, u_j) \equiv \frac{\left|\frac{du}{dx}\right|_{j,\text{DI-}L_2} - \left|\frac{u_i - u_j}{x_i - x_j}\right|}{\left|\frac{du}{dx}\right|_{j,\text{DI-}L_2}} \tag{12}$$

is positive only for data that should be weighted heavily in the data-dependent reconstruction. In smooth regions, $f = \mathcal{O}(\Delta x)$, whereas near jumps in the first derivative, $f = \mathcal{O}(1)$. To eliminate the total variation increase, a weighting function should give weights of $\mathcal{O}(1)$ to smooth data and weights that go asymptotically to zero for $f = \mathcal{O}(1)$. One such function is

$$w_{i,j} = w^0 \exp\left(A f(u_i, u_j) \Delta x^{-1/2}\right) \tag{13}$$

which implies that as $\Delta x \to 0$

$$w_{i,j} \to w^0 \begin{cases} 1 & \text{smooth} \\ \exp\left(\pm A \Delta x^{1/2}\right) & \text{nonsmooth} \end{cases} \tag{14}$$

The sign of the exponential is such that the data chosen always yields the lowest possible slope. In the nomenclature of Section 4,

$$s_j = \tanh \frac{Af}{\Delta x^{1/2}} \tag{15}$$

for a one-dimensional, uniform mesh.

Numerical experiments with function reconstruction show that this weighting scheme leads to uniformly second-order accurate reconstruction away from discontinuities in the first derivative. Because data that would increase the total variation is weighted by a value that goes exponentially to zero, the increase in total variation from slope discontinuities goes to zero exponentially fast as well. This is reflected in the total variation increase for the function of Figure 5, shown in Figure 12. This modified reconstruction scheme can reasonably be referred to as quasi-ENO.

8 Conclusions

A new method for function reconstruction has been described that combines the best features of least-squares and ENO reconstructions. This new scheme, DD-L_2, uses a data-dependent weighting in least-squares reconstruction to satisfy a relaxed form of the ENO properties. Like other ENO schemes, DD-L_2 is uniformly accurate, even in the presence of discontinuities, and prevents overshoots from exceeding the order of the truncation error asymptotically. The increase in total variation is bounded by $\mathcal{O}(\Delta x)$, and this increase occurs only in the neighborhood of discontinuities in the first derivative. For function reconstruction, the bound on total variation can be improved by using an exponential weighting function. Further investigation of the properties of this particular type of DD-L_2 reconstruction will determine its practical benefits.

Because DD-L_2 is a least-squares reconstruction procedure, existing efficient algorithms for least-squares reconstruction can be easily modified for DD-L_2. An algorithm for computing a linear DD-L_2 reconstruction on unstructured meshes is given. The data-dependent

weights used in the scheme vary continuously with the data and consequently good convergence behavior is obtained for steady-state calculations. The stencil for DD-L_2 is larger than that required by conventional ENO schemes, but no larger than required by k-exact least-squares reconstructions of the same order of accuracy.

The asymptotic behavior of the scheme in reconstructing smooth and piecewise smooth functions has been demonstrated. DD-L_2 produces uniformly high-order accurate reconstructions, even in the presence of discontinuities. Two-dimensional flow solutions using DD-L_2 reconstruction were shown. Subsonic and transonic flow solutions were virtually identical regardless of whether DD-L_2 or limited least-squares reconstruction was used. The absence of a limiter reduced the CPU time required for DD-L_2 solutions by 15-20% as compared with limited reconstruction, even though the DD-L_2 gradient computation is slightly more expensive than ordinary least-squares reconstruction.

Acknowledgments

This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38; and in part by the National Research Council while the author was a Research Associate at NASA Ames Research Center.

References

[1] Barth, T. J. and Frederickson, P. O., "Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction." AIAA paper 90-0013, Jan. 1990.

[2] Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady-State Solutions." AIAA paper 93-0880, Jan. 1993.

[3] Harten, A. and Osher, S., "Uniformaly High-Order Accurate Nonoscillatory Schemes," *SIAM Journal on Numerical Analysis*, vol. 24, pp. 279–309, Apr. 1987.

[4] Harten, A., Osher, S., Engquist, B., and Chakravarthy, S. R., "Some Results on Uniformlay High-Order Accurate Essentially Non-Oscillatory Schemes," *Applied Numerical Mathematics*, vol. 2, pp. 347–377, 1986.

[5] Harten, A., Enquist, B., Osher, S., and Chakravarthy, S. R., "Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III," *Journal of Computational Physics*, vol. 71, pp. 231–303, Aug. 1987.

[6] Abgrall, R., "Design of an Essentially Non-oscillatory Reconstruction Procedure on Finite-Element Type Meshes." ICASE Report No. 91-84, NASA Langley Research Center, 1991. NASA CR 189574.

[7] Durlofsky, L. J., Enquist, B., and Osher, S., "Triangle Base Adaptive Stencils for the Solution of Hyperbolic Conservation Laws," *Journal of Computational Physics*, vol. 98, pp. 64–73, Jan. 1992.

[8] Barth, T. J., "Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes." AIAA paper 93-0668, Jan. 1993.

[9] Barth, T. J., "Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier-Stokes Equations," in *Lecture Series 1994-05*, (Rhode-Saint-Genèse, Belgium), von Karman Institute for Fluid Dynamics, Mar. 1994.

[10] Casper, J., *An Extension of Essentially Non-Oscillatory Shock Capturing Schemes to Multi-Dimensional Systems of Conservation Laws*. PhD thesis, Old Dominion University, Dec. 1990.

[11] Ollivier-Gooch, C. F., "Multigrid Acceleration of an Upwind Euler Solver on Unstructured Meshes," *AIAA Journal*, vol. 33, pp. 1822–1827, Oct. 1995.

[12] Ollivier-Gooch, C. F., "Towards Problem-Independent Multigrid Convergence Rates for Unstructured Mesh Methods I: Inviscid and Laminar Viscous Flows," in *Proceedings of the Sixth International Symposium on Computational Fluid Dynamics*, Japan Society of Computational Fluid Dynamics, University of California at Davis, Sept. 1995.

[13] AGARD Fluid Dynamics Panel, *Test Cases for Inviscid Flow Field Methods*. AGARD, May 1985. AGARD Advisory Report AR-211.

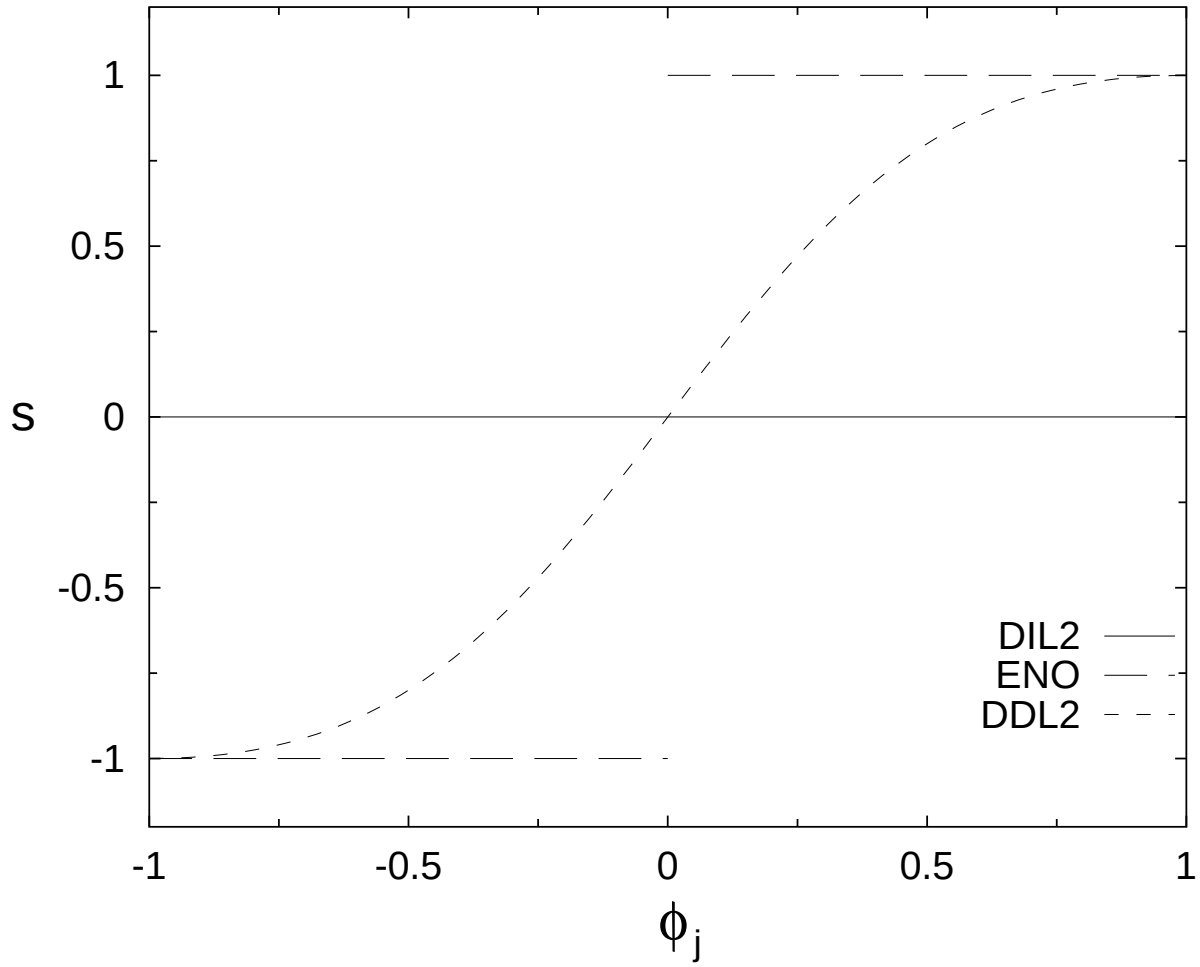


Figure 1: Influence of Forward Difference on First Derivative. Uniform Mesh in One Dimension.

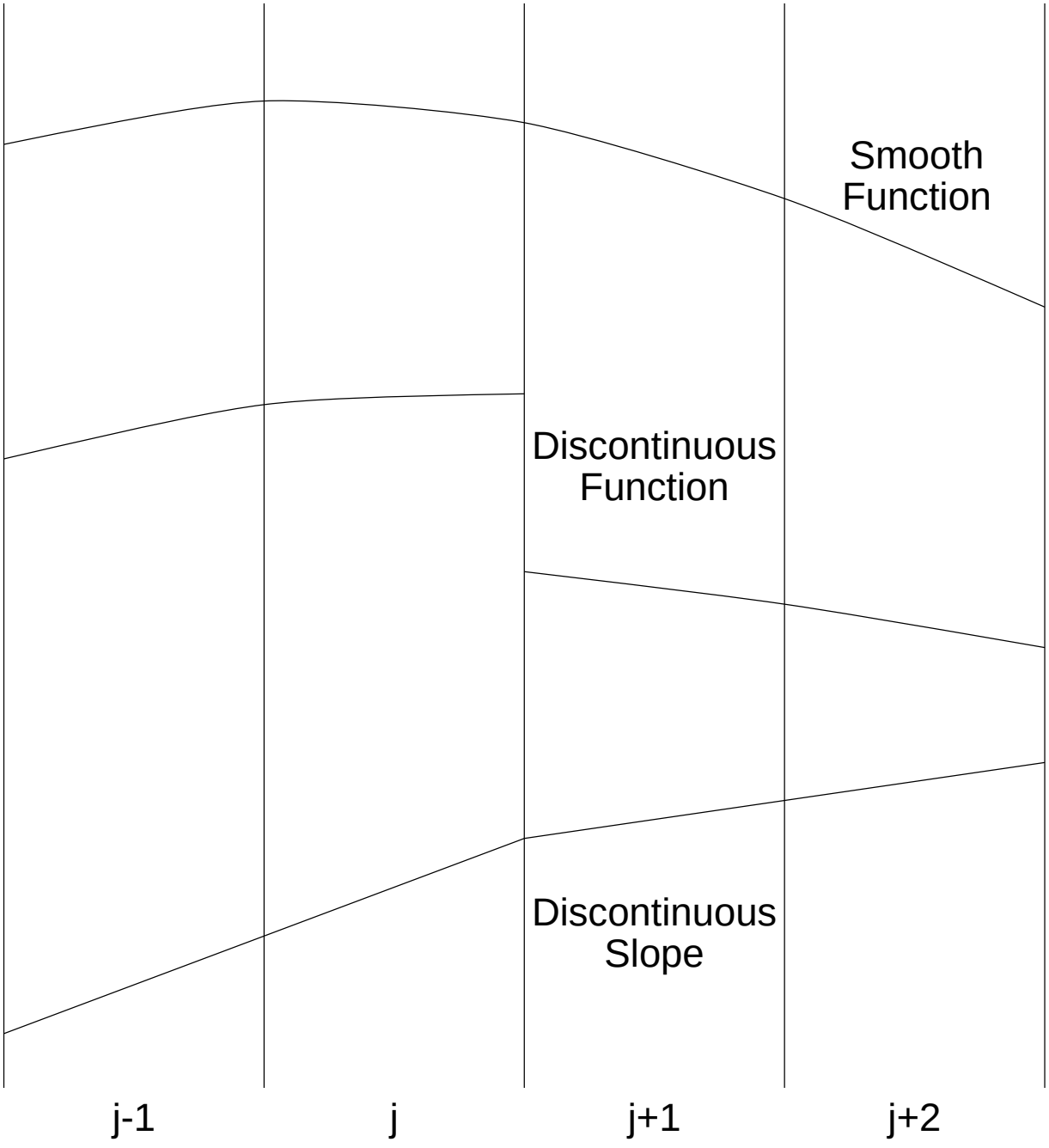
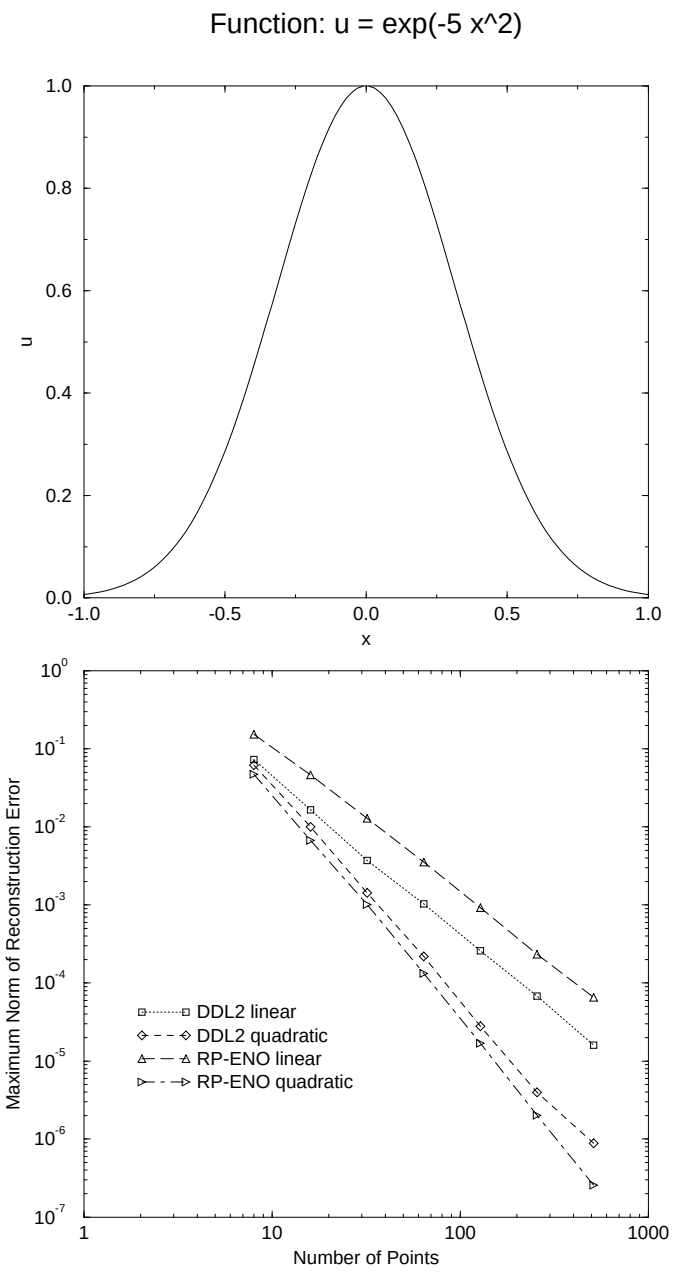


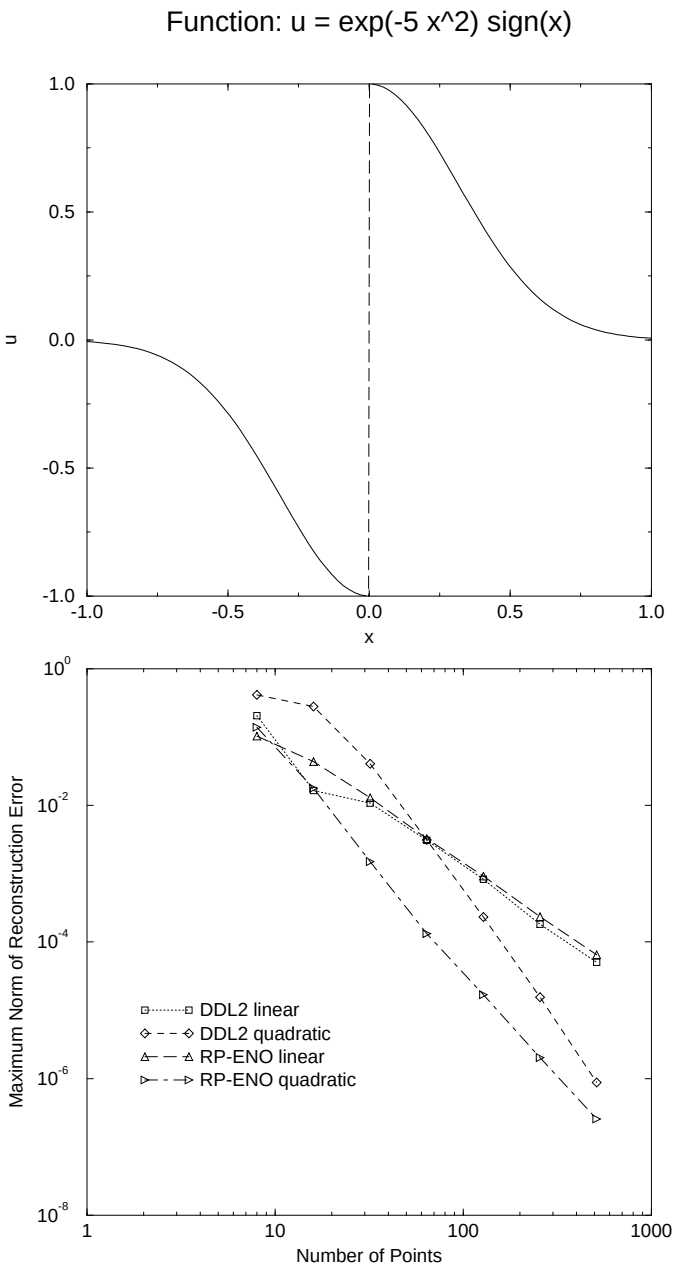
Figure 2: Example Functions for Reconstruction in Cell j



Convergence Order of Error Norms

Scheme	Order	L_1	L_2	L_∞
DD-L_2	2	2.05	2.10	1.97
DD-L_2	3	2.93	2.96	2.83
RP-ENO	2	2.02	2.01	1.91
RP-ENO	3	3.00	2.99	2.93

Figure 3: Reconstruction of Smooth Function



Convergence Order of Error Norms

Scheme	Order	L_1	L_2	L_∞
DD-L_2	2	2.13	2.19	1.96
DD-L_2	3	3.41	3.78	3.86
RP-ENO	2	2.02	2.01	1.91
RP-ENO	3	3.03	3.03	3.10

Figure 4: Reconstruction of Piecewise-Smooth Function with Discontinuity

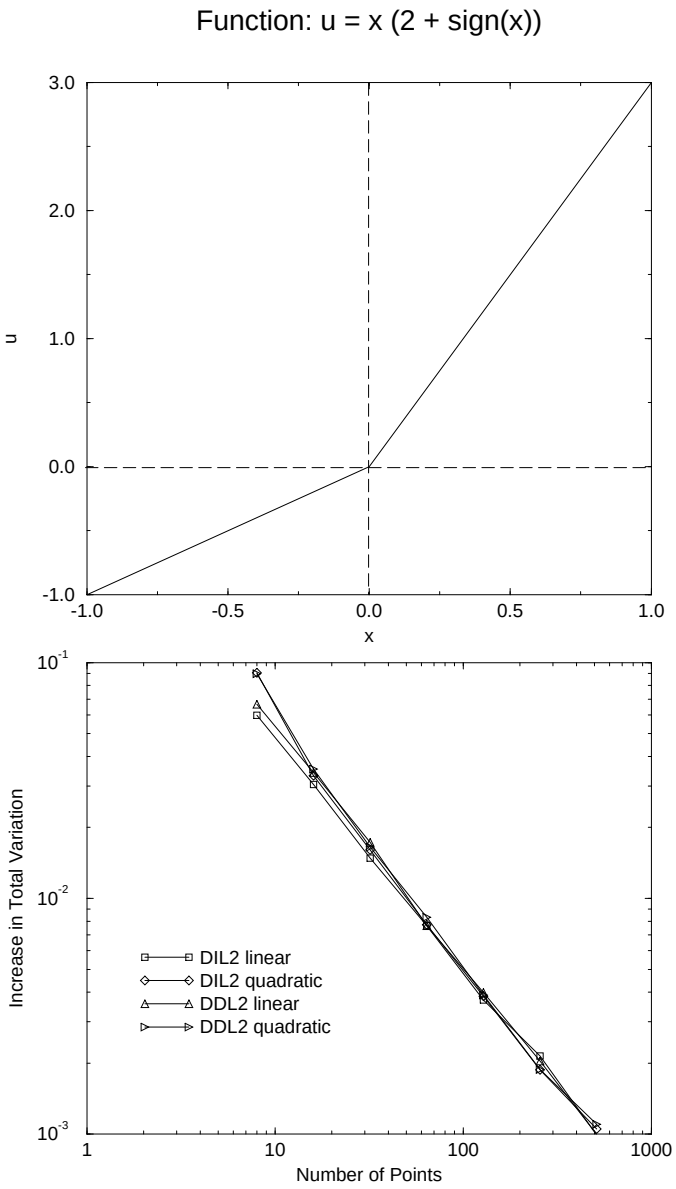


Figure 5: Increase in Total Variation for Least-Squares Reconstruction of Function with Discontinuous Slope

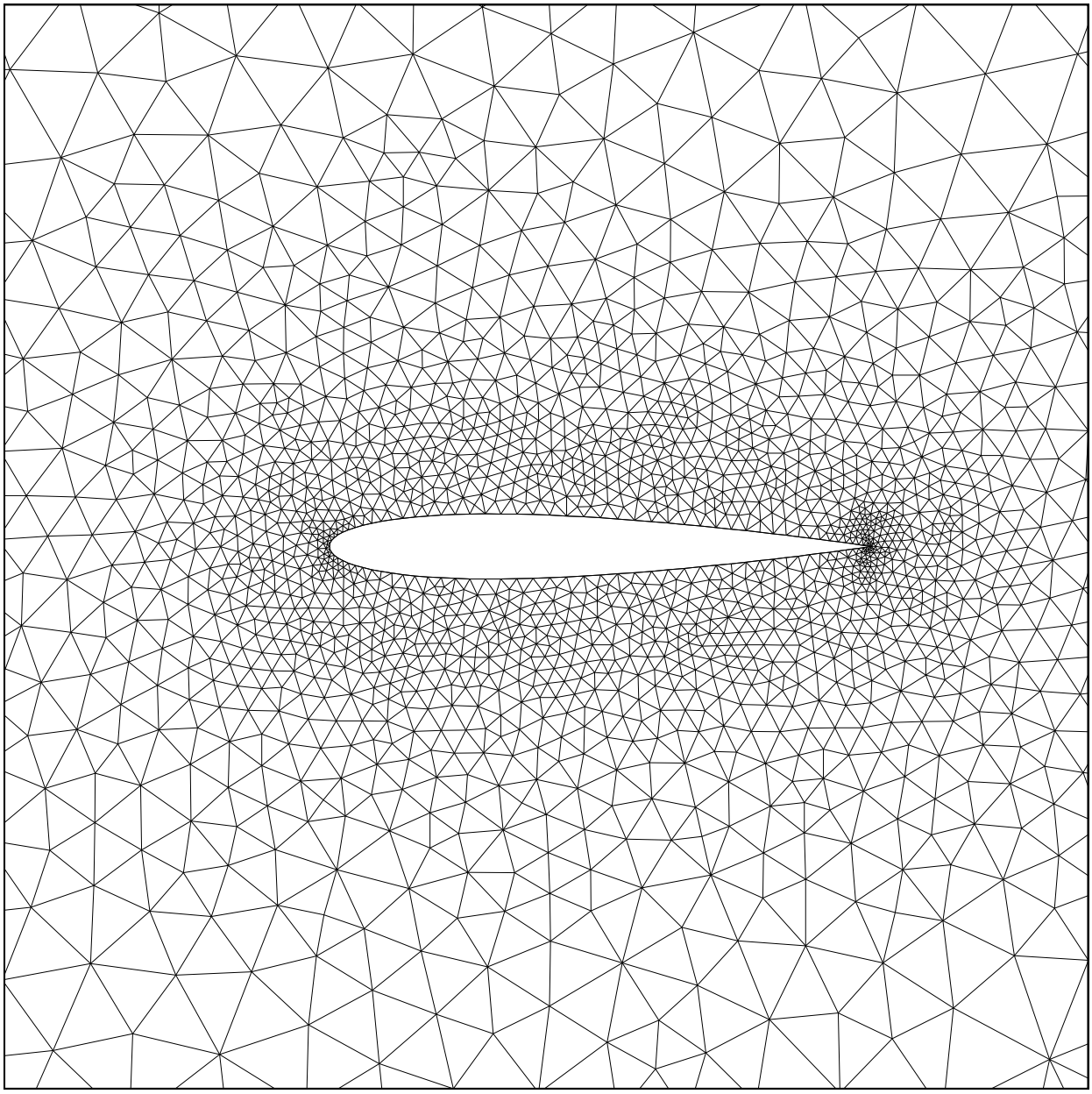


Figure 6: Mesh for High α Case. 3323 Vertices.

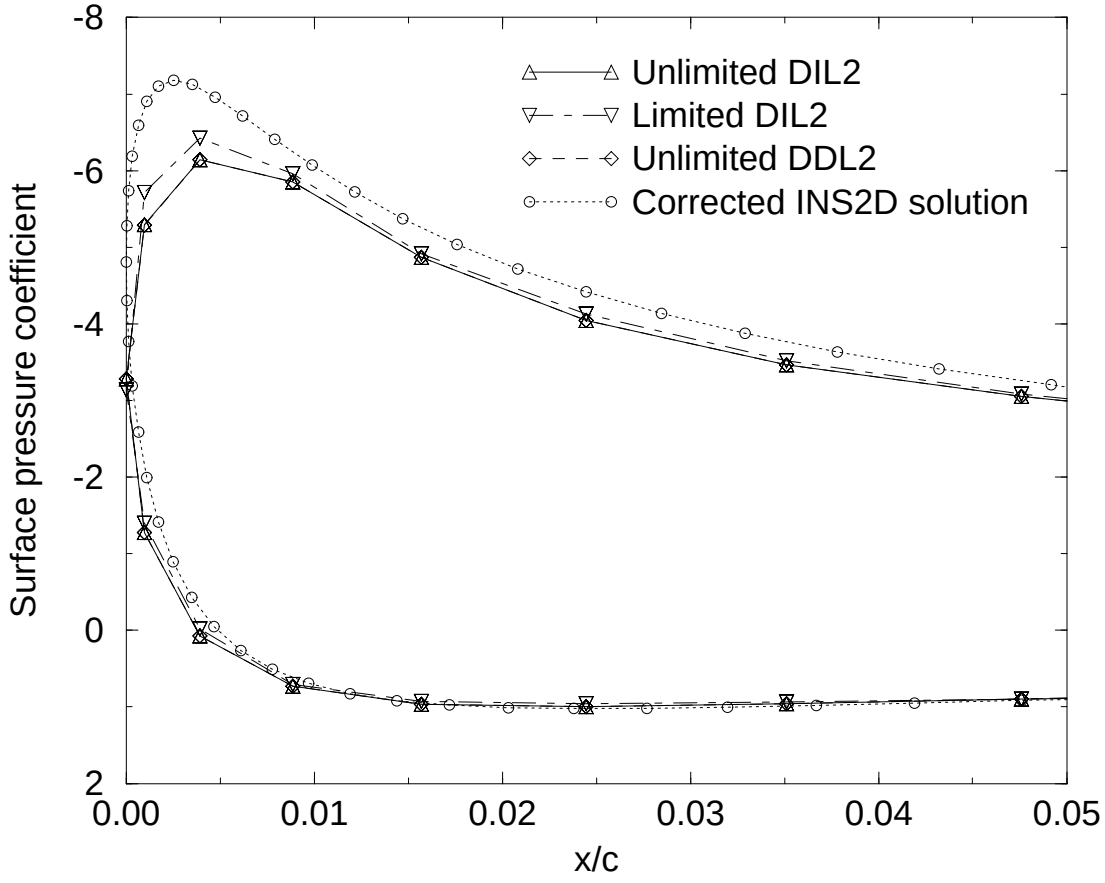


Figure 7: Surface C_p near Suction Peak

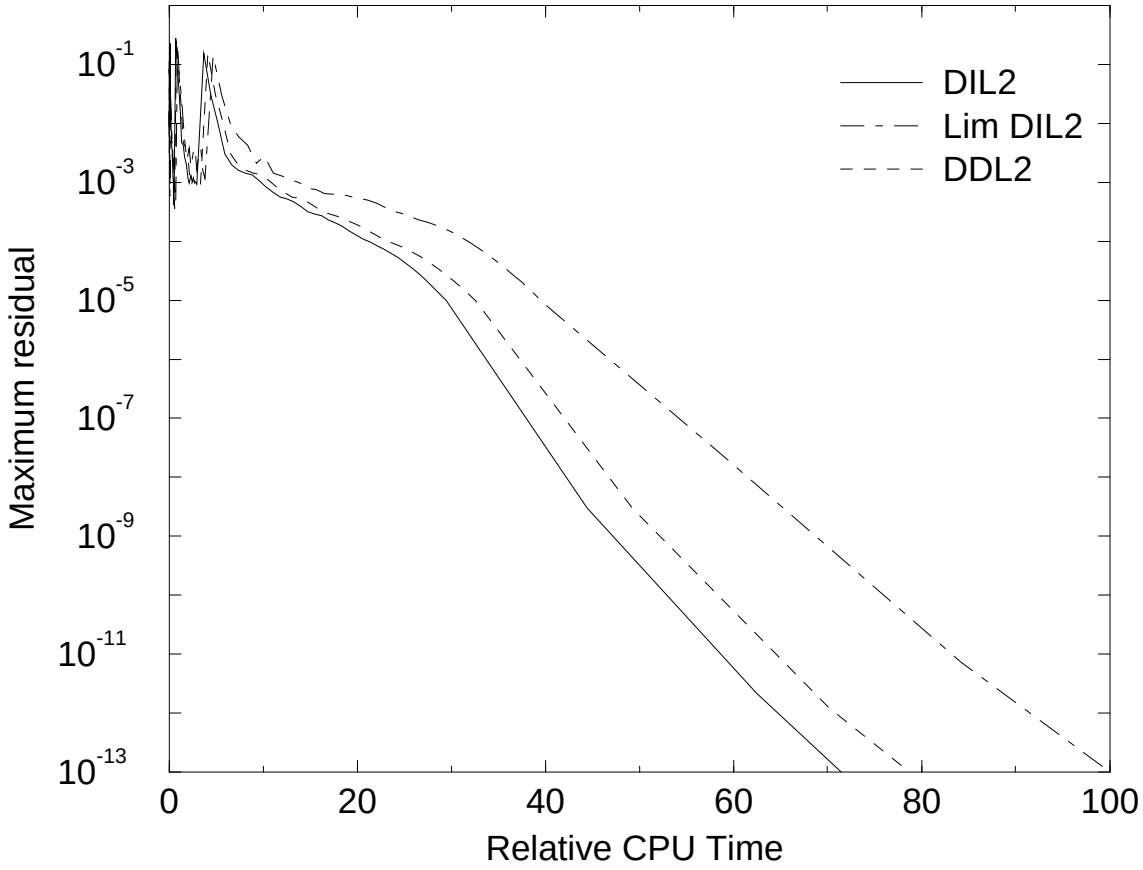


Figure 8: Convergence Histories for High α Case

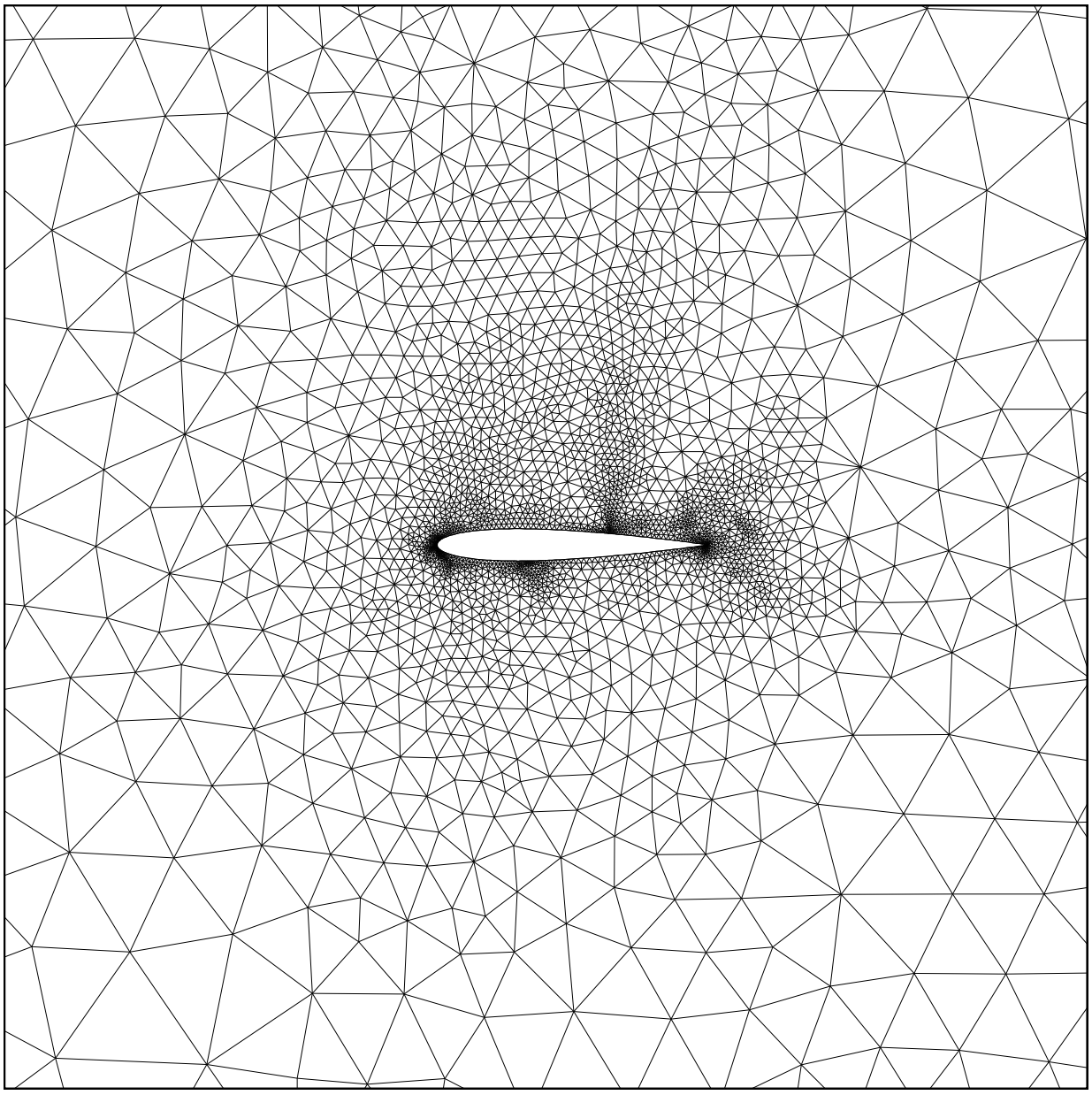


Figure 9: Mesh for AGARD Test Case 1. 4156 Vertices

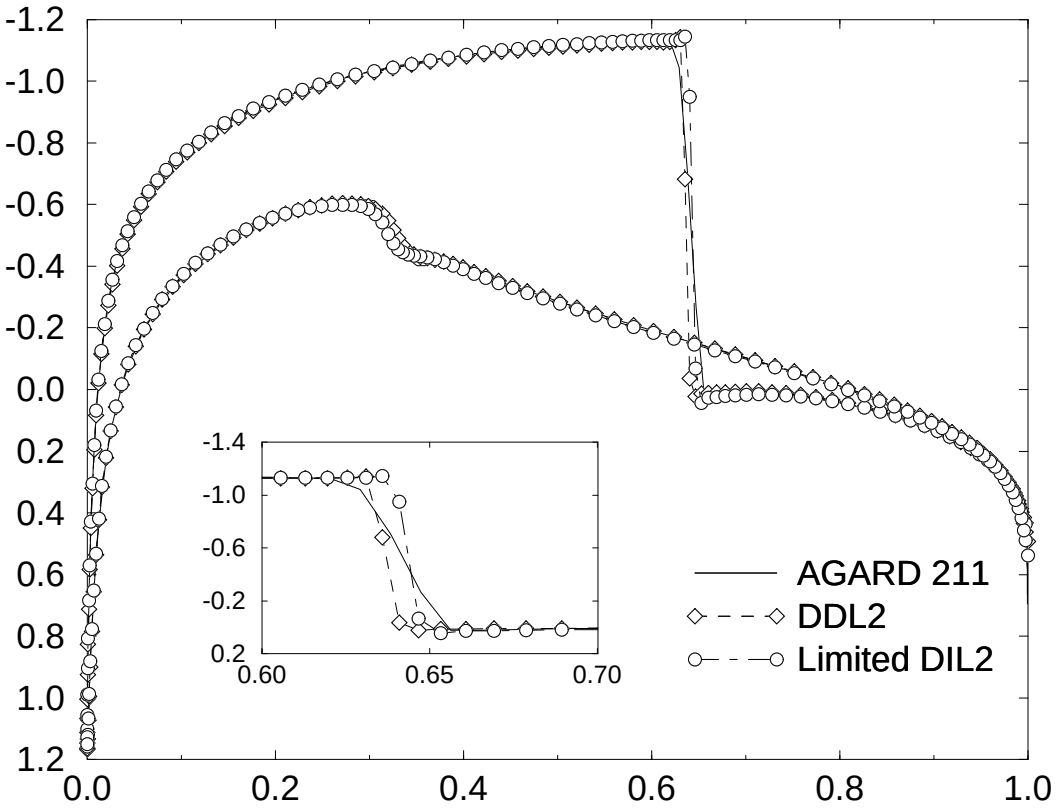


Figure 10: Surface Pressure Coefficient for AGARD Test Case 1

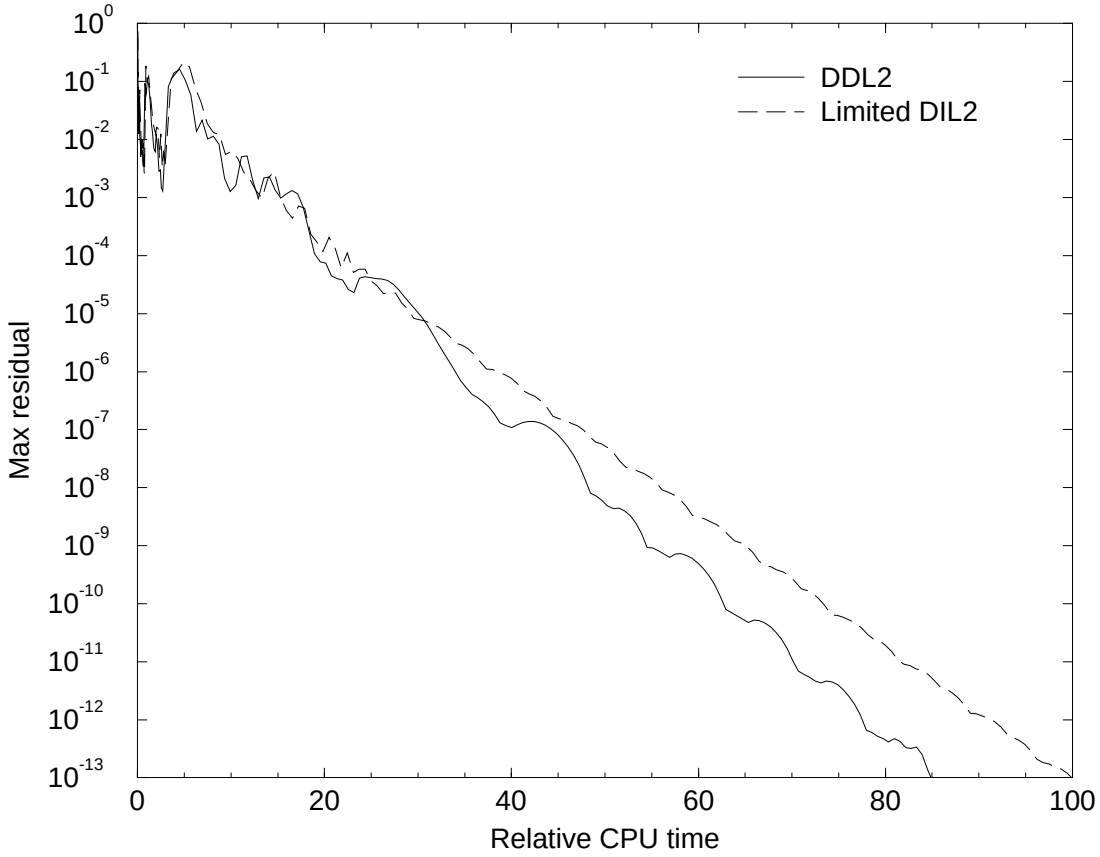


Figure 11: Convergence Histories for AGARD Test Case 1 Using Limited DI-L_2 and DD-L_2 Reconstruction

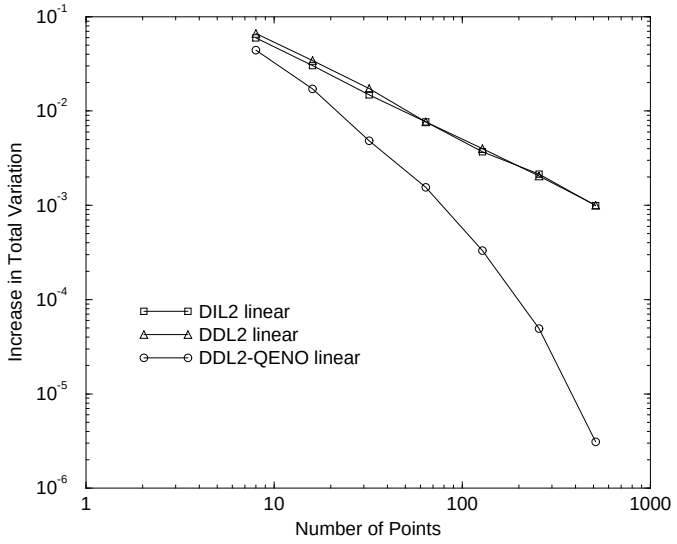


Figure 12: Increase in Total Variation for DD-L_2-QENO Reconstruction of Function with Discontinuous Slope