

Computations of Wall Distances Based on Differential Equations

Paul G. Tucker^a, Chris L. Rumsey^b, Philippe R. Spalart^c,
Robert E. Bartels^d and Robert T. Biedron^e

^a*Civil and Computational Engineering Center, School of Engineering, The University of Wales, Swansea, Singleton Park, Swansea SA2 8PP, UK*

^b*Computational Modeling and Simulation Branch, NASA Langley Research Center, Hampton, Virginia 23681, USA*

^c*Boeing Commercial Airplanes, Seattle, WA 98124, USA*

^d*Aeroelasticity Branch, NASA Langley Research Center, Hampton, Virginia 23681, USA*

The use of differential equations such as Eikonal, Hamilton-Jacobi and Poisson for the economical calculation of the nearest wall distance d , which is needed by some turbulence models, is explored. Modifications that could palliate some turbulence-modeling anomalies are also discussed. Economy is of especial value for deforming/adaptive grid problems. For these, ideally, d is repeatedly computed. It is shown that the Eikonal and Hamilton-Jacobi equations can be easy to implement when written in implicit (or iterated) advection and advection-diffusion equation analogous forms, respectively. These, like the Poisson Laplacian term, are commonly occurring in CFD solvers, allowing the re-use of efficient algorithms and code components. The use of the NASA CFL3D CFD program to solve the implicit Eikonal and Hamilton-Jacobi equations is explored. The re-formulated d equations are easy to implement, and are found to have robust convergence. For accurate Eikonal solutions, upwind metric differences are required. The Poisson approach is also found effective, and easiest to implement. Modified distances are not found to affect global outputs such as lift and drag significantly, at least in common situations such as airfoil flows.

Nomenclature

C_{DES}	=	turbulence modeling constant
C_L, C_D	=	lift and drag coefficients
c	=	wing chord
d	=	nearest wall distance
\tilde{d}	=	distance function
i, j, k	=	array or grid point indices
L	=	length scale function or modified Laplacian
M	=	metric term in transformed equations or Mach number
N	=	number of iterations
n	=	exponent, weighting/biasing parameter or outward facing normal
ns	=	marker in marching front procedure for seed, trial or points to be later solved

^aProfessor of Engineering, School of Engineering, University of Wales, Swansea. AIAA Member.

^bNASA Senior Researcher, NASA Langley, Hampton, Virginia. Associate AIAA Fellow.

^cSenior Technical Fellow, Boeing, Renton, Seattle. No AIAA Affiliation.

^dNASA Senior Researcher, NASA Langley, Hampton, Virginia. Senior AIAA Member.

^eNASA Senior Researcher, NASA Langley, Hampton, Virginia. No AIAA Affiliation.

R_u	= residual for \mathbf{U} equation
Re	= Reynolds number based on wing chord
\mathbf{U}	= front propagation velocity implied in Eikonal equation
u, v	= Cartesian velocity components for Eikonal front propagation
x, y	= Cartesian coordinates
y^+	= distance to nearest surface in ‘wall units’
α	= grid expansion or under-relaxation parameter
Γ, Γ^*	= diffusion analogous coefficients in equations (3,6,7)
ε	= constant in diffusion analogous coefficient evaluation (see Equation (10))
λ	= viscosity solution coefficient
ξ, η	= transformed coordinates
ϕ	= dependent variable in differential wall distance or distance function equations
Ω	= domain for wall distance or distance function computation
$\tilde{\Omega}$	= reduced domain for wall distance or distance function computation
ω	= solid angle

I. Introduction

Wall distances, d , are still a key parameter in many turbulence modeling approaches^{1,2} and also in peripheral applications incorporating additional solution physics^{3,4}. Also, their near wall iso-values can be used in grid generation^{5,6}. Far field d contours can be used as a rapid means of evaluating computational interfaces on unstructured overset meshes having relative movements⁷.

Surprisingly, with search procedures, the effort in calculating d can be significant. For example, even with Cray C90 class computers and time invariant meshes it can take 3 hours just to gain d in a large three-dimensional case⁸. Because of the expense and inter-block communication issues, in some codes the following dangerous approximations are made:

- i) computing distances down grid lines, not allowing for grid non-orthogonality;
- ii) computing d as the distance between a field point and the nearest wall *grid* point, instead of truly the surface; and
- iii) in multiblock grids, determining d on a purely block wise basis, ignoring the possibility that the nearest wall distance might be associated with another block. The latter can create large inaccuracies and also non-smooth, unhelpful-to-convergence, d distributions. In relation to point (iii), for overset grids the situation can arise where the same point in space has different equations depending on which block it is viewed from. In such a case, there is no reason why the solution should converge at all.

Clearly, inexact, non-smooth or grid-blocking-dependent wall distances \tilde{d} will mostly be unhelpful. However, the deliberate modification of d to some distance function \tilde{d} can alleviate certain secondary turbulence model anomalies or extend modeling potential⁹. For example, $\tilde{d} \gg d$ can alleviate the excessive influence of sharp convex features in the geometry on the turbulence model^{1,4}. For corners or bodies/surfaces in close proximity, the increased multiple surface turbulence damping effect (see Mompean et al.¹⁰, Launder et al.¹¹) can be crudely modeled by setting $\tilde{d} < d$.

A. Requirements of an ‘Ultimate’ Distance Function

Following Ref. 11 a preferable distance-function (\tilde{d}) behaviour is perhaps best captured in terms of the elemental solid angle, $d\omega$. This is the angle subtended by a patch, on say a solid surface of radius of curvature R , a distance d from a field point. Hence Ref. 11 proposed

$$\frac{1}{\tilde{d}} = \frac{1}{\pi} \int L \frac{d\omega}{d} \quad (1)$$

(here we correct the factor of 2 error in the equation of Ref. 11) where the continuous wall distance d can be found from say a nearest surface search (NSS). Here, the turbulence length scale function is $L = 1$, but for certain turbulence modeling requirements this may not be best¹¹. Turbulence models are calibrated with a single flat surface mostly in mind with $d \ll R$ or $\omega \sim 2\pi$. In that case, from Equation (1), $\tilde{d} = d$. For several surfaces at about the same distance, the blocking effects are weakly additive and so $\tilde{d} < d$ seems sensible. However, \tilde{d} should not greatly deviate from d . For instance, a reasonable effect, for two surfaces, would be to have the following harmonic related mean $1/\tilde{d}^n = 1/d_1^n + 1/d_2^n$ where d_1 and d_2 are the distances to the two walls and $n = 2$ (in Ref. 11, $n = 1$ is suggested). For a channel, helpfully, considering the harmonic type mean with $n = 2$, $\omega = 4\pi$ and for a flap cove $\omega > 2\pi$ i.e. extra turbulence destruction is naturally introduced.

The opposite situation is when the solid body is much smaller than its distance to the field point under consideration, or in other words the total solid angle it covers is much smaller than 1. An example is a thin wire, which clearly has a much weaker damping effect than a large flat surface at the same distance. We then need to have $\tilde{d} \gg d$. This also helps remedy the excessive modeled turbulence destruction that can be found around extreme convex features⁴.

The above are all accuracy, or physics, considerations, and rather preliminary. Numerically, d field smoothness and computing speed are always desirable. This is more the key focus of the current paper. In addition, the accurate integration of Equation (1) with high-aspect-ratio grid cells near the wall is far from trivial, and the approach of directly solving (1) may be quite unattractive in practice for that simple reason.

B. Differential d Equations

Motivated by the expense of the d solution Sethian⁵ considered the Eikonal equation below

$$|\nabla \phi| = 1 + \lambda \nabla^2 \phi \quad (2)$$

seeking viscosity solutions where $\lambda \rightarrow 0$. The dependent variable in equation (2), ϕ , models propagating front first arrival times. The right hand side implies (away from shock-like features) the front has unit velocity, i.e. there is some velocity field with $|\mathbf{U}| = 1$. This means first arrival times are equal to d . Fares and Schroder¹ essentially solve an Eikonal related equation for ϕ^{-1} . To enhance modeling potential, the Laplacian has some control over \tilde{d} . The Eikonal equation with an explicit Laplacian, as below, is called a Hamilton-Jacobi (HJ) equation.

$$|\nabla \phi| = 1 + \Gamma(\phi) \nabla^2 \phi \quad (3)$$

Solving for ϕ^{-1} as in Ref. 1 could overcomplicate programming and requires an additional arbitrary length scale d_0 to avoid infinite values at the wall. Also, in Ref. 1 no observation of the need for upwind differenced metric terms (see later) is made. The paper does not mention that the inverse \tilde{d} equation is connected to the Eikonal and HJ equations, and hence is amenable to specialized solution approaches.

Spalding⁴ approximately reconstructs d from solution of a more numerically benign Poisson equation ($\nabla^2 \phi = -1$). The $d(\phi)$ reconstruction involves an auxiliary analytically derived equation. The Eikonal, unlike the Poisson approach is challenging to code^{4,5}. This is especially the case for unstructured grids. Hence its implementation in established industrial CFD solvers represents a significant code developer time investment and hence cost. Therefore, here, use of an Eikonal equation form, amenable to general geometry CFD code implementation is explored. The form is reminiscent of the Euler/Navier-Stokes equations (the key equations modeled in CFD solvers). Also, since the Poisson d method is easy to implement in industrial codes, this approach is further studied. For the current work, as the base CFD solver, the NASA CFL3D¹² program is primarily used.

Operation counts show that differential approaches can be significantly faster than search procedures. For fixed mesh problems it is unusual for the search procedure to contribute significantly to the total solution cost.

Nonetheless, for increasingly-common non-stationary-mesh flow solutions the repeated search cost becomes highly significant. Then, differential approaches, which can make easy, safe use of previous d estimates, become especially attractive. The new d approaches presented here are best viewed in this moving mesh context. Although the approaches to be presented have successfully been tested for distorting meshes, only stationary mesh results are given here. Stationary meshes (where the starting/initialized d field is very different from the actual) most strongly test robustness and convergence traits and so seem appropriate. The equations considered are now more fully outlined.

II. Implicit/Iterated d Equations

In this paper, three different wall distance methods based on differential equations are used. These are: implicit Eikonal, implicit HJ and Poisson, defined below. The Eikonal (Equation (2)) is an exact d equation. When defining the vector

$$\mathbf{U} = \nabla \phi, \quad (4)$$

it can be rewritten in the following implicit (Eikonal_I) advection analogous form

$$\mathbf{U} \bullet \nabla \phi = 1. \quad (5)$$

In the above, for convenience $\lambda \nabla^2 \phi$ where $\lambda \rightarrow 0$ is omitted. The vector \mathbf{U} corresponds to the front propagation velocity implied in the Eikonal equation where $|\mathbf{U}| = 1$. With (4), the HJ (Equation (3)) can also be written in the following implicit (HJ_I) form

$$\mathbf{U} \bullet \nabla \phi = 1 + \Gamma(\phi) \nabla^2 \phi \quad (6)$$

The positive function $\Gamma(\phi)$ is discussed later. Since $(\nabla \phi)^2 = \nabla \bullet (\phi \nabla \phi) - \phi \nabla^2 \phi$ and $\mathbf{U} = \nabla \phi$ the following conservative form of (6) is also possible

$$\nabla \bullet (\phi \mathbf{U}) = 1 + \Gamma^* \nabla^2 \phi \quad (7)$$

where

$$\Gamma^* = (\Gamma + \phi) \quad (8)$$

To gain Eikonal_I solutions, using an initial ϕ field guess, equations (4) and (5) can be solved in an iterative sequence. HJ_I solutions can also be made using the same approach.

Along different lines, substitution of the Poisson based ϕ distribution arising from solving Equation (6) with $\mathbf{U} = 0$ and $\Gamma = 1$ into

$$d = -\sqrt{\sum_{j=1,3} \left(\frac{\partial \phi}{\partial x_j} \right)^2} \pm \sqrt{\sum_{j=1,3} \left(\frac{\partial \phi}{\partial x_j} \right)^2} + 2\phi \quad (9)$$

also gives distances⁴. This is called the Poisson method. The derivation of (9) assumes extensive (infinite) coordinates in the non-normal wall directions. Hence, unlike the Eikonal, d from Equation (9) is only accurate close to walls. However, turbulence models only need d accurate close to walls.

A. Laplacian form and role

Near a fine convex feature (for example a wire) for theoretical correctness accurate distances are needed so $\tilde{d} = d$. However, to prevent excessive far field influence $\tilde{d} \gg d$ can be required further away from the feature^{1,4}. Adjacent to a convex feature $\Gamma \nabla^2 \phi \gg 0$. Therefore, the positive Laplacian inclusion in (6) has the desired effect of enlarging/exaggerating $d(=\phi)$. Motivated by dimensional homogeneity, the need that as $d \rightarrow 0, \tilde{d} = d$ but $\nabla^2 \phi \rightarrow \infty$ suggests

$$\Gamma = \varepsilon d \quad (10)$$

where ε is a constant. Clearly more ‘aggressive’ functions than (10) (e.g. $\Gamma = \varepsilon(-1 + e^d)$) are possible but these are not explored.

At concave corners $\nabla^2 \phi < 0$, hence $\tilde{d} \ll d$. Therefore, with the Laplacian, the damping effects of ‘extra’ walls, discussed earlier, is naturally accounted for. Eikonal and hence Equation (5) distances will generally be discontinuous in gradient (this partly occurs where fronts effectively propagating from different walls collide). Therefore, the $\nabla^2 \phi$ term smoothing in Equation (6) has the potential to enhance convergence.

III. Numerical Modeling

For brevity, the Equations (4-7) numerical modeling is described. Modeling for the much less challenging Poisson d method is discussed in Ref. 4. Equations (4-7) can be solved on curvilinear and unstructured grids. For curvilinear grids they must be transformed using the chain rule for differential calculus in say an ξ, η system. When solving in this system, metric terms (M) such as $\xi_x (= \partial \xi / \partial x)$, η_x , ξ_y , and η_y are evaluated. This aspect will be discussed later. Equation (6-7) diffusive terms are discretized using second order central differences. The advection analogous derivatives in equations (5-7) are typically discretized, just considering an x coordinate direction for example, using the following 1st order upwind type of approximation

$$\frac{\partial \phi}{\partial x} \approx n_{i-1} \Delta_{i-1,j,k} \phi + n_{i+1} \Delta_{i+1,j,k} \phi \quad (11)$$

where

$$\Delta_{i-1,j,k} \phi = \frac{\phi_i - \phi_{i-1}}{\Delta x_{i-1}}, \quad \Delta_{i+1,j,k} \phi = \frac{\phi_{i+1} - \phi_i}{\Delta x_{i+1}} \quad (12)$$

and $n_{i-1} = 1$ for $u_{i-1} > -u_{i+1}$ and $u_{i-1} > 0$ else $n_{i-1} = 0$; and $n_{i+1} = 1$ for $-u_{i+1} > u_{i-1}$ and $-u_{i+1} > 0$ else $n_{i+1} = 0$. The propagation velocities in the above are evaluated from (4). Where, for example

$$u_{i-1} = \frac{\phi_i - \phi_{i-1}}{\Delta x_{i-1}}, \quad u_{i+1} = \frac{\phi_{i+1} - \phi_i}{\Delta x_{i+1}} \quad (13)$$

Setting $\Gamma = 0$ in (7,8) shows that the conservation form of the Eikonal₁ equation still has a Laplacian. Here the view is taken that the discretized *conservation* form of the equation is being solved. However, to allow optimal Eikonal₁ solution (using a hyperbolic advancing front approach described later), the extra Laplacian $\phi \nabla^2 \phi$ is ignored. The approximation of neglecting $\phi \nabla^2 \phi$ is equivalent to solving the non-conservation Equation (5) with the alternative approximation that

$$u_i = n_{i-1}u_{i-1} + n_{i+1}u_{i+1} \quad (14)$$

The above equation corresponds to using offset difference-based velocity components. These correspond to those needed in the conservative Equation (7) form. Either viewpoint has no great accuracy implications. Where d needs to be accurate, for turbulence models, $\phi \nabla^2 \phi \rightarrow 0$. Numerical tests confirm the validity of the above arguments.

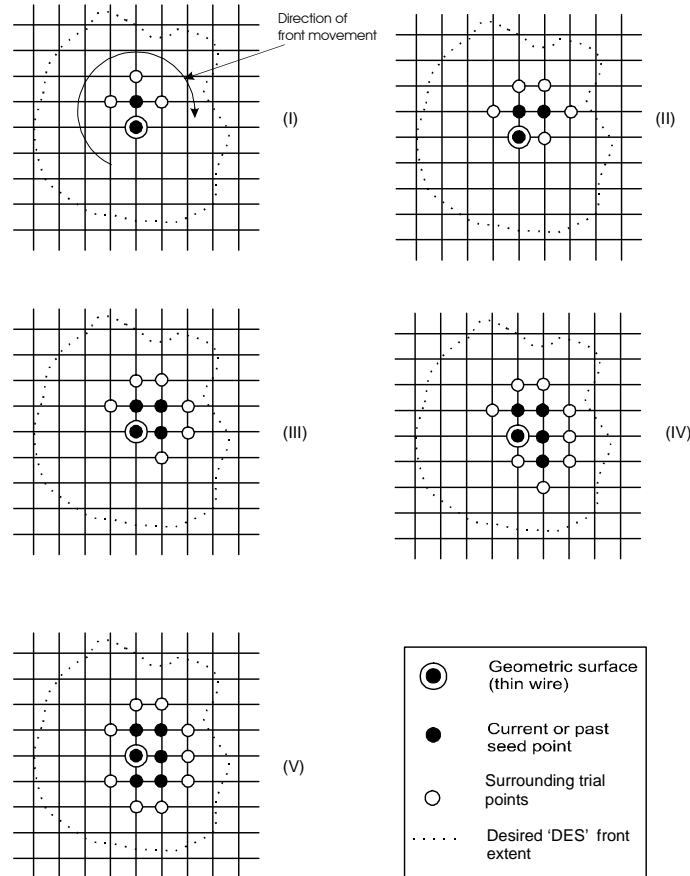


Figure 1: MF solution approach with one seed.

A. Metric Discretization

The metric terms (M) must be carefully discretized. In CFL3D, for the i index direction M_{i-1} and M_{i+1} are first evaluated. The latter, for example, involves geometric data at the i and $i+1$ grid points. These M_{i-1} and M_{i+1} difference based constructs are then, as with most CFD codes, averaged to give a single M_i value to be used in the discretized transport equations. However, for strongly expanding near wall grid spacings this standard approach results in d overestimations for the Eikonal_I and HJ_I. To remedy this the following metric formulation is used

$$M_i = n_{i-1}M_{i-1} + n_{i+1}M_{i+1} \quad (15)$$

The standard CFL3D implementation corresponds to $n_{i-1} = n_{i+1} = 1/2$ in the above.

B. Stabilizing Measures

To ensure stable solutions for the Eikonal_I and HJ_I velocity clipping and diagonal dominance enhancement are tried. Both use the observation that, in 2D (two-dimensions), the exact U field should satisfy

$$R_u = |u_{i,j}^2 + v_{i,j}^2 - 1| = 0 \quad (16)$$

where u and v are x and y direction velocity components, respectively. Therefore, to improve diagonal dominance R_u and R_v are added to both the discretized equation matrix diagonals and source terms, respectively. Based on Equation (16) the following velocity clipping is used

$$|u_{i,j}| \leq 1, \quad |v_{i,j}| \leq 1 \quad (17)$$

The Eikonal equation does not permit a backwards-front movement⁵. The implication of this is that if in (6), for example, $\Gamma \nabla^2 \phi < -1$ a theoretical violation has occurred (the sign of the Equation (6) right hand side gives the front propagation direction). Hence, for HJ_I solutions the Equation (6) Laplacian (L) is modified to

$$L = \max[-C, \Gamma \nabla^2 \phi] \quad (18)$$

where $0 \leq C \leq 1$. Here $C = 1$ is used. This is the theoretical correctness upper limit. Since anti-diffusion is associated with instability use of (18) should be viewed as a stability measure. As a further stability measure under-relaxation is used either through the following $\phi = (1 - \alpha)\phi^{old} + \alpha\phi^{new}$ (where α is an under-relaxation factor and the ‘new’ and ‘old’ superscripts indicate iterative states) or use of a pseudo time term.

When solving the Eikonal equation explicitly, for each grid point, evaluation of the discretized matrix coefficients needs around 160 operations. The Eikonal_I equation needs 90 operations but then iteration is also required (i.e. equations (4,5) must be contained in some iterative loop). Therefore, the key efficiency issue is the number of iterations N typically required.

C. Simultaneous Equation Solution

For Eikonal_I solutions either essentially crude global Gauss-Siedel (GGS) type iterations or a marching front (MF) approach are used. For HJ_I solutions the MF approach can be used to gain an initial d field and then GGS used. Alternatively, as for the Eikonal_I, purely GGS based solutions can be made. To solve the Poisson equation essentially a GGS approach is used.

The MF approach essence is illustrated using Fig. 1. In this, a ‘wire’ is represented on a Cartesian grid using a single node point. Points surrounding this are considered to be either seed, trial or those to be later solved. These points are labeled through the marker variable ns with the following respective values 2, 1 or 0. To start, a seed (or multiple seeds) point is defined. The Eikonal_I equations are applied N times (until convergence) to assign this seed point a d value. The seed is represented in Frame (I) by the closed symbol. The Eikonal_I equations are then solved for all immediate seed neighbors (trial points). These $ns = 1$ points are shown as open symbols. As shown in Frame (II), the trial point with the minimum d is taken as the next seed. Now, for this point, $ns = 2$. Then, as shown in Frame (III), d for all immediate neighbors to the new seed (for which $ns \neq 2$) are calculated. Hence the next seed point location is found. Frames (IV) and (V) show subsequent development stages for a clockwise moving circular front. In summary the procedure is as follows:

- a) Define a seed point;
- b) Iteratively solve for d at immediate neighbors to the seed (trial points) and
- c) Make the trial point with the minimum d the next seed and return to (a).

This procedure is continued until $ns = 2$ for all points in Ω or a smaller domain/extent $\tilde{\Omega}$. The latter domain is identified by the Fig. 1 dotted line. The Eikonal/MF method’s compatibility with smaller domains is useful. This is because, many turbulence models and DES (Detached Eddy Simulation)² only need d to a maximum of about $1/3^{\text{rd}}$ the boundary layer thickness. In a practical system, all surface adjacent points can be taken as seeds and what is called an ‘active front’ produced. For best efficiency a heap-sort procedure is required. This has not been incorporated as part of this work. Clearly, based on operation counts, $N < 2$ is needed for the Eikonal_I approach to match the Eikonal’s efficiency. However, a typical Eikonal_I N value is 4 (See Ref. 13) i.e. for stationary meshes the Eikonal_I equation has about twice the computational cost of the Eikonal. For moving meshes this difference greatly decreases.

D. Initial/Starting Conditions

For the Eikonal_I and HJ_I $1 < \phi < \infty$ in Ω is found adequate. However, for MF solutions $\phi \rightarrow \infty$ is used. Note, when using the MF approach to give a DES distance field $\phi = C_{DES}\Delta_i$ (C_{DES} is a turbulence modeling constant) can be used where $\Delta_i = \max(\Delta x, \Delta y, \Delta z)$. The d computation will naturally stop when $d = C_{DES}\Delta_i$. No DES d results are shown here. An example can be found in Ref. 13. For the Poisson approach $\phi \approx 0$ is adequate.

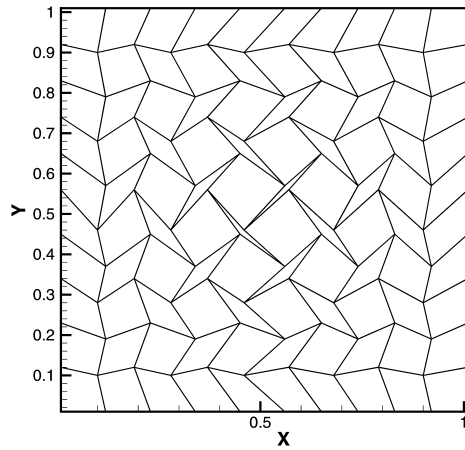


Fig. 2: Badly distorted flat plate grid.

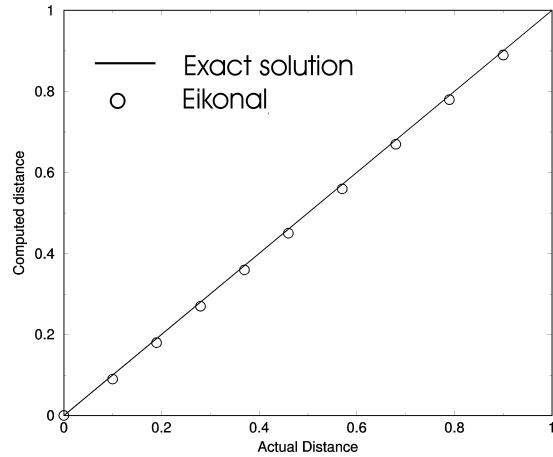


Fig. 3: Predicted against actual d at $x = 0.5$.

E. Boundary Conditions

Conditions on the domain boundaries are now described. At solid walls the following Dirichlet condition is applied

$$\phi = 0 \quad (19)$$

At flow/far field boundaries

$$\frac{\partial \phi}{\partial n} = 0 \quad (20)$$

can be used, where n is the boundary normal co-ordinate. However, if Ω is sufficiently large, (19) makes a stable, most computationally economical far field boundary condition. It is especially preferred for the Poisson method where it gives much faster convergence.

IV. Results and Discussion

The following geometries are considered: (a) Flat plate; (b) Single element airfoil (NACA4412); (c) Wing body and (d) Wing flap. Where d deviation ('error') values are given, these are relative to NSS distances. The deviation is defined using the equation below

$$Error = 100 \left(\frac{d_{NSS} - d}{d_{NSS}} \right) \quad (21)$$

where d_{NSS} is the distance coming from the nearest surface search. The d without the subscript in Equation (21) is the distance field for which the error is being evaluated. Flow solutions use the Spalart and Allmaras¹⁴ (SA) turbulence model. This probably has the most extreme d requirements of most standard turbulence models. It needs d accurate for around $1/3^{rd}$ the boundary layer thickness. Hence it is a good candidate for d modeling tests.

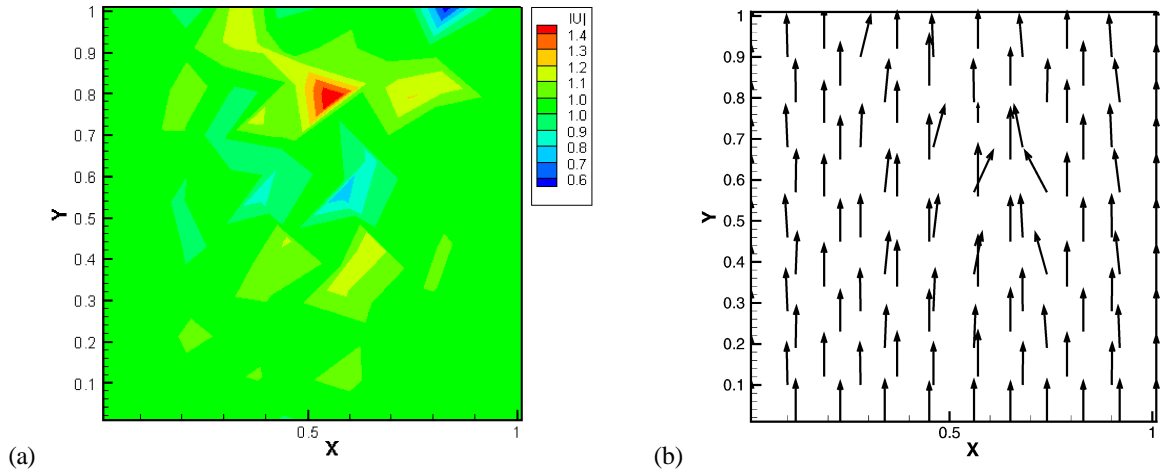


Fig. 4: Distorted grid $\nabla \phi$ data: (a) $|U|$ and (b) U vectors.

A. Flat Plate (Case (a))

Initially, for testing just the Eikonal_I method, a flat plate is used. This is at $y = 0$, in a 2D square domain Ω , with sides of unit length. First a grid (not shown) with strong grid expansion in the y direction is considered. Table 1 gives % d errors for different approximate geometric grid expansions α_y with offset and centered metrics.

α_y	% d Error	
	Offset metrics	Centered metrics
1.1	0.48	1.53
1.7	0.99	8.25
2.6	3.15	15.0

Table 1. Eikonal_I % d errors with offset and centered metrics.

Clearly, (see Table 1) with centered differences and larger α_y values serious errors arise. This is not surprising. Roache¹⁵ shows for accurate flow solutions $\alpha_y < 1.3$ is needed. The front propagation nature of the Eikonal_I (and Eikonal) makes errors additive. Furthermore, centered differences are inconsistent with a hyperbolic propagating front problem. Hence, all the remaining Eikonal_I results use offset metrics.

Next, the effect of using a badly distorted grid is explored. Fig. 2 shows the badly distorted ‘curvilinear’ grid used in this study. This grid is intended to severely test robustness of the Eikonal_I method. Fig. 3 plots predicted against actual distances (y) at $x=0.5$ for the Fig. 2 grid. The full line gives the exact solution. The symbols give the distances predicted by the Eikonal_I method. The figure shows that, even with an extremely distorted grid, reasonable distances can be obtained.

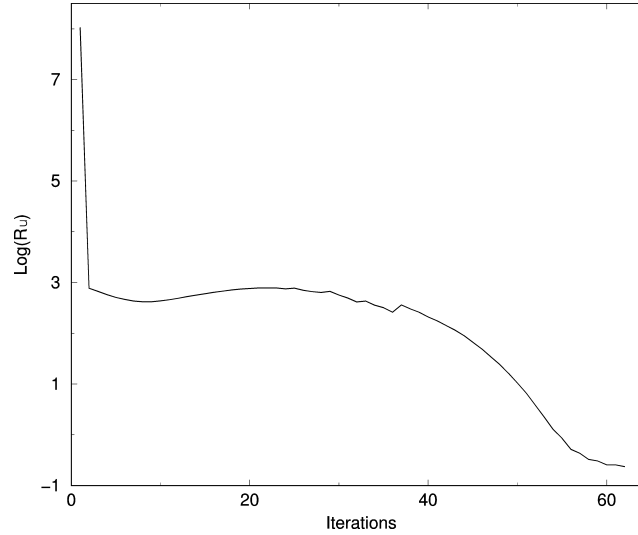


Figure 5: Distorted grid plot of $\log(R_u)$ against GGS iterations.

Fig. 4 shows Fig. 2 grid velocity data. Frames (a) and (b) give $|\mathbf{U}|$ contours and \mathbf{U} vectors, respectively. Considering the poor grid form the velocity magnitudes and direction are surprisingly good. Fig. 5 gives a Fig. 2 grid plot of $\log(R_u)$ against GGS iterations. Through its monotonic nature the plot reflects the stable convergence.

B. Single element airfoil (Case (b))

The Poisson, Eikonal_I and NSS procedures are now compared for a NACA4412 airfoil. The attack angle is 13.87° , $M = 0.2$ and $Re = 1.52 \times 10^6$ (based on the wing chord, c). Fig. 6 shows the 3 zone circa 25,000 cell overset grid used as part of the studies. Fig. 7 gives Eikonal_I \mathbf{U} vectors. Correctly, the vectors are surface normal. Their magnitude is close to unity. Overset Poisson solutions are made for Ω and also just the inner zone region ($\tilde{\Omega}$) labeled in Fig. 6. Obviously, the latter solution is most computationally efficient. At the $\tilde{\Omega}$ boundaries the Dirichlet condition $\phi = 0$ is used. Fig. 8 shows d contours representing in the following respective frames the: (a) NSS; (b) Eikonal_I; (c) Poisson and (d) $\tilde{\Omega}$ Poisson results.

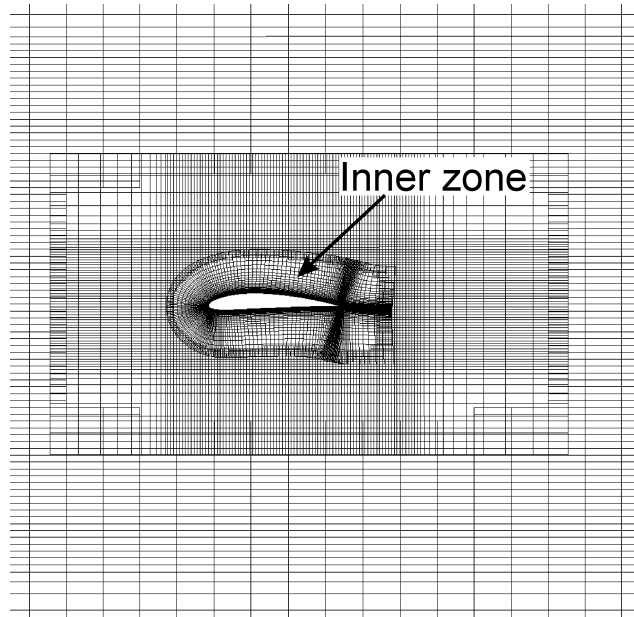


Figure 6: NACA4412 overset solution grid.

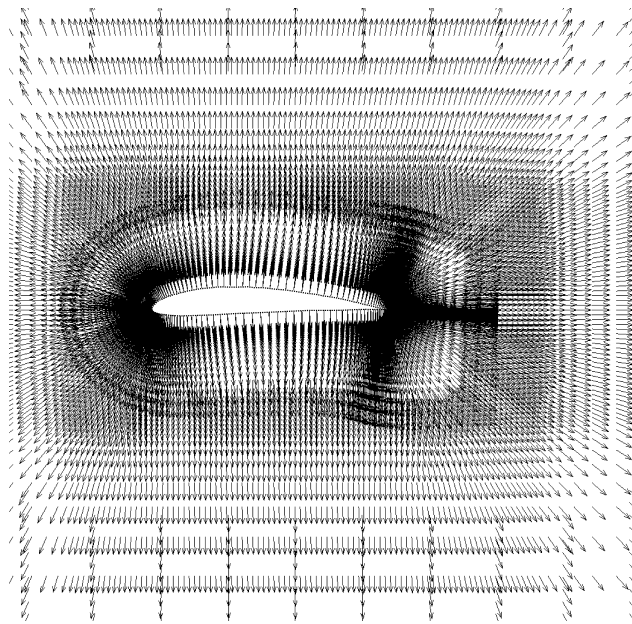


Figure 7: Eikonal_I U vectors.

Fig. 9 gives $y^+ < 400$, d deviation histograms. Frame (a) is for the Eikonal_I equation. The average deviation is 0.5 %. Frame (b) is the overset Poisson $\tilde{\Omega}$ domain solution. Although not shown by the histograms, the average d deviation for the Poisson is significantly higher (2.5%) than that for the Eikonal equation.

Fig. 10 gives a zoomed in view of trailing edge region Poisson d deviation contours. As can be seen, a key overestimation zone is the sharp convex trailing edge geometry region. This overestimation zone, which will also be produced with the HJ₁ equation, is potentially desirable (see later).

	C_L	C_D
Eikonal _I	1.704	0.03466
NSS	1.698	0.03496
Poisson	1.713	0.03543

Table 2. C_L and C_D 's for different d fields.

Table 2 gives lift (C_L) and drag (C_D) coefficients. Some minor differences are evident. For surface pressure plots and boundary layer velocity profiles lines/results effectively overlay. Fig. 11 gives SA, trailing edge region, turbulent viscosity (μ_t) contours for the NSS (Frame (a)), and Poisson (Frame (b)) approaches. The larger $x/c > 1$ Poisson distances, reduce the modeled turbulence destruction thus slightly increasing μ_t by just under 5%. Tests for flow over a thin wire⁴ suggest turbulence levels for the exaggerated distances provided by the HJ and Poisson methods, are more realistic than those provided by the NSS.

Clearly, for the current case, turbulence intensities downstream of the airfoil have insignificant influence on the parameters of interest in a design context. However, for multiple element airfoils it is not inconceivable that more significant solution differences could arise. For example, these can be caused by a peak in modeled turbulence energy convecting close to the center of the leading edge of a downstream element. Then small changes in the turbulence energy peak's position can give rise to different solutions.

The Poisson and NSS methods were also used as part of a single airfoil 'aeroelasticity' analysis comparing with AGARD Case 3 measurements¹⁶. The Poisson equation shows fast convergence and gives instantaneous C_L , C_D and moment coefficient values very similar to NSS. These results are reported in Ref. 13.

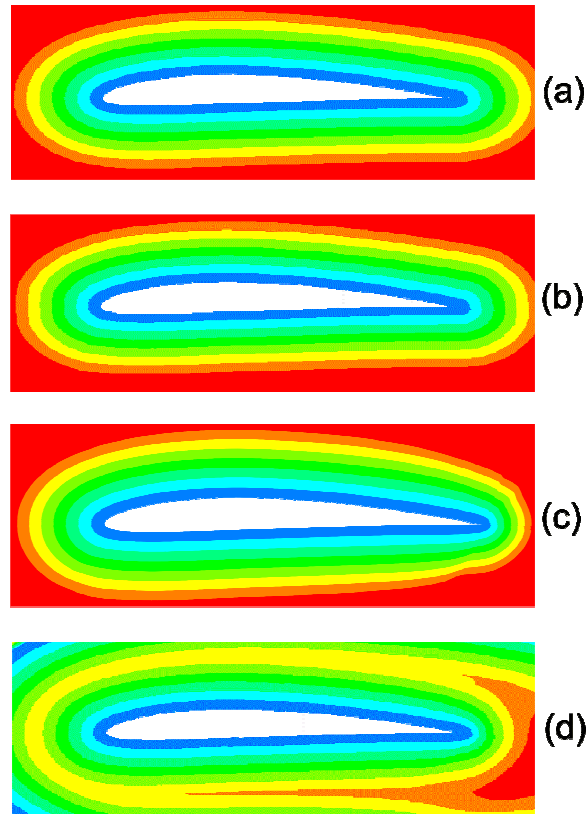


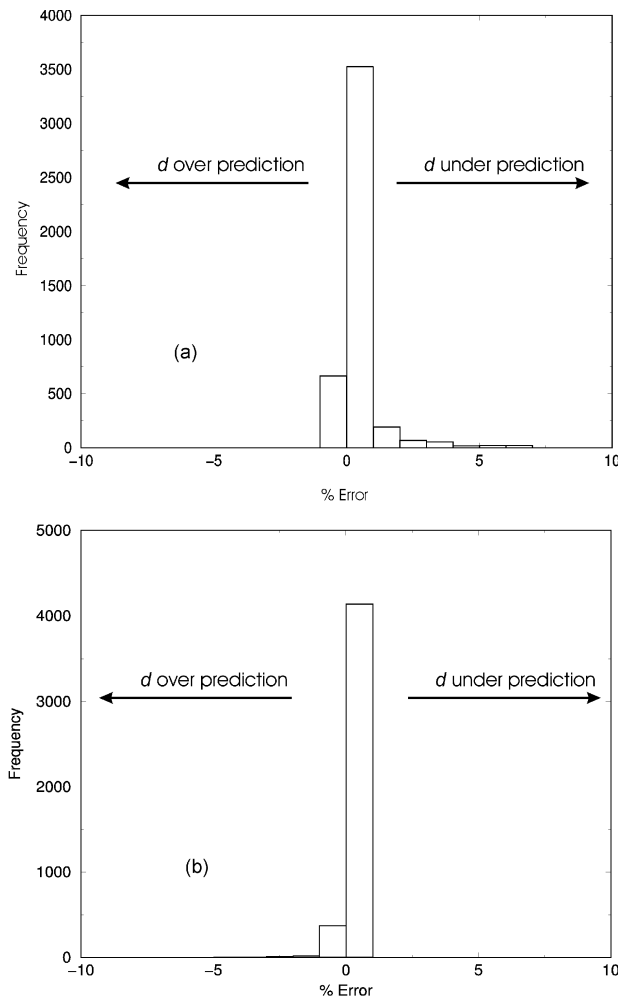
Figure 8: Case (b) (NACA4412) d contours: (a) NSS; (b) Eikonal; (c) Poisson and (d) inner block Poisson.

C. Wing body (Case (c))

For this case the angle of attack is 2.87° , $M = 0.802$ and $Re = 13.1 \times 10^6$ (based on c). The single block grid, for which the body surface zone is shown in Fig. 12, has around 0.9 million cells. For this case the Poisson, Eikonal_I and NSS approaches are tested. Fig. 13 gives the $y^+ < 400$ -region d deviation histogram for the Eikonal_I and Poisson methods. The average 'errors' are 3.13 % and 3.14 %, respectively. The Poisson's tendency to over predict d is evident in the histograms. Table 3 gives C_L and C_D values for the different d fields. For this more complex geometry case, the average differences between C_L and C_D is lower than for Case (b).

D. Wing Flap (Case (d))

For a multi-element airfoil case, the wing and flap angles are 2° and 40° , respectively. The wing-flap gap is 0.6% of c . The Reynolds number is 23×10^6 , based on c , and $M = 0.18$. A small part of the highly stretched 10 block, approximately 0.9 million-cell grid is shown in Fig. 14a. For this case the Poisson, NSS and Eikonal_I methods are tested. Fig. 14b gives Eikonal_I \mathbf{U} vectors. Fig. 15 gives a zoomed in view of area A. Correctly, the vectors are surface normal. Magnitudes are close to unity. Fig. 14c gives Eikonal_I d contours. Since they are virtually identical, NSS contours are not shown. Poisson contours can be found in Ref. 4.



**Figure 9: Case (b) $y^+ < 400$ d deviation histogram:
(a) Eikonal_I and (b) Poisson.**

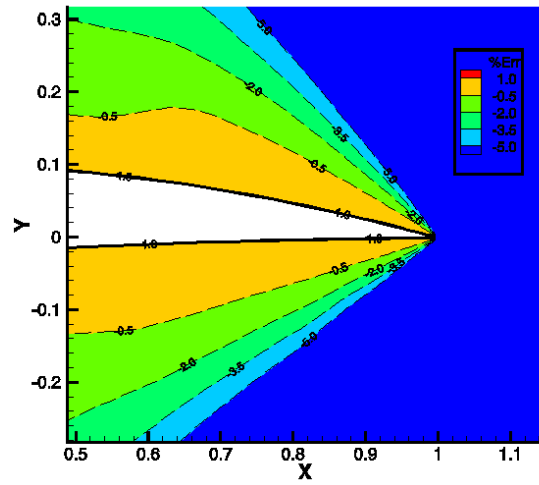


Figure 10: Case (b) (NACA4412) Poisson d deviation contours.

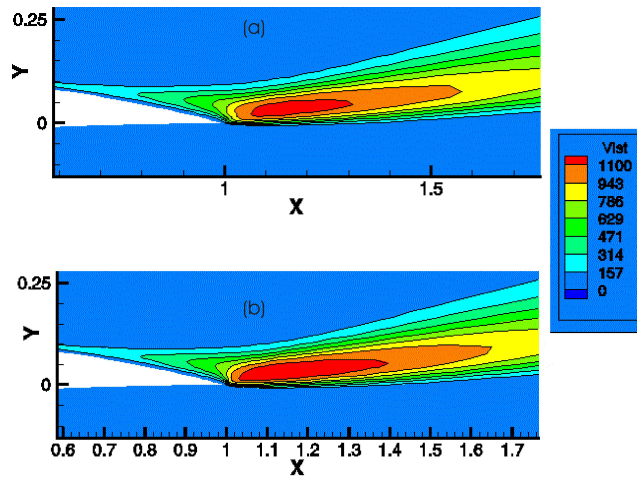


Figure 11: Turbulent viscosity contours: (a) NSS and (b) Poisson.

	C_L	C_D
Eikonal _I	0.6633	0.04839
NSS	0.6632	0.04855
Poisson	0.6635	0.04842

Table 3. Case (c) C_L and C_D 's for different d fields.

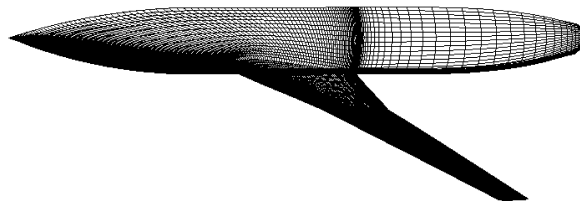


Figure 12. Case (c) (wing body) surface grid.

Fig. 16 gives $y^+ < 400$ d deviation histograms. Frame (a) is for the Eikonal_I. Frame (b) is for the Poisson. The average Frame (a) deviation is 2.1 %. The Poisson equation has a 1% deviation. This low Poisson ‘error’, relative to the Eikonal_I equation seems to contradict the histogram evidence. The contradiction is because the Eikonal equation has some large deviations at just a very few points. Clearly, the Frame (a) Eikonal deviation distribution has the least spread. For the Poisson method (see Frame (b)), it can be seen there is a slight tendency to over predict d . As can be inferred from Table 4, Poisson method lift and drag coefficients are within 0.05% of those for the NSS procedure. For the Eikonal_I equation the deviation is a little greater but not that significant.

Pleasingly, complex geometry tests show HJ_I GGS convergence rate is similar to the Eikonal_I. Hence the HJ_I Laplacian does not seem that important for stability, its value being more in offering the capability of controlling wall distances near convex or concave surfaces.

	C_L	C_D
Eikonal _I	2.815	0.0582
NSS	2.810	0.0584
Poisson	2.811	0.0584

Table 4. Case (d) C_L and C_D 's for different d fields.

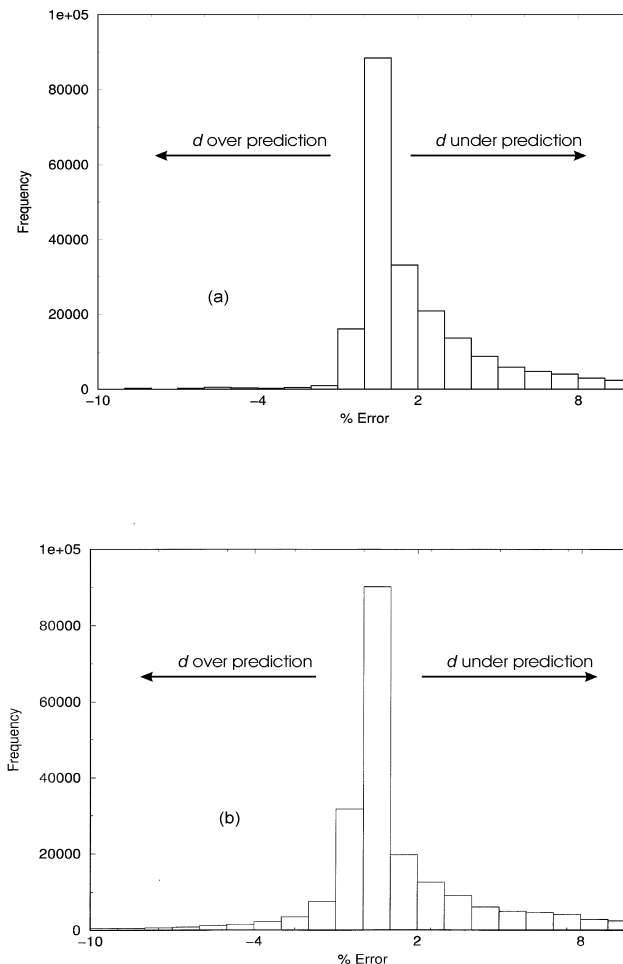


Figure 13: Case (c) (wing body) $y^+ < 400$ d deviation histogram: (a) Eikonal_I and (b) Poisson.

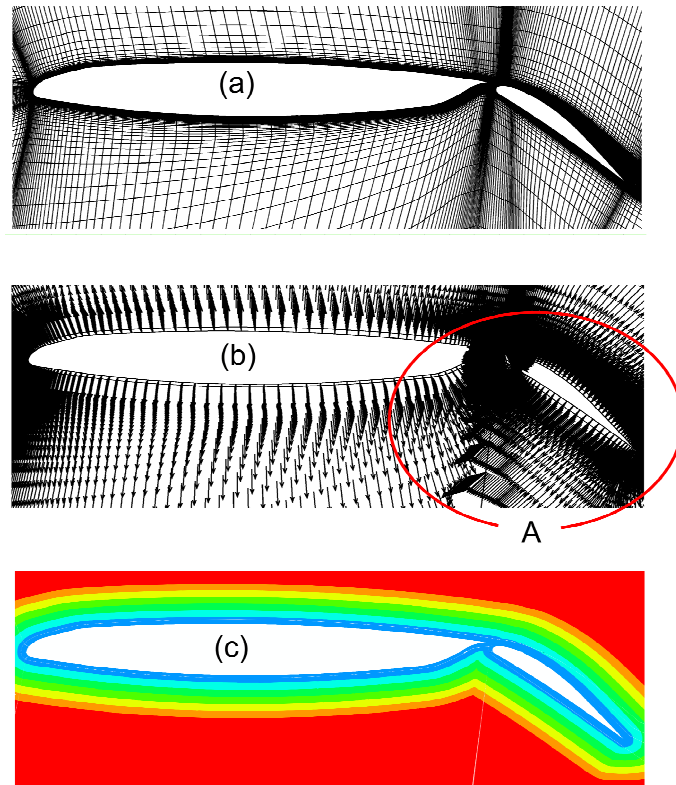


Figure 14: Wing flap geometry near surface information:
 (a) grid; (b) U vectors and (c) d contours.

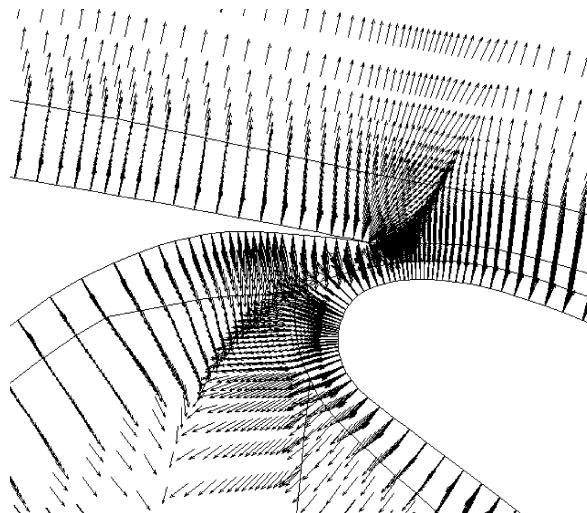


Figure 15: gives a zoomed in view of area A.

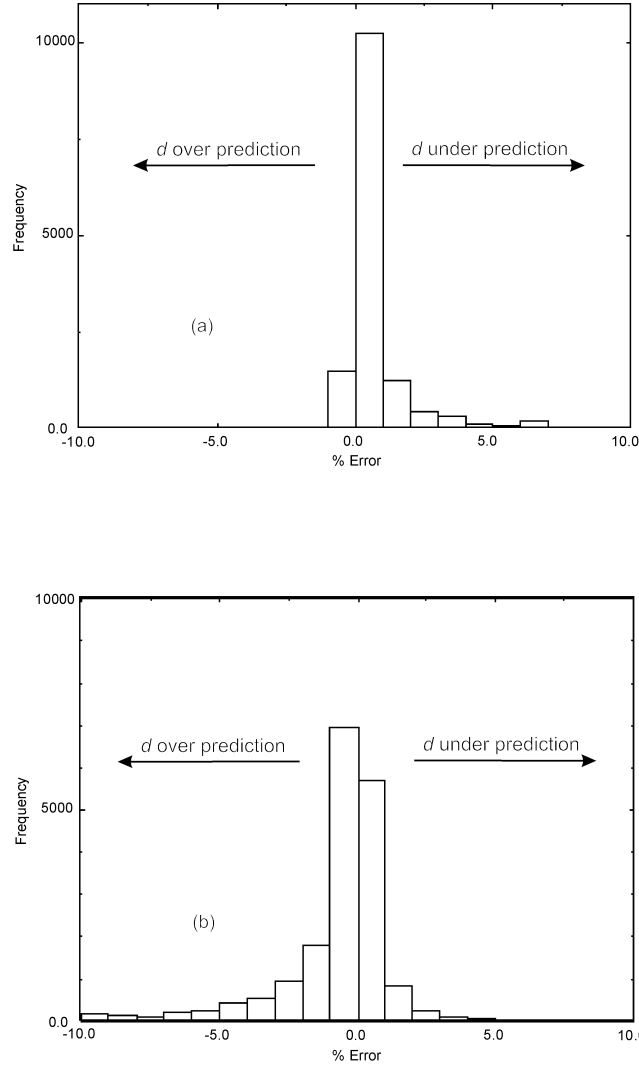


Figure 16: Case (d) (wing flap) $y^+ < 400$ d deviation histogram: (a) Eikonal_I and (b) Poisson.

V. Conclusions

Implicit forms of the Eikonal (Eikonal_I) and Hamilton-Jacobi (HJ_I) wall distance equations were presented. These are reminiscent of advection and advection-diffusion equations. Because of this, the Eikonal_I and HJ_I were found to be relatively easy to implement in an established industrial CFD solver. The implicit d equations were found to have robust convergence. Geometries studied included single and two element airfoils and a wing-body configuration. For HJ_I/Eikonal_I accuracy, offset metric differences are required. A simple, approximate, Poisson-equation-based distance approach was also found effective. Since it did not require offset metric evaluations it was easiest to implement. The sensitivity of flow solutions to plausible modifications of d was explored. Results were not greatly affected by wall distance traits, partly because peculiar geometries such as thin wires were not involved. Downstream of the airfoil trailing edge, intensities were altered by just under 5 %.

Acknowledgments

This work was carried out at NASA Langley. The visit was funded by the UK Royal Academy of Engineering and Engineering Physical Sciences Research Council (grant number GR/S28402/01). I am very grateful for these awards.

References

- ¹Fares E. and Schroder W., "A differential equation to determine the wall distance," *Int. J. for Numerical Methods in Fluids*, Vol. 39, 2002, pp. 743-762.
- ²Strelets M., "Detached eddy simulation of massively separated flows," AIAA Paper 2001-0879, 2001.
- ³Osher S. and Sethian J. A., "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation," *J. Comput. Phys.*, Vol. 79, 1988, pp. 12-49.
- ⁴Tucker P. G., "Differential equation based wall distance computation for DES and RANS," *J. of Comput. Phys.*, Vol. 190, Issue 1, 1st September, 2003, pp. 229-248.
- ⁵Sethian J. A., *Level set methods and fast marching methods*, Cambridge University Press, 1999.
- ⁶Barth T. J., "Steiner triangulation for isotropic and stretched elements," AIAA Paper 95-0213, 1995.
- ⁷Nakahashi K. and Togashi F., "Unstructured overset grid method for flow simulation of complex multiple body problems," *ICAS 2000 Congress*, Paper No. ICAS 0263, 2000.
- ⁸Wigton L. B., *Optimizing CFD codes and algorithms for use on Cray computers*, Frontiers of Computational Fluid Dynamics, edited by D. A Caughey and M. M. Hafez, World Scientific Publishing Co. Pte., Ltd. Chapter 16, 1998, pp. 1-15.
- ⁹Secundov N., Strelets M. Kh., Travin A. K., "Generalization of vt-92 turbulence model for shear-free and stagnation point flows," *J. of Fluids Engineering*, March, Vol. 123, 2001, pp. 111-15.
- ¹⁰Mompean G., Gavrilakis S., Machiels L. and Deville M. O., "On predicting the turbulence-induced secondary flows using non-linear k- ϵ models," *Phys. Fluids*, Vol. 8, No. 7, 1996, pp. 1856-1868.
- ¹¹Launder B. E., Reece G. J. and Rodi W., "Progress in the development of a Reynolds-stress turbulence closure," *J. Fluid Mech.*, Vol. 68, Part 3, 1975, pp. 537-566.
- ¹²Krist S. L., Biedron R. T. and Rumsey C. L., "CFL3D user's manual (version 5.0)," NASA/TM-1998-208444, 1998.
- ¹³Tucker P. G., Rumsey C. L., Bartels R. E. and Biedron R. T., "Transport equation based wall distance computations aimed at flows with time-dependent geometry," NASA/TM-2003-212680, December, 2003.
- ¹⁴Spalart P. R. and Allmaras S. R., "A one-equation turbulence model for aerodynamic flows," *La Rech. Aerospatiale*, Vol. 1, 1994, pp. 5-21.
- ¹⁵Roache P. J., *Computational fluid dynamics*, Hermosa, Albuquerque, N. M., 1976.
- ¹⁶Landon R. H., Compendium of unsteady aerodynamic measurements, "NACA 0012. Oscillatory and transient pitching," Data set 3, AGARD-R-702, 1982.