47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition
5 - 8 January 2009, Orlando, Florida

AIAA 2009-596

A Comparison of Various Meshless Schemes Within a Unified Algorithm

Aaron Katz* and Antony Jameson†

Department of Aeronautics and Astronautics, Stanford University, Stanford, CA

An algorithm for use with several meshless schemes is presented based on a local extremum diminishing property. The scheme is applied to the Euler equations in two dimensions. The algorithm is suitable for use with many meshless schemes, three of which are detailed here. First, a method based on Taylor series expansion and least squares is highlighted. Next, a similar least squares method is used, but using polynomial basis functions with fixed Gaussian weighting. A third method makes use of the Hardy multiquadric radial basis functions on a local cloud of points. Results indicate that all three methods perform essentially equally well for flows without shocks. For flows with shocks, the least squares methods perform significantly better than the radial basis method, which displays discrepancies in shock location and magnitude. All methods are compared to an established finite volume method for validation purposes.

I. Introduction

As computer power increases, researchers tackle problems of increasing complexity. A major bottleneck in computing flow over complex geometry is the ability to generate high quality meshes in a timely manner. Such difficulties have prompted great interest recently in meshless schemes. Meshless schemes differ from tradtional finite volume, finite difference, and finite element methods in that they do not require rigid domain discretization. Instead, meshless methods rely only on clouds of points, either local or global, upon which partial differential equations may be discretized. Full relief of mesh generation difficulties has not yet been shown by using a meshless method, although the work of Löhner,[1,2] in which efficient point generation schemes were developed, appears promising.

Two of the most popular classes of meshless schemes that have been applied to fluid simulation are based on least squares and radial basis functions. One of the earliest meshless works is that of Batina[3] in which he used least squares to compute inviscid and viscous flows in two and three dimensions. A much more extensive algorithm was soon developed by Oñate, Löhner, and others,[4–6] called the finite point method (FPM). The FPM method used a polynomial basis, echoing themes from the classical finite element method. Least squares methods based on Taylor series expansions have been used extensively by Deshpande and others[7–15] in the context of kinetic schemes for the Euler equations. They have developed extensive capabilities with the least squares kinetic upwind method (LSKUM). In a more traditional Riemann solver approach, Sridar and Balakrishnan[16,17] have developed an upwind scheme based on least squares, highlighting the order of accuracy of their scheme. Both the polynomial and Taylor series approaches have become quite popular to compute boundary conditions for non-body-conforming meshes.[18–23] Katz and Jameson[24] have also used a meshless method as a means of grid communication for overset grid systems.

While radial basis functions appear much more sparsely in the literature for compressible flow computations, they have found widespread use for incompressible flows and elliptic problems. Kansa[25,26] was the first to apply radial basis methods to computational fluid dynamics. Subsequently, Divo and Kassab[27,28] used radial basis methods to compute natural convection and heat transfer problems. Chinchapatnam[29] has computed steady incompressible flows with a radial method. Power and Barraco[30] compared symmetric and unsymmetric variants of radial basis methods. One of the few compressible flow applications of radial basis functions was that of Shu,[31] in which he computed flow in converging channels and shock tubes.

[*] PhD Candidate, Department of Aeronatics and Astronautics, Stanford, CA, AIAA Member.

[†] Thomas V. Jones Professor of Engineering, Department of Aeronautics and Astronautics, Stanford, CA, AIAA Member.

Copyright © 2009 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

While many meshless schemes for compressible flow have been proposed, a comprehensive comparison of major algorithms appears to be lacking in the literature. This work serves as a beginning for such a comparison. An important aspect of the present work is to formulate a flexible framework for various meshless schemes, which we use to solve the Euler equations. In this manner, valid comparisons between meshless methods may be made. We also compare the meshless schemes with traditional methods. The outline proceeds as follows: in section II, we formulate a common framework for meshless schemes based on a local extremum diminishing property. In section III, we present three different meshless procedures to compute partial derivatives on a cloud of points. In section IV, we compare the results of the three schemes for subsonic and transonic flows. Finally, in section V, we draw certain conclusion from the results obtained thus far.

II. An LED-based Algorithm for Meshless Schemes

A convenient basis for the construction of non-oscillatory schemes for gas dynamics is the Local Extremum Diminishing (LED) prinicple[32] for discrete approximations to conservation laws. Given a two-dimensional scalar conservation law of the form

$$\frac{\partial v}{\partial t} + \frac{\partial f(v)}{\partial x} + \frac{\partial g(v)}{\partial y} = 0, \tag{1}$$

nearly any spatial discretization at a node j may be cast into the following form:

$$\frac{\partial v_j}{\partial t} = \sum_k c_{jk} v_k, \tag{2}$$

where the summation is over the nearest neighbors, k. A consistent approximation which meets the minimal requirement to approximate a constant function exactly should possess the property that

$$\sum_k c_{jk} = 0. \tag{3}$$

It follows with no loss of generality, that Equation 2 may be written as

$$\frac{\partial v_j}{\partial t} = \sum_{k \neq j} c_{jk}(v_k - v_j). \tag{4}$$

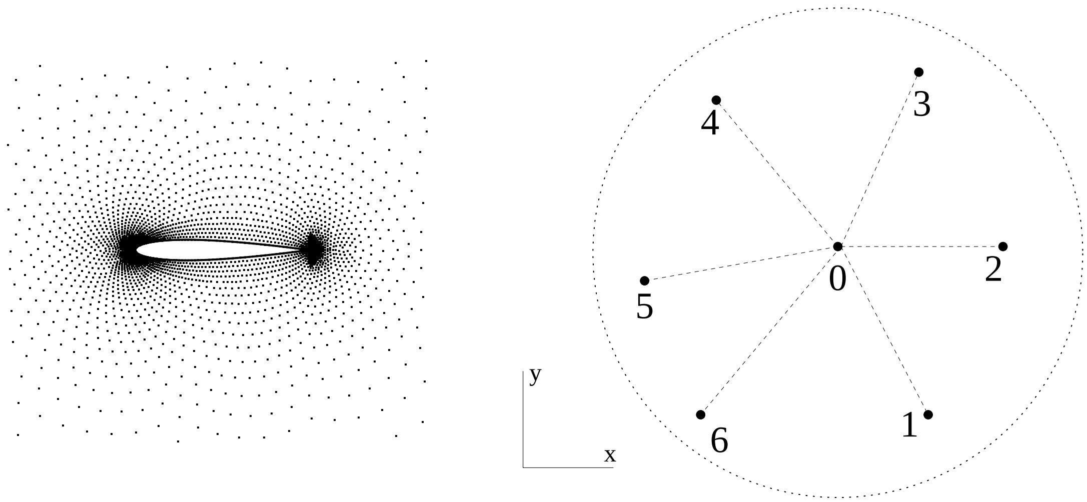
By constructing discrete schemes which ensure the coefficients, c_{jk}, are positive, a local minimum cannot decrease and a local maximum cannot increase. This is the essential principle of LED schemes. The LED principle assures positivity and leads to the class of TVD schmes in one dimension proposed by Harten.[33] The requirement to produce positive coefficients has led to a variety of schemes with additive artificial diffusion terms. Jameson[32] showed that a variety of popular diffusion schemes including JST,[34] scalar, and matrix schemes satisfy the LED criterion. The majority of LED schemes to date have been constructed for traditional mesh-based discretizations including finite volume, finite element, and finite difference.

In this work we show that the LED principle is flexible and general enough to extend to many meshless schemes. Given a global cloud of points conforming to some aerodynamic shape of interest, as shown in Figure 1(a), one may define for each point a set of nearest neighbors. These n nearest neighbors form a local cloud of points and constitute the connectivity for the meshless discretization, as shown in Figure 1(b). Suppose for now that a meshless discretization procedure may be used to approximate partial derivatives of a function, $\phi = \phi(x, y)$, at (x_0, y_0) as

$$\frac{\partial \phi_0}{\partial x} \approx \sum_{j=0}^{n} a_{0j}\phi_j, \quad \frac{\partial \phi_0}{\partial y} \approx \sum_{j=0}^{n} b_{0j}\phi_j, \tag{5}$$

where a_{0j} and b_{0j} are metric weights, which only depend on the mesh. Again, suppose that

$$\sum_{j=0}^{n} a_{0j} = 0, \quad \sum_{j=0}^{n} b_{0j} = 0. \tag{6}$$



(a) Global point cloud around NACA 0012

(b) Local point cloud

Figure 1. Point clouds for meshless schemes.

This zero sum property is actually a result of all partition of unity schemes, elaborated by Duarte and Oden.[35] All the meshless schemes in this work may be considered partition of unity schemes. With no loss of generality, Equation 5 may be expressed as

$$\frac{\partial \phi_0}{\partial x} \approx \sum_{j=1}^{n} a_{0j} \Delta \phi_{0j}, \quad \frac{\partial \phi_0}{\partial y} \approx \sum_{j=1}^{n} b_{0j} \Delta \phi_{0j}, \tag{7}$$

where $\Delta \phi_{0j} = \phi_j - \phi_0$.

The meshless method above may be used to discretize Equation 1, resulting in

$$\frac{\partial v_0}{\partial t} + \sum_{j=1}^{n} \left(a_{0j} \Delta f_{0j} + b_{0j} \Delta g_{0j} \right) = \frac{\partial v_0}{\partial t} + \sum_{j=1}^{n} \Delta F_{0j} = 0, \tag{8}$$

where $F = af + bg$ is a directed flux along the metric weight vector (a, b). Assuming the existence of a relation $\Delta F = \alpha \Delta v$, Equation 8 may be recast as

$$\frac{\partial v_0}{\partial t} + \sum_{j=1}^{n} \alpha_{0j} \Delta v_{0j} = 0, \tag{9}$$

which is of the form of Equation 4.

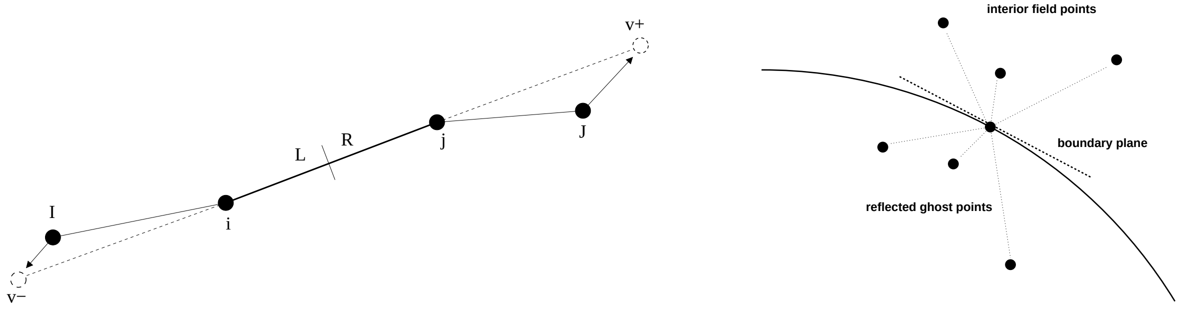
The LED criterion may be satisfied by adding an artificial diffusion term to the right hand side of Equation 9 of the form

$$\sum_{j=1}^{n} d_{0j}, \tag{10}$$

where

$$d_{ij} = \beta_{ij} \Delta v_{ij}, \quad \beta_{ij} \geq |\alpha_{ij}|. \tag{11}$$

Schemes of this sort are far too diffusive and may be made more accurate by the addition of anti-diffusive terms. Anti-diffusion may be conveniently added via the reconstruction of left and right states at the midpoint of the line connecting points i and j. A proven reconstruction method is the symmetric limited positive (SLIP) scheme of Jameson,[32] which leads to high order accuracy in smooth regions and reverts to first order accuracy in the vicinity of discontinuities to avoid oscillations. While the SLIP scheme was originally designed for structured grids, it has also proven effective for unstructured meshes. Here the SLIP scheme is extended to meshless methods.



(a) Reconstruction procedure.

(b) Boundary ghost points.

Figure 2. Aspects of the meshless discretization.

The use of left and right reconstructed states replaces d_{ij} with $d_{LR} = \beta_{LR}\Delta v_{LR}$, where

$$v_L = v_i + \frac{1}{2}s(\Delta v^+, \Delta v^-)\overline{\Delta v}, \quad v_R = v_j - \frac{1}{2}s(\Delta v^+, \Delta v^-)\overline{\Delta v}. \tag{12}$$

Here, $\overline{\Delta v} = \frac{\Delta v^+ + \Delta v^-}{2}$ is the average of the change in the solution on either side of the edge connecting points i and j, as shown in Figure 2(a). First, points I and J, the points in the local clouds of i and j which are most closely aligned with the edge in question are identified. Estimates with $\Delta v^+ = v_J - v_j$ and $\Delta v^- = v_i - v_I$ may be used. However an increase in accuracy is observed by making corrections of the form

$$\Delta v^+ = v_J - v_j + l_j \cdot \nabla v_J, \quad \Delta v^- = v_i - v_I - l_i \cdot \nabla v_I, \tag{13}$$

where the gradients of v are computed using the same meshless scheme of Equation 5 and l_i and l_j are the position vectors from I and J to the psuedo-points past the edge.

The Δv estimates obtained in this way lead to sharp capturing of discontinuities, turning on the limiter,

$$s(u, v) = 1 - \left|\frac{u - v}{|u| + |v|}\right|^q, \tag{14}$$

in the presence of shocks. Here q is a positive integer, usually taken to be $q = 3$. With the limiter turned on, the left and right states revert back to the i and j states, maintaining the LED quality at a local extremum. The key is the accurate detection of local extrema and shocks, which the above procedure performs quite well in practice.

The LED principle may be enforced at domain boundaries by using the same interior scheme for boundary points. At domain boundaries, interior points are reflected across tangent boundary planes to form ghost points, as shown in Figure 2(b). The ghost points serve two purposes. First, they balance the local clouds of the boundary points, improving the condition of meshless discretization procedures. Second, solution values at ghost points may be set consistent with any number of physical boundary conditions to obtain a solution.

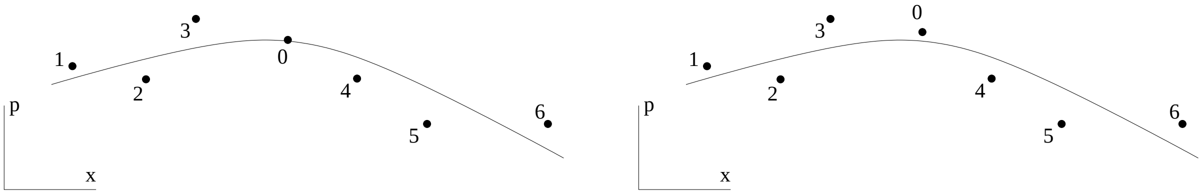
The above method for discretizing interior and boundary points completes the meshless spatial discretization for a scalar conservation law. Integration in time may be performed via any number of well established ODE methods. In this work, we acheive steady-state solutions using the modified Runge-Kutta scheme of Mavriplis, Jameson, and Martinelli.[36] In addition local time-stepping and the multicloud procedure of Katz and Jameson[37] has been used to accelerate convergence.

This work highlights the application of the meshless procedure to the Euler equations:

$$\frac{\partial w}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0, \tag{15}$$

where

$$w = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad f = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho u v \\ \rho u H \end{pmatrix}, \quad g = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + P \\ \rho v H \end{pmatrix},$$



(a) Taylor approximation passes through node 0.

(b) Polynomial fit not required to pass through node 0.

Figure 3. Two meshless approaches based on least squares.

and the equation of state and enthalpy definition are

$$E = \frac{P}{(\gamma-1)\rho} + \frac{1}{2}(u^2+v^2), \quad H = E + \frac{P}{\rho}.$$

In the above notation, ρ, u, v, P, E, and H are the density, velocity components, pressure, total energy, and total enthalpy.

Since the Euler equations are a system of conservation laws, it is necessary to generalize the scalar diffusion of Equation 11 to vectors. Many diffusion schemes have been proposed for the Euler equations including scalar, matrix, and mixed schemes. Here, the convective upwind split pressure (CUSP) scheme of Jameson[38] is used. The CUSP scheme is highly accurate, obtaining one-point shocks in one dimension, while minimizing computational cost. The CUSP scheme may be cast as

$$d_{LR} = \alpha^* c(w_R - w_L) + \beta(F_R - F_L). \tag{16}$$

Details of computing the coefficients α^* and β may be found in the work on artificial diffusion schemes by Jameson.[38] The CUSP scheme may be formulated to produce a steady state which admits constant stagnation enthalpy, a defining characteristic of steady inviscid flow. Moreover, enthalpy damping[34] may be used in such a scheme to accelerate convergence to steady state.

No slip boundary conditions were prescribed at the ghost nodes by a simple reflection of the velocity at the corresponding field node. Pressure and density at ghost nodes were extracted from the local boundary node. At the far field, ghost nodes were set via one-dimensional Riemann invariants, while maintaining constant stagnation enthalpy.[39] Extrapolation from the interior was facilitated by using the reflected interior point.

III. Three approaches to Meshless Discretization

The meshless scheme described in section II made use of certain metric weights to obtain the partial derivative estimates of Equation 5. This section details three methods to obtain these weights. The first is a least squares method based on Taylor series expansions. The second method is also a least squares method, but based on polynomial basis functions. The third method is a local radial basis collocation method. It will be shown that all three methods may be used to estimate partial derivatives and possess the zero sum property of Equation 6.

III.A. Taylor Series Least Squares Method

The Taylor series least squares method (TLS) involves expanding a function from the cloud center, (x_0, y_0), to the j^{th} member of its local cloud. In the TLS, any number of terms may be retained in the Taylor series, with increasing matrix size as the number of retained terms increases. Truncating after the linear terms, we obtain

$$\Delta\phi_{0j} = \Delta x_{0j}\frac{\partial\phi_0}{\partial x} + \Delta y_{0j}\frac{\partial\phi_0}{\partial y}, \quad j = 1,\ldots,n \tag{17}$$

The weighted least squares problem to obtain the unknown partial derivatives may be expressed as[40]

$$\text{minimize} \sum_{j=1}^{n} w_{0j}\left[\Delta\phi_{0j} - \Delta x_{0j}\frac{\partial\phi_0}{\partial x} - \Delta y_{0j}\frac{\partial\phi_0}{\partial y}\right]^2, \quad \text{wrt} \quad \frac{\partial\phi_0}{\partial x}, \frac{\partial\phi_0}{\partial y} \tag{18}$$

A simple inverse distance weighting function of the following form may be used to improve accuracy and conditioning:[41]

$$w_{0j} = \frac{1}{(\Delta x_{0j}^2 + \Delta y_{0j}^2)^{p/2}}, \quad p \geq 0. \tag{19}$$

In this work, a value of $p = 1$ was used. By constraining $n > 2$ for a two-dimensional linear fit, a non-square system of equations arising from Equation 18 results. In practice, the normal equations have been used to solve the least squares problem, which leads to simple explicit formulas for the derivative weight coefficients:

$$\frac{\partial \phi_0}{\partial x} \approx \sum_{j=1}^{n} a_{0j}\Delta\phi_{0j}, \quad a_{0j} = \frac{w_{0j}\Delta x_{0j}\sum w\Delta y^2 - w_{0j}\Delta y_{0j}\sum w\Delta x\Delta y}{\sum w\Delta x^2 \sum w\Delta y^2 - \left(\sum w\Delta x\Delta y\right)^2} \tag{20}$$

$$\frac{\partial \phi_0}{\partial y} \approx \sum_{j=1}^{n} b_{0j}\Delta\phi_{0j}, \quad b_{0j} = \frac{w_{0j}\Delta y_{0j}\sum w\Delta x^2 - w_{0j}\Delta x_{0j}\sum w\Delta x\Delta y}{\sum w\Delta x^2 \sum w\Delta y^2 - \left(\sum w\Delta x\Delta y\right)^2}, \tag{21}$$

Here, the summations are all $j = 1, \ldots, n$.

The TLS weights are the cheapest computationally to obtain of the three methods described here. Only the inverse of a $2x2$ matrix is needed, which is trivially computed. It can be seen from Equations 20-21 that the TLS method does not require the use of the coefficients at the cloud center, (a_{00}, b_{00}). This is a result of the fact that the approximation passes directly through the node at the cloud center, as shown in Figure 3(a). Therefore, we may set these weights to obtain the zero sum property of the form of Equation 6 if we like. This is done by setting

$$a_{00} = -\sum_{j=1}^{n} a_{0j}, \quad b_{00} = -\sum_{j=1}^{n} b_{0j}. \tag{22}$$

III.B. Polynomial Basis Least Squares Method

The polynomial basis least squares method (PLS) also uses least squares to obtain derivative weights, but makes use of polynomial basis functions instead of a Taylor series expansion. While the formulation of the PLS and TLS methods are conceptually different, they result in very similar approximations. The TLS approach solves directly for the partial derivatives, while the PLS approach fits a function to discrete data, which is then differentiated to compute partial derivatives. The PLS method is easily visualized in one dimension, as shown in Figure 3(b). Note that the computed polynomial is not required to pass through the function values at the nodes, including the cloud center at node 0. This is a subtle difference from the TLS approach.

The PLS method begins by selecting a polynomial basis, which determines the order of accuracy of the method. Here, a basis up to linear terms in two dimensions is used. The polynomial basis is then used to generate an approximate solution, $\hat{\phi}(x, y)$, over a local cloud of points:

$$\hat{\phi}(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y. \tag{23}$$

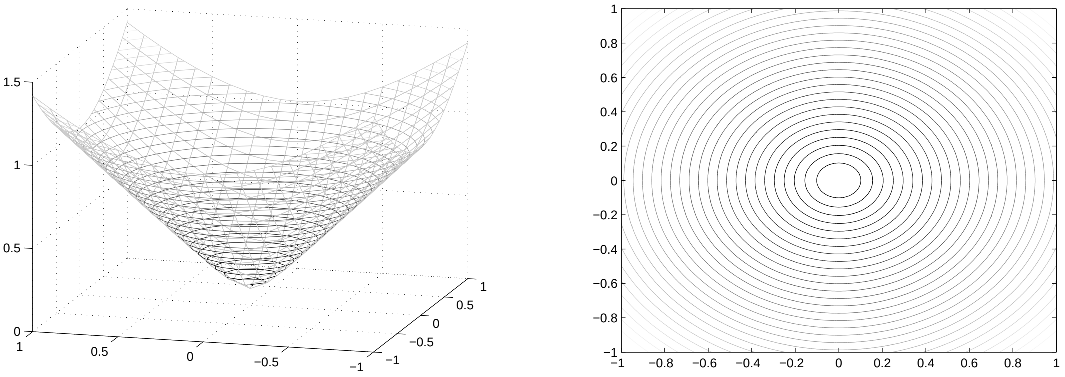
It is helpful to define a local origin at $(x_0, y_0) = (0, 0)$. The weighted least squares problem we seek to solve is

$$\text{minimize} \sum_{j=1}^{n} w_{0j}\left[\phi_j - \alpha_1 - \alpha_2 x_j - \alpha_3 y_j\right]^2 \quad \text{wrt} \quad \alpha_1, \alpha_2, \alpha_3. \tag{24}$$

While many weighting schemes have been used,[4] here a fixed weighting based on a normalized Gaussian function is used:

$$w_{0j} = \frac{e^{-\left(\frac{r_j}{c}\right)^2} - e^{-\left(\frac{r_m}{c}\right)^2}}{1 - e^{-\left(\frac{r_m}{c}\right)^2}}. \tag{25}$$

Here, r_j is the Euclidean distance of point j from the local origin, $r_m = (1 + \epsilon)r_{max}$ is a multiple of the maximum nodal distance in the cloud, and $c = \kappa r_m$. In practice, $\epsilon = 1$ and $\kappa = \frac{1}{2}$ have been used.



(a) Surface plot.

(b) Contour plot.

Figure 4. Multiquadric radial basis function.

Unlike the solution to the TLS least squares problem, the solution to the PLS least squares problem is tedious to write explicitly since it involves the inversion of a 3x3 matrix. However, it may be solved quite easily using Cramer's rule. The derivatives for which we seek are obtained easily by noting that

$$\frac{\partial \phi_0}{\partial x} = \alpha_2, \quad \frac{\partial \phi_0}{\partial y} = \alpha_3, \tag{26}$$

and may be cast into the form of Equation 5. Oñate[4] showed that the approximation, $\hat{\phi}(x, y)$, can represent all the functions in the chosen polynomial basis exactly. This implies that for constant ϕ,

$$\frac{\partial \phi_0}{\partial x} = \phi_0 \sum_{j=1}^{n} a_{0j} = 0.$$

Since ϕ_0 is arbitrary, this implies that the zero sum property of Equation 6 is satisfied. The same applies to the coefficients b_{0j}.

III.C. Radial Basis Function Collocation Method

Both the TLS and PLS methods rely on least squares to obtain derivative weights, but the Radial Basis Function (RBF) method does not. Instead, it seeks to fit a function which passes through all points in the local cloud. Many RBF methods attempt to fit the function through a single global cloud, but this results in a very large, non-sparse matrix for CFD problems with large numbers of unknowns. This work utilizes the RBF method on local clouds of points. While there are many types of radial basis functions, multiquadric functions are used here. The multiquadric function centered at node i may be expressed as

$$\theta_i(x, y) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + c^2}, \tag{27}$$

where c is a free parameter. Unfortunately, there exists no analysis to guide in the selection of c. Usually it is empirically determined based on accuracy and convergence considerations. Surface and contour plots of the multiquadric function of Equation 27 are shown in Figure 4.

The radial basis approximating function, $\hat{\phi}(x, y)$, takes the form

$$\hat{\phi}(x, y) = \boldsymbol{t}^T(x, y)\boldsymbol{\lambda} + \boldsymbol{p}^T(x, y)\boldsymbol{\alpha}, \tag{28}$$

where $\boldsymbol{t}$ and $\boldsymbol{p}$ are the vectors of radial basis functions and optionally appended polynomials respectively:

$$\boldsymbol{t}^T(x, y) = \begin{bmatrix} \theta_0(x, y) & \theta_1(x, y) & \cdots & \theta_n(x, y) \end{bmatrix},$$

$$\boldsymbol{p}^T(x, y) = \begin{bmatrix} p_1(x, y) & p_2(x, y) & \cdots & p_m(x, y) \end{bmatrix}.$$

The appended polynomials are necessary in some cases to ensure a non-singular approximation, and improve consistency. For a partition of unity scheme with the zero-sum property, a polynomial of at least degree 0 must be added. However, Schaback and Wendlend[42] showed at least a linear polynomial should be added to the multiquadric approximation to ensure an invertible system.

The unknown weights, $\boldsymbol{\lambda}$ and $\boldsymbol{\alpha}$, may be obtained by invoking the following $n+1$ equations and m orthogonality constraints:

$$\hat{\phi}(x_i, y_i) = \phi_i, \quad i = 0, \ldots, n, \qquad \sum_{i=1}^{n} \lambda_i p_j(x_i, y_i) = 0, \quad j = 1, \ldots, m. \tag{29}$$

The equations in 29 may be recast as

$$\begin{bmatrix} T & P^T \\ P & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{\phi} \\ \mathbf{0} \end{Bmatrix}, \tag{30}$$

where the submatrices, T and P, are defined as

$$T^T = \begin{bmatrix} \boldsymbol{t}(x_0, y_0) & \boldsymbol{t}(x_1, y_1) & \cdots & \boldsymbol{t}(x_n, y_n) \end{bmatrix},$$

$$P = \begin{bmatrix} \boldsymbol{p}(x_0, y_0) & \boldsymbol{p}(x_1, y_1) & \cdots & \boldsymbol{p}(x_n, y_n) \end{bmatrix},$$

and the vector of nodal values, $\boldsymbol{\phi}$, is defined as

$$\boldsymbol{\phi}^T = \begin{bmatrix} \phi_0 & \phi_1 & \cdots & \phi_n \end{bmatrix}.$$

Writing the inverse of the symmetric matrix in Equation 30 as

$$\begin{bmatrix} T & P^T \\ P & 0 \end{bmatrix}^{-1} = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix}, \tag{31}$$

the approximation of Equation 28 becomes

$$\hat{\phi}(x, y) = \left(\boldsymbol{t}^T(x, y) A + \boldsymbol{p}^T(x, y) B \right) \boldsymbol{\phi}. \tag{32}$$

Once again, since we are interested in derivatives for PDE discretization, we seek the derivatives of $\hat{\phi}$, which may be obtained from

$$\frac{\partial \hat{\phi}(x_0, y_0)}{\partial x} = \left(\frac{\partial \boldsymbol{t}^T(x_0, y_0)}{\partial x} A + \frac{\partial \boldsymbol{p}^T(x_0, y_0)}{\partial x} B \right) \boldsymbol{\phi} = \boldsymbol{a}^T \boldsymbol{\phi},$$

$$\frac{\partial \hat{\phi}(x_0, y_0)}{\partial y} = \left(\frac{\partial \boldsymbol{t}^T(x_0, y_0)}{\partial y} A + \frac{\partial \boldsymbol{p}^T(x_0, y_0)}{\partial y} B \right) \boldsymbol{\phi} = \boldsymbol{b}^T \boldsymbol{\phi},$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are the vectors of the derivative metric terms for which we seek.

Thus, we arrive at the same form for the derivatives as the PLS and TLS methods. Because all methods fit into the same convenient summation form for derivatives, we can easily compare the performance the schemes.

IV. Comparison of Meshless Schemes for Subsonic and Transonic Flow

All three meshless schemes of section III were implemented into the common algorithm of section II. The meshless schemes were compared with an inviscid structured finite volume code based on the CUSP algorithm of Jameson,[32,38] denoted "FV" in the following tables. The finite volume code uses a conservative approach of numerical fluxes, taking on the form,

$$V_{ij} \frac{d\boldsymbol{w}_{ij}}{dt} + \boldsymbol{h}_{i+\frac{1}{2},j} - \boldsymbol{h}_{i-\frac{1}{2},j} + \boldsymbol{h}_{i,j+\frac{1}{2}} - \boldsymbol{h}_{i,j-\frac{1}{2}} = 0, \tag{33}$$

where

$$\boldsymbol{h}_{i+\frac{1}{2},j} = \frac{1}{2}(\boldsymbol{f}_{i,j} + \boldsymbol{f}_{i+1,j}) - \frac{1}{2}(\boldsymbol{d}_{i+\frac{1}{2},j}) \qquad (34)$$

is the numerical flux. The numerical flux, $\boldsymbol{h}$, is based on the average of the directed Euler fluxes, $\boldsymbol{f}$, augmented with a diffusive flux, $\boldsymbol{d}$, which is based on the CUSP formulation. Similar procedures were used for the finite volume algorithm to reach steady state, including explicit Runge-Kutta local time stepping, implicit residual smoothing, enthalpy damping, and multigrid. Similar point densities for both the finite volume and meshless schemes were used, with 160 points on the surface of airfoils, and roughly 6400 nodes in the interior of the domain. The point clouds were extracted from unstructured meshes obtained with the *Delaundo*[43] package.

Figures 5-6 show meshless pressure curves overplotted with the finite volume results for shock-free solutions. All methods perform essentially equally well, computing lift and drag closely to the finite volume results, as shown in Tables 2-3. Figure 5 shows flow over a NACA 0012 at $M = 0.5$, $\alpha = 3^o$, which is computed quite accurately by all methods. Additionally, Figure 6 includes the KORN airfoil, which is designed to be shock free at $M = 0.75$, and $\alpha = 0^o$. The least squares methods give a shock free upper surface, while the radial basis method produces a very weak shock. Still all meshless methods performed well for this case.

However, discrepancies arise in the presence of shock waves, as shown in Figures 7-8 and Tables 4-5. The least squares methods appear to give the correct shock locations and jumps. However, significant differences in both shock location and magnitude are present with the radial basis method. Instead of differences of less than 3% in lift and drag coefficients as with the least squares methods, the RBF method produces differences in the 10-15% range. A close examination of the pressure curves reveals discrepancies at shocks. A variety of shape parameters in Equation 27 were tested to try to improve the accuracy of the shock capturing with the RBF method with little success.

It is possible that the incorrect shock capturing may be due to the non-conservative nature of the radial basis scheme. To investigate this, a study was performed to assess the ability of the scheme to satisfy the Rankine-Hugoniot shock jump conditions:

$$\begin{aligned} (\rho u)_L &= (\rho u)_R, \\ (P + \rho u^2)_L &= (P + \rho u^2)_R, \\ H_{tL} &= H_{tR} \end{aligned}$$

Here the left (L) and right (R) states must satisfy these conditions on either side of the shock. The error in each of these conditions for the FV method, the TLS method, and the RBF method was examined. The results for a transonic case are shown in Table 1.

Table 1. Error in Rankine-Hugoniot condtions, NACA 0012, $M = 0.8$, $\alpha = 1.25^o$.

	$\lvert(\rho u)_L - (\rho u)_R\rvert$	$\lvert(P + \rho u^2)_L - (P + \rho u^2)_R\rvert$	$\lvert H_{tL} - H_{tR}\rvert$
FV	0.0023	0.0035	0.0000
TLS	0.0026	0.0036	0.0000
RBF	0.0015	0.0047	0.0000

As the table shows, there is no clear evidence that the radial basis method is actually miscapturing shocks. The error in the Rankine-Hugoniot conditions is on the same order for all methods. Tests with other transonic cases showed similar results. Therefore, it is possible that some source other than non-conservation is the cause of the misplaced shocks. After all, the least squares methods are also non-conservative, but consistently capture shocks correctly. Possible explanations may include a failure to properly satisfy the Kutta condition or boundary conditions. Further tests are needed to determine the cause of the poor accuracy for transonic flow using the radial basis method.

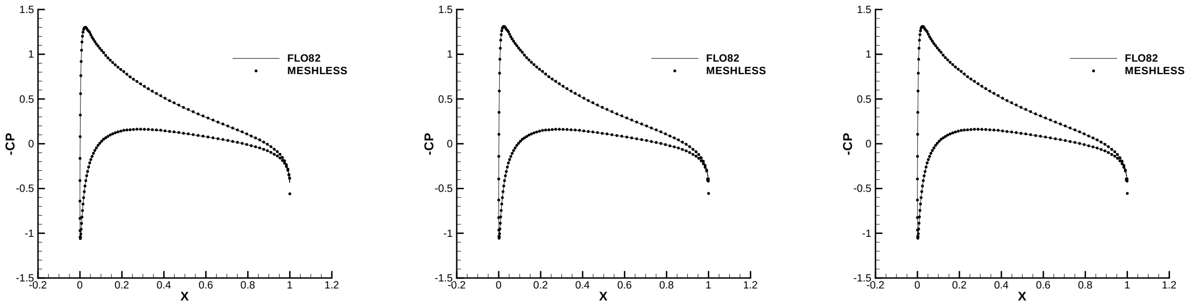
V. Conclusions

The results obtained so far indicate that for transonic flows with shocks, least squares methods give more accurate results than radial basis methods. The RBF method showed significant discrepancies in both shock location and magnitude when compared with finite volume solutions. The most accurate of the schemes tested was the PLS method, which performed extremely well compared with the finite volume results. For smooth flows without shocks, all schemes performed essentially equally well.

Future work will include the addition of higher order terms for use in the PLS and TLS methods. While these schemes are generally considered "meshless", they may also be used transparently on arbitrary grid types. Thus, these methods may present an avenue towards high order grid-based methods, especially on unstructured grids. These schemes also have applications for reconstruction with upwind finite volume schemes on meshes. Additionally, all schemes tested here can easily be extended to three dimensions.

VI. Acknowlegements

Development was performed primarily under a National Defense Science and Engineering Graduate (NDSEG) fellowship. Aaron Katz has benefited greatly from the NDSEG program administered through the High Performance Computing Modernization Office of the Department of Defense. Likewise, Professor Jameson has benefited greatly from the long term and continuing support of the AFOSR Computational Mathematics Program directed by Dr. Fariba Fahroo. We would also like to acknowledge the members of the HPC Institute for Advanced Rotorcraft Modeling and Simulation (HIARMS) team located at NASA Ames Research Center, whose continuing support has facilitated certain aspects of this work.

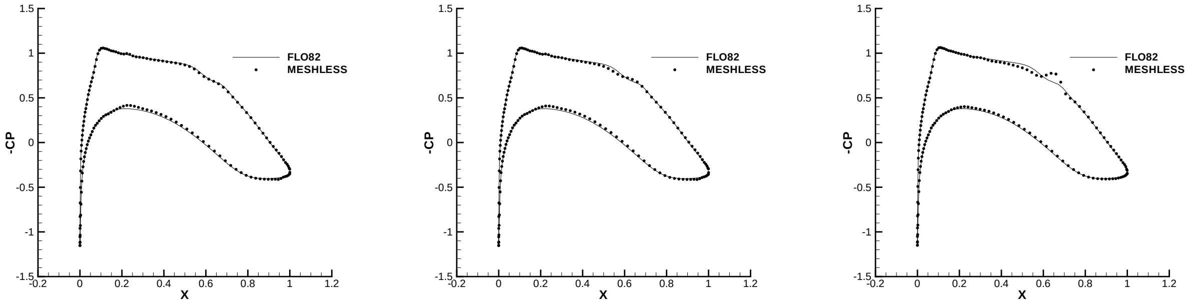


(a) TLS (b) PLS (c) RBF

Figure 5. Surface pressure coefficient NACA 0012, $M = 0.5$, $\alpha = 3.0^o$.

Table 2. Lift and drag coefficients, NACA 0012, $M = 0.5$, $\alpha = 3.0^o$.

	c_l	% difference	c_d	% difference
FV	0.4313	-	0.0000	-
TLS	0.4312	0.0	0.0007	-
PLS	0.4326	0.3	0.0005	-
RBF	0.4340	0.6	0.0002	-

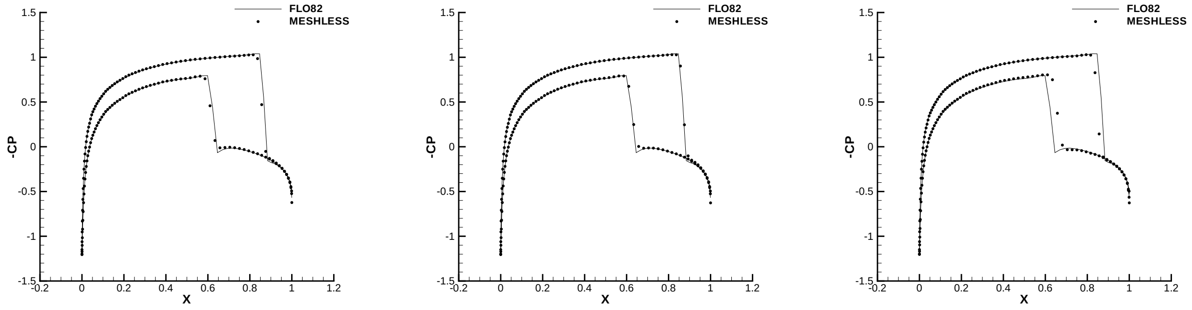


(a) TLS (b) PLS (c) RBF

Figure 6. Surface pressure coefficient KORN airfoil, $M = 0.75$, $\alpha = 0.0^o$.

Table 3. Lift and drag coefficients, KORN airfoil, $M = 0.75$, $\alpha = 0.0^o$.

	c_l	% difference	c_d	% difference
FV	0.6308	-	0.0000	-
TLS	0.6178	2.1	0.0009	-
PLS	0.6164	2.3	0.0011	-
RBF	0.6200	1.7	0.0010	-

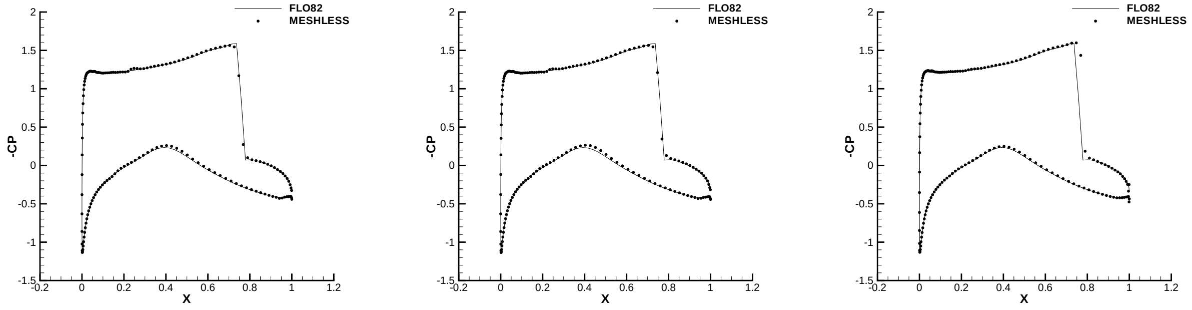


(a) TLS (b) PLS (c) RBF

Figure 7. Surface pressure coefficient NACA 0012, $M = 0.85$, $\alpha = 1.0^o$.

Table 4. Lift and drag coefficients, NACA 0012, $M = 0.85$, $\alpha = 1.0^o$.

	c_l	% difference	c_d	% difference
FV	0.3891	-	0.0582	-
TLS	0.3830	1.6	0.0565	2.9
PLS	0.3883	0.2	0.0593	1.9
RBF	0.3343	14.1	0.0570	2.1



(a) TLS (b) PLS (c) RBF

Figure 8. Surface pressure coefficient RAE 2822, $M = 0.75$, $\alpha = 3.0^o$.

Table 5. Lift and drag coefficients, RAE 2822, $M = 0.75$, $\alpha = 3.0^o$.

	c_l	% difference	c_d	% difference
FV	1.1481	-	0.0486	-
TLS	1.1319	1.2	0.0481	1.0
PLS	1.1347	0.9	0.0491	1.0
RBF	1.1744	2.3	0.0545	12.1

References

[1] Löhner, R. and nate, E. O., "An Advancing Front Point Generation Technique," *Communications in Numerical Methods in Engineering*, Vol. 14, 1998, pp. 1097–1108.

[2] Löhner, R., Sacco, C., and nate, E. O., "A General Advancing Front Technique for Filling Space with Arbitrary Objects," *Int. J. Numer. Meth. Engng.*, Vol. 61, 2004, pp. 1977–1991.

[3] Batina, J. T., "A Gridless Euler/Navier-Stokes Solution Algorithm for Complex Aircraft Applications," *AIAA paper* 1993-0333, AIAA 31st Aerospace Sciences Meeting and Exhibit, Reno, NV, January 1993.

[4] Onate, E., Idelsohn, S., Zienkiewicz, O. C., Taylor, R. L., and Sacco, C., "A Stabilized Finite Point Method for Analysis of Fluid Mechanics Problems," *Comput, Methods Appl. Mech. Engrg.*, Vol. 139, 1996, pp. 315–346.

[5] Onate, E. and Idelsohn, S., "A Mesh-free Finite Point Method for Advective-diffusive Transport and Fluid Flow Problems," *Computational Mechanics*, Vol. 21, 1998, pp. 283–292.

[6] Löhner, R., Sacco, C., nate, E. O., and Idelsohn, S., "A Finite Point Method for Compressible Flow," *Int. J. Numer. Meth. Engng.*, Vol. 53, 2002, pp. 1765–1779.

[7] Ghosh, A. K. and Deshpande, S. M., "Least Squares Kinetic Upwind Method for Inviscid Compressible Flows," *AIAA paper* 1995-1735, AIAA 12th Computational Fluid Dynamics Conference, San Diego, CA, June 1995.

[8] Anandhanarayanan, K. and Nagarathinam, M., "Parallelisation of a Gridfree Kinetic Upwind Solver," *AIAA paper* 2005-4628, AIAA 17th Computational Fluid Dynamics Conference, Toronto, June 2005.

[9] Harish, G. and Pavanakumar, M., "Store Separation Dynamics using Grid-free Euler Solver," *AIAA paper* 2006-3650, AIAA 24th Applied Aerodynamics Conference, San Francisco, CA, June 2006.

[10] Srinarayana, N., Gonzalez, L. F., Whitney, E. J., and Srinivas, K., "Aerodynamic Optimisation using a Robust Evolutionary Algorithm and Grid-free Flowsolver," *AIAA paper* 2006-52, AIAA 44th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2006.

[11] Kumar, G. N. S., Mahendra, A. K., and Rao, S. V. R., "Shape Optimization Using Hybrid GA-ACO Method and Grid-free CFD Solver," *AIAA paper* 2007-3830, AIAA 18th Computational Fluid Dynamics Conference, Miami, FL, June 2007.

[12] Ramesh, V. and Deshpande, S. M., "Euler Computations on Arbitrary Grids Using LSKUM," *Computational Fluid Dynamics 2000: Proceedings of the First International Conference on Computational Fluid Dynamics*, edited by N. Satofuka, Springer-Verlag, 2000, pp. 783–784.

[13] Deshpande, S. M., Anil, N., Arora, K., Malagi, K., and Varma, M., "Some Fascinating New Develpments in Kinetic Schemes," *Proceedings Workshop on Modeling and Simulation in Life Sciences, Materials and Technology*, edited by A. Avudainayagam, P. Misra, and S. Sundar, 2004, pp. 43–64.

[14] Deshpande, P. C. S. M., "A New Grid-free Method for Conservation Laws," *Computational Fluid Dynamics 2002: Proceedings of the Second International Conference on Computational Fluid Dynamics*, edited by S. Armfield, P. Morgan, and K. Srinivas, 2002, pp. 128–133.

[15] Deshpande, S. M., Anandhanarayanan, K., Praveen, C., and Ramesh, V., "Theory and Application of 3-D LSKUM Based on Entropy Variables," *International Journal for Numerical Methods in Fluids*, Vol. 40, 2002, pp. 47–62.

[16] Sridar, D. and Balakrishnan, N., "Convergence Acceleration of an Upwind Least Squares Finite Difference based Meshless Solver," *AIAA Journal*, Vol. 44, 2006, pp. 2189–2196.

[17] Sridar, D. and Balakrishnan, N., "An Upwind Finite Difference Scheme for Meshless Solvers," *Journal of Computational Physics*, Vol. 189, 2003, pp. 1–29.

[18] Kirshman, D. J. and Liu, F., "Gridless Boundary Condition Treatment for a Non-Body-Conforming Mesh," *AIAA paper* 2002-3285, AIAA 32nd Fluid Dynamics Conference, St. Louis, MO, June 2002.

[19] Koh, E. P. C. and Tsai, H. M., "Euler Solution Using Cartesian Grid with Least Squares Technique," *AIAA paper* 2003-1120, AIAA 41st Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2003.

[20] Kirshman, D. J. and Liu, F., "Cartesian Grid Solution of the Euler Equations Using a Gridless Boundary Treatment," *AIAA paper* 2003-3974, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.

[21] Luo, H. and Baum, J. D., "A Hybrid Cartesian Grid and Gridless Method for Compressible Flows," *AIAA paper* 2005-0492, AIAA 43rd Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2005.

[22] Kamatsuchi, T., "Turbulent Flow Simulation Around Complex Geometries with Cartesian Grid Method," *AIAA paper* 2007-1459, AIAA 45th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 2007.

[23] Kirshman, D. J. and Liu, F., "A Gridless Boundary Condition Method for the Solution of the Euler Equations on Embedded Cartesian Meshes with Multigrid," *Journal of Computational Physics*, Vol. 201, 2004, pp. 119–147.

[24] Katz, A. and Jameson, A., "Edge-based Meshless Methods for Compressible Viscous Flow with Applications to Overset Grids," *AIAA paper* 2008-3989, AIAA 38th Fluid Dynamics Conference, Seattle, WA, June 2008.

[25] Kansa, E. J., "Multiquadrics–A Scattered Data Approximation Scheme with Applications to Computational Fluid-Dynamics–I," *Computers Math. Applic.*, Vol. 19, 1990, pp. 127–145.

[26] Kansa, E. J., "Multiquadrics–A Scattered Data Approximation Scheme with Applications to Computational Fluid-Dynamics–II," *Computers Math. Applic.*, Vol. 19, 1990, pp. 147–161.

[27] Divo, E. and Kassab, A. J., "Efficient Localized Meshless Modeling of Natural Convective Viscous Flows," *AIAA paper* 2006-3089, 9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, San Francisco, CA, June 2006.

[28] Divo, E. and Kassab, A. J., "A Meshless Method for Conjugate Heat Tranfer Problems," *Engineering Analysis with Boundary Elements*, Vol. 29, 2005, pp. 136–149.

[29] Chinchapatnam, P. P., Djidjeli, K., and Nair, P. B., "Meshless RBF Collocation for Steady Incompressible Viscous Flows," *AIAA paper* 2006-3525, AIAA 36th Fluid Dynamics Conference and Exhibit, San Francisco, CA, June 2006.

[30] Power, H. and Barraco, V., "Comparison Analysis Between Unsymmetric and Symmetric Radial Basis Function Collocation Methods for the Numerical Solution of Partial Differential Equations," *Computers Math. Applic.*, Vol. 43, 2002, pp. 551–583.

[31] Shu, C., Ding, H., Chen, H. Q., and Wang, T. G., "An Upwind Local RBF-DQ Method for Simulation of Inviscid Compressible Flows," *Comput, Methods Appl. Mech. Engrg.*, Vol. 194, 2005, pp. 2001–2017.

[32] Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics 1 Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence," *International Journal of Computational Fluid Dynamics*, Vol. 4, 1995, pp. 171–218.

[33] Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 49, 1983, pp. 357–393.

[34] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," *AIAA paper* 1981-1259, AIAA 14th Fluid and Plasma Dynamic Conference, Palo Alto, CA, June 1981.

[35] Duarte, C. and Oden, J., "Hp clouds - a meshless method to solve boundary-value problems," Technical Report 95-05, Texas Institute for Computational and Applied Mathematics, Austin, TX, 1995.

[36] Mavriplis, D. J., Jameson, A., and Martinelli, L., "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes," *AIAA paper* 1989-0120, AIAA 27th Aerospace Sciences Meeting, Reno, NV, January 1989.

[37] Katz, A. and Jameson, A., "Multicloud convergence Acceleration for Complex Applications on Arbitrary Grids," *IEEE Proceedings of the 2008 DoD High Performance Computing Modernization Program*, 2009.

[38] Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics 2 Artificial Diffusion and Discrete Shock Structure," *International Journal of Computational Fluid Dynamics*, Vol. 5, 1995, pp. 1–38.

[39] Pulliam, T., "Efficient Solution Methods for the Navier-Stokes Equations," *Lecture Notes for the Von Karman Institute for Fluid Dynamics Lecture Series*, 1986.

[40] Chandrashekar, P., "Development and Application of Kinetic Meshless Methods for Euler Equations," Phd thesis, IISc, 2004.

[41] Mavriplis, D. J., "Revisiting the Least-Squares Procedure for Gradient Reconstruction on Unstructured Meshes," *AIAA paper* 2003-3986, AIAA 16th Computational Fluid Dynamics Conference, Orlando, FL, June 2003.

[42] Schaback, R. and Wendlend, H., "Characterization and Construction of Radial Basis Functions," *Multivariate Approximations and Applications*, edited by N. Dyn, D. Leviatan, D. Levin, and A. Pinkus, Cambridge University Press, 2001, pp. 1–24.

[43] Müller, J., "On Triangles and Flow," Phd thesis, The University of Michigan, 1996.