Meshfree Euler Solver using local Radial Basis Functions for inviscid Compressible Flows

Prasad V. Tota[1]
Flow Science Inc, Santa Fe, NM, 87505

Zhi J. Wang [2]
Iowa State University, Ames, IA, 50011

The existing computational techniques use a mesh to discretize the domain and approximate the solution. Hence the accuracy of the method depends on the quality of mesh. Meshfree methods make an attempt to address these problems due to their mesh dependence or sensitivity. The present meshless method uses differential quadrature (DQ) technique to approximate the derivatives at a point using the information at a set of scattered nodes in its neighborhood. Radial basis functions (RBFs) are used as basis functions. The RBF-DQ technique is used to develop a meshfree Euler solver for inviscid compressible flows. The solver is applied to and validated by various steady state compressible flows.

Nomenclature

c	=	shape parameter
C_p	=	pressure coefficient
dt	=	time step
f	=	generic functions
$\mathbf{F_1}$	=	Flux vector in x-direction
$\mathbf{F_2}$	=	Flux vector in y-direction
$\mathbf{G}$	=	Flux vector along a given direction
i	=	reference node
j	=	supporting node
k	=	index of neighboring nodes
M	=	Mach number
N_I	=	number of supporting nodes
n	=	time index during navigation
$\mathbf{Q}$	=	Vector of conservative variable
r	=	distance between two nodes
T	=	temperature
$\mathbf{x}$	=	position vector
α	=	limiter
φ	=	radial basis function

I. Introduction

ONE of the major challenges in Computational Fluid Dynamics (CFD) is the generation of a suitable mesh. For a complex configuration, generation of a good quality mesh can be very expensive in terms of human labor and CPU time. For practical problems the geometries encountered can be highly irregular and not strictly convex. Hence it is desirable to be able to solve partial differential equations (PDEs) over an irregular domain and discretize it. In order to overcome this problem a number of numerical schemes have been proposed in the past two decades, which are referred to as gridless or meshless schemes. They are also known as meshfree methods. In future the terms

[1] CFD Engineer, Flow Science Inc, 683 Harkle Rd. Suite A, and AIAA Member
[2] Associate Professor, Dept. of Aerospace Engg, 2249 Howe Hall, Iowa State University, Associate editor of AIAA

gridless, meshless and meshfree will be used synonymously. These schemes completely discard the idea of a mesh for the spatial discretization of the PDEs governing the flow. The meshfree term not only suggests that they do not depend on any mesh, but also implies that they can be applied to any kind of mesh- structured, unstructured or hybrid.
In order to overcome this problem a number of numerical schemes have been proposed in the past two decades, which are referred to as gridless or meshless schemes. They are also known as meshfree methods. In future the terms gridless, meshless and meshfree will be used synonymously. These schemes completely discard the idea of a mesh for the spatial discretization of the PDEs governing the flow. The meshfree term not only suggests that they do not depend on any mesh, but also implies that they can be applied to any kind of mesh- structured, unstructured or hybrid.

The numerical solution of partial differential equations (PDEs) has been dominated by finite difference methods (FDM), finite Element methods (FEM) and finite volume methods (FVM). The common feature among all these methods is that they all require a mesh to discretize the PDEs. The objective of meshless methods is to eliminate at least part of this mesh dependence by constructing the approximation entirely in terms of nodes. In these methods moving discontinuities can usually be treated without remeshing with slight compromise with accuracy. Hence it is possible to solve a large class of problems computationally using meshless methods more accurately and some times efficiently than the conventional mesh based methods. The nodes can be created in a fully automated manner without any human intervention and hence the time spent in mesh generation is reduced[2].

The origin of meshless methods can be traced back to about thirty years ago, but very little research was done until the past decade. The starting point which seems to have the longest history is the smooth particle hydrodynamics (SPH) method[3] by Lucy, who used it to model astrophysical phenomenon. One common characteristic among all the meshfree methods is that they can construct the functional approximation or interpolation entirely from the information at a set of scattered nodes or points. These methods don't need to store any prespecified connectivity or relationship among these scattered nodes. Some of the well known meshless methods are smooth particle hydrodynamics (SPH) method[3], the diffuse element method[4], the element free Galerkin method (EFGM)[2,5], the reproducing kernel particle method (RKPM)[6], the partition of unity method[7], the hp-clouds method[8], the finite point method[9], the meshless Petrov-Galerkin (MLPG) method[10], and the general finite difference method. One of the main advantages of meshfree methods is that it is computationally easy to add or remove nodes from a preexisting set of nodes. On the contrary in conventional methods addition or removal of a point or an element would lead to heavy remeshing and hence computationally difficult to implement.

A. Meshless solvers using radial basis functions

In the past decade researchers have been trying to develop another group of meshless methods which are the radial basis functions (RBFs) and have become attractive for solving partial differential equations (PDEs). Although RBFs were initially developed for multivariate data and function interpolation, their truly meshfree nature has motivated researchers to employ them in solving PDEs. Kansa[13] has done some pioneering work in application of RBFs to solve PDEs. Other great contributions in the area of RBFs come from Fornberg[13], Hon[25] and Wu, Chen[21] and Tanaka. RBFs have found applications in various engineering problems like structural dynamics[19], fluid dynamics[21, 22] and fluid structure interaction[20] to name a few.

RBFs when used as base functions for multi-variate data interpolation show favorable properties like high efficiency and good quality. RBFs naturally have the ability of dealing with scattered data. Another advantage is that they have high-order of accuracy than the typical finite difference schemes on a scattered distribution of nodes. In the present study a meshless Euler solver based on radial basis functions has been developed to solve inviscid compressible fluid flows. The algorithm consists of two parts, first part deals with the derivative approximation using differential quadrature (DQ) method with RBFs as basis functions. The latter part consists of implementing a suitable upwind scheme to evaluate the fluxes. RBF based solver is applied to Euler equations to solve compressible flow problems.

II. Differential quadrature technique using Radial basis functions

The local RBF-differential quadrature (DQ) method is an interpolation technique in which the radial basis functions are used as basis functions. The function approximation is by RBFs and the derivative approximation by differential quadrature (DQ). The partial derivative at a reference point can be approximated by a weighted linear sum of function values at a set of discrete neighboring nodes within its support domain. These weighting coefficients at the supporting nodes are determined by a set of basis functions.

Consider Ω as on open domain of $\Re^d$, d = 1 and 2. We want to approximate the derivative of a continuous function $f:\Omega\to\Re$ at node $\mathbf{x_I}$ where the function values at node $\mathbf{x_I}$ and its supporting nodes $\mathbf{x_I^j}$, j =1, 2... N_I are known as shown in Fig.1. The DQ approximation of the m[th] order derivative of a function *f(x)* in the *x*-direction at the node $\mathbf{x_I}$ can be expressed as

$$\left.\frac{\partial^m f}{\partial x^m}\right|_{x_I} = \sum_{j=0}^{N_I} w_{I,j}^{(m)} f\left(x_I^j\right), \qquad j = 0,1,2,...,N. \tag{1}$$

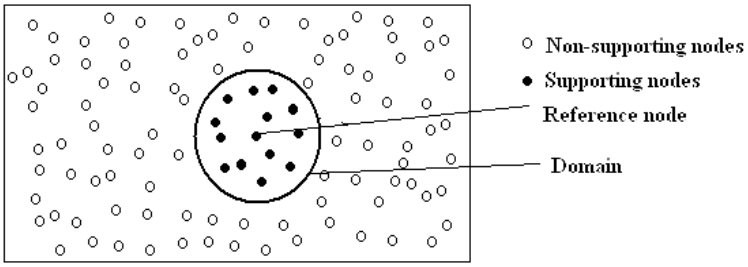


Figure 1. Supporting domain and nodes around a reference node

$f\left(x_I^j\right)$ = function values at the scattered nodes

$w_{I,j}^{(m)}$ = weight coefficients at the nodes

$\mathbf{N_I}$ = number of supporting points within the domain

$\mathbf{x_I^j}$ = the coordinates of supporting nodes of $\mathbf{x_I}$

RBFs are used as basis functions to determine the weighting coefficients. The most commonly used RBFs are multiquadrics (MQs): $\varphi(r)=\sqrt{r^2+c^2}, c>0,$ Thin-Plate Splines (TPS): $\varphi(r)=r^2\log(r),$ Gaussians: $\varphi(r)=e^{-\alpha r^2}, \alpha>0,$ Inverse MQs: $\varphi(r)=1/\left(\sqrt{r^2+c^2}\right), c>0$. MQs are most extensively used RBFs and were proposed by Hardy. Franke studied all the RBFs and found that MQs generally perform better than other RBFs for the interpolation of 2D scattered data. The exponential convergence of MQs makes them superior to other RBFs such as thin plate splines (TPS) or Gaussians. In the present work we will be using the MQ-RBFs to determine the weight vector $\left|w_{I,j}^{(m)}\right|$.

$$\frac{\partial\varphi_k(\mathbf{x_I})}{\partial x} = \sum_{j=0}^{N_I} w_{I,j}^{(1x)}\varphi_k\left(\mathbf{x}^j\right) \quad k = 0,1,2,...,N_I. \tag{2}$$

The terms on the left of Eq.2 can be obtained analytically as shown below (Eq.3).

$$\frac{\partial\varphi_k(\mathbf{x_I})}{\partial x} = \frac{(x_I - x_k)}{\sqrt{(x_I - x_k)^2 + (y_I - y_k)^2 + c^2}} \tag{3}$$

For simplicity of notation $\varphi_k(\mathbf{x})$ is used to replace $\varphi\left(\left\|\mathbf{x}-\mathbf{x}_k\right\|\right)$, where $\left\|\mathbf{x}-\mathbf{x}_k\right\|$ is the Euclidean norm. The system of equations can be written clearly in matrix form as represented by Eq.4.

$$\underbrace{\begin{bmatrix} \frac{\partial \varphi_0(\mathbf{x}_I)}{\partial x} \\ \frac{\partial \varphi_1(\mathbf{x}_I)}{\partial x} \\ \vdots \\ \frac{\partial \varphi_{N_I}(\mathbf{x}_I)}{\partial x} \end{bmatrix}}_{\left\{\frac{\partial \varphi(\mathbf{x}_I)}{\partial x}\right\}} = \underbrace{\begin{bmatrix} \varphi_0(\mathbf{x}_I^0) & \varphi_0(\mathbf{x}_I^1) & \cdots & \\ \varphi_1(\mathbf{x}_I^0) & \varphi_1(\mathbf{x}_I^1) & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{N_I}(\mathbf{x}_I^0) & \varphi_{N_I}(\mathbf{x}_I^1) & & \varphi_{N_I}(\mathbf{x}_I^N) \end{bmatrix}}_{[A]} \cdot \underbrace{\begin{bmatrix} w_{I,0}^{(1x)} \\ w_{I,1}^{(1x)} \\ \vdots \\ w_{I,N_I}^{(1x)} \end{bmatrix}}_{\{w\}}. \tag{4}$$

RBFs are globally supported shape functions and the resulting system matrix **[A]** becomes dense if locally supported RBFs are not used. If the collocation matrix **[A]** is non-singular, the coefficient vector $\{w\}$ can be obtained by Eq.5.

$$\{w\} = [A]^{-1} \left\{ \frac{\partial \varphi_k(\mathbf{x_I})}{\partial x} \right\}. \tag{5}$$

The behavior of the collocation matrix [**A**] depends on the type of RBFs used, however it is known that the matrix [**A**] is positive definite for MQ-RBFs. The matrix [**A**] is hence non singular and thus $[\mathbf{A}]^{-1}$ exists for distinct supporting points. The accuracy of a RBF based scheme depends on various parameters, local distribution of points, number of supporting points N_I and free shape parameter *c* which is studied in the subsequent sections.

A. Shape parameter c in local MQ-DQ method

The shape parameter *c* strongly influences the accuracy of the MQ-RBF method. The choice of the shape parameter *c* has been a topic of lot of discussion in the community of RBF researchers. Franke[12] suggested a formula to find the optimum shape parameter *c* as $c = \frac{1.25 \cdot D}{\sqrt{N_I}}$ where D is the radius of the smallest circle and N_I is the number of nodes in the support domain. Hardy suggested another formula for evaluating the shape parameter, $c = 0.815 \cdot d$, where $d = \frac{1}{N_I} \cdot \sum_{i=1}^{N_I} d_i$ and d_i is the distance between the i[th] data point and its nearest neighbor. Kansa[14] suggested a variable shape parameter which increases accuracy up to five orders of magnitude for many monotonic functions (Eq.6).

$$(c_j)^2 = (c_N)^2 \left[\frac{(c_M)^2}{(c_N)^2} \right]^{\frac{(j-1)}{(N-1)}} \qquad \text{where } c_M \text{ and } c_N \text{ are input parameters} \tag{6}$$

A general theoretical analysis of how the shape parameter *c* is associated with the accuracy of the approximation is difficult. Hence a numerical study is performed on first order derivatives of a function of two

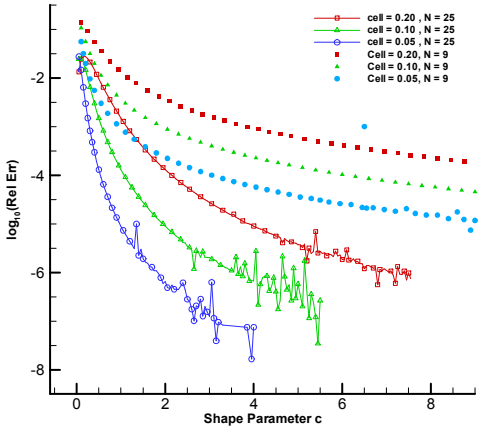


Figure 2. Parameters affecting the accuracy of RBF approximation

variables $f(x,y) = x^2 + y^2$. The numerical error was studied for varying c for three different uniform nodal distributions. It was found that the error decreases as the density of nodal distribution around the reference node increases.

The approximation error is high for low values of *c* and decreases for increasing values until a certain value of $c = c_{max}$, beyond which the numerical error increases/oscillates as seen in Fig.2. Hence there is a specific range of c within which the approximation is numerically stable and shows consistent behavior. Another interesting observation is that the value of c_{max} decreases with increase in number of supporting nodes N_I and/or density of nodes. Similar trend was observed for higher order polynomials in both their first order and higher order derivatives. However for higher order polynomials the numerical error was found to be more sensitive to changes in value of *c*.

III. RBF-differential quadrature to solve elliptic PDEs

In this section RBFs will be implemented to solve PDEs with special attention to elliptic PDEs. In an elliptic problem over a closed domain the boundary conditions are specified and we need to solve for the function $f(x,y)$ over the domain. Nodes are generated within the domain, Ω and on the boundary, $\partial\Omega$. The total number of nodes is N_T, boundary nodes are N_B and internal nodes is N_{Int}. RBF-DQ method is applied to solve poisson equation (Eqn.7) a particular case of elliptic PDE.

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = q(x,y) \text{ where } q(x,y) \text{ is the source term.} \tag{7}$$

The resulting equations are shown in Eqn.8. The superscript term in $w_{I,j}^{2x}$ implies that the weights are corresponding to the second order derivatives with respect to x.

$$\frac{\partial^2 f}{\partial x^2} = \sum_{j=0}^{j=N_B} w_{I,j}^{2x} f\left(\mathbf{x}_I^j\right) \quad \text{and} \quad \frac{\partial^2 f}{\partial y^2} = \sum_{j=0}^{j=N_B} w_{I,j}^{2y} f\left(\mathbf{x}_I^j\right). \tag{8}$$

We can combine Eqn.7 and Eqn.8 to obtain a simplified form Eqn.9

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \sum_{j=0}^{j=N_B} \left(W_{I,j}\right) f\left(\mathbf{x}_I^j\right) \qquad \text{where} \quad W_{I,j} = \left(w_{I,j}^{2x} + w_{I,j}^{2y}\right) \tag{9}$$

Using Eq.7 and Eq.8 we have obtained a system of linear equations (Eq.10) and the value of the function at the internal nodes can be obtained by solving this system of equations. In the present work the unknowns were solved using Gauss-elimination with no pivoting.

$$[\mathbf{W}][\mathbf{f}] = [\mathbf{q}] \tag{10}$$

The resulting system matrix [W] is highly sparse. If we decrease the number of nodes within the support domain the sparseness of the matrix increases. A large bandwidth of matrix [W] results in small errors but is computationally expensive. On the other hand a small bandwidth is computational easier to solve but leads to numerical errors[17]. Hence it is important to optimize between computational efficiency and numerical accuracy when the RBF-method is used.

A. 2D Poisson problem

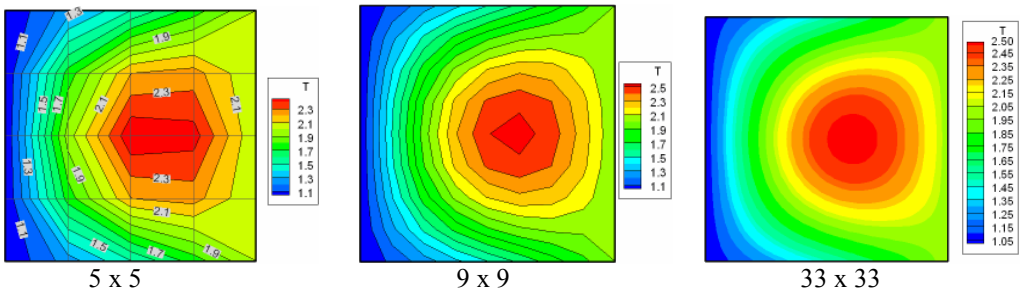
Consider the Poisson equation governing temperature distribution *T*(x,y) in a unit square domain $\Omega(0 < x < 1, 0 < y < 1)$ with a source term Eqn.11. The boundary conditions are $T(x,y) = 1 + x$ on the boundary, $\partial\Omega$.

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = -2\pi^2 (\sin \pi x)(\sin \pi y) \tag{11}$$

The analytical solution for this problem is given by

$$T(x, y) = 1 + x + (\sin \pi x)(\sin \pi y) \tag{12}$$

The numerical solutions computed for various uniform node distributions are presented in Fig.3. As the node density increases the accuracy of the solution increases exhibiting convergence. The computed results are compared with the analytical solutions and the accuracy/convergence is measured by the L_2 norm of relative error.



5 x 5 9 x 9 33 x 33

Figure3. Numerical solution of poisson equation for various uniform node distributions

The accuracy of RBF based method should be less dependent on whether the node distribution is uniform or random. Each of the nodes in a uniform distribution is slightly disturbed in a random direction to get a random distribution of nodes as shown in Fig.4.

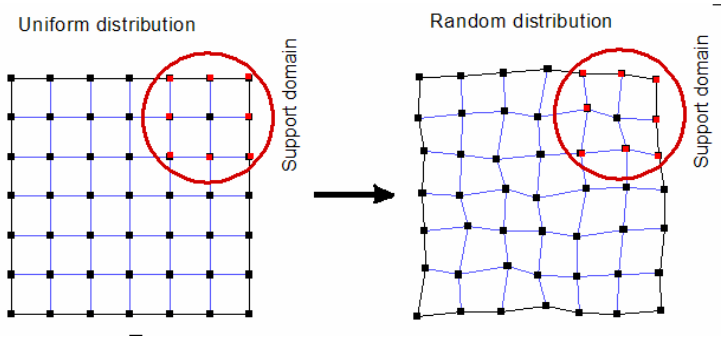


Figure 4. Schematic representation of disturbing nodes in a uniform distribution

T(x,y) obtained from both the distributions are close (Fig.5), hence the solver is less sensitive to node distribution.

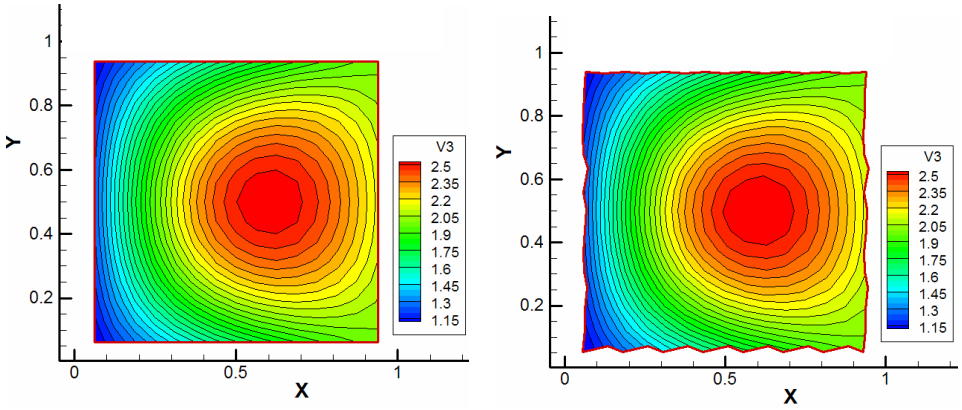


Figure 5. Temperature distribution for 17x17 nodes (a) Uniform (b) Random

IV. RBF-DQ based Euler Solver

Euler equations are a set of hyperbolic equations which govern the inviscid fluid dynamics. An exact analytical solution to these equations is intensive and cumbersome; hence the PDEs are discretized to obtain algebraic equations which are numerically solved to obtain an approximate solution. The classical methods which have been used to discretize the Euler equations are finite difference method (FDM), finite volume method (FVM) and finite element method (FEM). Kansa[13] was the first person to have applied RBF-based methods to solve problems in computational fluid dynamics. Later many researchers have shown great interest in using RBF-based methods to computationally solve a variety of engineering problems in fluid mechanics, heat transfer and structural dynamics. An RBF-based meshfree Euler solver to solve inviscid compressible flows is presented in this work.

A. Upwind method for the flux evaluation at the mid-point between two nodes

When solving hyperbolic PDEs such as Euler equations, it is important to employ a suitable discretization method, which not only can accurately approximate the smooth region of flow but also have the ability of capturing the possible discontinuities like shocks in the flow field. The basic framework of local RBF-DQ method is only suitable for solving incompressible flows or smooth compressible flows without any discontinuities. When shock wave occurs in the compressible flow region, either artificial dissipation or upwind schemes must be brought into the flow solver to capture the discontinuity. In the present scheme an upwind scheme is developed which accurately takes into account the direction of wave propagation associated with the hyperbolic equations. Such a method is required to suppress the oscillatory behavior of solution around the discontinuities. The mesh free upwind scheme is described for the two-dimensional (2D) compressible flows.

The 2D unsteady Euler equations can be written in the differential form in Cartesian coordinates as

$$\frac{\partial}{\partial t}\mathbf{Q}+\frac{\partial}{\partial x}\mathbf{F}_1(\mathbf{Q})+\frac{\partial}{\partial y}\mathbf{F}_2(\mathbf{Q})=0 \tag{13}$$

$$\mathbf{Q}=\begin{pmatrix}\rho\\ \rho\cdot u\\ \rho\cdot v\\ e\end{pmatrix},\ \mathbf{F}_1=\begin{pmatrix}\rho\\ \rho\cdot u\\ \rho\cdot v\\ e\end{pmatrix},\ \mathbf{F}_2=\begin{pmatrix}\rho\\ \rho\cdot u\\ \rho\cdot v\\ e\end{pmatrix} \text{ and the flux vector is } \mathbf{F}=\left[\mathbf{F}_1,\mathbf{F}_2\right]$$

where and $\mathbf{u}=(u,v)^T$ is the velocity vector, e stands for the total energy $e=\rho\left[\varepsilon+\left(u^2+v^2\right)/2\right]$ and ε is the specific internal energy. For a thermally perfect gas the static pressure p can be computed by the equation of state.

$$p=(\gamma-1)\left(e-\rho\frac{\mathbf{u}^2}{2}\right) \tag{14}$$

The divergence of flux vector in Eqn.13 is evaluated using the local RBF-DQ method discussed in the previous sections. However it is important to note that the points used for the discretization are not located at the supporting nodes. Instead they are located at the mid-points between the reference node and its supporting nodes (Fig. 6).

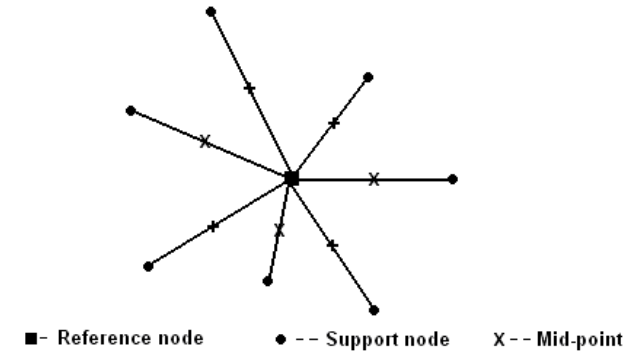


Figure 6. Scattered nodes around a given node and the corresponding mid points

After spatial discretization by RBF-DQ the Euler equations take the form as below:

$$\left.\frac{d\mathbf{Q}}{dt}\right|_i^n = -\sum_{k=0}^{N_I}\left[w_{i,k}^{(1x)}\mathbf{F}_1\left(\mathbf{Q}_{i,k}\right)+ w_{i,k}^{(1y)}\mathbf{F}_2\left(\mathbf{Q}_{i,k}\right)\right]^n, \tag{15}$$

where $\mathbf{Q}_{i,k}$ are the conservative variables at the mid points between the reference nodes *i* and its k^{th} supporting node. The terms $w_{i,k}^{(1x)}$ and $w_{i,k}^{(1y)}$ in Eqn.15 are the corresponding weighting coefficients for the first order derivatives in the x and y-direction respectively. By inspecting Eqn.15 we notice that at each mid-point a new flux can be defined, based on a unit vector $\overrightarrow{l_w} = \left(\alpha_{i,k}, \beta_{i,k}\right)^T$, which is associated with the weighting coefficients for first order derivatives in x and y-direction respectively. The new flux $\mathbf{G}_{i,k}$ can be written as

$$\mathbf{G}_{i,k} = \alpha_{i,k}\cdot\mathbf{F}_1\left(\mathbf{Q}_{i,k}\right)+ \beta_{i,k}\cdot\mathbf{F}_2\left(\mathbf{Q}_{i,k}\right), \tag{16}$$

where the elements of the unit vector $\alpha_{i,k}$ and $\beta_{i,k}$ are given by

$$\alpha_{i,k} = \frac{w_{i,k}^{(1x)}}{\sqrt{\left(w_{i,k}^{(1x)}\right)^2 + \left(w_{i,k}^{(1y)}\right)^2}} \quad \text{and} \quad \beta_{i,k} = \frac{w_{i,k}^{(1y)}}{\sqrt{\left(w_{i,k}^{(1x)}\right)^2 + \left(w_{i,k}^{(1y)}\right)^2}} \tag{17}$$

If we define a new variable $W_{i,k} = \sqrt{\left(w_{i,k}^{(1x)}\right)^2 + \left(w_{i,k}^{(1y)}\right)^2}$, then Eq. (4.4) takes the form

$$\left.\frac{d\mathbf{U}}{dt}\right|_i = -\sum_{k=0}^{N_I} W_{i,k}\mathbf{G}_{i,k}. \tag{18}$$

Eqn.18 can be interpreted in such a way that the variation of conservative variables at the reference point can be measured by a linear sum of the new fluxes at the reference point and the mid-points. How to evaluate the fluxes at the mid-points is a very critical issue in this scheme.

The RBF-DQ method described above to evaluate the flux derivatives cannot distinguish the influence from upstream or downstream. Hence to make sure that the scheme is upwind, appropriate evaluation of the new fluxes at the mid-point should take the directions of wave propagation of the hyperbolic system into consideration. Otherwise it can result in non-physical oscillations near steep gradients. Upwind schemes in the line of Godunov's method are quite popular where the numerical flux at the mid-point is obtained by exactly solving locally one dimensional (1D) Euler equations for discontinuous states i.e. a 1D Riemann problem. Godunov type scheme is very appropriate for the evaluation of new flux at the mid point by supposing that the functional values at the reference node *i* and its supporting node *k* form a local Riemann problem. However it is important to note that the evaluation of new fluxes still holds the meshfree property.

Euler equations are non-linear in behavior hence the solution of Riemann problem needs iteration and is very time-consuming. In order to reduce the computational cost the new fluxes at mid-point are evaluated using approximate Riemann solvers. In the present work Rusanov solver is used which assumes that all the waves

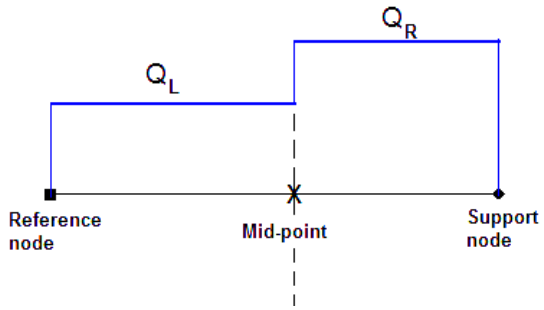


Figure 7. One dimensional Riemann problem

associated with the hyperbolic system travel with the maximum wave speed. Hence using Rusanov's scheme the new flux vector at the mid point $\mathbf{G}(Q_L, Q_R)$ can be evaluated as

$$\mathbf{G}(Q_L, Q_R) = \frac{1}{2}[\mathbf{G}(Q_L) + \mathbf{G}(Q_R)] - \frac{1}{2}\left|\hat{\mathbf{A}}\right|(\mathbf{Q}_R - \mathbf{Q}_L), \tag{19}$$

where $$\hat{\mathbf{A}} = \begin{bmatrix} |V_n| + c & 0 & 0 & 0 \\ 0 & |V_n| + c & 0 & 0 \\ 0 & 0 & |V_n| + c & 0 \\ 0 & 0 & 0 & |V_n| + c \end{bmatrix}$$

$|V_n|$ is absolute value of the normal velocity and c is the average local speed of sound.

B. Second order Rusanov solver with limiter

It is important to note that the flux solver described assumes that the flux between the mid-point and the related reference node remains a constant as shown in Fig.8, which is a first order spatial approximation. In order to obtain higher order accuracy we need to construct a higher order Rusanov solver by higher order spatial approximation of the solution. We try to use linear interpolation to obtain the fluxes on either side of the mid-point. The fluxes at the reference node and supporting node are indicated using a subscript and the extrapolated values on either side of the mid-point are represented with a superscript. Higher order approximation of the numerical flux at mid-point is obtained as shown below Eqn.20

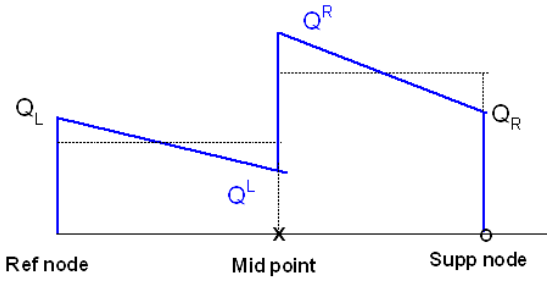


Figure 8. Linear interpolation for 2nd order accuracy

$$\mathbf{G}(Q_L, Q_R) = \frac{1}{2}[\mathbf{G}(\mathbf{Q}^L) + \mathbf{G}(\mathbf{Q}^R)] - \frac{1}{2}\left|\mathbf{A}^*(\mathbf{Q}^R, \mathbf{Q}^L)\right|(\mathbf{Q}^R - \mathbf{Q}^L), \tag{20}$$

where the conservative variables on the left and right of the mid-point are obtained by linear interpolation of the conservative variables as shown in Eqn.21.

$$\mathbf{Q}^L = \mathbf{Q}_L + (\Delta\mathbf{Q}_L) \qquad \text{where } \Delta\mathbf{Q}_L = \frac{\partial\mathbf{Q}_L}{\partial x}\cdot\left(\frac{x_R - x_L}{2}\right) + \frac{\partial\mathbf{Q}_L}{\partial y}\cdot\left(\frac{y_R - y_L}{2}\right)$$

$$\mathbf{Q}^R = \mathbf{Q}_R + (\Delta\mathbf{Q}_R) \qquad \text{where } \Delta\mathbf{Q}_R = \frac{\partial\mathbf{Q}_R}{\partial x}\cdot\left(\frac{x_R - x_L}{2}\right) + \frac{\partial\mathbf{Q}_R}{\partial y}\cdot\left(\frac{y_R - y_L}{2}\right) \tag{21}$$

In the above equation, matrix $\mathbf{A}^*$ also is a constant diagonal matrix but the maximum eigen values and averaged values are evaluated using the interpolated variables. To avoid spurious or non-physical numerical oscillations near the discontinuities, which general characteristic of higher order schemes we introduced a flux limiter. In the present scheme after the implementation of limiter the conservative variables on either side of the mid-point are evaluated as below (Eqn.22).

$$\mathbf{Q}^L = \mathbf{Q}_L + \alpha(\Delta\mathbf{Q}_L) \text{ and } \mathbf{Q}^R = \mathbf{Q}_R + \alpha(\Delta\mathbf{Q}_R) \tag{22}$$

The constant α is maximum possible α >0 such that

$$\min(\mathbf{Q}_k) < (\mathbf{Q}^L, \mathbf{Q}^R) < \max(\mathbf{Q}_k) \tag{23}$$

Hence numerically we will have 2nd order accuracy (α = 1) in the regions of smooth or continuous flow. In the regions of discontinuity maximum possible value of α (α < 1) is chosen such that Eqn.23 is satisfied with lower than 2nd order of accuracy but avoid undesired numerical oscillations.

Another interesting feature is observed by comparing the present upwind mesh-free scheme and the finite volume method with Rusanov flux approximation at the cell interface. It can be noted from Eqn.15 that the present meshfree scheme can be interpreted as a finite volume method with a non standard formulation. Firstly the coefficients $W_{i,0}$ associated with the reference node were found to be approximately zero, which implies negligible flux contribution from the reference node to itself which is physically true. Secondly, the unit vector $\overrightarrow{l_w} = (\alpha_{i,k}, \beta_{i,k})^T$ of the new flux $\mathbf{G}_{i,k}$ defined in Eqn.16 has the direction along the line joining the reference node $\mathbf{x}_I$ and the supporting node $\mathbf{x}_k$. Due to this fact Eqn.18 resembles closely to the flux evaluation $\oint_s \vec{F} \cdot \vec{n}$ in the finite volume method, and hence the present scheme is conservative. However the flux term at the reference node in the present RBF method generally does not vanish and makes a contribution to the flux gradients. This can be interpreted as a compensation for irregular cloud of supporting points, and differentiates the present scheme with the finite volume method.

V. Results and discussion

This section presents the results obtained using RBF-DQ based Euler solver for inviscid flows. All the test cases studied are compressible flows and are steady state. For all the configurations chosen the nodes within the domain were generated using commercial grid generation software "CFD-GEOM". For simplicity the nodes were generated using a structured mesh generator, though the nodes are stored and accessed in an unstructured format within the program. The flow variables are updated in time by first order forward Euler time stepping as shown in Eqn.24.

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n - \Delta t \left(\sum_{k=0}^{N_I} W_{i,k} \mathbf{G}_{i,k} \right) \tag{24}$$

A. Boundary conditions

1. Wall and symmetry boundary conditions

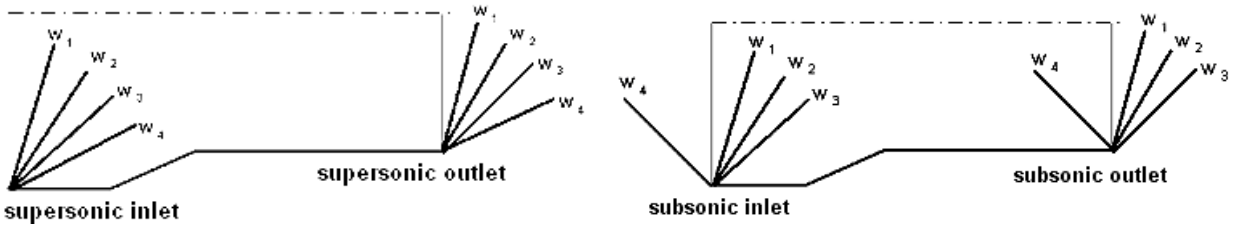
The flow variables are updated in time for interior nodes only since the boundary conditions are constant in time. The flow is inviscid hence the inviscid wall and symmetry boundary conditions are implemented identically. At both these boundaries we impose no penetration condition in other words the velocity normal to the wall is set to zero i.e. $u_n = 0$. Also the gradients normal to the wall and symmetry boundary condition are set to zero.

2. Inlet and outlet boundary conditions

At the inlet and outlet careful consideration needs to be taken in numerically implementing the boundary conditions. We use the theory of characteristics to set the inlet and outlet boundary conditions by solving a Riemann problem at both the inlet and exit. The Riemann invariants are

$$w_1 = \frac{p}{\rho^\gamma}, \quad w_2 = u_t, w_3 = u_n + \frac{2c}{(\gamma - 1)} \text{ and } w_4 = u_n - \frac{2c}{(\gamma - 1)} \tag{25}$$

where u_n and u_t are the normal and tangential velocities at the boundary, respectively.



(a) **(b)**

Figure 9. Characteristic waves at inlet and outlet (a) supersonic (b) subsonic

If the flow is supersonic, we can fix all the four Riemann invariants in Eqn.25 to freestream conditions since they all are moving with positive speed into the domain (Fig.9.a). By similar argument we can fix all the four variables at the domain outlet since they all are leaving the domain. Numerically this is equivalent to fixing all the four primitive variables ρ, p, u and v by copying their values from freestream conditions at the inlet. At the outlet all the characteristics are leaving the domain and hence the flow parameters ρ, p, u and v are extrapolated from interior.

For subsonic inlet all the characteristics w_1, w_2 and w_3 are traveling with positive speed and hence can be fixed at the inlet using freestream flow variables and w_4 traveling with negative speed is extrapolated from the interior. On the other hand at the outlet characteristics w_1, w_2 and w_3 moving out of the domain (Fig.9.b) are extrapolated from the interior and w_4 moving into the domain is fixed at the exit using freestream values.

B. Supersonic flow in a convergent nozzle with a ramp on the floor

This test case is ideal for testing the RBF-DQ based Euler solver. The channel consists of a 15^0 compression ramp followed by a 15^0 expansion corner along the lower and upper walls (Fig.10). At the inlet the flow is supersonic with an inlet Mach number of 2.0. The convergent nozzle is symmetric hence only the lower half of the channel is chosen as the computational domain with symmetry boundary condition along the centerline. The use of this symmetry nature brings down the computational cost. To study the effect of refinement we have used two nodal distributions 97x33 and 193x65 nodes. The Mach number flood contours in the channel for the both the nodal distributions are presented in Fig.11. The coarser distribution fails to capture the small region of subsonic flow but the denser nodal distribution successfully does as reported in Ref.21.

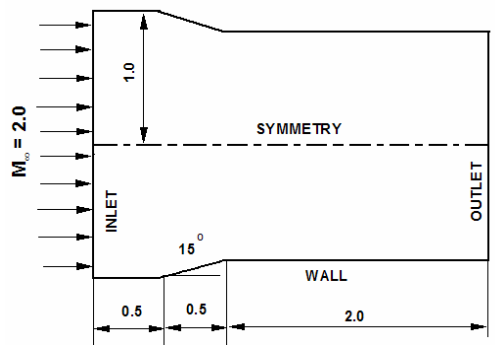
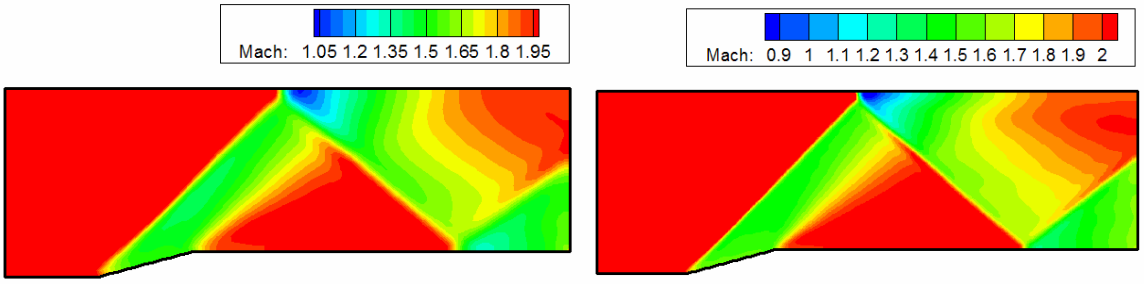


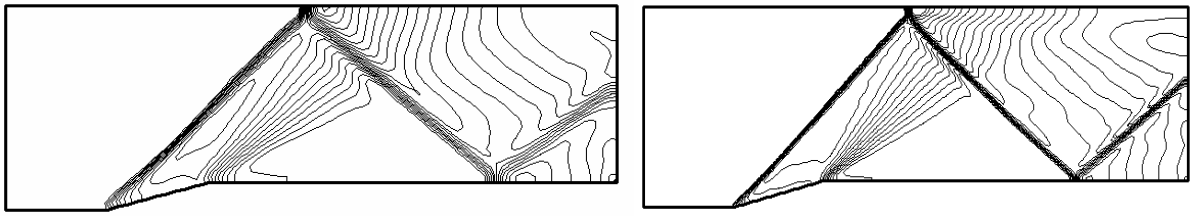
Figure 10. Supersonic flow in a convergent nozzle



(a) **(b)**

Figure 11. Mach number flood contours (a) 97x33 (b)193 x 65

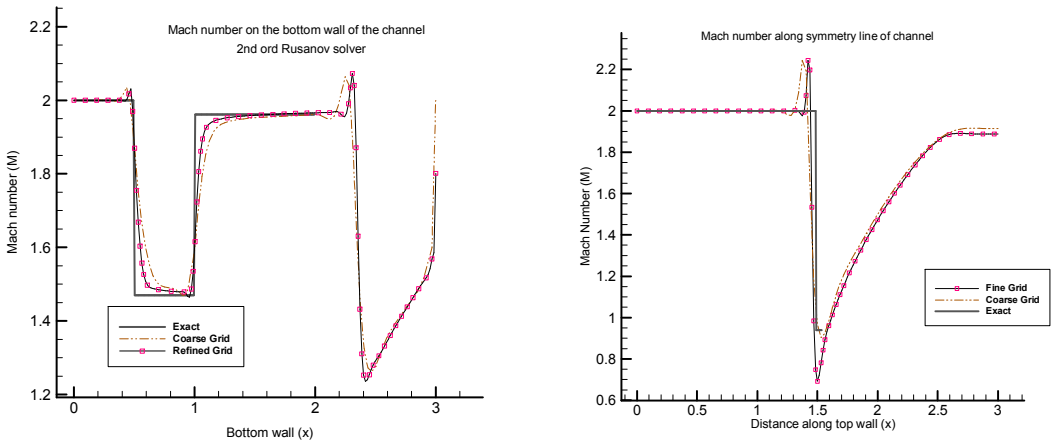
The incident and reflected shocks get sharper (Fig.12) as the nodal density increases showing that the resolution increases with mesh refinement. The expansion also becomes sharper with increase in nodal density (Fig.12.b).



(a) **(b)**

Figure 11. Mach number contours (a) 97 x 33 (b) 193 x 65

The Mach number computed on the channel floor and along the symmetry line is compared with the analytical solution (Fig.13). The Mach number after the incident shock Ma_2 and after expansion fan Ma_3 calculated analytically by compressible flow theory are compared with the numerical results in Table 1.



(a) **(b)**

Figure 13. Mach number (a) along channel floor (b) along symmetry line

Table 1. Results for the supersonic convergent nozzle with a ramp.

	Theory	Computed (coarse)	% Error	Computed (refined)	% Error
Ma_2	1.4457	1.47	1.68	1.443	0.19
Ma_3	1.9614	1.947	0.73	1.956	0.28
Shock angle(deg)	45.34388	44.51	1.84	44.93	0.91

C. Supersonic Flow over a circular bump

The next test case is supersonic flow (M=1.40) over a 5% circular bump in a channel. The computational domain and boundary conditions are shown in Fig.14. Both the inlet and outlet are supersonic as in the previous case. The domain is normalized using the length of the bump, L. The length of domain is 3L and height of domain equals L. The nodes are generated by a structured grid for simplicity. The supersonic flow causes a shock at the leading edge and at the trailing edge of the circular bump. This leading edge shock is reflected off the top wall boundary, crosses the trailing edge shock, is reflected again and it finally merges with the trailing edge shock. The solver accurately predicts the shock, expansion waves on the bump and the shock interaction occurring behind the bump. The decrease in strength of the reflected shock due to interaction with expansion waves can be observed in both the flood and line contour plots in Fig.15.

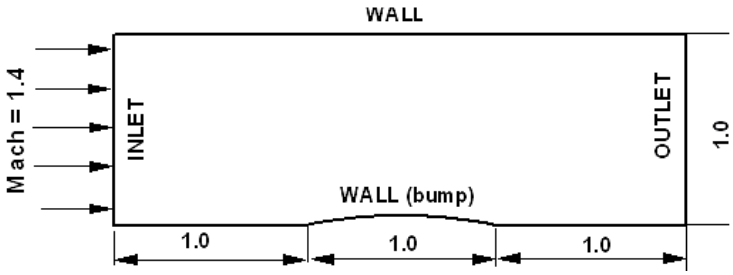


Figure 14. Supersonic flow over a 5 % thick circular bump

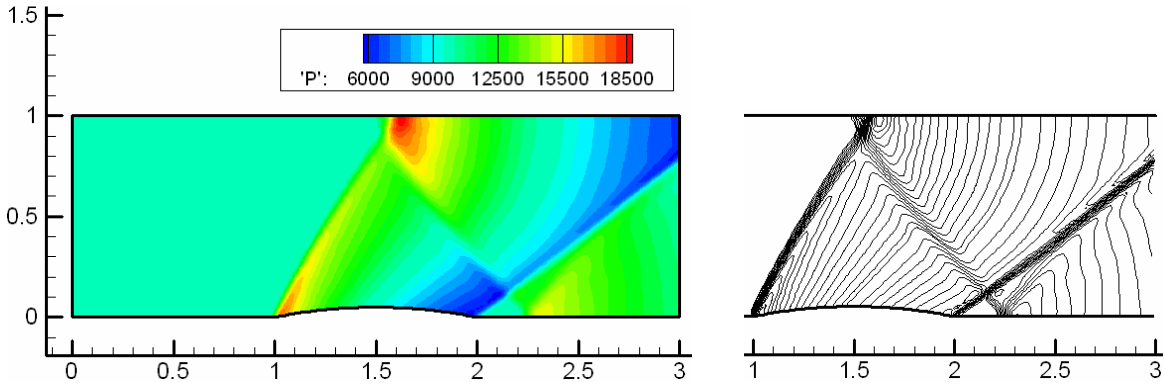
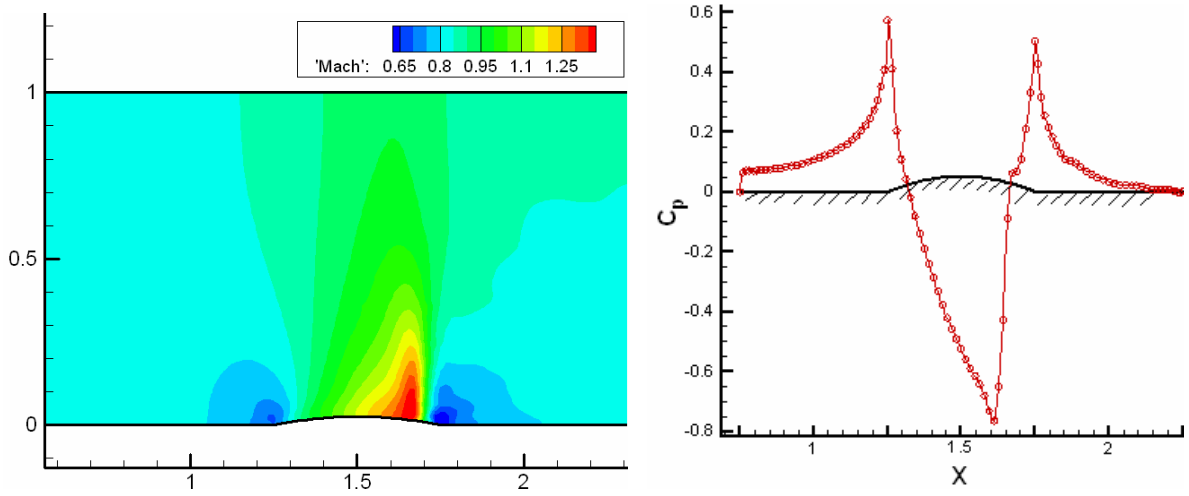


Figure 15. Pressure distribution for supersonic flow over a bump

D. Transonic flow over a circular bump

A slightly challenging test case for the present Euler solver was transonic flow M_{in} = 0.84 over a 5% circular bump in a channel. In transonic flow the magnitude of flow disturbance is an order higher than the characteristic dimensions of the body. The computational domain chosen for this case is larger than the supersonic case as disturbances travel both upstream and downstream as against supersonic flow. An important feature of this test case is that in part of the domain the flow is subsonic and in some parts it becomes supersonic. The flow expands over the bump and becomes supersonic leading to a transonic shock on the bump (Fig.16.a). The pressure coefficient on the floor of the channel shows the transonic shock on the bump as expected in theory (Fig.16.b).



(a) **(b)**

Figure 16. (a) Mach flood contours over the bump (b) Coefficient of pressure on the channel floor

VI. Conclusions

For complex geometries the process of grid generation can prove quite time consuming and cumbersome. Researchers have shown interest in meshfree methods which do not require any kind of mesh to be generated to solve the governing equations. In the present work an attempt has been made to develop a computational technique based on meshfree methods using RBFs. The present scheme can work on a random distribution of scattered nodes with no prespecified connectivity or relationship. Differential quadrature technique is coupled with RBFs to develop a meshfree Euler solver to solve inviscid compressible flows. The solver developed is validated by applying it to various 2D compressible flows. The RBF based meshfree solver models the flow phenomenon for compressible flows both qualitatively and quantitatively.

References

[1]T.Belytschko, Y.Kronganz, and D.Organ, "Meshless methods: An overview and Recent Developments",Comput. Methods Appl. Mech. Engrg.139. 3-47, 1996.

[2]G.R.Liu, "Meshfree methods: Moving beyond the Finite Element method", CRC Press, Florida 2003.

[3]L.B.Lucy, "A numerical approach to the testing of the fission hypothesis, The Astron. J. 8(12) (1977) 1013-1024.

[4]Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. Comput Mech 1992;10:307–18.

[5]Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. Int J Numer Meth Eng, 1994;37:229–56.

[6]Liu W, Jun S, Zhang Y. Reproducing kernel particle methods. Int J Numer Meth Fluids 1995;20:1081–106.

[7]Babuska I, Melenk J. The partition of unity method. Int J Numer Meth Eng 1997;40:727–58.

[8]Liszka T.J., Duarte C.A.M., and Tworzydlo W.W., "hp-Meshless cloud method", Computer methods in applied mechanics and engineering, 139, 263-288, 1996

[9]Onate E, Idelsohn S, Zienkiewicz OC, Taylor RL. "A stabilized finite point method for analysis of fluid mechanics problems". Computer methods in applied mechanics and engineering, 39, 3839–66, 1996.

[10]Atluri SN, Zhu T. "New meshless local Petrov–Galerkin (MLPG) approach in computational mechanics." Comput. Mech 1998;22(2):117–27.

[11]Buhmann M. D., "Radial basis Functions", Cambridge University Press, Cambridge, United kingdom, 2003.

[12]W. K. Liu Et al., "Multiresolution reproducing kernel particle method for computational fluid dynamics", International journal for numerical methods in fluids, vol. 24, 1391-1415 (1997)

[13]E.J. Kansa, "Multiquadrics-A scattered data approximation scheme with applications to computational fluid dynamics II. Solutions to parabolic, hyperbolic, and elliptic partial difference equations", Computers Math. Applic. 19 (8/9), 147-161, (1990).

[14]Franke C. and Schaback R ,"Solving partial differential equations by collocation using radial basis functions", Applied Mathematics and Computation 93,73-82, 1998.

[15]Wang J.G. and Liu G.R., "On the optimal shape parameter of radial basis functions used for 2-D meshless methods", Computer methods in applied mechanics and engineering, 191, 2611-2630, 2002.

[16]Maithili S, Kansa E.J, and Gupta S, "Application of the multiquadric method for numerical simulation of elliptic partial differential equations", Applied mathematics and computations 84, 275-302, 1997.

[17]Fasshauer E.G., "Solving differential equations with radial basis functions: multilevel methods and smoothing", Advances in computational mathematics, 11, 139-159, 1999.

[18]E. Larsson, B. Fornberg, "A numerical study of some radial basis function based solution methods for elliptic PDEs", Computers and Mathematics with Applications, 46,891-902, 2003.

[19]P.A. Ramachandran, K. Balakrishnan, "Radial basis functions as approximate particular solutions: review of recent progress", Engineering Analysis with Boundary Elements, 24 575–582, 2000.

[20]Shu C., Ding H., Chen H.Q., and Wang T.G., "An upwind local RBF-DQ method for simulation of inviscid compressible flows", Computer methods in applied mechanics and engineering, 194, 2001-2017, 2001.

[21]Ding H., Shu C., Yeo K.S. and Xu.D., "Development of least square based two dimensional finite difference schemes and their application to simulate natural convection in a cavity", Computers and Fluids, 33, 137-154, 2004.

[22]D. Sridar, N. Balakrishnan, "An upwind finite difference scheme for meshless solvers", Journal of Computational Physics, 189 (2003) 1–29.

[23]W.Chen and Tanaka M., " A Meshles, Integration free, and Boundary-Only RBF Technique", Computer and Mathematics with Applications, Vol 43, pp 379-391, 2002

[24]Fornberg et al., "Observations on the behaviour of radial Basis function Approximations Near Boudnaries", Computer and Mathematics with Applications, Vol 43, pp 473-490, 2002.