Some Fascinating New Developments in Kinetic Schemes

S. M. Deshpande[1], N. Anil[2], K. Arora[2], K. Malagi[2] and M. Varma[2]

[1]*Engineering Mechanics Unit, Jawaharlal Nehru Centre for Advanced Scientific Research, Jakkur, Bangalore-560064*
[2]*AR&DB CFD Centre, Department of Aerospace Engg., Indian Institute of Science, Bangalore - 560012*

1 Introduction

Least squares Kinetic Upwind Method (LSKUM) which is a grid free or meshless method is a subject of intense R & D activity in ARDB CFD Centre for a number of years. After the first report by Ghosh, Mandal and Deshpande [5]where basic idea and theory are described, it has seen rapid developments. Ghosh [8]studied the stability, accuracy, defect correction and solved a number of 2 D flow problems thus establishing the method. The LSKUM requires point distribution and a connectivity preprocessor to generate connectivity $N(P_o)$ for each node or point P_o in the computational domain. Connectivity $N(P_o)$ of P_o is a set of neighbouring nodes or points. The LS formula operates on the data at these points to obtain discrete approximation to $(grad\, f)_o$ where f is any function of x,y,z. For 2 D, the least squares formulae are

$$f_{x_o}{}^{(1)} = \frac{\left(\sum w_i \Delta y_i{}^2\right)\left(\sum w_i \Delta x_i \Delta f_i\right) - \left(\sum w_i \Delta x_i \Delta y_i\right)\left(\sum w_i \Delta y_i \Delta f_i\right)}{det}$$

$$f_{y_o}{}^{(1)} = \frac{\left(\sum w_i \Delta x_i{}^2\right)\left(\sum w_i \Delta y_i \Delta f_i\right) - \left(\sum w_i \Delta x_i \Delta y_i\right)\left(\sum w_i \Delta x_i \Delta f_i\right)}{det} \tag{1.1}$$

where

$$det = \left(\sum w_i \Delta x_i{}^2\right)\left(\sum w_i \Delta y_i{}^2\right) - \left(\sum w_i \Delta x_i \Delta y_i\right)^2 \tag{1.2}$$

'Invited talk in "Workshop on Modelling & Simulation in Life Sciences, Materials and Technology" Dec 7-8,2004, Dept. of Mathematics. I.I.T. Chennai - 600036, India

$$\Delta x_i = x_i - x_o, \qquad \Delta y_i = y_i - y_o, \qquad \Delta f_i = f_i - f_o \qquad (1.3)$$

and w_i are the weights to be chosen by the user. Further, xi, y_i are the Cartesian coordinates of $P_i \in N(P_o)$ and f_i is the value of f at P_i. The superscript 1 on f_{x_o}, f_{y_o} in **(1.1)** indicates that these formulae are first order accurate in space.

Ramesh [16] studied various issues such as boundary conditions, various types of clouds of points around many 2 D shapes and computed flow for bodies with these point distributions. Anandhanarayanan [1] classified these clouds into simple clouds, chimera clouds and multiple chimera clouds. A simple cloud is a cloud of points generated by a simple grid generator such as elliptic grid generator in a geometrically simple block or by a triangular grid generator for simple shapes. A chimera cloud is a cloud of points obtained by overlapping two simple grids. For example, for a 2 D cascade we may obtain chimera cloud by overlapping two O-grids around each aerofoil in the cascade. A multiple chimera cloud is obtained by overlapping of several simple clouds. Ramesh [16] successfully applied LSKUM to computation of 2 D flows with these clouds. Central to the application of LSKUM is therefore generation of connectivity for all types of clouds. Anandhanarayanan [1] used a gradient search mthod for chimera clouds and successfully computed flow field and aerodynamic coefficients for flight vehicles with deflected control surfaces. Praveen [15] studied FAME cloud, one of the toughest clouds of points for computation of flow by any algorithm. FAME cloud consists of several overlapping simple clouds with a high degree of clustering near sharp geometric features. Praveen faced unprecedented difficulties in dealing with bad connectivity [15]. Connectivity is called nearly degenerate when the *det* defined by (1.2) is very close to zero. This is just one example of bad connectivity which causes code divergence or "blowing up" - a term commonly used by code developers. Other examples of bad connectivity are : highly anisotropic distribution of points; large difference between $d_{min}(P_o, P_i)$ and $d_{max}(P_o, P_i)$, $P_i \in N(P_o)$; nearly empty quadrants, etc. Large inaccuracy in computation of derivatives on such connectivity sets gives very inaccurate results and in extreme cases causes code divergence.

Mahendra [13] applied LSKUM to computation of strongly rotating viscous flows involving Ekman and Stewartson layers. Highly clustered grids near end plates and walls of rotating cylinder are necessary for resolution of fine scales in flow and these in turn lead to highly anisotropic connectivity. Dauhoo [4] used entropy variables called q-variables which greatly

simplify defect correction step in obtaining higher order accuracy. Very accurate computations have been done using q-LSKUM by Dauhoo [4]and Deshpande etal.[6]. Anandhanarayanan [1] has also used q-LSKUKM for computation of subsonic,transonic and supersonic flows around practical flight vehicles.

Improvements in solution algorithm are required in many directions. One direction of research as mentioned before is a robust solver and better connectivity.Yet another direction is low dissipation flux splitting and further introduce rotational invariance, that is, develop upwinding without symmetry breaking. Another line of research is a new method of point generation without use of grid generator.

2 LSKUM with eigenvector basis

The matrix A (w) associated with weighted Least Squares method for *2* D is

$$A(w) = \begin{bmatrix} \sum w_i \Delta x_i{}^2 & \sum w_i \Delta x_i \Delta y_i \\ \sum w_i \Delta x_i \Delta y_i & \sum w_i \Delta y_i{}^2 \end{bmatrix} \tag{2.4}$$

Here the summation is over P_i in $N(P_o)$. The corresponding linear algebraic equation is

$$A(w)\,(grad\ f)^T = \begin{bmatrix} \sum w_i \Delta x_i \Delta f_i \\ \sum w_i \Delta y_i \Delta f_i \end{bmatrix} \tag{2.5}$$

The solution of ***(2.5)*** is given by **(1.1).** The formulae **(1.1)**are standard *2* D formulae and use of these formulae in LSKUM based state update has been found to be quite useful for a variety of problems. However there are some examples of bad connectivity related to code divergence and there is a need to understand what is bad connectivity. While this analysis is going on, it is also essential to improve the robustness of LS formulae employed for spatial derivatives [3]. Following Arora and Deshpande [3], we want to determine weights w_i such that

$$\sum w_i \Delta x_i \Delta y_i = 0 \tag{2.6}$$

In such a case the matrix $A(w)$ becomes diagonal and x, y directions are along eigenvectors of A (w). *2* D formulae then collapse to 1 D formulae

$$f_{x_o}{}^{(1)} = \frac{(\sum w_i \Delta x_i \Delta f_i)}{\sum w_i \Delta x_i{}^2}, \qquad f_{y_o}{}^{(1)} = \frac{(\sum w'_i \Delta y_i \Delta f_i)}{\sum w'_i \Delta y_i{}^2} \tag{2.7}$$

The weights required for x and y derivative are different as will become clear later. The weights can be very easily determined by referring to the quadrants shown in Fig.(1). Introduce the notation $C^I_{xy}, C^{II}_{xy}, C^{III}_{xy}$ and

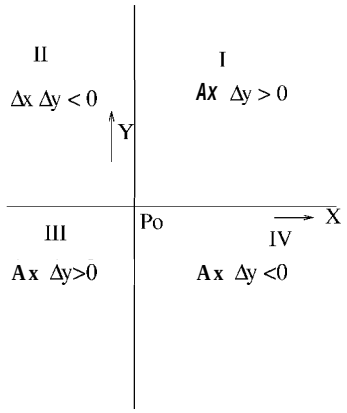


Figure 1: Quadrants for the split stencil of a node

C^{IV}_{xy} for cross terms over the quadrants I, II, III, IV respectively, that is

$$\begin{aligned} C^I_{xy} &= \sum\nolimits_I \Delta x_i \Delta y_i,\ C^{II}_{xy} = \sum\nolimits_{II} \Delta x_i \Delta y_i, \\ C^{III}_{xy} &= \sum\nolimits_{III} \Delta x_i \Delta y_i,\ C^{IV}_{xy} = \sum\nolimits_{IV} \Delta x_i \Delta y_i \end{aligned} \tag{2.8}$$

The summation in (2.8) is over respective quadrants. Obviously

$$C^I_{xy} > 0,\ \ C^{III}_{xy} > 0 \quad \text{and} \quad C^{II}_{xy} < 0,\ \ C^{IV}_{xy} < 0 \tag{2.9}$$

For upwinding, stencil division is normally done, that is for x-derivative we take sub stencil (SS) $= I + IV$ with $\Delta x_i > 0$ and also SS $= II + III$ depending on the direction of information propagation. Taking SS $= I + IV$ we now enforce

$$w_I C^I_{xy} + w_{IV} C^{IV}_{xy} = 0 \tag{2.10}$$

giving

$$w_I \quad - \quad \frac{C^I_{xy}}{C^{IV}_{xy}} > 0 \tag{2.11}$$

Thus for sub stencil $I + IV$, the condition

$$\sum\nolimits_{I+IV} w_i \Delta x_i \Delta y_i = 0$$

can be easily satisfied with

$$\begin{aligned} w_i &= w_I \qquad \text{for } i \in I \\ w_i &= w_{IV} \qquad \text{for } i \in \text{I V} \end{aligned} \tag{2.12}$$

Evidently, we can choose $w_I = 1$ and then obtain

$$w_I = 1, \qquad w_{IV} = -\frac{C_{xy}^I}{C_{xy}^{IV}} \tag{2.13}$$

The x-derivative of f given by Eq.(2.7) therefore becomes

$$f_{x_o}{}^{(1)} = \frac{\sum_{i\in I} w_I \Delta x_i \Delta f_i + \sum_{i\in IV} w_{IV} \Delta x_i \Delta f_i}{\sum_{i\in I} w_I \Delta {x_i}^2 + \sum_{i\in IV} w_{IV} \Delta {x_i}^2} \tag{2.14}$$

For the y-derivative, we take the SS $= I + II$ or SS $= III + IV$ again depending on the direction of velocity. Choosing SS $= I + II$, we want to enforce

$$\sum_{I+II} w'_i \Delta x_i \Delta y_i = 0$$

Again using the cross terms, we obtain

$$\frac{w'_{II}}{w'_I} = -\frac{C_{xy}^I}{C_{xy}^{II}} > 0 \tag{2.15}$$

These weights are different from those given by (2.11). The y-derivative in Eq.(2.7) now becomes

$$f_{y_o}{}^{(1)} = \frac{\sum_{i\in I} w'_I \Delta y_i \Delta f_i + \sum_{i\in II} w'_{II} \Delta y_i \Delta f_i}{\sum_{i\in I} w'_I \Delta {y_i}^2 + \sum_{i\in II} w'_{II} \Delta {y_i}^2} \tag{2.16}$$

It is clear that given any sub stencil (SS) we can determine positive weights which make $A(w)$ diagonal. Second order accuracy can be easily obtained by using defect correction technique **[3]** and we have

$$f_{x_o}{}^{(2)} = \frac{\sum w_i \Delta x_i \widetilde{\Delta f_i}}{\sum w_i \Delta {x_i}^2} \qquad f_{y_o}{}^{(2)} = \frac{\sum w'_i \Delta y_i \widetilde{\Delta f_i}}{\sum w'_i \Delta {y_i}^2} \tag{2.17}$$

where

$$\widetilde{\Delta f_i} = \Delta f_i - \frac{\Delta x}{2} \Delta f_{xi} - \frac{\Delta y}{2} \Delta f_y \tag{2.18}$$

Konark and Deshpande **[3]** have used the above formulae in LSKUM for making 2 D calculations. Figs.(2) and (3) show the residue plots for computation of subsonic flow past NACA 0012 aerofoil at $M_\infty = 0.63$ and A.O.A. $\alpha = 2°$. Both first and second order computations have been performed. The residue for the second order calculations with weighted LSKUM falls more than that for unweighted LSKUM. To demonstrate the ability of the new method to work with bad connectivity, a point distribution generated by Delaunay triangulation as shown in Fig.4 has been used. The triangulation near the aerofoil has been intentionally tampered so as to make it an extreme case of bad connectivity. The initial connectivity of

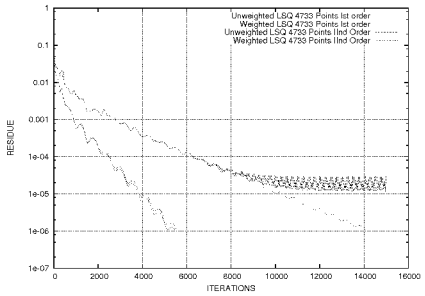


Figure 2: Residue drop for weighted and unweighted LSKUM : 4733 Points

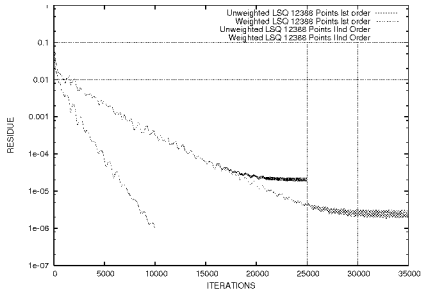


Figure 3: Residue drop for weighted and unweighted LSKUM : 12388 Points

the node is shown in Fig.5. The node under consideration is indicated by a circle in Fig.5. After tampering with triangulation, the good connectivity has been converted into an extremely bad connectivity as shown in Fig.6. Then, the tampered point distribution (with bad connectivity) was used as an input for the code using weighted LSKUM and unweighted LSKUM. As was expected, the unweighted LSKUM code was unable to run on this extreme case of bad connectivity as evident from the Fig.7. However, the weighted LSKUM code did not encounter any problems and was successful in generating results for the subsonic test case as is evident from the residue plot shown in Fig.8. The pressure contours are shown in Fig 9.

3 Low dissipation KFVS

KFVS has been extensively used for doing Euler computations [1, 13, 15] of flows past many flight vehicle configurations. Viscous computations of laminar flows have been done by Mathur, Anandhanarayanan and Mahendra [13, 1, 14]. It is still pertinent to study whether numerical dissipation inherent in first order accurate schemes using KFVS can be reduced either for

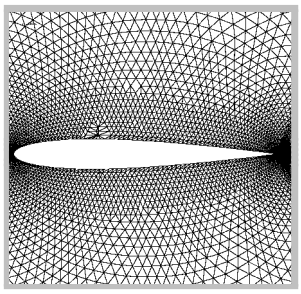


Figure 4: Bad connectivity on aerofoil : 7269 points in domain

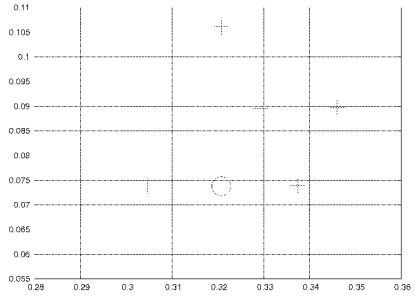


Figure 5: Original good connectivity

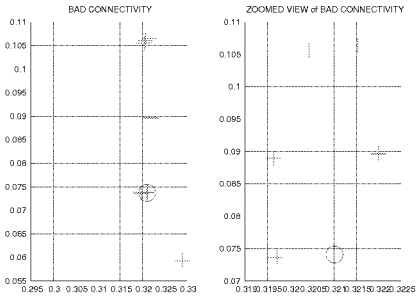


Figure 6: Bad connectivity

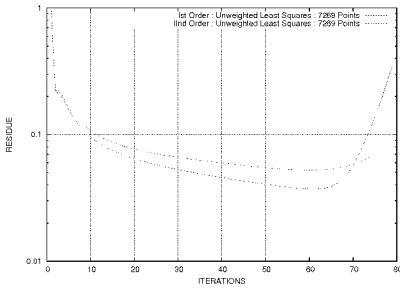


Figure 7: Residue drop for unweighted least squares using bad connectivity : First Order and Second Order

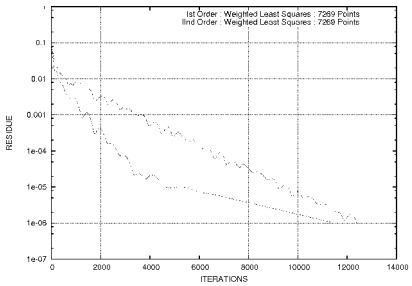


Figure 8: Residue drop for weighted least squares using bad connectivity : First Order and Second Order

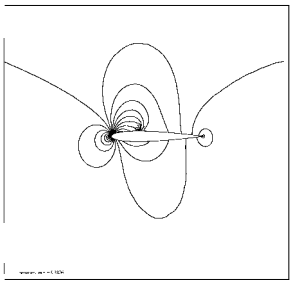


Figure 9: Pressure contours : Weighted LSKUM : Second order : Bad connectivity

obtaining single point shock capture or for getting low dissipation schemes. For this purpose we have developed Modified CIR (MCIR) splitting method. To explain the basic principle behind MCIR (for details see [17, 2]) .

Consider the $1D$ linear hyperbolic pde for scalar f

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}(vf) = \frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} = 0 \qquad \textbf{(3.19)}$$

MCIR splitting gives

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}\left(\frac{v+|v|\phi}{2}f\right) + \frac{\partial}{\partial x}\left(\frac{v-|v|\phi}{2}f\right) = 0 \qquad (3.20)$$

Here ϕ is a dissipation control parameter which can be a function of v. Considering a uniform grid of points with spacing Ax and using first order forward and backward different formula for the derivatives in eq. (3.20), we get

$$\left(\frac{\partial f}{\partial t}\right)_j + \frac{v+|v|\phi}{2}\frac{f_j - f_{j-1}}{\Delta x} + \frac{v-|v|\phi}{2}\frac{f_{j+1} - f_j}{\Delta x} = \qquad (3.21)$$

The corresponding modified pde (mpde) is

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}(vf) = \frac{\Delta x}{2}|v|\phi\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2}{\partial x^2}\left(|v|\frac{\Delta x}{2}\phi f\right) \qquad (3.22)$$

giving the kinematic numerical viscosity ν'_{NUM} as

$$\nu'_{NUM} = |v|\frac{\Delta x}{2}\phi \qquad \textbf{(3.23)}$$

Evidently, kinematic numerical viscosity is a product of length scale $\frac{1}{2}\Delta x$ and velocity scale $|v|$ and a factor ϕ which can be used for controlling the numerical viscosity. It is interesting to note that $\phi = 0$ gives central difference scheme while $\phi = 1$ gives first order upwind scheme known to contain a large amount of numerical diffusion leading to smearing of contact discontinuities and shocks. If we identify f with Maxwellian distribution F given by

$$F = \frac{\rho}{\pi}\sqrt{\frac{\beta}{\pi}}e^{-\beta(v-u)^2 - I/I_0} \qquad (3.24)$$

and take Ψ moments of eq. (3.20), we get modified kinetic split fluxes (m-$kfvs$)

$$Gm^{\pm} = \int_{\mathbb{R}^+\times\mathbb{R}} \frac{v \pm |v|\phi}{2}\psi F dv dI \qquad \textbf{(3.25)}$$

where Ψ is the moment function vector given by

$$\Psi = \begin{bmatrix} 1 \\ v \\ I + \frac{v^2}{2} \end{bmatrix}$$

The notation here has standard meaning used so often in KFVS [12]. The corresponding split Euler equations are

$$\frac{\partial U}{\partial t} + \frac{\partial}{\partial x}\left(Gm^+\right) + \frac{\partial}{\partial x}\left(Gm^-\right) = 0 \tag{3.26}$$

Backward and forward differencing of spatial derivatives in eq. (3.26) gives modified *KFVS (m-KFVS)* differenced Euler equations as

$$\underline{Gm_j^+ - Gm_{j-1}^+} \perp \underline{Gm_{j+1}^- - Gm_j^+} \tag{3.27}$$

which can be cast in semidiscrete conservation law

$$\Delta x \left(\frac{dU}{dt}\right)_j + G_{j+\frac{1}{2}} - G_{j-\frac{1}{2}} = 0 \tag{3.28}$$

The fluxes $G_{j\pm\frac{1}{2}}$ on cell faces $j \pm \frac{1}{2}$ are given by

$$\begin{aligned} G_{j+\frac{1}{2}} &= Gm_j^+ + Gm\bar{I}+1 \\ G_{j-\frac{1}{5}} &= Gm_{j-1}^+ + Gm_j^- \end{aligned} \tag{3.29}$$

The coresponding mpde for the eq. (3.27) is given by

$$\frac{\partial U}{\partial t} + \frac{\partial G}{\partial x} = \frac{\partial^2}{\partial x^2}\left(\int\limits_{\mathbb{R}^+\times\mathbb{R}} |v|\frac{\Delta x}{2}\phi\Psi F dv dI\right) \tag{3.30}$$

which again shows that $\phi = \phi(v)$ can be regarded as a dissipation controlling function. Taking the first component of U, i.e., ρ we get from eq. (3.30)

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}\left(\rho u\right) = \frac{\partial^2}{\partial x^2}\left| \quad \rho|v|\frac{\Delta x}{2}\phi\sqrt{\frac{\beta}{\pi}}e^{-\beta(v-u)^2}dv \right| \tag{3.31}$$

which clearly shows that

$$\nu_{NUM} = \qquad |v|\frac{\Delta x}{2}\phi\sqrt{\frac{\beta}{\pi}}e^{-\beta(v-u)^2}dv \tag{3.32}$$

Obviously, when $\phi = 1$, maximum contribution to the integral comes from particles with $v \simeq u$ and the contribution becomes vanishingly small as $|v| \to \infty$. This suggests that we should choose ϕ so that it is small near

$|v| \simeq u$. It is not required to make ϕ small as $|v| \to \infty$. One possible choice is

$$\phi = e^{-\frac{\alpha}{|v|}} \tag{3.33}$$

which gives $\phi = 1$ as $|v| \to \infty$ and $\phi = 0$ as $v \to 0$. The condition $\frac{\alpha}{|v|} \gg 1$ gives $|v| \ll a$, that is particles having $|v| \ll a$ will contribute very little to ν_{NUM}. Detailed arguments in favour of the choice **(3.33)** based on BGK model have been given by Anil and Deshpande [2].

Yet another choice is

$$\phi(v) = e^{-\alpha|v|} \tag{3.34}$$

which gives $\phi \to 0$ as $|v| \to \infty$ and $\phi = 1$ as $|v| \to 0$ which gives another weighting in velocity space. By suitably adjusting α in (3.34) we get the same effect as the choice **(3.33)** so far as ν_{NUM} is concerned. The choice **(3.34)** as shown by Anil and Deshpande [ref]leads to much simpler expressions for $Gm^{\pm}$. Infact, the formulae for $Gm*$ with $\phi(v) = e^{-\alpha|v|}$ are

$$Gm^{\pm} = \frac{G}{2} \pm \frac{1}{2}\left[e^{\left(\frac{\alpha^2}{4\beta}-\alpha u\right)} G^{+}\left(u - \frac{\alpha}{2\beta}\right) - e^{\left(\frac{\alpha^2}{4\beta}+\alpha u\right)} G^{-}\left(u + \frac{\alpha}{2\beta}\right)\right] \tag{3.35}$$

where G is the unsplit flux, $G^{\pm}$ are the usual *kfvs* split fluxes for the $1D$ Euler equations. The first order accurate (in time and space) state update is obtained by time discretisation of 3.27, that is,

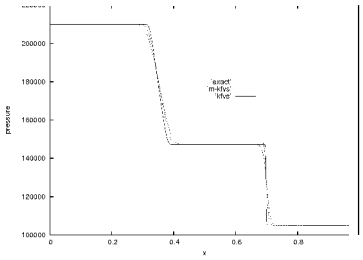
$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}\left(G^n_{j+\frac{1}{2}} - G^n_{j-\frac{1}{2}}\right) \tag{3.36}$$

Where $G^n_{j\pm\frac{1}{2}}$ are given by eq. **(3.30).** It is interesting to note that by suitably choosing a, that is by reducing numerical dissipation one can make **(3.36)** nearly second order accurate in space. This is one of the significant advantages of modified KFVS (m-KFVS) expressions (3.35).
Anil and Deshpande[2] have applied the above method to standard $1D$ shock tube problem, so often used in CFD for testing a new algorithm. The initial conditions are chosen with a pressure jump of 2 across the shock. Numerical solutions are performed on the computational domain $0 \leq x \leq 1$ with 500 equally spaced cells. Near the discontinuity we have chosen $\alpha = 0.4\sqrt{\beta_r}$ and in smooth regions, the value of a is given by $a = 0.91\sqrt{\beta_r}$.

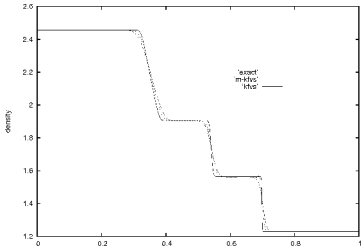
The *m-kfvs* scheme has been applied to a convergent-divergent nozzle problem [10]. This is a steady state test case with flow being quasi-one-dimensional:

inviscid and compressible. Numerical simulations are performed on a uniform grid of 61 cells with $a = 0.76\sqrt{\beta_r}$ near the discontinuity and $a = 0.794\sqrt{\beta_r}$ in smooth regions. Note that, β_r is the reference value of β based on free stream conditions.

It is clear from figures 10 and 11 that the *m-kfvs* based update scheme becomes nearly second order accurate and captures shock far more crisply than the usual first order *kfvs* method. Infact, the potential value addition of the above idea to existing finite volume or grid free codes based on *kfvs* is immense. Just by changing the expressions for the split fluxes in the flux calculation subroutine will yield *m-kfvs* based codes having less numerical disipation, resulting in accurate capture of leading edge suction, crisp shocks, negligible loss of stagnation pressure in isentropic regions and accurate prediction of vortex dominated flows.



(a) Pressure Plot

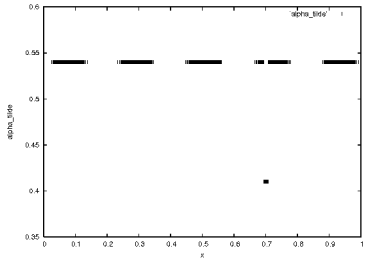


(b) Density Plot

4 Rotationally Invariant LSKUM

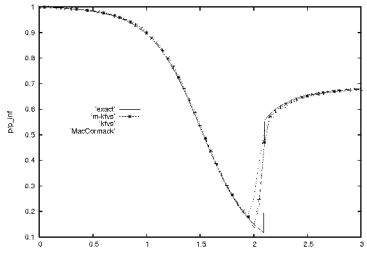
Deshpande [7]has studied the question of symmetry of differential operators and their discrete approximations. In kinetic schemes one often deals with differential operator

$$D(f) = v_1 \frac{\partial f}{\partial} + v_2 \frac{\partial f}{\partial} \tag{4.37}$$

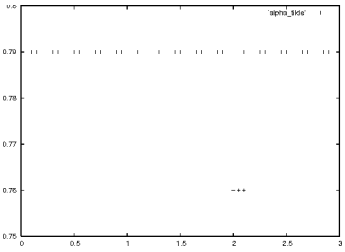


(c) Variation of $\tilde{\alpha}$

Figure 10: Shock tube problem using m-$kfvs$ with $\phi = e^{-\alpha|v|}$. Computed results are compared with $kfvs$ scheme and exact solution.



(a) Pressure distribution through the nozzle



(c) Variation of $\tilde{\alpha}$

Figure 11: Convergent divergent nozzle problem. Computed results are compared with second order accurate MacCormack scheme and exact solution.

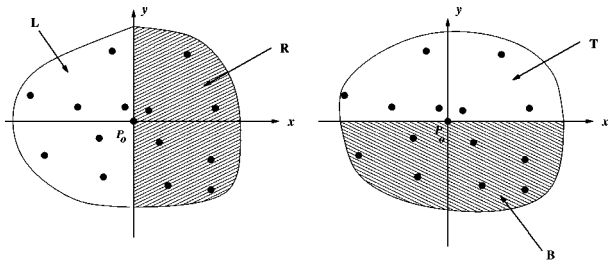


Figure 12: stencil division

which arises in Boltzmann equation.The differential operator $D(f)$is invariant under 2-D rotation group $\mathbf{O}(2)$.For rotation of axes through 6 we have

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.38}$$

giving co-ordinates of a point P in rotated frame in terms of $x-$ y frame. The matrix

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{4.39}$$

is an element of $\mathbf{O}(2)$. Obviously R **(0)** satisfies the relation

$$R(\theta)\,R(\phi) = R(\theta+\phi)$$

Let us now consider LS approximation to f_{xO},f_{yO} at point P_O. Then the discrete differential operator is

$$DD(f) = v_1 f_{x0}^{(1)} + v_2 f_{y0}^{(1)} \tag{4.40}$$

which as Deshpande [7]has shown is invariant under $\mathbf{O}(2)$. Here $\mathbf{O}(2)$ is a symmetry group of $D(f)$. However we use finite difference scheme to discretise f_{xj} and f_{yj}, then the resultant approximation

$$FD(f) = v_1 \frac{f_{i+1,j} - f_{i-1,j}}{2\mathrm{Ax}} + v_2 \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta y} \tag{4.41}$$

on a 2-D regular mesh is not invariant under $\mathbf{O}(2)$. It is invariant under a subgroup of $\mathbf{O}(2)$ and this property is called symmetry breaking by Deshpande [7]. Deshpande [7]has also shown that upwinding generally breaks symmetry. For example, upwinding is enforced in LSKUM by stencil division. Consider a stencil or connectivity of P_0 shown in Fig. 15.

The division of stencil is also shown. R consists of points for which $\Delta x_i > 0$

and L consists of points for which $\Delta x_i < 0$. Similarly,

$$\begin{aligned} T &= \{P_i \in N(P_0) \ / \ \Delta y_i > 0\} \\ B &= \{P_i \in N(P_0) \ / \ \Delta y_i < 0\} \end{aligned} \tag{4.42}$$

The upwind approximation to $D(f)$ used in LSKUM is given by

$$\begin{aligned} UD(f) \ = \ & \frac{v_1 + |v_1|}{2} \left[f_{x0}^{(1)}\right]_L + \frac{v_1 - |v_1|}{2} \left[f_{x0}^{(1)}\right]_R \\ & + \frac{v_2 + |v_2|}{2} \left[f_{y0}^{(1)}\right]_B + \frac{v_2 - |v_2|}{2} \left[f_{y0}^{(1)}\right]_T \end{aligned} \tag{4.43}$$

where L, R, B and T indicate the substencils over which summation is taken in leastsquares approximation. Define a subgroup of $\mathbf{O}(2)$ consisting of

$$E = \{ R(0), R(\pi/2), R(\pi), R(3\pi/2) \} \tag{4.44}$$

The upwind approximation equation(4.43) is invariant under $E \subset \mathbf{O}(2)$. Thus upwinding along co-ordinate directions based on stencil division is symmetry breaking. A question therefore arises: Can we construct an upwind approximation to eqn(37) which is also rotationally invariant i.e., no symmetry breaking is allowed. Keshav and Deshpande [11] have precisely constructed a discrete upwind rotationally invariant least squares method. This method uses the easily provable identity

$$\frac{\partial}{\partial x}(GX) + \frac{\partial}{\partial y}(GY) = \frac{1}{\pi} \int_0^{2\pi} \frac{\partial}{\partial s} \left(\vec{Q} \cdot \hat{\ell}\right) \mathrm{d}\theta \tag{4.45}$$

where GX and GY are flux vectors, $\vec{Q} = (GX)\hat{i} + (GY)\hat{j}$,
$\hat{\ell}$ is unit vector making an angle θ with $x-axis$ and s is the distance along $\hat{\ell}$.

It may be noted that the divergence of flux vector, div$\vec{Q}$, is an integral of derivative of $\vec{Q} \cdot \hat{\ell}$, this derivative can be approximated by 1-D formula. Thus multidirectional upwind discretisation of div$\vec{Q}$ is an integral over 1-d approximations in all directions. Following this argument we write 2-D Euler equations as,

$$\frac{\partial U}{\partial t} + \frac{1}{\pi} \int_0^{2\pi} \left[\frac{\partial}{\partial s} \left(\vec{Q} \cdot \hat{\ell}\right)^+ + \frac{\partial}{\partial s} \left(\vec{Q} \cdot \hat{\ell}\right)^- \right] \mathrm{d}\theta = 0 \tag{4.46}$$

where 1-D derivative w.r.t. 's' has been split into two derivatives of split fluxes. The next step consists of approximating these split flux derivatives. Denote .

$$\zeta(\theta) = \frac{\partial}{\partial s} \left(\vec{Q} \cdot \hat{\ell}\right)^+ + \frac{\partial}{\partial s} \left(\vec{Q} \cdot \hat{\ell}\right)^- \tag{4.47}$$

Let us take a point P_0 sorrounded by nodes P_i in its connectivity $N(P_0)$ as shown below. The directional derivative ζ **(0)** along $\hat{\ell}$ can be determined

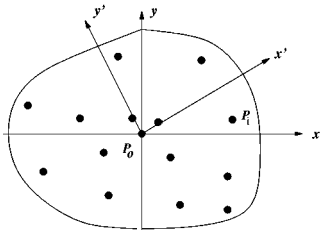


Figure 13: Rotated co-ordinate frame

by using least squares. Let x'-y' be a rotated frame with x' along θ_i and y' perpendicular to it. Then,

$$\zeta(\theta) = \frac{\partial}{\partial x'}\left(\vec{Q}\,.\,\hat{\ell}\right)^{+} + \frac{\partial}{\partial x'}\left(\vec{Q}\,.\,\hat{\ell}\right)^{-} \tag{4.48}$$

and $\vec{Q} = (GX)\hat{\imath} + (GY)\hat{\jmath}$
Let.

$$E^{+} = \left(\vec{Q}\,.\,\hat{\ell}\right)^{+},\ E^{-} = \left(\vec{Q}\,.\,\hat{\ell}\right)^{-}$$

The components of E'^{+} and E'^{-} are denoted as,$E'(i)^{+}$ and $E'(i)^{-}$, $i = 1,2,3,4$ respectively. *The special split fluxes* E^{+} and E^{-} are given by,

$$\begin{aligned}
E'(1)^{\pm} &= GX'(1)^{\pm} \\
E'(2)^{\pm} &= \cos\theta GX'(2)^{\pm} - \sin\theta GX'(3)^{\pm} \\
E'(3)^{\pm} &= \sin\theta GX'(2)^{\pm} + \cos\theta GX'(3)^{\pm} \\
E'(4)^{\pm} &= GX'(4)^{\pm}
\end{aligned} \tag{4.49}$$

The split fluxes $GX'^{\pm}$ in (4.70) are the same as usual KFVS fluxes of Mandal and Deshpande [12]. Note the prime notation in the above formula, it refers to x'-y' frame i.e., the fluxes involved should be computed using values of variables in x'-y' frame. The $GX'^{\pm}$ are given by,

$$\begin{aligned}
GX'(1)^{\pm} &= \rho u_1 A_1^{\pm} \pm \rho B_1 \\
GX'(2)^{\pm} &= \left(p + \rho u_1^2\right) A_1^{\pm} \pm \rho u_1 B_1 \\
GX'(3)^{\pm} &= \rho u_1 u_2 A_1^{\pm} \pm \rho u_2 B_1 \\
GX'(4)^{\pm} &= \left\{\frac{\gamma}{\gamma-1}p + \frac{1}{2}\rho(u_1^2+u_2^2)\right\} u_1 A_1^{\pm} \pm \left\{\frac{\gamma+1}{2(\gamma-1)}p + \frac{1}{2}\rho(u_1^2+u_2^2)\right\} B_1
\end{aligned} \tag{4.50}$$

$$A_1^{\pm} = \frac{1 \pm erf(s_1)}{2} \qquad B_1 = \frac{e^{-s_1^2}}{2\sqrt{\pi\beta}}$$
$$s_1 = u_1\sqrt{\beta} \qquad \beta = \frac{1}{2RT}$$

In terms of $E^{\pm}$ the directional derivative $\zeta(\theta_i)$ can be determined using least squares as,

$$\zeta(\theta_i) = \left\{ \frac{\sum(\Delta E^{+}\Delta x_i')\sum \Delta y_i'^2 - \sum(\Delta E^{+}\Delta y_i')\sum \Delta x'_i \Delta y'_i}{det(L)} \right\}_{i \epsilon L(P_0)} + \left\{ \frac{\sum(\Delta E^{-}\Delta x_i')\sum \Delta y_i'^2 - \sum(\Delta E^{-}\Delta y_i')\sum \Delta x'_i \Delta y'_i}{det(R)} \right\}_{i \epsilon R(P_0)} \tag{4.51}$$

where.

$$det = \left(\sum \Delta x_i'^2\right)\left(\sum \Delta y_i'^2\right) - \left(\sum \Delta x'_i \Delta y'_i\right) \tag{4.52}$$

and det(L) and det(R) are obtained by taking summation in (i) over $i \epsilon L(P_0)$ and $i \epsilon N(P_0)$ respectively.

RESULTS: The method has been tested on shock reflection problem.The mach no is 2.9 and $\theta = 11$. The computational domain consists of 312 x 112 points equally divided in a domain of $0 \leq x \leq 3$ and $0 \leq y \leq 1$.One can observe the good residue fall.

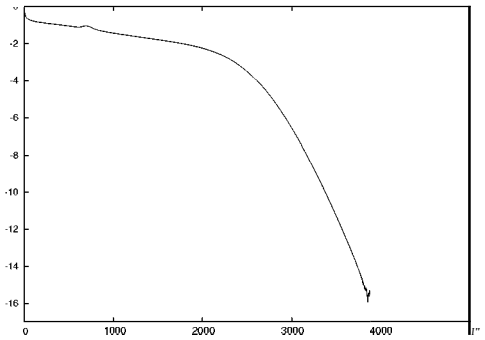


Figure 14: Residue plot

5 Point Generation

Anandanarayanan[1] and Praveen[15] used q-LSKUM to solve flows over many complex geometries. Anandanarayanan developed algorithms to ob-

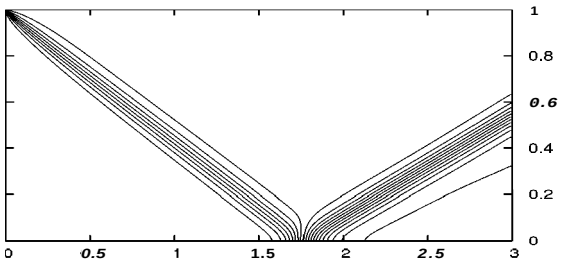


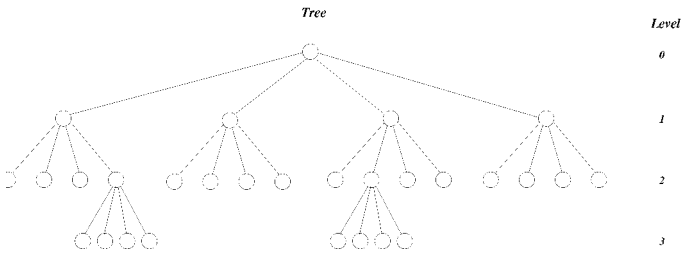
Figure 15: pressure contour plot

tain point distributions and point neighborhoods from overlapping grids. He generated overlapping meshes by decomposing the complex body into geometrically simple components and generating grids around each component independently. He found that, for flows involving fin deflections, use of q-LSKUM with overlapping grids with an appropriate transformation for each individual grid reduces the overall turnaround time for flow simulation considerably. Praveen deviced algorithms to obtain the point distributions and point neighborhoods from a FAME data structure. Though, flows around many complex geometries are solved using q-LSKUM, a need for a fast and fully automatic algorithm to obtain point distributions and point neighborhoods is felt. Effective use of the meshless methods is possible only with the development of such algorithms. This work is a step in that direction.

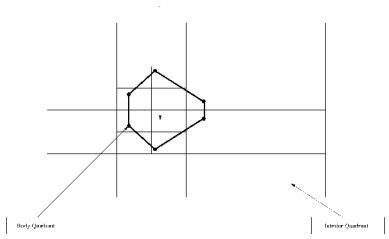
This algorithm uses quadtree in two dimensions and octree in three dimensions for generating point distributions and point neighborhoods. Quadtree and Octree are hierarchical data structures which are found to have applications in many fields of engineering[19]. In mesh generation, these are used generate structured and unstructured meshes[20] and hierarchically refined Cartesian meshes[21]. Octree based Cartesian mesh generation provides a fast and highly automatic mesh generation algorithm. However, meshes obtained using this procedure contain cut-cells of various shapes and sizes in the vicinity the solid boundaries. **A** fair amount of book-keeping is required to keep track of cut-cells of hundreds of types. The process of clipping in which the cut-cell faces are computed is com-

putationally very expensive. These cut-cells pose special problems in the implementation of state update as well as boundary condtions. This results in poor solution accuracy and convergenece in the vicinity of solid boundaries. Point distribution generation using quadtree and octree data structures is expected to eliminate many of these difficulties. It eliminates the book-keeping and requirement of clipping completely. The problems of solution accuracy and convergence near the solid boundaries is also expected to be reduced.

5.1 Quadtree Data Structure



(a) Quadtree



(b) Physical Space

Figure 16: A quadtree example

The basic principle of a quadtree/octree is to cover the region of interest with a rectangle/box, then recursively partition the region into smaller regions until each region satisfies a suitable uniformity criterion. This recursive subdivision is represented by a tree structure. An example quadtree is shown in fig. 16. The root node[1] represents the whole region of interest.

[1] We use the word node to represent the nodes of the tree but not the points of the distribution

This region is tested for the uniformity criterion. If it fails to satisfy the uniformity criteria, then it is divided into four disjoint congruent regions called quadrants. This subdivision is represented by creating four children nodes for the root node. Each of these children nodes represents one of the created quadrants. These new quadrants are now checked for the uniformity criterion. All the quadrants which doesn't satisfy the uniformity criterion are further divided by creating children nodes for the corresponding nodes. This process continues till each of the disjoint quadrants satisfies the uniformity criterion. The nodes of the tree corresponding to these disjoint quadrants are called leaf nodes. They contain no children. Each node also contains the information about its level. The level of the root node is given by 0. The level of any other node in the tree is determined by adding one to the level of its parent node. The neighbor quadrants of any given quadrant can be obtained by traversing the tree. The algorithms for the neighbor search are given by [23, 24].

5.2 Generation of Quadtree

This quadtree generation algorithm has two inputs. The first input is the area of interest which is needed to construct the root node of the quadtree. This area of interest is specified by giving the coordinates of its bottom-left and top-right corners. The quadtree generation algorithm also requires a uniformity criterion. In the point generation algorithm, we used two uniformity criteria. When a desired point density distribution is given, the point density is interpolated at the centroids of each quadrant. If A is the area of a quadrant and d is the point density at its centroid, then the uniformity criterion is specified as $A * d <= 1.5$. The quadrants which are not satisfying this inequality are partitioned further. One can also generate point distributions, by just specifying the desired point distribution on the solid boundaries. These input boundary points are not a part of the final distribution. They are used only to determine the uniformity criterion. In this case, the uniformity criterion is that a quadrant should not have more than one input solid boundary point. The quadtree is refined till each leaf quadrant contains atmost one solid boundary point.

The refined quadtree obtaining by using the above mentioned uniformity criterion usually contains abrupt jumps in the level difference of two neighbor quadrants. To obtain smooth distributions these jumps should be removed. We achieve this by specifying another uniformity criterion. Any quadrant which is coarser by more than one level to one of its neighbors is refined.

5.3 Extracting the Point Distribution and Point Neighborhoods from the Quadtree

The quadtree generated using the above procedure is then used to obtain a point distribution. The quadrants of the tree are classified into three categories. The quadrants which are completely inside the solid bodies are called as solid quadrants. The quadrants which are cut by the solid bodies are called body quadrants. The quadrants which are in the flow field and which are not cut by the solid bodies are called interior quadrants. The interior quadrants and body quadrants provide one point each to the distribution. The interior quadrants supply their centroids to the point distribution. The centroids of the body quadrants are projected to the solid body. These projections are then added to the point distribution. For each point in the distribution, the neighbor quadrants of its corresponding quadrant are obtained by traversing the tree. The points provided by these quadrants are then added to the neighborhood of the original point.

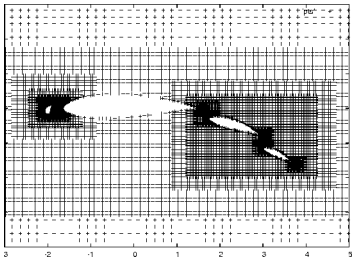
5.4 Generating Adaptive Point Distributions

Adaptivity is essential to obtain solutions of higher accuracy. The quadtree based point generation algorithm can be used to obtain adaptive point distributions also. Meshless methods are used to compute the flow solution on the initial point distribution obtained by using the above described algorithm. The quadtree structure of the initial grid is then refined using a uniformity criterion based on the flow solution. We use D^2 distance as the sensor for adaptation[?, Mohan] For each quadrant in the quadtree, the primitive variables at its corresponding point are interpolated from the initial solution. The D^2 distance is then calculated between the points of two neighboring quadrants. When this D^2 distance exceeds a threshold value, these quadrants are then divided. After refining the quadtree using this criterion, the quadtree corresponding to the adaptive point distribution is obtained. The adaptive point distribution is then obtained as described in the previous section. Similarly, using the quadtree of first level adaptive point distribution and a flow solution over it, one can obtain a second level adaptive point distribution. This procedure can be repeated till one gets a solution of desired accuracy.

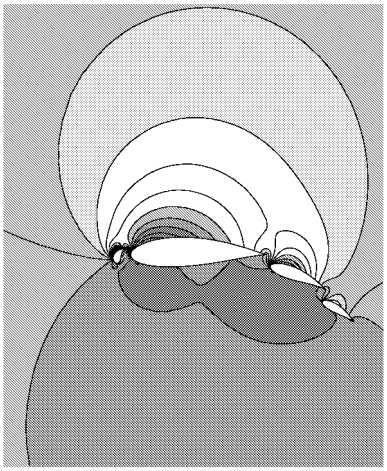
5.5 Results

We used these algorithms to generate flow solutions over standard test cases. Fig (17) shows a point distribution and q-LSKUM solution on this

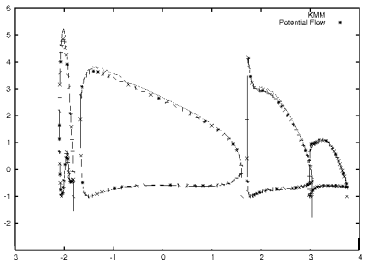
distribution for flow over a Suddho-Hall airfoil. The flow conditions are $M = 0.15$ and $\alpha = 0.0$. The C_p comparison with a potential flow solution is shown in the figure. Fig (18) shows an adaptive point distribution and q-LSKUM solution on this distribution over Bi-Naca0012 configuration. Two levels of adaptation are used in this case. The flow conditions for this case are $M = 0.85$ and $\alpha = 0$. The C_p plot shows a comparison with a solution on unstructured grid. One can observe that the results obtained using this procedure are satisfactory. Work is in progress to validate this method for **3-D** configurations.



Point distribution

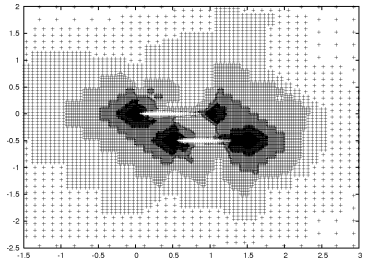


Pressure distribution

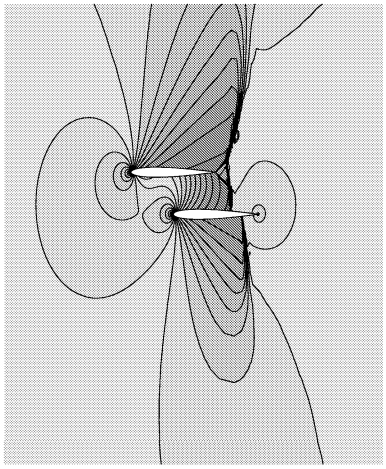


C_p distribution

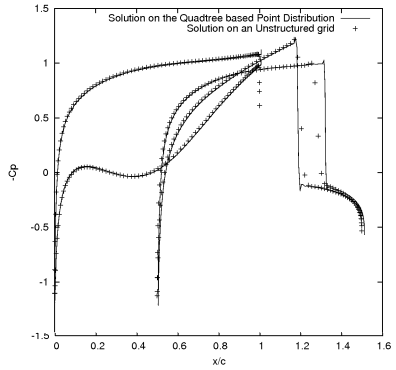
Figure 17: Flow over Suddho-Hall Airfoil



Point distribution



Pressure contours



C_p distribution

Figure 18: Flow over Bi-NACA-0012 - after two levels of adaptation

References

[1] Anandhanarayanan, K. (2003) *Development and Applications of a Gridfree Kinetic Upwind Solver to Multibody Configurations*, PhD. Thesis, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India.

[2] Anil N. and Deshpande S.M. Low Dissipative Modified KFVS (M-KFVS) Method. *Repo rt 2004 FM 22*, Dept. of Aerospace Engg., IISc, Bangalore, India.

[3] Arora, Konark and Deshpande, S.M. (2004) *Weighted Least Squares Kinetic Upwind Method using Eigenvector Basis*, FM Report No. 2004 FM 17, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India , 2004.

[4] Dauhoo, M.Z., Ghosh, A.K., Ramesh, V. and Deshpande, S.M.(1999) *q-LSKUM : A new higher order accurate kinetic method for Euler equations using entropy variables*, ISCFD-99, Bremen, Germany, Sept. 1999.

[5] Deshpande, S.M., Ghosh, A.K. and Mandal, J.C.(1989) *Least Squares Weak Upwind Method for Euler Equations*, FM Report No. 89 FM 4, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India, 1989.

[6] Deshpande, S.M., Anandhanarayanan, K., Praveen, C. and Ramesh, V. (2001) *Theory and Applications of 3-D LSKUM Based on Entropy Variables*, First ICFD, Oxford, March 2001.

[7] Deshpande, S.M. (2003) *Meshless Method, Accuracy Symmetry Breaking, Upwinding and LSKUM*, FM Report No. 2003 FM 1, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India, January, 2003.

[8] Ghosh, A.K. (1996) *Robust Least Squares Kinetic Upwind Method for Inviscid Compressible Flows*, PhD. Thesis, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India.

[9] Ghosh, A.K. and Deshpande, S.M. (1995) *Least Squares Kinetic Upwind Method for Inviscid Compressible Flows*, AIAA Paper No. 95-1735.

[10] John D. Anderson Jr. Computational Fluid Dynamics - The Basics with Applications. *McGraw Hill International Editions.*

[11] Malagi, K.S., Kulkarni, P.S. and Deshpande, S.M. (2004) *Rotationally Invariant Kinetic Upwind Method,* FM Report under preparation, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India, 2004.

[12] Mandal, J.C. and Deshpande, S.M. (1994) *Kinetic Flux Vector Splitting for Euler Equations,* Computers and Fluids, Vol 23. No. 2, pp. 447-478.

[13] Mahendra, A.K. (2003) *Application* of *Least Squares Kinetic Upwind Method to Strongly Rotating Viscous Flows,* MSc. (Engg.) Thesis, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India.

[14] Mathur, J.S. (1997) *Application of kinetic schemes to structured, unstructured, cartesian and hybrid grids,* PhD. Thesis, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India, 1997.

[15] Praveen, C. (2004) *Development and Applications* of *Kinetic Meshless Methods for Euler Equations,* PhD. Thesis, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India.

[16] Ramesh, V. (2001) *Least Squares Grid-Free Kinetic Upwind Method,* PhD. Thesis, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India.

[17] Ramesh V. and Deshpande S.M. Low dissipation grid free upwind kinetic scheme with modified CIR splitting. *Report 2004 FM 20,* Dept. of Aerospace Engg., IISc, Bangalore, India.

[18] Varma, Mohan U., Raghurama Rao, S.V. and Deshpande, S.M. (2003) *Point generation using Quadtree Data Structures for Meshless Solvers,* FM Report No. 2003 FM 08, Department of Aerospace Engg., Indian Institute of Science, Bangalore, India.

[19] *http://www.ics.uci. edu/ eppstein/gina/quadtree.html*

[20] *http://www.andrew.cmu.edu/user/sowen/topics/octree.html*

[21] Aftosmis, M.J., *Solution adaptive Cartesian Grid Methods,* in Lecture Series 1997-02, von Karman Instutute for Fluid Dyanmics, March 1997

[22] Samet H., "*The Quadtree and Related Hierarchical Data Structures*", Computing Surveys, 16(2), 187-285, 1984.

[23] Samet H., "*Neighbor Finding Techniques for Images represented by Quadtrees*", Computer Graphics and Image Processing, 18(1) pp. 367-386, Jun. 1989.

[24] Samet H., "*Neighbor Finding in Images represented by Octrees*", Computer Graphics and Image Processing, 46(3) pp. 37-58, 1982.