

# Weighted Least Squares Kinetic Upwind Method (WLSKUM) using Eigenvector Basis

Konark Arora<sup>1</sup> arora@aero.iisc.ernet.in

N.K.S. Rajan<sup>2</sup> nksr@cgpl.iisc.ernet.in S. M. Deshpande<sup>3</sup> smd@jncasr.ac.in

<sup>1</sup> ARDB CFD Centre, Department of Aerospace Engineering,  
Indian Institute of Science, Bangalore 560012.

<sup>2</sup> Combustion Gasification and Propulsion Laboratory, Department of Aerospace Engineering,  
Indian Institute of Science, Bangalore 560012.

<sup>3</sup> Engineering Mechanics Unit,  
Jawaharlal Nehru Centre for Advanced Scientific Research, Jakkur, Bangalore 560064.

**Abstract.** The least squares grid-free method, though having the ability to work effectively on any distribution of points is limited by the requirement of a good connectivity around a node. This paper deals with a fundamental improvement over the usual least squares grid-free method to overcome this limitation of the grid-free method. The new approach involves the use of the weights to diagonalize the least squares matrix  $A$  such that the  $x$  and  $y$  directions become the eigen directions along which the higher dimensional least squares formulae reduce to the one dimensional formulae. A very important advantage of this approach (apart from improving the convergence characteristics of the grid-free solver) is that it helps in tackling the problems of code divergence due to the degenerate and other cases of bad connectivity.

**Keywords:** Gridfree, Connectivity, Eigenvector basis, Defect correction, LED

## 1 Introduction

Gridfree or meshless methods are slowly gaining acceptance as a useful tool in computation of flows around practical configurations. Using LSKUM (Least Squares Kinetic Upwind Method), Anandhanarayanan [1] and Praveen [7] have computed inviscid supersonic and transonic flows around flight vehicles and M165 configuration with as complex mesh as FAME. All grid free methods operate on a point distribution in the computational domain  $\Omega$  and require connectivity  $N(P_o)$  of every point or node  $P_o \in \Omega$ . The connectivity is a set of points in the neighbourhood of  $P_o \in \Omega$ . The convergence and accuracy of solution depends on connectivity. Praveen [7] has done detailed analysis of "bad connectivity" which often leads to code divergence or inaccurate solution. While connectivity characteristics are continuously improved to avoid bad connectivity related code divergence, it is also essential to make solver more robust for reliable convergence. The convergence characteristics of the solver also depend on the discrete approximation used for computation of spatial derivatives, which in case of LSKUM are determined by LS (least squares) approximation. The present paper considers a new approach based on eigenvectors and weights in LS approximation. Numerical experiments do show improvement of convergence characteristics of LSKUM and this improvement is due to the ability of LS approximation in working on degenerate and other cases of bad connectivity [2].

## 2 Weighted Least Squares Method in 2-D

Consider Fig.1 showing connectivity of  $P_o$ . To get the derivative of the function  $F(x, y)$  at point  $P_o$ , we expand  $F_i$  around point  $P_o$  in terms of Taylor series

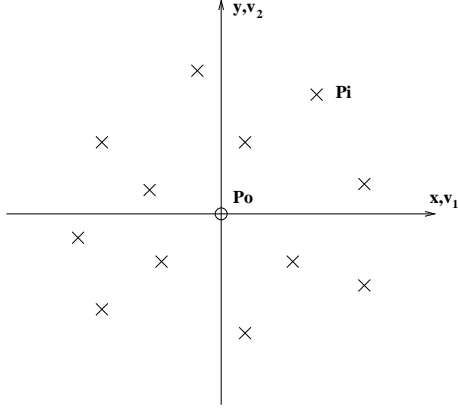
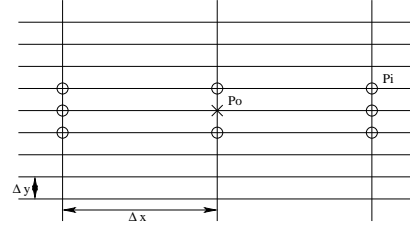


Figure 1: Connectivity for node o in 2 dimension



Uniform cartesian grid, node Po denoted by cross, with its connectivity denoted by circ

Figure 2: Highly stretched Cartesian grid showing a node and its connectivity

$$F_i = F_o + (x_i - x_o)F_{x_o} + (y_i - y_o)F_{y_o} + O(\Delta x^2, \Delta y^2) \quad (1)$$

Define the differences

$$\Delta x_i = x_i - x_o, \quad \Delta y_i = y_i - y_o, \quad \Delta F_i = F_i - F_o$$

The weighted sum of the squares of errors at point  $P_o$  (after truncating Taylor Series in Eq.(1)) is given by

$$E = \sum_{i=1}^p w_i (\Delta F_i - \Delta x_i F_{x_o} - \Delta y_i F_{y_o})^2 \quad (2)$$

Minimizing the error  $E$  in Eq.(2) with respect to  $F_{x_o}$  and  $F_{y_o}$ , we get the following system of linear algebraic equations

$$A(w) (\text{grad } F)_o^T = b(w) \quad (3)$$

where

$$A(w) = \begin{bmatrix} \sum w_i \Delta x_i^2 & \sum w_i \Delta x_i \Delta y_i \\ \sum w_i \Delta x_i \Delta y_i & \sum w_i \Delta y_i^2 \end{bmatrix}, \quad (\text{grad } F)_o^T = \begin{bmatrix} F_{x_o} \\ F_{y_o} \end{bmatrix}, \quad b(w) = \begin{bmatrix} \sum w_i \Delta x_i \Delta F_i \\ \sum w_i \Delta y_i \Delta F_i \end{bmatrix}$$

It is observed that for  $w_i = 1$ , the weighted least squares matrix  $A(w)$  reduces to the usual least squares matrix  $A$ . The least squares matrix  $A(w)$  has several interesting mathematical and geometrical characteristics.

- (a) The matrix  $A(w)$  is purely a geometric matrix. All the elements of this matrix are functions of coordinates of nodes in the connectivity. This matrix must be inverted to find the derivatives at a node. This directly explains the importance of the connectivity of the node in grid-free solvers. The bad connectivity [7] can degrade the solution accuracy due to singularity or ill-conditioning of  $A$  or due to some other cause. Sometimes, even a well conditioned matrix coupled with a bad algorithm can give unacceptable solution. Consider the least squares formula (with  $w_i = 1$ ) for the x-derivative in  $(x, y)$  frame in 2-D

$$\left( \frac{\partial F}{\partial x} \right)_o = \frac{\sum \Delta y_i^2 \sum \Delta x_i \Delta F_i - \sum \Delta x_i \Delta y_i \sum \Delta y_i \Delta F_i}{\sum \Delta x_i^2 \sum \Delta y_i^2 - (\sum \Delta x_i \Delta y_i)^2} \quad (4)$$

The denominator in the Eq.(4) is the determinant of the least squares matrix  $A$ . If the matrix  $A$  becomes singular or nearly singular, then Eq.(4) is unable to determine the value of the derivatives accurately. This can happen in various cases of bad connectivity [7]. Moreover, if the connectivity points are highly anisotropic as in the case of stretched cartesian grid shown in Fig.2, the least squares matrix  $A$  becomes highly illconditioned because  $\sum \Delta x_i \Delta y_i = 0$  and  $\Delta y \ll \Delta x$ . Even then 1-D finite difference formulae work perfectly fine for such cases. This suggests that the 2-D formula (4) ought to be reduced to 1-D formula for enhancing robustness and accuracy.

- (b) The matrix  $A$  is a real symmetric matrix and is orthogonally diagonalizable. From the matrix theory, we know that a real symmetric matrix has all real eigenvalues and a complete set of real distinct eigenvectors [10]. If we rotate the local frame of reference along the eigendirections and use them as the basis vector for the least squares formula, it is observed that the two dimensional least squares formula reduces to the one dimensional formula along each of the eigendirections as the off diagonal terms in the least squares matrix vanish. We then get

$$F_{x_o} = \frac{\sum \Delta x'_i \Delta F_i \sum \Delta y'_i{}^2}{\sum \Delta x'_i{}^2 \sum \Delta y'_i{}^2} = \frac{\sum \Delta x'_i \Delta F_i}{\sum \Delta x'_i{}^2} \quad (5)$$

As a result, it is expected that the above problem of loss of accuracy will not be faced if the co-ordinate frame is locally rotated along the eigendirections.

The weights used in the weighted least squares formulae serve various purposes:

- (i) The weights can be used to increase the accuracy of the least squares formulae [3].
- (ii) The weights can be selected so as to increase the spectral resolution of the least squares formulae [9].
- (iii) The weights can help to maintain positivity and local extremum diminishing (LED) property of the least squares formulae [2].
- (iv) The weights can help in improving the convergence characteristics of the grid-free solver.

It is interesting to note that an appropriate choice of the weights can even change the nature of the matrix  $A(w)$ . So an interesting question naturally arises : Whether suitable weights can be chosen which favourably change the least squares matrix such that the solution accuracy and robustness are improved ? The answer is affirmative. The weights can be suitably determined such that the weighted least squares matrix  $A(w)$  becomes a diagonal matrix [2]. It has been observed earlier that the diagonalization of the least squares matrix reduces the two and three dimensional least squares formulae of the derivatives to the corresponding one dimensional least squares formulae along the eigen directions. It is therefore expected that  $A(w)$  with suitable weights will help in overcoming the problems of bad connectivity to a great extent. When weights are chosen in the above fashion  $x, y$  directions become the eigendirections of  $A(w)$ . The method of determining the weights is described in the next section.

## 2.1 Calculation of weights for two dimensional least squares formulae

Consider Fig.3 which shows the four quadrants of the split stencil normally used for upwinding. The idea is to make  $x$ -axis parallel to an eigenvector of  $A(w)$  for split stencil by suitably determining weights. It is observed that the product of  $\Delta x$  and  $\Delta y$  is always positive in quadrants I and III, while it is

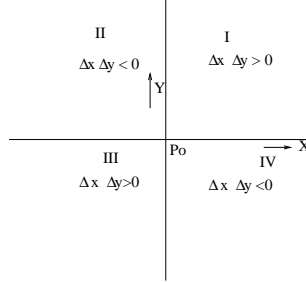


Figure 3: Quadrants for the split stencil of a node

always negative in quadrants II and IV. Whenever we are using x-y splitting, each split stencil involves two quadrants. One of the quadrants always contributes to the positive product  $\Delta x \Delta y$  while the other quadrant always contributes to the negative product  $\Delta x \Delta y$ . Suppose we want to find the weights for the least squares formula when we are using the point distribution in the left stencil ( $II + III$ ) only. Making use of the above observation, we can easily obtain the weights for the points in the left stencil such that  $\sum_{II+III} w_i (\Delta x_i \Delta y_i) = 0$  while ensuring that the weights calculated are always positive. It must be kept in mind that a primary requirement of a good connectivity of a node is that none of the quadrants shown in Fig.3 should be empty. Let  $w_{II}$  be the weight assigned to the points lying in the quadrant  $II$  of the stencil while  $w_{III}$  be the weight assigned to the points lying in the quadrant  $III$  of the stencil. We then enforce

$$w_{III} \left( \sum \Delta x_i \Delta y_i \right)_{III} + w_{II} \left( \sum \Delta x_i \Delta y_i \right)_{II} = 0 \quad (6)$$

Introduce the notation for cross products,

$$C_{xy}^{III} = \sum_{III} \Delta x_i \Delta y_i, \quad C_{xy}^{II} = \sum_{II} \Delta x_i \Delta y_i$$

where the superscripts  $II$  and  $III$  on  $C$  denote the quadrants over which the summation is taken and subscript  $xy$  to  $C$  stands for the fact the cross products  $\Delta x_i \Delta y_i$  in the  $x - y$  plane. The Eq.(6) can now be re-written as

$$w_{III} C_{xy}^{III} + w_{II} C_{xy}^{II} = 0 \quad (7)$$

As per the observations made before

$$C_{xy}^{II} < 0, \quad C_{xy}^{III} > 0$$

The solution of Eq. (7) is

$$\frac{w_{III}}{w_{II}} = -\frac{C_{xy}^{II}}{C_{xy}^{III}} > 0 \quad (8)$$

From the Eq.(8) above, it is seen that the ratio of the weights obtained above is always positive as the product  $\Delta x \Delta y$  is positive in quadrant III while it is negative in quadrant II. The ratio goes to infinity when the denominator in the Eq.(8) goes to zero. This can only happen when the quadrant III is empty, which goes against our requirement that the quadrant contributing to the derivative at a node must have atleast a point. Thus, the ratio in Eq.(8) can never go to infinity. Now, choosing one of the weights as 1, the other weight can be obtained by this calculated ratio. The similar procedure can be applied to find the weights for obtaining derivatives using other split stencils, namely the split stencil  $I + IV$ , or the split

stencils for  $\Delta y_i \leq 0$ . In each of these cases, it is observed that the two dimensional least squares formula reduces to the corresponding one dimensional formula in the respective directions. Use of appropriate weights help in mitigating the effect of bad connectivity thus avoiding presumably loss of accuracy of the least squares formulae. It is therefore reasonable to assume that if the weights are chosen such that the two dimensional least squares formulae reduce to the one dimensional formulae in the corresponding eigendirections, the problem of code divergence due to bad connectivity can be effectively tackled. Thus the definition of bad connectivity to some extent depends on which formula for the derivative we are using. We have a choice between the least squares formula with and without weights. A suitable choice of weights reduces a multi-dimensional formula to a one dimensional formula for the derivative and this as will be shown later helps in avoiding code divergence caused by "bad connectivity". Second order accuracy in case of least squares formulae is obtained by the defect correction method [3]. To derive second order accurate weighted least squares formulae for the derivatives, the Taylor series in Eq.(1) is truncated to  $O(\Delta x^3, \Delta y^3)$ . On minimizing the modified weighted sum of the squares of error at  $P_o$  wrt  $F_{x_o}$  and  $F_{y_o}$ , we get the second order accurate weighted least squares formula for the derivatives

$$\left(\frac{\partial F}{\partial x}\right)_o = \frac{\sum w_i \Delta y_i^2 \sum w_i \Delta x_i \widetilde{\Delta F}_i - \sum w_i \Delta x_i \Delta y_i \sum w_i \Delta y_i \widetilde{\Delta F}_i}{\sum w_i \Delta x_i^2 \sum w_i \Delta y_i^2 - (\sum w_i \Delta x_i \Delta y_i)^2} \quad (9)$$

where

$$\widetilde{\Delta F}_i = \Delta F_i - \frac{\Delta x_i}{2} \Delta F_{x_i} - \frac{\Delta y_i}{2} \Delta F_{y_i} \quad (10)$$

If the weights in Eq.(9) are chosen as described above in the previous section, the second order weighted LS 2-D formulae for the derivative reduces to the second order 1-D formulae given by

$$F_{x_o} = \frac{\sum w_i \Delta x_i \widetilde{\Delta F}_i}{\sum w_i \Delta x_i^2}, \quad F_{y_o} = \frac{\sum w'_i \Delta y_i \widetilde{\Delta F}_i}{\sum w'_i \Delta y_i^2} \quad (11)$$

It is important to note that the 1-D formulae in Eq.(11) are having the modified  $\widetilde{\Delta F}_i$  given by Eq.(10). The weights in each of the formulae for  $F_{x_o}$  and  $F_{y_o}$  are different, depending upon the split stencils used to calculate weights in each formula as described in the previous section.

### 3 Results and Discussion

Results of test case of Subsonic flow past NACA 0012 aerofoil at Mach number 0.63 and angle of attack of  $2^\circ$  have been shown to support the claims made above about the code divergence, for which a point distribution generated by Delaunay triangulation shown in Fig.4 has been used. The triangulation at one point on the aerofoil has been intentionally tampered so as to make it an extreme case of a bad connectivity. The initial connectivity of the node is shown in Fig.5. The node under consideration is indicated by a star in Fig.5. After tampering the 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 is evident from the Fig.8. 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.9. The pressure contours for the above computation are shown in Fig 7. Fig.12 shows the comparison of the residue drop for the second order accurate computations by employing weighted and unweighted LSKUM on a computational domain with 4733 nodes. The weights have been calculated

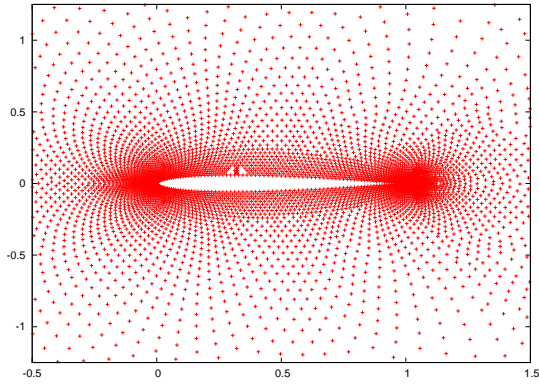


Figure 4: Bad connectivity on aerofoil : 7269 Nodes in the domain

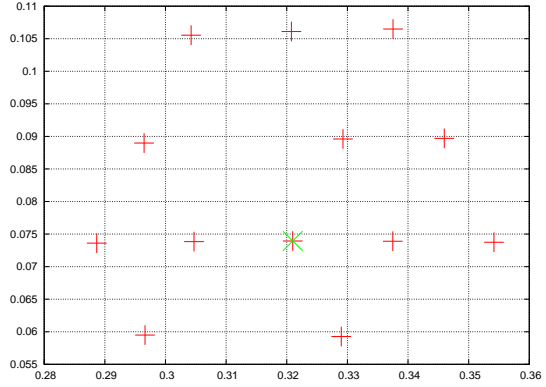


Figure 5: Original good connectivity

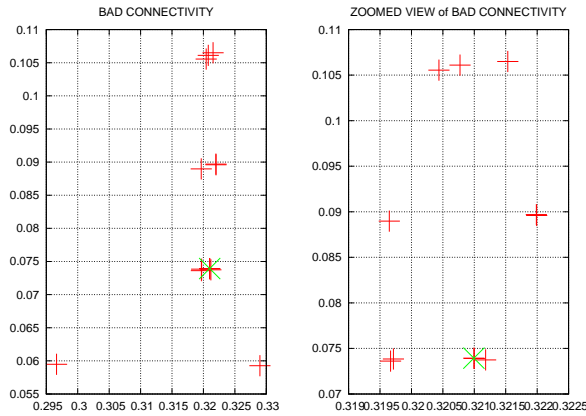


Figure 6: Bad connectivity

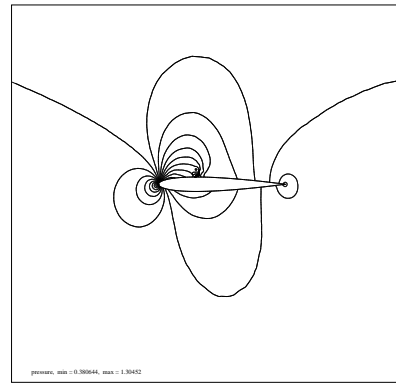


Figure 7: Pressure contours : weighted least squares : second order : bad connectivity

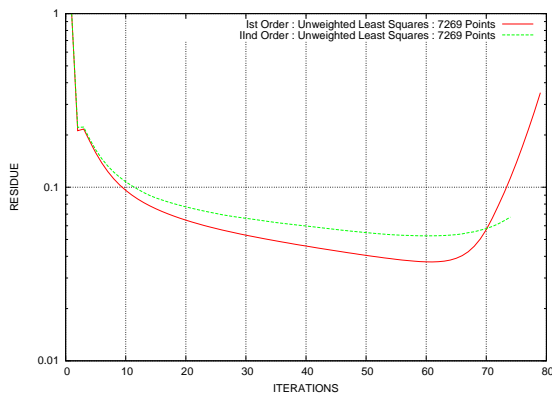


Figure 8: Unweighted least squares method

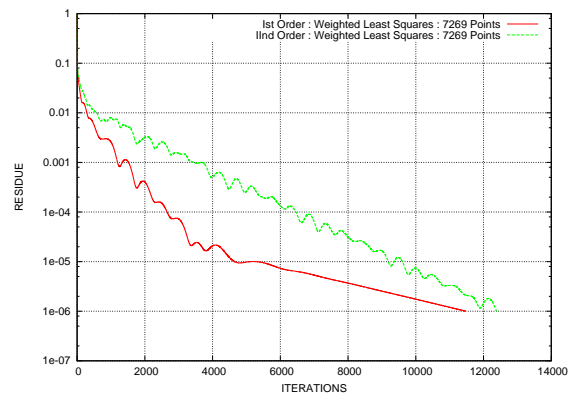


Figure 9: Weighted least squares method

Residue Drop : first and second order computations with bad connectivity

such that the two dimensional formulae reduce to the one dimensional formulae in the corresponding eigendirections. Since the weighted least squares method is LED [2], the second order computations have been performed without the use of limiters. A greater decade fall in residue is observed even for the second order accurate computations done using the weighted LSKUM as against the computations done using second order accurate unweighted LSKUM where a saturation state is reached at a much earlier stage. Similar observations are made when the same test case is run on a finer cloud of points comprising of 10938 nodes as shown in the Fig.13. The comparison of CP plots for the test case also reveals the capture of better suction peak in case of the weighted LSKUM as shown in Figs. 10 and 11.

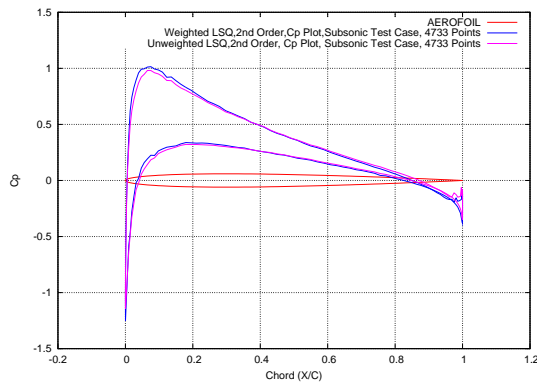


Figure 10:  $C_P$  comparison : 4733 points

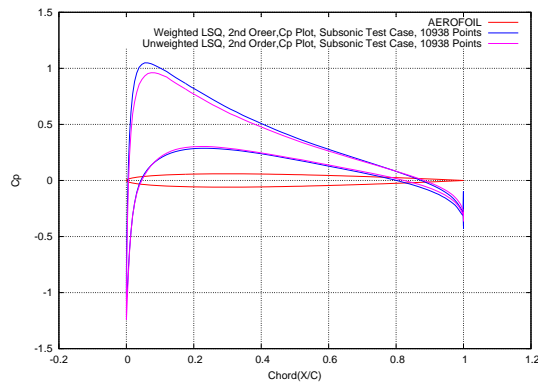


Figure 11:  $C_P$  comparison : 10938 points

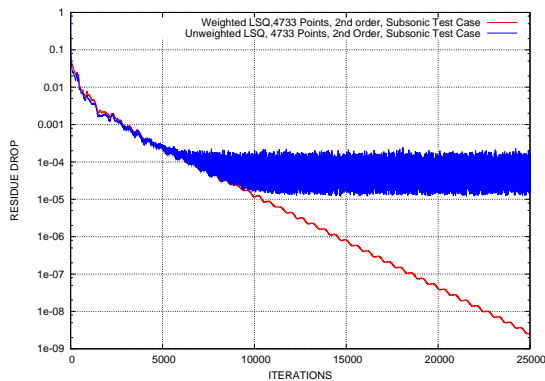


Figure 12: 4733 points in the computational domain

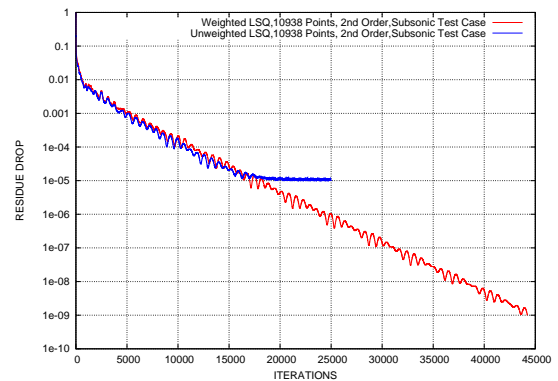


Figure 13: 10938 points in the computational domain

Residue Drop for weighted and unweighted least squares method : second order computations

## 4 Conclusion

A new weighted least squares formulation based on the eigenvectors has been developed as a result of which the least squares formulae for derivatives in multidimensions reduce to the 1-D formulae thus considerably reducing the problem of code divergence faced by the standard LSKUM. The weights are easily determined from cross products  $C_{xy}$  over quadrants for diagonalizing the least squares matrix  $A(w)$ . Positivity of weights ensure LED property [2] of the multidimensional state update formula. The whole

analysis smoothly carries over to 3 D for which  $A(w)$  can be diagonalized to give 1-D LS formulae for spatial derivatives [2]. This approach can offer tremendous advantage in LSKUM in terms of robustness and accuracy and give the user a kind of "care-free flexibility" while generating connectivity.

## References

- [1] Anandhanarayanan, K. (2003) *Development and Applications of a Gridfree Kinetic Upwind Solver to Multibody Configurations*, PhD. Thesis, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India.
- [2] 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 Engineering, Indian Institute of Science, Bangalore, India.
- [3] Ghosh, A.K. (1996) *Robust Least Squares Kinetic Upwind Method for Inviscid Compressible Flows*, PhD. Thesis, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India.
- [4] Ghosh, A.K. and Deshpande, S.M. (1995) *Least Squares Kinetic Upwind Method for Inviscid Compressible Flows*, AIAA Paper No. 95-1735.
- [5] 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.
- [6] Mahendra, A.K. (2003) *Application of Least Squares Kinetic Upwind Method to Strongly Rotating Viscous Flows*, MSc. ( Engg. ) Thesis, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India.
- [7] Praveen, C. (2004) *Development and Applications of Kinetic Meshless Methods for Euler Equations*, PhD. Thesis, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India.
- [8] Ramesh, V. (2001) *Least Squares Grid-Free Kinetic Upwind Method*, PhD. Thesis, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India.
- [9] Sashi Kumar, G.N., Mahendra, A.K. and Deshpande, S.M. (2004) *Spectral Resolution and Order of Accuracy of Least Squares Scheme*, FM Report (under preparation), Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India.
- [10] Strang, Gilbert *Linear Algebra And Its Applications*, Third Edition.