ARDB Centre of Excellence for Aerospace CFD
Department of Aerospace Engineering
Indian Institute of Science

Fluid Mechanics Report

A positive meshless method for hyperbolic equations

Praveen C
ARDB Centre of Excellence for Aerospace CFD
Department of Aerospace Engineering
Indian Institute of Science

Report 2004-FM-16

July 2004
Bangalore 560 012

A positive meshless method for hyperbolic equations

Praveen. C

CFD Center, Dept. of Aerospace Engg.
Indian Institute of Science, Bangalore 560012
email: `praveen@aero.iisc.ernet.in`

Abstract

A positive meshless method is constructed using least squares approximation for derivatives, upwinding and artificial dissipation. Godunov-type ideas and limiting procedures from finite-volume methods can be used in the present scheme. The method is applied to inviscid Euler equations in 2-D, and discontinuous flows are computed which show good oscillation-free solutions. Also, two new approaches to deriving a meshless finite difference formula are proposed which will avoid the need for using artificial dissipation terms.

Keywords: Meshless method, positive scheme, least squares, Euler equations.

1 Introduction

Meshless methods are receiving increasing attention in CFD because of their ability to work on just a distribution of points. These methods are meshless in the sense that the points need not have to be joined in any particular way to form a mesh. What is required is only the local connectivity information: for each point we need to find a set of nearby points which will form the stencil or connectivity for approximating the equations. The set of points used in a meshless methods is called a `finite-point mesh` While grid-based methods have demonstrated their usefulness in practical applications, the grid-generation part of the solution process is the most time consuming; for a complex 3-D configuration the time required to generate a grid can be in the order of several months and requires a lot of user intervention. On the other hand, the generation of a finite-point mesh is expected to be easier and faster since we do not have to worry about the quality of the mesh like orthogonality, smoothness, skewness of elements/volumes, etc., since no grid-lines, elements or volumes are required for meshless methods. The absence of these constraints can lead to greater automation in the point generation step.

There are two approaches to obtaining a point distribution for meshless methods. In the first approach we can use existing grid generation tools to generate a grid and take the points from this grid. Note that these grid generators need not be very sophisticated since we are not interested in obtaining a `nice` grid but only a reasonable point distribution. Another attractive alternative is to use the chimera approach where simple meshes are generated around geometrically simple parts of a complex configuration and the finite-point mesh is obtained by combining the points from all the individual meshes [20, 1].

In the second approach we can directly generate the point distribution without generating a mesh. According to Löhner [12], generating a finite-point mesh is an order of magnitude faster as compared to an unstructured mesh for a 3D configuration. Löhner has developed an advancing front point generation technique using ideas from unstructured grid generation techniques. Another powerful approach is to generate points using a quad-tree/oct-tree technique [22], as in Cartesian grid methods [3]. With a meshless method we do not have to worry about cut-cells near boundaries since we do not need any cells in the first place.

Whenever designing a new algorithm, it pays to ensure that the algorithm mimics all the properties of the governing equations like conservation, stability, smoothness of solution, TVD, rotational invariance, etc. For hyperbolic equations whose solutions can develop discontinuities, conservation is a desirable property. A conservative meshless method can be developed using the integral formulation but this requires the evaluation of certain integrals over the computational domain, which is not easy since we do not have a mesh. A background mesh has to be constructed for numerical quadrature which introduces more complexity into the algorithm. Hietel et al. [7, 10] have developed Finite Volume Particle Method (FVPM) for hyperbolic conservations laws which uses an integral formulation and is hence conservative. The integral formulation requires the evaluation of certain coefficients using approximate numerical quadrature and some correction of these coefficients is necessary to achieve consistency [8]. In the present work we do not use an integral formulation but we solve the equations in the conservation form. A growing body of numerical evidence from meshless solution of conservation laws [6, 15, 13, 17, 21] indicates that strict conservation property in the sense of finite volume schemes (telescopic collapse of interfacial fluxes) is not required to resolve discontinuous flows. It is however interesting to note that all these meshfree schemes solve the governing equations in their conservation form. Recently in the context of LSKUM [6], Deshpande [4] has given some arguments to show that a consistent meshless scheme for a conservation law leads to correct steady jump conditions.

Another important property for discontinuous solutions is TVD which is however valid in 1-D only. For higher dimensions Jameson [9] has proposed the local extremum diminishing (LED) property which is equivalent to TVD in 1-D. A scheme is said to be LED if it can be written as

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_j c_{ij}(u_j - u_i) \tag{1}$$

with each $c_{ij} \geq 0$ for all j. Under a CFL-condition this leads to a `positive` update scheme

$$u_i^{n+1} = \sum_j k_{ij} u_j^n \tag{2}$$

where each $k_{ij} \geq 0$ for all j. The positivity of the coefficients implies that the solution at the new time-level is bounded between the minimum and maximum of the solution at the previous time-level.

In this report we construct a positive meshless method based on the ideas in KMM [16, 17, 18]. The new method uses a combination of least squares approximation for derivatives, upwinding and artificial dissipation to ensure the positivity of the coefficients. The contribution of the artificial dissipation component is usually quite small. Higher order scheme is obtained by a reconstruction procedure similar to finite-volume methods but the positivity cannot be proved in this case. A Van Albada type limiter is used which leads to oscillation-free solutions for the test cases considered here. Results are presented for subsonic, transonic and supersonic flow in 2-D by solving the Euler equations which demonstrate good shock capturing ability.

2 Model conservation law

We will describe the meshless scheme for the prototypical hyperbolic scalar conservation law

$$\frac{\partial u}{\partial t} + \operatorname{div} \vec{Q}(u) = 0 \tag{3}$$

where the flux is given by $\vec{Q}(u) = \vec{a}u$ with $\vec{a} = (a_x, a_y)$ being a constant vector. The basic building block of our meshless method is the least squares approximation for the spatial derivatives, which allows us to approximate derivatives using arbitrarily scattered data.

3 Least squares approximation

Assume that the computational domain is discretized by finite-point mesh and the nodes are numbered $i = 1, 2, ..., n_p$. For each node N_i, we must find the set of nearby nodes C_i which will be used as a stencil for estimating the derivatives. Unlike in grid-based methods the stencil of a meshless method does not have any particular topology and the size of the stencil can vary from point to point. The only constraint on the stencils will be that the least squares problem be solvable. This means that

1. there must be more than d points in each stencil, d being the number of spatial dimensions, and,
2. all the points in the stencil must not lie on the same straight line

The least squares approximation is based on Taylor's formula. Assuming that $u(x, y)$ is sufficiently smooth we can write

$$u_j = u_i + (x_j - x_i) \left.\frac{\partial u}{\partial x}\right|_i + (y_j - y_i) \left.\frac{\partial u}{\partial y}\right|_i + O(h^2), \quad j \in C_i \tag{4}$$

The derivatives are determined by using a weighted least squares fit

$$\min \sum_{j \in C_i} w_{ij} \left(u_j - u_i - \Delta x_{ij} \left.\frac{\partial u}{\partial x}\right|_i - \Delta y_{ij} \left.\frac{\partial u}{\partial y}\right|_i \right)^2, \quad \text{wrt} \quad \left.\frac{\partial u}{\partial x}\right|_i, \left.\frac{\partial u}{\partial y}\right|_i$$

where $\Delta x_{ij} = x_j - x_i$, $\Delta y_{ij} = y_j - y_i$ and in general $\Delta(\cdot)_{ij} = (\cdot)_j - (\cdot)_i$. The weights $w_{ij} = w(\Delta r_{ij})$ are functions of the distance between N_i and N_j; the weight function w is a monotonically decreasing function and some possible choices are $w(r) = 1/r^p$, $p \geq 0$ or $w(r) = \exp(-\kappa r^2)$ with $\kappa \geq 0$. The weights give greater bias to neighbouring data which is closer to the node N_i than to those which are further away. The solution of this minimization problem leads to a matrix equation called the *normal equations*

$$\begin{bmatrix} \sum_j w_{ij} \Delta x_{ij}^2 & \sum_j w_{ij} \Delta x_{ij} \Delta y_{ij} \\ \sum_j w_{ij} \Delta x_{ij} \Delta y_{ij} & \sum_j w_{ij} \Delta y_{ij}^2 \end{bmatrix} \begin{bmatrix} \partial_x u\big|_i \\ \partial_y u\big|_i \end{bmatrix} = \begin{bmatrix} \sum_j w_{ij} \Delta x_{ij} (u_j - u_i) \\ \sum_j w_{ij} \Delta y_{ij} (u_j - u_i) \end{bmatrix} \tag{5}$$

These equations can be solved explicitly and the solution can be written in a generalized finite difference notation as follows:

$$\left.\frac{\partial u}{\partial x}\right|_i = \sum_{j \in C_i} \alpha_{ij} (u_j - u_i), \quad \left.\frac{\partial u}{\partial y}\right|_i = \sum_{j \in C_i} \beta_{ij} (u_j - u_i) \tag{6}$$

where the coefficients (α, β) are given by

$$\begin{aligned} \alpha_{ij} &= \frac{(\sum_k w_{ik} \Delta y_{ik}^2) w_{ij} \Delta x_{ij} - (\sum_k w_{ik} \Delta x_{ik} \Delta y_{ik}) w_{ij} \Delta y_{ij}}{D_i} \\ \beta_{ij} &= \frac{(\sum_k w_{ik} \Delta x_{ik}^2) w_{ij} \Delta y_{ij} - (\sum_k w_{ik} \Delta x_{ik} \Delta y_{ik}) w_{ij} \Delta x_{ij}}{D_i} \end{aligned} \tag{7}$$

and D_i is a determinant given by

$$D_i = \sum_{j \in C_i} w_{ij} \Delta x_{ij}^2 \cdot \sum_{j \in C_i} w_{ij} \Delta y_{ij}^2 - \left(\sum_{j \in C_i} w_{ij} \Delta x_{ij} \Delta y_{ij} \right)^2 \tag{8}$$

Note that by Cauchy-Schwarz inequality this determinant is zero iff $\Delta y_{ij} =$ const. Δx_{ij} for all $j \in C_i$, i.e., iff N_i, $\{N_j\}_{j \in C_i}$ all lie on the same straight line.

There is another useful way in which the solution can be written. Neglecting the $O(h^2)$ terms in (4)

$$\begin{bmatrix} \sqrt{w_{i_1}}(x_{i_1} - x_i) & \sqrt{w_{i_1}}(y_{i_1} - y_i) \\ . & . \\ . & . \\ . & . \\ . & . \\ . & . \\ \sqrt{w_{i_n}}(x_{i_n} - x_i) & \sqrt{w_{i_n}}(y_{i_n} - y_i) \end{bmatrix} \begin{bmatrix} \left.\frac{\partial u}{\partial x}\right|_i \\ \\ \left.\frac{\partial u}{\partial y}\right|_i \end{bmatrix} = \begin{bmatrix} \sqrt{w_{i_1}}(u_{i_1} - u_i) \\ . \\ . \\ . \\ . \\ . \\ \sqrt{w_{i_n}}(u_{i_n} - u_i) \end{bmatrix}$$

or

$$A_i \nabla u_i = \Delta u_i$$

where we have used a local numbering for the nodes in C_i. The solution can be obtained by first multiplying both sides by $A_i^\top$ and then taking the inverse

$$\nabla u_i = \underbrace{(A_i^\top A_i)^{-1} A_i^\top}_{M_i} \Delta u_i \tag{9}$$

In terms of our previous notation in equation (6), we see that

$$M_i = \begin{bmatrix} \alpha_{i,i_1} & . & . & . & . & \alpha_{i,i_n} \\ \beta_{i,i_1} & . & . & . & . & \beta_{i,i_n} \end{bmatrix} \tag{10}$$

The above matrix formula is useful for computer coding since it allows easy extension to higher orders by simply adding additional columns to the matrix A_i.

4 Designing the meshless method

The naive discretization for the divergence would be

$$\text{div } \vec{Q}(u)_i = a_x \sum_j \alpha_{ij}(u_j - u_i) + a_y \sum_j \beta_{ij}(u_j - u_i) \tag{11}$$

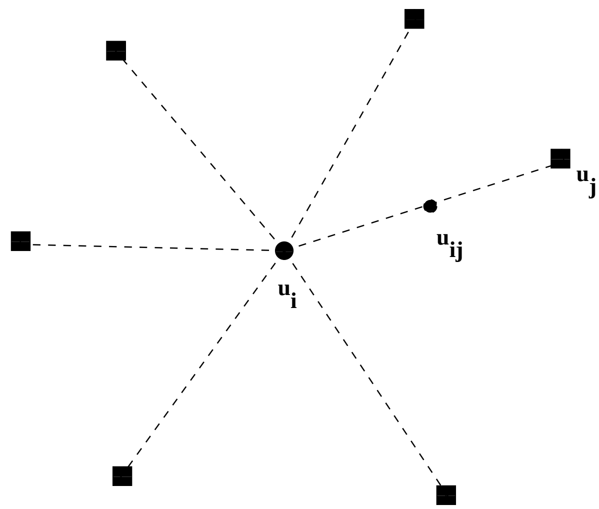


Figure 1: Definition of stencil and mid-point

but this will be unstable since it is a centered-type discretization and does not respect the hyperbolic nature of the pde which involves propagation of information in certain definite directions.

To obtain a stable discretization, we will assume that we know the value of u at the midpoint of $\overrightarrow{N_i N_j}$ as shown in figure (1) and which is as yet unspecified. We can then write the divergence as

$$\text{div } \vec{Q}(u)_i = 2a_x \sum_j \alpha_{ij}(u_{ij} - u_i) + 2a_y \sum_j \beta_{ij}(u_{ij} - u_i) \tag{12}$$

where the factor "2" naturally comes because now the stencil is scaled by 1/2, i.e., all the data are at mid-points. The mid-point values u_{ij} must be specified in an upwind-manner so as to obtain a stable scheme.

Now let θ_{ij} be the angle between $\overrightarrow{N_i N_j}$ and the positive x-axis, $\hat{n}_{ij} = (\cos\theta_{ij}, \sin\theta_{ij})$ be the unit vector along $\overrightarrow{N_i N_j}$ and $\hat{s}_{ij} = (-\sin\theta_{ij}, \cos\theta_{ij})$ be the unit vector normal to $\hat{n}_{ij}$ so that $(\hat{n}_{ij}, \hat{s}_{ij})$ form a right-handed coordinate system. The wave vector $\vec{a}$ can be written in terms of this coordinate system after a rotational transformation

$$\begin{aligned} a_x &= (\vec{a} \cdot \hat{n}_{ij}) \cos\theta_{ij} - (\vec{a} \cdot \hat{s}_{ij}) \sin\theta_{ij} \\ a_y &= (\vec{a} \cdot \hat{n}_{ij}) \sin\theta_{ij} + (\vec{a} \cdot \hat{s}_{ij}) \cos\theta_{ij} \end{aligned} \tag{13}$$

Using these formulae in (12) and after some rearrangement we obtain

$$\operatorname{div} \vec{Q}(u)_i = 2\sum_j \{\bar{\alpha}_{ij}(\vec{a}\cdot\hat{n}_{ij})(u_{ij}-u_i) + \bar{\beta}_{ij}(\vec{a}\cdot\hat{s}_{ij})(u_{ij}-u_i)\} \tag{14}$$

where we have defined

$$\begin{bmatrix} \bar{\alpha}_{ij} \\ \bar{\beta}_{ij} \end{bmatrix} = \begin{bmatrix} \cos\theta_{ij} & \sin\theta_{ij} \\ -\sin\theta_{ij} & \cos\theta_{ij} \end{bmatrix} \begin{bmatrix} \alpha_{ij} \\ \beta_{ij} \end{bmatrix} \tag{15}$$

Note that in (14) the first and second terms contain the mid-point flux approximations along $\hat{n}_{ij}$ and $\hat{s}_{ij}$ respectively. In finite volume methods we have to only approximate the flux along $\hat{n}_{ij}$ (normal to the edge or face of the finite volume) because the divergence theorem contains only the flux along the normal. Now $(\vec{a}\cdot\hat{n}_{ij})$ is the component of the wave-velocity along $\overrightarrow{N_iN_j}$ and we can use an upwind approximation for the first term in (14)

$$(\vec{a}\cdot\hat{n}_{ij})u_{ij} = \frac{\vec{a}\cdot\hat{n}_{ij} + |\vec{a}\cdot\hat{n}_{ij}|}{2}u_i + \frac{\vec{a}\cdot\hat{n}_{ij} - |\vec{a}\cdot\hat{n}_{ij}|}{2}u_j \tag{16}$$

so that

$$(\vec{a}\cdot\hat{n}_{ij})(u_{ij}-u_i) = \frac{\vec{a}\cdot\hat{n}_{ij} - |\vec{a}\cdot\hat{n}_{ij}|}{2}(u_j-u_i) =: (\vec{a}\cdot\hat{n}_{ij})^-(u_j-u_i)$$

The semi-discrete scheme can be written as

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -2\sum_{j\in C_i}\left\{\bar{\alpha}_{ij}(\vec{a}\cdot\hat{n}_{ij})^- + \bar{\beta}_{ij}(\vec{a}\cdot\hat{s}_{ij})\frac{(u_{ij}-u_i)}{(u_j-u_i)}\right\}(u_j-u_i) \tag{17}$$

We will show in the next section that $\bar{\alpha}_{ij} > 0$ so that the first term within the curly braces is negative. If we can ensure that the second term is also negative then we get a positive scheme. We have to specify the flux along $\hat{s}_{ij}$ in such a way that the second term is also negative. But we do not have left and right states along $\hat{s}_{ij}$ so that we cannot use an upwind formula for this flux. The simplest option is to take the arithmetic average which leads to

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -2\sum_{j\in C_i}\left\{\bar{\alpha}_{ij}(\vec{a}\cdot\hat{n}_{ij})^- + \frac{1}{2}\bar{\beta}_{ij}(\vec{a}\cdot\hat{s}_{ij})\right\}(u_j-u_i) \tag{18}$$

This does not give a positive scheme since the second term can be positive making the total term within the curly braces positive. The other option is to take the artificial dissipation approach and write

$$(\vec{a}\cdot\hat{s}_{ij})u_{ij} = (\vec{a}\cdot\hat{s}_{ij})\frac{u_i+u_j}{2} - \lambda_{ij}(u_j-u_i) \tag{19}$$

where λ_{ij} is a dissipation coefficient which has to be determined. If we take[1]

$$\lambda_{ij} = \mathrm{sign}(\bar{\beta}_{ij}) \frac{|\vec{a} \cdot \hat{s}_{ij}|}{2} \tag{20}$$

then the second term becomes

$$\begin{aligned}
\bar{\beta}_{ij}(\vec{a} \cdot \hat{s}_{ij}) \frac{(u_{ij} - u_i)}{(u_j - u_i)} &= \bar{\beta}_{ij} \left(\frac{\vec{a} \cdot \hat{s}_{ij}}{2} - \lambda_{ij} \right) \\
&= \frac{\bar{\beta}_{ij}(\vec{a} \cdot \hat{s}_{ij})}{2} - \bar{\beta}_{ij} \mathrm{sign}(\bar{\beta}_{ij}) \frac{|\vec{a} \cdot \hat{s}_{ij}|}{2} \\
&= \frac{\bar{\beta}_{ij}(\vec{a} \cdot \hat{s}_{ij}) - |\bar{\beta}_{ij}(\vec{a} \cdot \hat{s}_{ij})|}{2} \\
&\leq 0
\end{aligned}$$

With the choice (19), (20) the semi-discrete scheme can be written as

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_{j \in C_i} c_{ij}(u_j - u_i) \tag{21}$$

where the coefficients are given by

$$c_{ij} = -\bar{\alpha}_{ij}(\vec{a} \cdot \hat{n}_{ij})^- - \bar{\beta}_{ij} \left(\frac{\vec{a} \cdot \hat{s}_{ij}}{2} - \lambda_{ij} \right) \geq 0 \tag{22}$$

Equation (21) is in the LED form of Jameson and the positivity of the coefficients ensures that local extrema are not amplified, ie., local minimum will not decrease and local maximum will not increase. Using forward Euler time discretization, the update scheme is

$$u_i^{n+1} = [1 - \Delta t \sum_{j \in C_i} c_{ij}] u_i^n + \Delta t \sum_{j \in C_i} c_{ij} u_j^n \tag{23}$$

Provided the CFL-condition

$$\Delta t < \frac{1}{\sum\limits_{j \in C_i} c_{ij}} \tag{24}$$

is satisfied we get a positive update scheme which ensures stability in l_∞-norm, ie.,

$$\max_i |u_i^{n+1}| \leq \max_i |u_i^n|, \quad \forall n \tag{25}$$

Remark: On a Cartesian point distribution the above scheme reduces to finite volume method and the artificial dissipation component of the flux is absent because $\beta_{ij} \equiv 0$.

[1] $\mathrm{sign}(x) = -1$ if $x < 0$, is 0 if $x = 0$ and is 1 if $x > 0$

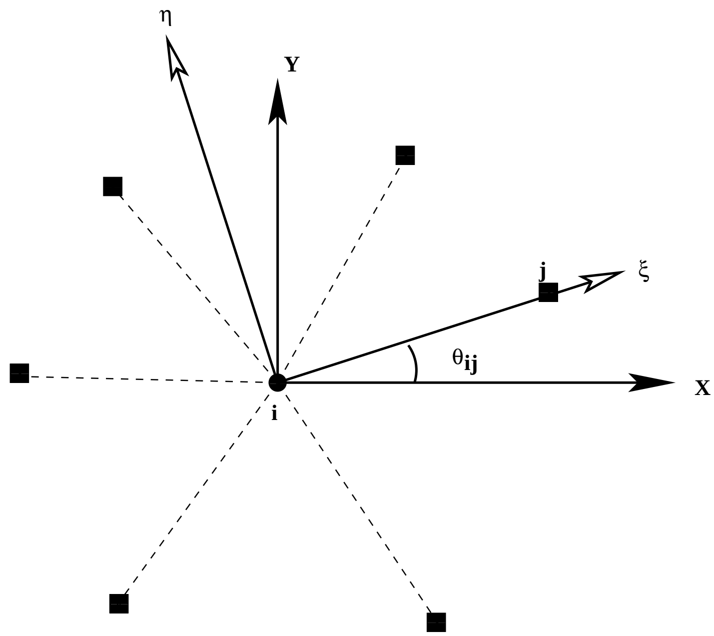


Figure 2: Proof of inequality $\bar{\alpha}_{ij} > 0$

5 Proof of inequality $\bar{\alpha}_{ij} > 0$

The form of equation (15) suggests that we should look at rotational transformations. Assume that the coordinate system is rotated by an angle ϕ. Then the matrix A_i transforms as

$$\bar{A}_i = A_i R(\phi)$$

where

$$R(\phi) = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \tag{26}$$

and the matrix M_i transforms as

$$\bar{M}_i = R(\phi) M_i$$

This simply means that each pair of numbers $(\alpha_{ij}, \beta_{ij})$ transforms like a vector (and hence is a vector). If we choose $\phi = \theta_{ij}$ then we see that $(\bar{\alpha}_{ij}, \bar{\beta}_{ij})$ is the vector $(\alpha_{ij}, \beta_{ij})$ expressed in the $(\hat{n}_{ij}, \hat{s}_{ij})$ coordinate frame. If (ξ, η) as the coordinates in this frame, see figure (2), then

$$\bar{\alpha}_{ij} = \frac{(\sum_k w_{ik} \Delta\eta_{ik}^2) w_{ij} \Delta\xi_{ij} - (\sum_k w_{ik} \Delta\xi_{ik} \Delta\eta_{ik}) w_{ij} \Delta\eta_{ij}}{D_i}$$

But since $\Delta\eta_{ij} = 0$ we have

$$\bar{\alpha}_{ij} = \frac{(\sum_k w_{ik}\Delta\eta_{ik}^2)w_{ij}\Delta\xi_{ij}}{D_i} > 0$$

There are other ways of obtaining the coefficients α_{ij}, β_{ij}, for example by using higher order least squares where the quadratic terms in the Taylor's series are included or but using radial basis functions. While we dont have a proof of the inequality it seems very plausible that it will still be satisfied.

6 Extension to non-linear equations and systems

Let the flux function $\vec{Q}(u) = (F(u), G(u))$ be nonlinear. Then the semi-discrete scheme can be written as

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\sum_{j\in C_i}\{\alpha_{ij}(F_{ij} - F_i) + \beta_{ij}(G_{ij} - G_i)\} \tag{27}$$

where the mid-point fluxes (F_{ij}, G_{ij}) are given by

$$\begin{bmatrix} F_{ij} \\ G_{ij} \end{bmatrix} = R^{-1}(\theta_{ij}) \begin{bmatrix} F(u_i, u_j, \hat{n}_{ij}) \\ G(u_i, u_j, \hat{s}_{ij}) \end{bmatrix} \tag{28}$$

ie. we are calculating the fluxes in the $(\hat{n}_{ij}, \hat{s}_{ij})$-frame and then transforming them back to the global (x, y)-frame. The flux along $\hat{n}_{ij}$ is obtained by an upwind formula

$$F(u_i, u_j, \hat{n}_{ij}) = \frac{\vec{Q}(u_i) + \vec{Q}(u_j)}{2} \cdot \hat{n}_{ij} - \frac{1}{2}|\vec{a}_{ij} \cdot \hat{n}_{ij}|(u_j - u_i) \tag{29}$$

and

$$\vec{a}_{ij} = \begin{cases} \frac{\vec{Q}(u_j) - \vec{Q}(u_i)}{u_j - u_i} & \text{if } u_j \neq u_i \\ \\ \frac{\mathrm{d}}{\mathrm{d}u}\vec{Q}(u_i) & \text{if } u_j = u_i \end{cases} \tag{30}$$

while the flux along $\hat{s}_{ij}$ is given by

$$G(u_i, u_j, \hat{s}_{ij}) = \frac{\vec{Q}(u_i) + \vec{Q}(u_j)}{2} \cdot \hat{s}_{ij} - \mathrm{sign}(\bar{\beta}_{ij})\frac{|\vec{a}_{ij} \cdot \hat{s}_{ij}|}{2}(u_j - u_i) \tag{31}$$

In the case of system of equations, we use an upwind formula for the flux along $\hat{n}_{ij}$ and use equation (31) for the flux along $\hat{s}_{ij}$ with $(\vec{a}_{ij} \cdot \hat{s}_{ij})$ being replaced by some average Jacobian of the flux along $\hat{s}_{ij}$.

7 On the coefficients α, β

In some special situations the connectivity is such that

$$\sum_{j \in C_i} \Delta x_{ij}^2 = \sum_{j \in C_i} \Delta y_{ij}^2, \quad \sum_{j \in C_i} \Delta x_{ij} \Delta y_{ij} = 0 \tag{32}$$

then

$$\alpha_{ij} = \frac{\Delta x_{ij}}{\sum_k \Delta x_{ik}^2}, \quad \beta_{ij} = \frac{\Delta y_{ij}}{\sum_k \Delta y_{ik}^2} \quad \Longrightarrow \quad \frac{\alpha_{ij}}{\beta_{ij}} = \frac{\Delta x_{ij}}{\Delta y_{ij}}$$

and the vector $(\alpha_{ij}, \beta_{ij})$ is parallel to $(\Delta x_{ij}, \Delta y_{ij})$ so that $\bar{\beta}_{ij} = 0$. We then do not need the $\hat{s}_{ij}$ component of the flux. This happens for example on a uniform Cartesian mesh and in a Delaunay triangulation where all N_j in C_i are at the same distance from N_i. Even when these conditions are not exactly satisfied, the vector $(\alpha_{ij}, \beta_{ij})$ is usually almost parallel to $(\Delta x_{ij}, \Delta y_{ij})$, at-least for sufficiently uniform point distributions. We show some examples in figures (3). In (a) the stencil is almost uniform and isotropic (angular distribution) and we see that the vectors (α, β) are practically aligned with the corresponding vectors $(\Delta x, \Delta y)$. In (b), the alignment is still pretty good while in (c) none of the vectors are aligned properly. In (d) we see a boundary stencil where we again see lack of alignment due to anisotropy of the stencil.

The above facts explain why even simple averaging for the $\hat{s}_{ij}$ component of the flux works quite well because usually $|\bar{\beta}_{ij}| << \bar{\alpha}_{ij}$ so that this component makes very little contribution to the divergence and positivity may not be violated. This also gives some justification for the methods of Morinishi [15], Löhner et al. [13] and Hietel et al. [10, 7], where an upwind flux formula along $(\alpha_{ij}, \beta_{ij})$ is used avoiding the need to specify any other flux. While this leads to a positive scheme it is physically not correct since $(\alpha_{ij}, \beta_{ij})$ may not be parallel to $\hat{n}_{ij}$ as we have seen in the examples above. In the appendices, we attempt to derive a formula for the flux derivatives in which only the $\hat{n}_{ij}$ component is present. This approach is still under investigation.

8 Higher order scheme

The scheme (27)-(31) has zeroth order consistency. First order consistency is however lost due to upwinding and is characteristic of most upwind schemes[2]. First order consistency can be restored by using linear reconstruction to the mid-points in the spirit of finite volume methods. Hence we define two reconstructed

[2] Upwind finite difference schemes are consistent in the sense of Taylor series but this usually comes at the expense of conservation. The LSKUM is a generalized upwind finite difference scheme which is consistent, i.e., the formal order of accuracy is equal to the order of the least squares approximation.

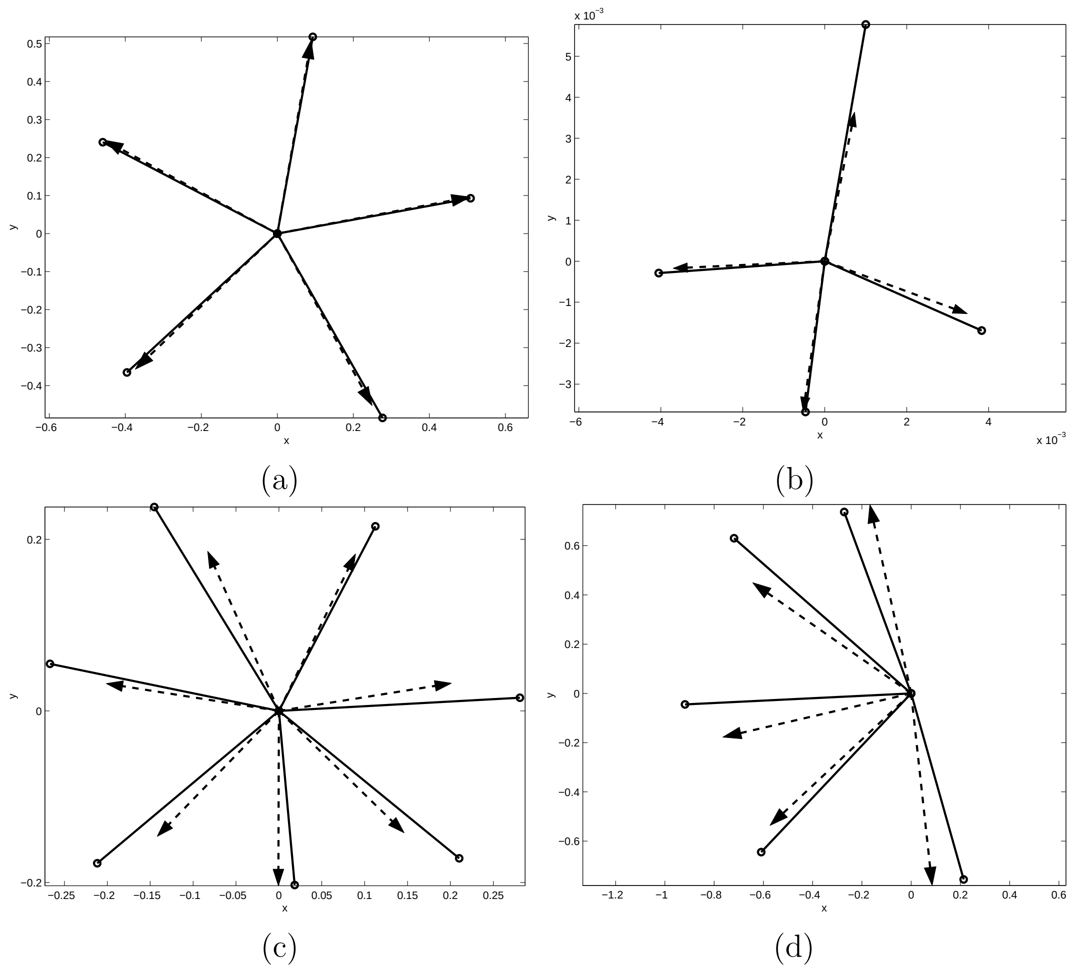


Figure 3: The broken lines with arrows are the vectors (α, β) while the circles denotes the nodes forming the stencil. The length of the arrows does not indicate their magnitude.

values at the mid-point

$$u_{ij}^+ = u_i + \frac{1}{2}\Delta\vec{r}_{ij}\cdot\nabla u_i, \quad u_{ij}^- = u_j - \frac{1}{2}\Delta\vec{r}_{ij}\cdot\nabla u_j \tag{33}$$

and calculate the mid-point fluxes as

$$\left[\begin{array}{c} F_{ij} \\ G_{ij} \end{array}\right] = R^{-1}(\theta_{ij}) \left[\begin{array}{c} F(u_{ij}^+, u_{ij}^-, \hat{n}_{ij}) \\ G(u_{ij}^+, u_{ij}^-, \hat{s}_{ij}) \end{array}\right] \tag{34}$$

When the solution is discontinuous the gradients in the reconstruction have to be limited to avoid spurious wiggles in the results. We use a MUSCL-type reconstruction with a Van Albada limiter which is defined as follows [14],

$$\begin{array}{rcl} u_{ij}^+ & = & u_i + \frac{s_i}{4}[(1-ks_i)\Delta_{ij}^- + (1+ks_i)(u_j-u_i)] \\ u_{ij}^- & = & u_j - \frac{s_j}{4}[(1-ks_j)\Delta_{ij}^+ + (1+ks_j)(u_j-u_i)] \end{array} \tag{35}$$

where

$$\Delta_{ij}^- = 2\Delta\vec{r}_{ij}\cdot\nabla u_i - (u_j - u_i), \qquad \Delta_{ij}^+ = 2\Delta\vec{r}_{ij}\cdot\nabla u_j - (u_j - u_i) \tag{36}$$

and

$$s_i = \max\left[0, \frac{2\Delta_{ij}^-(u_j-u_i)+\epsilon}{(\Delta_{ij}^-)^2 + (u_j-u_i)^2+\epsilon}\right]$$

$$s_j = \max\left[0, \frac{2\Delta_{ij}^+(u_j-u_i)+\epsilon}{(\Delta_{ij}^+)^2 + (u_j-u_i)^2+\epsilon}\right]$$

where ϵ is a small positive number which prevents null division. Note that the positivity of the scheme is not guaranteed with the higher order scheme. Numerical results however do not show any spurious wiggles indicating that the limiter is able to suppress the oscillations near discontinuities. This limiter may not be sufficient for flows with strong discontinuities in which case a stronger limiter like that of Venkatakrishnan [23] or even the very strong min-max limiter of Barth and Jesperson [2] can be used.

9 Data structure and coding

The simplicity of data structure is an attractive feature of meshless methods. An example of a data structure is given below:

```
struct node {
       float x, y, angle;
       int type, non, conn[non];
       float alpha[non], beta[non];
       } p[np];
```

and the different elements are explained below.

`x, y`	coordinates of the point
`angle`	orientation of tangent, only for boundary points
`type`	type of point - interior, boundary, etc.
`non`	number of neighbours
`conn[non]`	array containing indices of neighbours
`alpha[non], beta[non]`	coefficients for calculating derivatives

Note that the last two elements `alpha, beta`, in the data structure can be calculated from the remaining elements. It is more efficient to calculate them once and store them since they do not change during the solution process except when the point distribution or connectivity changes, for example when an adaptation is performed. The efficiency is gained because the calculation of divergence is faster due to smaller number of operations as the following code indicates.

```
/* Calculate flux divergence for scalar conservation law */
for i=1 to np
    xyflux(p[i].u, Fi, Gi)
    for j=1 to p[i].non
        neigh = p[i].conn[j]
        dx    = p[neigh].x - p[i].x
        dy    = p[neigh].y - p[i].y
        theta = atan(dy, dx)
        flux(p[i].u, p[neigh].u, theta, Fj, Gj)
        div[i] += p[i].alpha[j]*(Fj - Fi) + p[i].beta[j]*(Gj - Gi)
    end
end
```

We see that once the flux is calculated the divergence is obtained by a single summation and we do not solve the least squares problem, which involves a matrix inversion, for every time iteration. Also, due to the absence of stencil splitting, many `if` statements (eg. `if dx < 0`) are avoided.

10 Application to Euler equations

The first two test cases solve the flow around NACA-0012 airfoil. Two different point distributions G_1 and G_2 obtained from unstructured grids are used and the number of points in each case is given in table (1). A close-up view of the point distributions is given in figure (6). The neighbouring points are obtained from the edge connectivity of the unstructured grid and the average number of neighbours is five though a few points have as less as three neighbours.

	On airfoil	Total	C_l	C_d
G_1	120	4385	0.3298	0.00054
G_2	200	8913	0.3326	0.00005
GAMM	-	-	0.329-0.336	0.003-0.07

Table 1: Point distributions and C_l, C_d for the NACA-0012 subsonic test case

The first test case is a subsonic flow over over NACA-0012 at $M_\infty = 0.63$ and $AOA = 2$ deg. The pressure and Mach number contours are shown in figures (7) while the pressure coefficient and convergence history are given in figures (8). The pressure coefficient shows that the solution is accurate even on the smaller point distribution. The lift and drag coefficients are shown in table (1) and we see an improvement in these values are the point distribution is increased; the drag coefficient on G_2 is only 0.00005 while the exact value is zero.

The second test case is the transonic flow over NACA-0012 airfoil at $M_\infty = 0.8$ and $AOA = 1.25$ deg which is computed using Roe flux. The pressure and the Mach contours are shown in figure (9), both of which indicate that there are no spurious wiggles in the solution. The C_p on the airfoil is shown in figure (10) which again shows that there are no wiggles in the solution. The weak compression on the bottom surface is not captured accurately with G_1, whereas it is better resolved with G_2 as seen in contour and C_p plot in figure (11). The C_p plots show that the shock is captured within two points.

The third test case is supersonic flow past a 2-D semi-cylinder at $M_\infty = 3$ which is computed using kinetic flux. The pressure and Mach contours are shown in figure (12) indicating the good resolution of the bow shock without any oscillations. The MUSCL-type limiter is used in this case also. When the s-component of the flux was averaged then the code blew-up which shows that the positivity of the scheme is really making a difference. The Mach number and total pressure variation along the stagnation streamline of the cylinder is shown in figure (13) which shows wiggle-free solutions except for a small jump in the total pressure. The total pressure ratio at the stagnation point is found to be 0.3264 while the exact value computed from normal shock relation is 0.3283 which has an error of only 0.58%.

11 Summary

A positive meshless method on arbitrary point distributions is constructed using the least squares approximation. A combination of upwind fluxes and artificial dissipation is used to achieve positivity. The key step is the introduction of the sign function in the artificial dissipation. The constraints on the point distribution are very minimal; the connectivity of each point must have more than two nodes and they must not lie on a straight line. Limiters are used for the higher order scheme and the results demonstrate wiggle-free solutions for transonic and supersonic test cases.

A An alternative to least squares

In this section we propose an alternative approach to obtain a formula for the derivative which is also meshless in nature. This approach gives the same formula as the least squares technique. The advantage of this new approach is that it is possible to specify additional constraints on the coefficients $(\alpha_{ij}, \beta_{ij})$.

A.1 Approximation in 1-D

We start by assuming a formula of the form

$$\left.\frac{\partial u}{\partial x}\right|_i = \sum_{j \in C_i} \alpha_{ij}(u_j - u_i) \tag{37}$$

and this is consistent to first order if

$$\sum_{j \in C_i} \alpha_{ij}(x_j - x_i) = 1 \tag{38}$$

We have only one equation for determining the coefficients α_{ij} which is clearly insufficient. To determine these coefficients we solve the following minimization problem:

$$\min \frac{1}{2} \sum_{j \in C_i} \frac{\alpha_{ij}^2}{w_{ij}}, \quad \text{wrt} \quad \{\alpha_{ij}\} \tag{39}$$

subject to the constraint (38). This minimization problem can be solved using Lagrange multipliers, i.e.,

$$\min \frac{1}{2} \sum_{j \in C_i} \frac{\alpha_{ij}^2}{w_{ij}} + \lambda \sum_{j \in C_i} \alpha_{ij}(x_j - x_i), \quad \text{wrt} \quad \{\alpha_{ij}\}, \lambda \tag{40}$$

Differentiating the above functional wrt α_{ij} for some fixed j and equating it to zero, we get

$$\frac{\alpha_{ij}}{w_{ij}} + \lambda(x_j - x_i) = 0$$

or

$$\alpha_{ij} = -\lambda w_{ij}(x_j - x_i)$$

Now multiply both sides by $(x_j - x_i)$ and sum up over all $j \in C_i$, we get, after using the constraint (38)

$$\lambda = -\frac{1}{\sum_{j \in C_i} w_{ij}(x_j - x_i)^2} \tag{41}$$

and

$$\alpha_{ij} = \frac{w_{ij}(x_j - x_i)}{\sum_{k \in C_i} w_{ik}(x_k - x_i)^2}$$

Substituting this in equation (37) we see that we get the standard least squares formula as derived using Taylor's series. Thus the two formulations are equivalent. The advantage of the new formulation is that we can impose additional conditions on the coefficients.

Remark: The present approach can be considered as a discrete analogue of the Backus-Gilbert approach [11] to moving least squares. In this approach we try to approximate an unknown function u, given its value at some discrete locations x_i, $i = 1, ..., N$. The approximation is given as follows:

$$u(x) = \sum_{i=1}^{N} a_i(x) u(x_i)$$

The coefficients a_i are determined so that the approximation will reproduce some set of basis functions p_j, $j = 1, ..., M$, i.e.,

$$\sum_{i=1}^{M} a_i(x) p_j(x_i) = p_j(x), \quad j = 1, ..., M) \tag{42}$$

The a_i are determined by solving the following minimization problem

$$\min \frac{1}{2} \sum_{i=1}^{N} \eta(|x - x_i|) a_i^2(x), \qquad \text{wrt} \quad \{a_i(x)\} \tag{43}$$

subject to the constraints (42). In the above minimization, $\eta : [0, \infty) \to [0, \infty)$ is an increasing function, which is consistent with our notation, since w is a decreasing function. We will refer to this new least squares approach as the Backus-Gilbert (BG) technique.

A.2 Extension to 2-D

The BG technique can be extended to 2D least squares in a straight-forward manner. We assume that the derivatives are given by

$$\left.\frac{\partial u}{\partial x}\right|_i = \sum_{j \in C_i} \alpha_{ij}(u_j - u_i), \quad \left.\frac{\partial u}{\partial y}\right|_i = \sum_{j \in C_i} \beta_{ij}(u_j - u_i) \tag{44}$$

and the coefficients α_{ij}, β_{ij} must satisfy the following consistency conditions

$$\begin{array}{ll} \sum_{j\in C_i} \alpha_{ij}(x_j - x_i) = 1, & \sum_{j\in C_i} \beta_{ij}(y_j - y_i) = 1 \\ \sum_{j\in C_i} \alpha_{ij}(y_j - y_i) = 0, & \sum_{j\in C_i} \beta_{ij}(x_j - x_i) = 0 \end{array} \tag{45}$$

As in 1D we solve a minimization problem to determine the coefficients:

$$\min \frac{1}{2} \sum_{j\in C_i} \frac{\alpha_{ij}^2 + \beta_{ij}^2}{w_{ij}} \quad \text{wrt} \quad \{\alpha_{ij}\}, \{\beta_{ij}\} \tag{46}$$

subject to the constraints (45). It is easy to solve this problem explicitly and we again see that the coefficients are exactly what we would obtain from the standard 2-D least squares formulation. The minimizations problems for determining α and β can actually be decoupled in this case, but in the next section we impose additional constraints which lead to a coupling of the two problems.

B A derivative formula which leads to a positive scheme

Suppose we are able to approximate the gradient by a formula of the type

$$\nabla u_i = \sum_{j\in C_i} c_{ij}(u_j - u_i)\hat{n}_{ij} \tag{47}$$

where each of the coefficients $c_{ij} > 0$, then the approximation to the flux divergence can be written as

$$\vec{a} \cdot \nabla u_i = 2 \sum_{j\in C_i} c_{ij}(\vec{a} \cdot \hat{n}_{ij})(u_{ij} - u_i)$$

The mid-point flux $(\vec{a} \cdot \hat{n}_{ij})u_{ij}$ can be approximated by the upwind formula (16) which leads to a positive scheme given by

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -2 \sum_{j\in C_i} c_{ij}(\vec{a} \cdot \hat{n}_{ij})^-(u_j - u_i)$$

We see that there is no $\hat{s}_{ij}$-component of the flux in this approximation and we get an upwind scheme which avoids the need to use artificial dissipation.

If the coefficients $(\alpha_{ij}, \beta_{ij})$ in equations (44) are parallel to $(\Delta x_{ij}, \Delta y_{ij})$ then we obtain equation (47). The BG technique for determining the coefficients in equation (44) allows us to introduce extra constraints on the coefficients so as to satisfy the parallelism. Hence, using the BG approach we pose the following minimization problem:

$$\min \frac{1}{2} \sum_{j\in C_i} \frac{1}{w_{ij}}(\alpha_{ij}^2 + \beta_{ij}^2), \quad \text{wrt} \quad \{\alpha_{ij}\}, \{\beta_{ij}\}$$

subject to the constraints

$$\sum_{j\in C_i} \alpha_{ij}\Delta x_{ij} = 1 \tag{48}$$

$$\sum_{j\in C_i} \alpha_{ij}\Delta y_{ij} = 0 \tag{49}$$

$$\sum_{j\in C_i} \beta_{ij}\Delta y_{ij} = 1 \tag{50}$$

$$\alpha_{ij}\Delta y_{ij} - \beta_{ij}\Delta x_{ij} = 0, \quad \text{for all} \quad j \in C_i \tag{51}$$

The last constraint enforces the parallelism between $(\alpha_{ij}, \beta_{ij})$ and $(\Delta x_{ij}, \Delta y_{ij})$. Note that the consistency condition $\sum \beta_{ij}\Delta x_{ij} = 0$ is not included since this is already satisfied by the constraints. If C_i contains N points then there are $N + 3$ equations (constraints) and $2N$ unknowns. If $N = 3$ then the constraints will *probably* determine the solution without the need to solve a minimization problem. If $N > 3$, we have to solve the minimization problem subject to all the constraints. We take the standard approach of Lagrange multipliers and write the un-constrained problem as follows

$$\begin{aligned} \min \qquad & \frac{1}{2}\sum_{j=1}^{N} \frac{1}{w_j}(\alpha_j^2 + \beta_j^2) + \lambda_1 \sum_{j=1}^{N} \alpha_j \Delta x_j + \lambda_2 \sum_{j=1}^{N} \alpha_j \Delta y_j + \lambda_3 \sum_{j=1}^{N} \beta_j \Delta y_j \\ & + \sum_{j=1}^{N} \lambda_{j+3}(\alpha_j \Delta y_j - \beta_j \Delta x_j), \quad \text{wrt} \quad \{\alpha_j\}, \{\beta_j\}, \{\lambda_j\} \end{aligned} \tag{52}$$

where for simplicity we have suppressed the i-index and used a local numbering of the nodes in C_i. Differentiating the above functional wrt α_j and β_j for some j and equating to zero we get

$$\alpha_j = -\lambda_1 w_j \Delta x_j - \lambda_2 w_j \Delta y_j - \lambda_{j+3} w_j \Delta y_j \tag{53}$$

$$\beta_j = -\lambda_3 w_j \Delta y_j + \lambda_{j+3} w_j \Delta x_j \tag{54}$$

Using the parallel condition (51) we get

$$\lambda_{j+3} = -\frac{(\lambda_1 - \lambda_3)\Delta x_j \Delta y_j + \lambda_2 \Delta y_j^2}{\Delta x_j^2 + \Delta y_j^2} \tag{55}$$

Next using (53) in (48) and (49) we obtain

$$\begin{bmatrix} \sum w_j \Delta x_j^2 & \sum w_j \Delta x_j \Delta y_j \\ \sum w_j \Delta x_j \Delta y_j & \sum w_j \Delta y_j^2 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = - \begin{bmatrix} 1 + \sum \lambda_{j+3} w_j \Delta x_j \Delta y_j \\ 0 \end{bmatrix} \tag{56}$$

The above equation can be solved provided the determinant of the matrix on the left is non-zero, ie.,

$$\sum w_j \Delta x_j^2 \sum w_j \Delta y_j^2 - (\sum w_j \Delta x_j \Delta y_j)^2 \neq 0$$

By Cauchy-Schwarz inequality the determinant is ≥ 0. Equality holds if $\Delta y_j =$ const. Δx_j for all $j = 1, \ldots, N$, ie., if and only if all the points lie on a straight line.

Using equation (54) in (50) we get an equation for λ_3

$$\lambda_3 = \frac{\sum \lambda_{j+3} w_j \Delta x_j \Delta y_j - 1}{\sum w_j \Delta y_j^2} \tag{57}$$

Note that the equations for $\{\lambda_j\}$ are decoupled from $\{\alpha_j, \beta_j\}$. An iterative scheme can be set up as follows:

1. Set all $\lambda_j = 0$
2. Solve equation (56) for λ_1, λ_2 and equation (57) for λ_3
3. Solve equation (55) for $\lambda_4, \ldots, \lambda_{N+3}$
4. If the λ_j's have not converged then go to (2).
5. Compute α_j, β_j from equations (53), (54).

The above iterative technique for finding the coefficients has been applied to point distributions obtained from unstructured grids with partial success. Two issues have to be resolved:

1. What is the solvability criteria for the above minimization problem ? Is there some condition on the number of points and their spatial distribution ?
2. The condition (51) does not ensure that (α_j, β_j) and $(\Delta x_j, \Delta y_j)$ are parallel since it is satisfied even when they are anti-parallel.

C A second approach

Another approach to determining a formula of the type (47) is inspired by the remarks in section (7). If equations (32) hold then the coefficients determined by the standard least squares technique of section (3) will lead to an equation of the form (47). But in general equations (32) will not hold. This can be easily seen on a Cartesian point distribution with $\Delta x \neq \Delta y$. We can hope to enforce (32) by using appropriate weights, ie., we first determine the weights w_j such that

$$\sum_j w_j \Delta x_j \Delta y_j = 0 \qquad \sum_j w_j (\Delta x_j^2 - \Delta y_j^2) = 0 \tag{58}$$

Once the weights are determined they can be used in the least squares approximation which automatically leads to coefficients (α_j, β_j) satisfying the parallel condition. An interesting point to note is that condition number of the least squares problem is identitically one if the weights satisfy (58). Let $w_g(r)$ be a geometric weight function and let the weight be taken as $w_j = \omega_j w_g(\Delta r_j)$ where ω_j is another weight which has to be determined so that equations (58) are satisfied. Since there are less number of equations than unknowns, we solve a minimization problem to determine $\{\omega_j\}$. Since $\omega_j = 1$ in the least squares approximation we require that it should be as close to unity as possible and hence we pose the following minimization problem:

$$\min \frac{1}{2}\sum_j(\omega_j - 1)^2 + \lambda_1 \sum_j \omega_j p_j + \lambda_2 \sum_j \omega_j q_j \quad \text{wrt} \quad \{\omega_j\}, \lambda_1, \lambda_2 \tag{59}$$

where $p_j = w_g(\Delta r_j)\Delta x_j \Delta y_j$ and $q_j = w_g(\Delta r_j)(\Delta x_j^2 - \Delta y_j^2)$. Differentiating the above function wrt ω_j and equating it to zero we obtain

$$\omega_j = 1 - \lambda_1 p_j - \lambda_2 q_j \tag{60}$$

The system of equations in matrix form is

$$\begin{bmatrix} I & M \\ M^\top & 0 \end{bmatrix} \begin{bmatrix} \{\omega\} \\ \{\lambda\} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{61}$$

where I is an $N \times N$ identity matrix and

$$M = \begin{bmatrix} p_1 & p_2 & \dots & p_N \\ q_1 & q_2 & \dots & q_N \end{bmatrix}^\top$$

Equation (61) can be easily reduced using row operations so that $M^\top$ is eliminated. The last two equations will contain only λ_1 and λ_2 and are given by

$$\begin{bmatrix} \sum p_j^2 & \sum p_j q_j \\ \sum p_j q_j & \sum q_j^2 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} \sum p_j \\ \sum q_j \end{bmatrix} \tag{62}$$

The above equation can be solved provided the determinant is non-zero. In fact by Cauchy-Schwarz inequality the determinant is ≥ 0 and equality holds if and only if

$$p_j = C q_j$$

for some non-zero constant C. The above condition leads to

$$\tan 2\theta_j = \text{constant}$$

where θ_j is the angle between $(\Delta x_j, \Delta y_j)$ and the positive x-axis. This condition will be satisfied if all points in the connectivity lie on a straight line or they lie on a cross as in figure (4). The first case is not acceptable since we

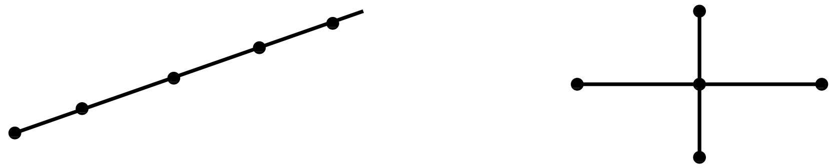


Figure 4: Degenerate cases

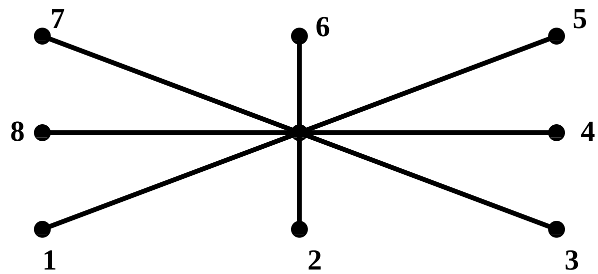


Figure 5: Cartesian stencil

cannot determine the derivatives using one-dimensional data; hence the connectivity has to be modified. In the second case if $\Delta x = \Delta y$ then the standard least squares already yields the desired solution. Otherwise, we have to modify the connectivity either by perturbing the coordinates or by adding extra nodes to the connectivity. While the solvability condition is known in this case, the conditions under which the weights are positive has to be investigated. Consider a point connectivity on Cartesian point distribution as shown in figure (5). The connectivity points are numbered from 1 to 8 and we compute the weights ω_j for different values of the ratio $\Delta x/\Delta y$ and using a geometric weight of $w_g(r) = 1/r^2$. The computed weights are given in table (2). On a uniform Cartesian point distribution, $\Delta x/\Delta y = 1$, we see that all the weights are equal to one. As the

$\Delta x/\Delta y$	1,3,5,7	4,8	2,6
1	1	1	1
2	0.73529	0.55882	1.4412
10	0.51000	0.50010	1.4999
100	0.50010	0.50000	1.5000
1000	0.50000	0.50000	1.5000

Table 2: Computed weights ω_j for Cartesian stencil of figure (5)

stencil gets elongated, $\Delta x/\Delta y > 1$, we see that the weights change and reach an asymptotic value. What is important to note is that none of the weights becomes very small or negative even when the point distribution is highly stretched. In the general case the conditions under which the weights are strictly positive are not known and this aspect needs further investigation. In numerical experiments loss of positivity was encountered when the connectivity was close to a cross-type distribution as discussed before.

References

[1] Anandhanarayanan K, *Development and Applications of A Gridfree Kinetic Upwind Solver to Multibody Configurations*, PhD thesis, Dept of Aerospace Engg., Indian Institute of Science, Bangalore.

[2] Barth TJ and Jesperson DC, "The Design and Application of Upwind Schemes on Unstructured Meshes", AIAA paper 89-0366, 1989.

[3] Berger MJ and LeVeque RJ, "Stable boundary conditions for Cartesian grid calculations", ICASE Report No. 90-37, May 1990.

[4] Deshpande SM, "Meshless method, accuracy, symmetry breaking, upwinding and LSKUM", FM Report 2003-FM-1, Dept. of Aerospace Engg., IISc, Bangalore, Jan 2003.

[5] Furst J and Sonar Th, "On meshless collocation approximations of conservation laws: preliminary investigations on positive schemes and dissipation models", ZAMM Z Angew. Math., **81**(6), pp. 403-415, 2001.

[6] Ghosh AK and Deshpande SM, "Least Squares Kinetic Upwind Method for Inviscid Compressible Flows", AIAA Paper 95-1735, 1995.

[7] Hietel D, Steiner K and Struckmeier J, "A finite-volume particle method for compressible flows", Math. Models Methods Appl. Sci, **10**, 1363-1382, 2000.

[8] Hietel D and Rainer Keck, "Consistency by coefficient-correction in the finite-volume-particle method", in *Meshfree methods for partial differential equations*, ed. Michael Griebel and Marc A Schweitzer, LNCSE, Springer, 2000

[9] Jameson A, "Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence in transonic and hypersonic flow", AIAA paper 93-3359, July 1993.

[10] Junk M, "Do finite volume methods need a mesh ?", in *Meshfree Methods for Partial Differential Equations*, ed. Michael Griebel and Marc A Schweitzer, LNCSE, Springer, 2000

[11] Levin, David (1998): "The approximation power of moving least-squares", *Math. Comp.*, vol. 67, no. 224, pp. 1517-1531.

[12] Löhner R, "An advancing front point generation technique", Commun. Numer. Meth. Engng, **14**, 1097-1108, 1998.

[13] Löhner R, Sacco C, Onate E and Idelsohn, "A finite point method for compressible flow", Int. J. Num. Meth. Engg., **53**, 2002.

[14] Löhner R, *Applied CFD Techniques: An introduction based on Finite Element Methods*, Wiley, 2001.

[15] Morinishi K, "An implicit gridless type solver for the Navier-Stokes equations", CFD J., **9**(1), 2000.

[16] Praveen C and Deshpande SM, "Rotationally Invariant Grid-less Upwind Method for Euler Equations", FM Report, 2001-FM-08, Dept. of Aerospace Engg., IISc, Bangalore, 2001. Available online at `http://eprints.iisc.ernet.in/archive/00000117/`

[17] Praveen C and Deshpande SM, "A new grid-free method for conservation laws", Second ICCFD, Sydney, July 2002. Available online at `http://eprints.iisc.ernet.in/archive/00000112/`

[18] Praveen C and Deshpande SM, "Kinetic meshless method", FM Report No. 2003-FM-10, Dept. of Aerospace Engg., IISc, Bangalore. Available online at `http://eprints.iisc.ernet.in/archive/00000157/`

[19] Praveen C and Deshpande SM, "New developments in kinetic meshless method", under preparation.

[20] Ramesh V, "Least squares grid-free kinetic upwind method", PhD Thesis, Dept. of Aerospace Engg., IISc, Bangalore, July 2001.

[21] Sridar D and Balakrishnan N, "An upwind finite difference scheme for meshless solvers", JCP, **189**, pp. 1-29, 2003.

[22] Varma MU, Raghuramarao SV and Deshpande SM, "Point generation using the quadtree data structure for meshless solvers", FM Report 2003-FM-08, Dept. of Aerospace Engg., Indian Institute of Science, Bangalore, September 2003.

[23] Venkatakrishnan V, "Convergence to steady state solutions of the Euler equations on unstructured grids with limiters", JCP, 118:120-130, 1995.

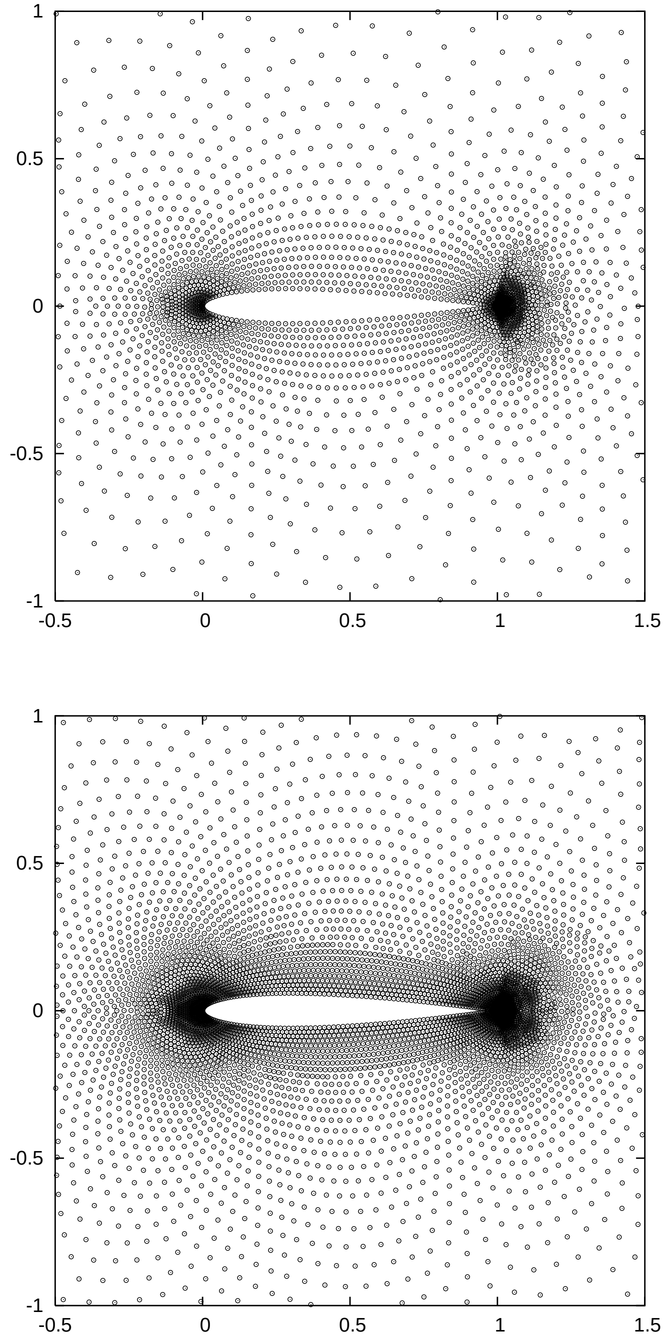


Figure 6: Point distributions for NACA-0012 with 4733 (G_1) and 4385 (G_2) points

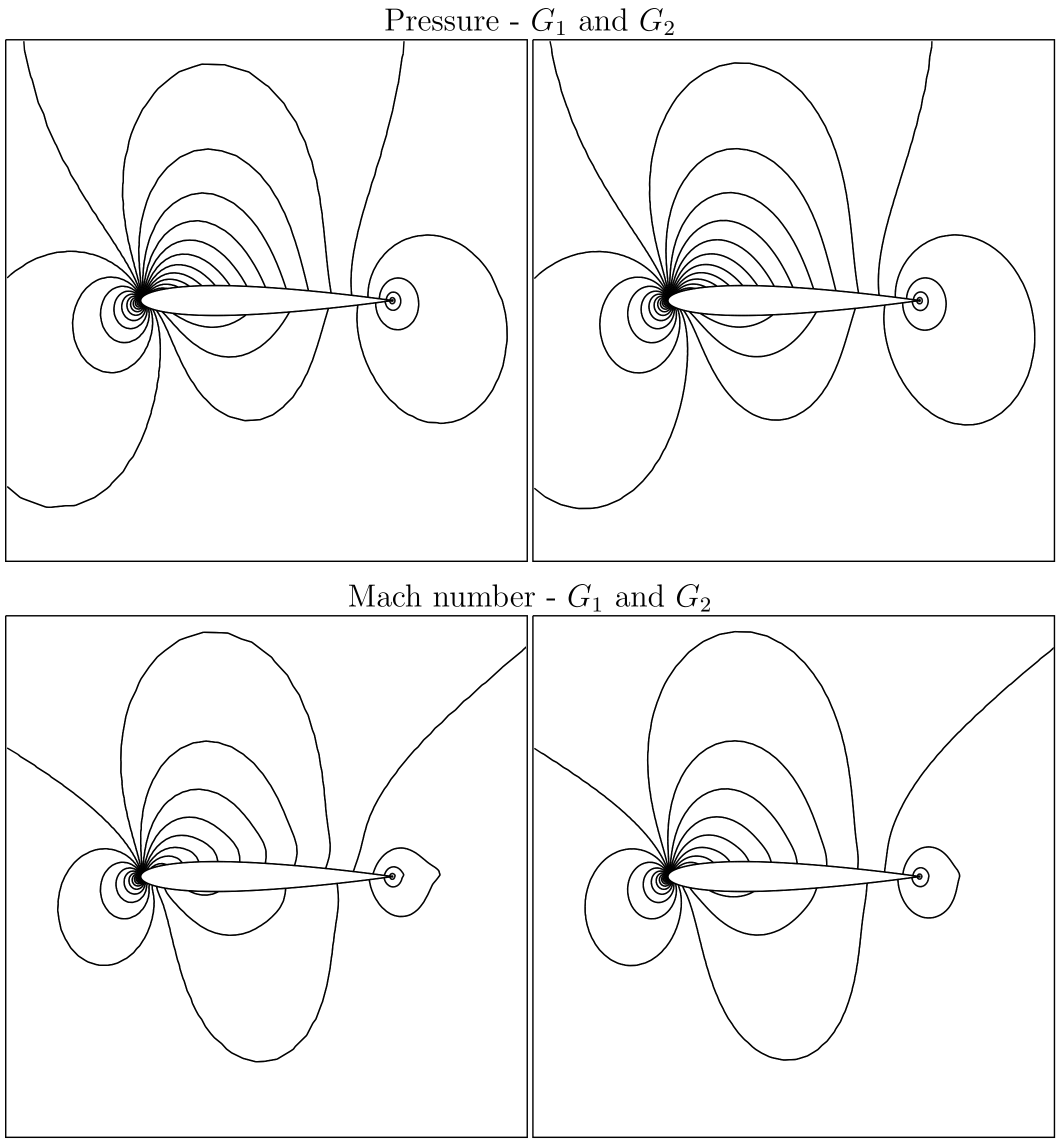
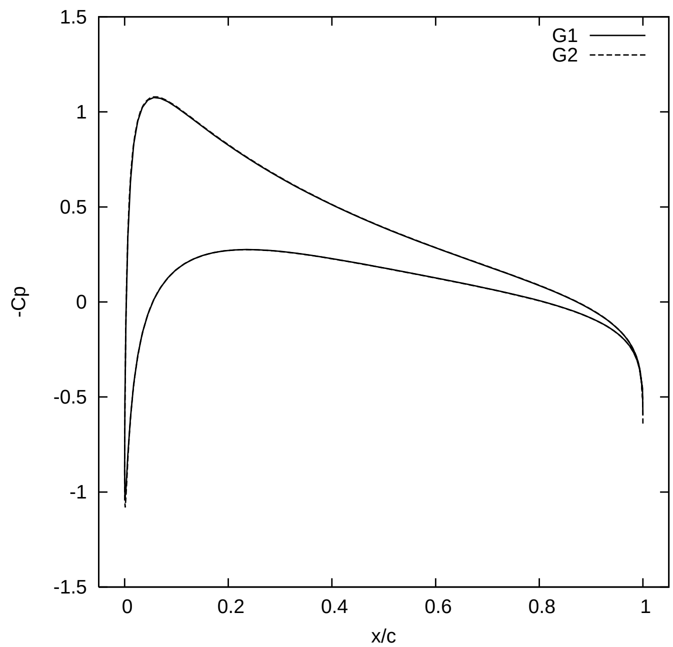


Figure 7: Subsonic flow over NACA-0012



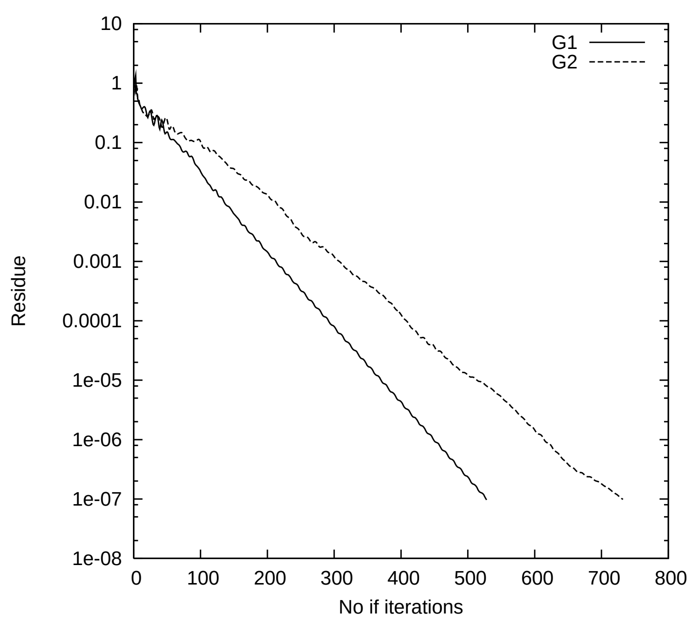


Figure 8: Pressure coefficient and convergence history for subsonic flow over NACA-0012

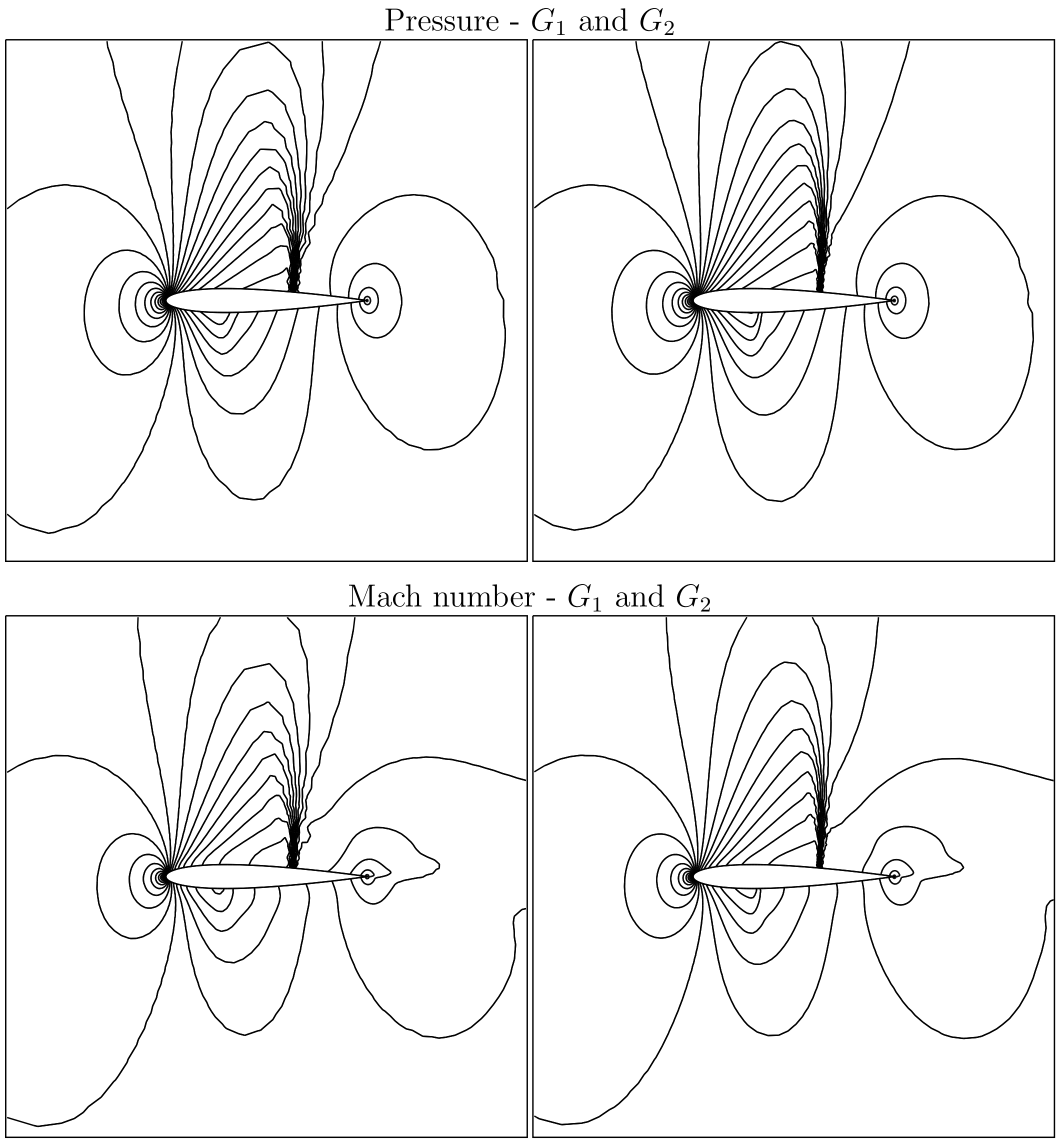


Figure 9: Transonic flow over NACA-0012

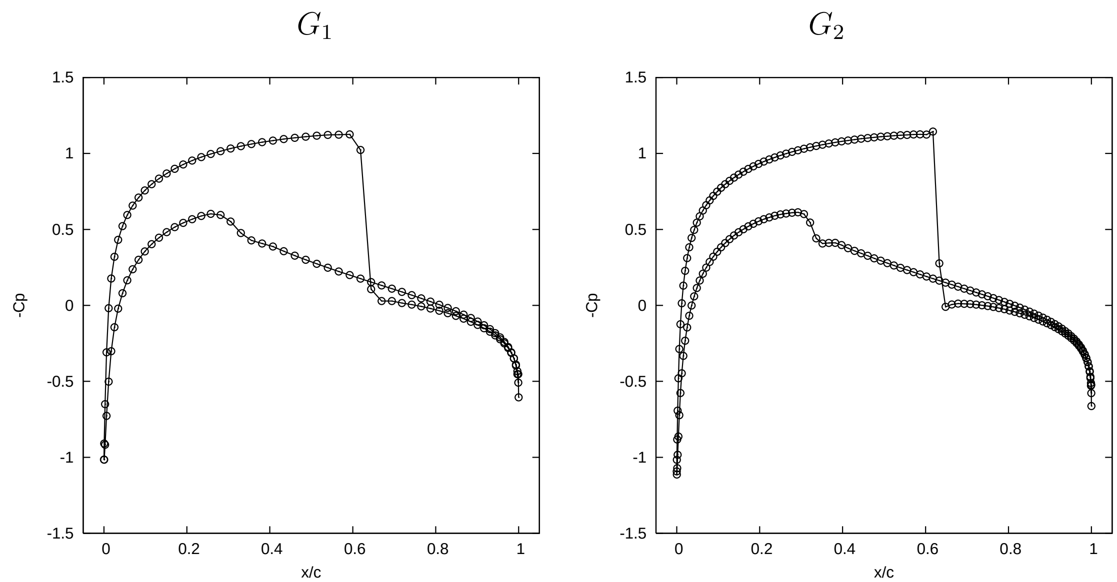
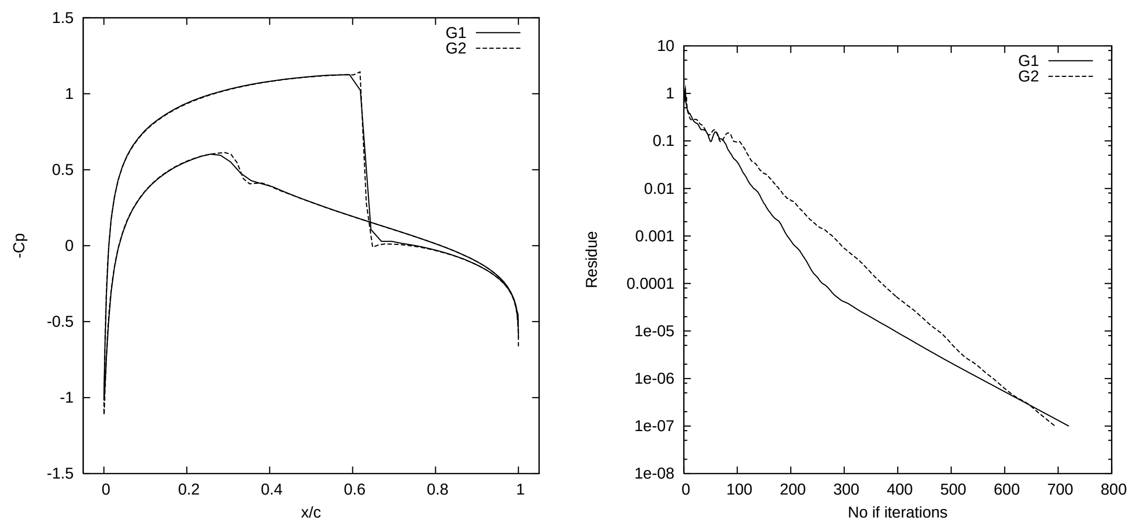


Figure 10: Pressure coefficient for transonic flow over NACA-0012



(a) (b)

Figure 11: Pressure coefficient and convergence history for transonic flow over NACA-0012

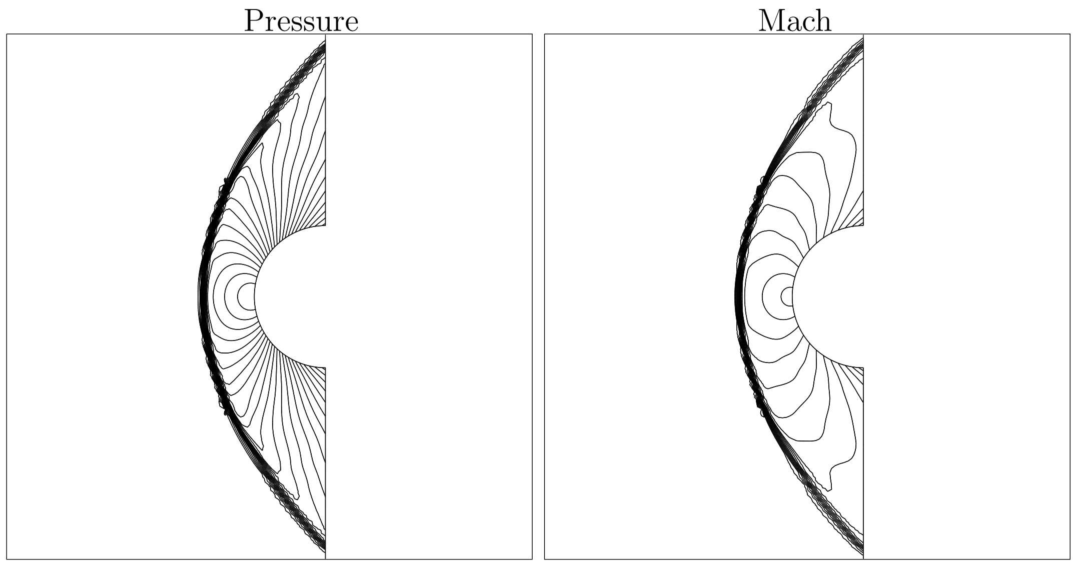


Figure 12: Mach 3 flow over a semicylinder

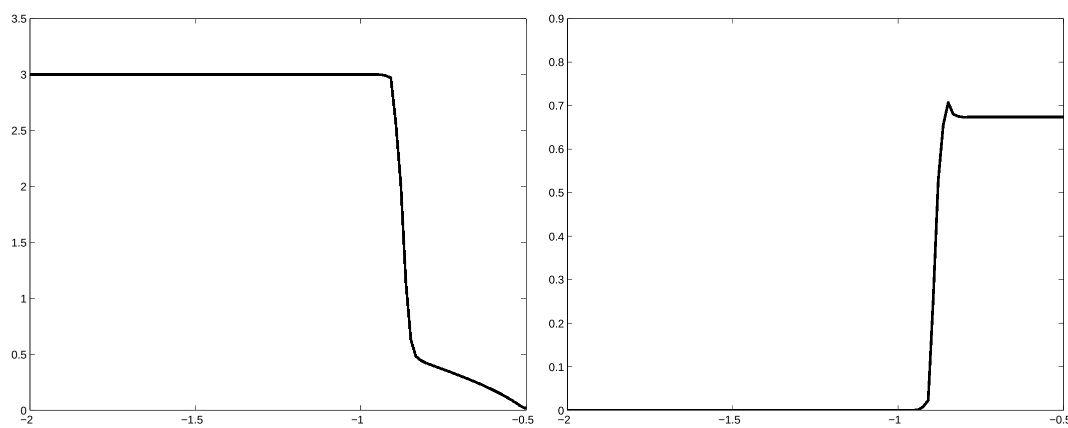


Figure 13: Mach number and $(p_{o1} - p_{o2})/p_{o1}$ along the stagnation streamline of semi-cylinder