Assessment of Implicit and Adaptive Mesh-free CFD Modelling

Tao Zhang*
University of Leicester, Leicester, England, United Kingdom, LE1 7RH

George N. Barakos†
University of Glasgow, Glasgow, Scotland, United Kingdom, G12 8QQ

This work presents detailed investigations of implicit and adaptive mesh-free CFD modelling approaches to alleviate laborious mesh generation in modern CFD process. A least-square based mesh-free discretisation scheme was derived for the compressible RANS equations, and the implicit dual-time stepping was adopted for improved stability and convergence. The spatial accuracy and convergence properties were verified using simulations of 2D and 3D benchmark cases. Impacts of irregular point distributions and various point collocation choices were then systematically investigated. The mesh-free scheme showed strong flexibility accommodating various point distributions and collocation configurations, but the modelling was sensitive to the regularity and reciprocity of the collocation, especially in critical flow regions e.g. boundary layers. The mesh-free flexibility was exploited for adaptative modelling, and various adaptation strategies was assessed, combining two different point refinement mechanisms and three different collocation search methods. Their performance was carefully evaluated using simulations of the isentropic vortex. The adaptative mesh-free modelling was then successfully applied to simulations of the transonic RAE 2822 aerofoil with automated point generation. A novel weighted pressure gradient metric prioritising high gradient region with large point sizes was introduced to drive the adaptation. The mesh-free adaptation effectively improved the shock resolution.

I. Introduction

Modern CFD (Computational Fluid Dynamics) workflow depends on mesh generation that affects modelling capabilities to address complex applications [1]. The meshing process is difficult due to geometric complexity, and time-consuming due to poor automation. The user dependent meshing is also a major source of uncertainty in modern CFD workflows. The meshing process is also iterative due to the lack of foreknowledge of the flow solution. Approaches such as Adaptive Mesh Refinement (AMR) were proposed to tackle this problem by actively evolving the mesh while the flow solution is in progress, but complex data structures and algorithms must be designed for efficient implementation. Apart from the meshing difficulty, the stiff connectivity of mesh points is incompatible with problems involving large domain motions or deformations, such as multi-body dynamics. Complementary techniques such as sliding or overlapping grids are needed to enable the modelling, although the modelling fidelity may be severely compromised. Meshing has hence been recognised as a major challenge in the CFD 2030 vision [1].

Mesh-free CFD methods [2–4] can be an effective solution to this mesh-associated bottleneck. Mesh-free methods solve the governing flow equations on a set of discrete points. Without the need for connectivity, the meshing process can be simplified as point generation and the mesh-free modelling is naturally compatible with large domain motions/deformations. Over the past two decades, several studies on mesh-free methods [4–6] have been published , yet relevant applications are still significantly under-developed at present. Many of the studies focused on verification of mesh-free discretisation schemes using simple explicit time marching, while complementary research on e.g. point cloud generation, collocation search and adaptivity has often been ignored. In many cases, the mesh-free solver would still require connectivity information extracted from meshes to enable stable and high-fidelity simulations. These have considerably undermined the flexibility of the mesh-free modelling and hindered its practical applications.

This paper presents a detailed investigation of mesh-free CFD modelling, with focuses on implicit time marching and impacts of point distribution and collocation configurations. A least-square based mesh-free scheme was first derived and its accuracy was verified using benchmark flow simulations. Random point offsets were then introduced to

*Lecturer, School of Engineering, MAIAA, tz77@leicester.ac.uk.
†Professor, James Watt School of Engineering, SMAIAA, george.barakos@glasgow.ac.uk.

examine irregular point clouds, and various collocation configurations were also tested. Various adaptative modelling strategies were explored combining two different point refinement mechanisms (Cartesian and mid-point) and three collocation search methods (distance-based, optimum selection, local Voronoi). Their performance was assessed through simulations of the isentropic vortex. Adaptive simulations were then applied to transonic simulations of the RAE 2822 aerofoil with a weighted pressure gradient metric. The adaptation effectively improved the shock resolution.

II. Numerical Methodologies

A. Generalised Finite Differences using Least-square Approximations

The principle of Generalised Finite Differences (GFD) [] is to fit local field data with shape functions upon discrete points, and to construct the differential operations using the shape function derivatives. Regardless of implementation details e.g. Radial Basis Functions (RBF) or least-square fitting with various basis functions, GFD mesh-free schemes face common challenges regarding collocation selection, flow conservation, implicit schemes, adaptation strategies etc. The current work focuses on the least-square-based GFD as a first step to assess the mesh-free modelling performance and adaptative simulations.

Given a function $\phi(x)$ with $x = (x, y, z)$ in a Cartesian system, known at a set of n_i discrete points $\phi_i = \phi(x_i)$ in a local cloud, and approximation function $\hat{\phi}(x)$ can be constructed using polynomials:

$$\hat{\phi}(x) = p(x)^T \alpha, \tag{1}$$

where $p(x)$ is the polynomial basis with m terms and α is a coefficient vector. By enforcing known values at local cloud points, the coefficient vector α can be determined through the following linear system:

$$\hat{\phi}(x_i) = p(x_i)^T \alpha = \phi_i, \; i = 1, 2, ..., n_i, \tag{2}$$

where the vector $p(x_i)$ contains the basis polynomial values at point x_i. The linear system is of the size $n_i \times m$ and is not necessarily a square system. We assume that $n_i \geq m$ so that the system is not under-determined, in which case there may exist many equally good solutions. In most case, we have $n_i > m$ so that the system is over-determined.

If the point cloud is carefully design to avoid singularity, we can use the weighted least-square approach to determine a best fit with the minimum L2 normal and solve for the coefficients α:

$$\alpha = (P_i^T W_i P_i)^{-1} P_i^T W_i \phi_i = C_i \phi_i, \tag{3}$$

where $C_i = (P_i^T W_i P_i)^{-1} P_i^T W_i$. The subscript i denotes values calculated at the n_i cloud points. ϕ_i is the $n_i \times 1$ vector of known values at supporting cloud points, and P_i is the $n_i \times m$ polynomial matrix. W_i is a $n_i \times n_i$ diagonal weighting matrix adjusting contributions from cloud points. The choice of the weighting function varies, and the present work adopted inverse-distance weighting functions.

The computational cost of the matrix inversion is negligible due to its small size, but the point cloud topology or collocation should be designed with caution so that the inversion operation is feasible. The choice of collocation shapes is investigated in later sections. Inserting Equation 3 into Equation 1, the function $\phi(x)$ is then approximated by $\hat{\phi}(x)$ as

$$\hat{\phi}(x) = p^T(x)\alpha = p^T(x) C_i \phi_i = N(x)\phi_i, \tag{4}$$

where $N(x) = p^T(x) C_i$ is regarded as the shape function.

The partial derivatives of $\phi(x)$ can be determined through the approximation:

$$\frac{\partial \phi(x)}{\partial x} \approx \frac{\partial \hat{\phi}(x)}{\partial x} = \frac{\partial N(x)}{\partial x} \phi_i, \tag{5}$$

where the partial derivatives now depend on the partial derivatives of the shape function $N(x)$, which is written as follows:

$$\frac{\partial N(x)}{\partial x} = \frac{\partial p^T(x)}{\partial x} C_i \phi_i, \tag{6}$$

here $\frac{\partial p^T(x)}{\partial x}$ are the partial derivatives of the basis polynomial. Note that the matrix C_i was considered as a constant while determining the derivatives. This is often known as the fixed least square approach and it reduces the computational cost.

B. Spatial Discretisation of Governing Equations

With the least-square approximation of derivatives, the governing flow equations can be discretised in a manner that is similar to the classic Finite Difference scheme. The inviscid part of the governing flow equations can be written in the following strong form in a Cartesian frame:

$$\frac{\partial W}{\partial t} + \nabla \cdot F = 0, \tag{7}$$

where W is the vector of conservative flow variables, and F is the convective flux term. Assuming a local point could Ω_i with the centre point i and j supporting points, the flow equation can be discretised at point i through the following approximation of the derivatives:

$$\frac{dW_i}{dt} = -\sum_{j\in\Omega_i}\sum_{v=1}^{d}\frac{\partial N}{\partial x^v}(x_j)F_j^v, \tag{8}$$

where the superscript v denotes the number of spatial dimensions from 1 up to d. However, this scheme is not stable due to the hyperbolic nature of the governing flow equations, and upwind schemes should be used for stability. This is realised by evaluating the flux terms at the middle point of i and j, which denoted as $j-\frac{1}{2}$:

$$\frac{dW_i}{dt} = -\sum_{j\in\Omega_i}\sum_{v=1}^{d}\frac{\partial N}{\partial x^v}(x_{j-\frac{1}{2}})F^v_{j-\frac{1}{2}}, \tag{9}$$

where the flux term $F_{j-\frac{1}{2}}$ are the reconstructed flux values at the middle point and Riemann solvers can be used here to account for the upwind properties. In this work, the Osher's flux scheme [7] was adopted, where the mid-point flux is reconstructed as:

$$f_{j-\frac{1}{2}} = \frac{1}{2}(f(p_L)+f(p_R)) - \frac{1}{2}|\tilde{A}(p_L,p_R)|(p_R-p_L), \tag{10}$$

where $\tilde{A}$ is the Roe-averaged Jacobian matrix [8]. p_L and p_R are the primitive variable vectors on the left- and right-hand sides of the middle point, respectively. There are various ways to determine of the middle point values. A first-order reconstruction would simply take the values of the two neighbouring points i and j, while higher-order reconstruction such as MUSCL can also be incorporated here. The present work adopted the MUSCL reconstruction.

Note that in Equation 9, the flux term between points i and j is not reciprocal, as point i may not necessarily be in the point cloud centring point j. This leads to the lack of flow conservation and this is an important pitfall of this scheme.

It should also be noted that Equation 9 resembles the popular finite volume scheme in the local point cloud, if we take the term $\frac{\partial N}{\partial x^v}(x^v_{j-\frac{1}{2}})$ as a virtual edge. This similarity with FVM schemes is an important advantage for numerical implementation and especially for the realisation of hybrid mesh-based/mesh-free methods.

As for the viscous terms, the flow variables are simply averaged at the midpoint for the flux computation. The gradient variables, however, adopted slight modifications [9] to avoid the odd-even node decoupling:

$$\nabla\phi_{ij} = \frac{1}{2}(\nabla\phi_i+\nabla\phi_j) - \left[\frac{1}{2}(\nabla\phi_i+\nabla\phi_j)\cdot\frac{\delta x}{|\delta x|} - \frac{\phi_j-\phi_i}{|\delta x|}\right]\cdot\frac{\delta x}{|\delta x|} \tag{11}$$

where $\delta x = (x_j - x_i)$ is the difference between the two points.

RANS turbulence closures are also included in the discretisation as additions sources terms. The one-equation Spalart-Allmaras model [10] was used in the present work. The convective and diffusive terms were evaluated following the same procedures described for the inviscid and viscous terms, respectively.

C. Implicit Temporal Discretisation

The temporal discretisation in this work adopted implicit schemes that are rarely used for mesh-free schemes due to complexity. The classic dual time-stepping concept [] was used here. Taking the RHS of Equation 9 as a residual vector $-R$, and applying an implicit temporal discretisation to the ODE, we shall have:

$$D_t(W) = -R^{n+1}, \tag{12}$$

where $D_t()$ is an implicit finite difference operator about the real time t, and the superscript $n+1$ denotes the $n+1$ time step in real time. Regrouping this equation to form a pseudo steady residual as follows:

$$R^* = D_t(W) + R^{n+1} = 0, \tag{13}$$

where R^* is a pseudo residual term. By adding a temporal term and apply another temporal discretisation D_τ, both about the pseudo time τ, a steady flow problem can be formulated:

$$D_\tau(W) = -R^{*(l+1)}, \tag{14}$$

$(l+1)$ is the $l+1$ time step in the pseudo time step. The $R^{*(l+1)}$ can be approximated using second-order linearisation:

$$R^{*(l+1)} \approx R^{*l} + \frac{\partial R^*}{\partial p}(p^l)\Delta p, \tag{15}$$

where $\frac{\partial R^*}{\partial p}$ is the Jacobian matrix relative to the primitive variables $p = [\rho, u, v, w, P]^T$, and $\Delta p = p^{l+1} - p^l$. The steady discretisation then becomes:

$$D_\tau(W) = R^{*l} + \frac{\partial R^*}{\partial p}(p^l)\Delta p, \tag{16}$$

where the operator D_τ can take simple forms such as the backward Euler since this is essentially a steady problem, and the term Δp can be easily solved as a large linear system. A particular advantage here is that acceleration techniques can be adopted to accelerated the solution process. In this work, the local time step approach is adopted. When handling essentially steady problems, the temporal discretisation can directly follow Equation 13 by dropping the first level of discretisation in Equation 12.

The major difficulty here is the determination of the Jacobian matrix $\frac{\partial R_i}{\partial p_j}$ at each supporting point j, which can be expanded as

$$\frac{\partial R_i}{\partial p_j} = \sum_{j \in \Omega_i} \sum_{v=1}^{d} \frac{\partial N}{\partial x^v}(x^v_{j-\frac{1}{2}}) \frac{\partial F^v_{j-\frac{1}{2}}}{\partial p_j}, \tag{17}$$

where we can notice that the shape function terms are irrelevant to the Jacobian computation as they are purely geometric. While following Eulerian descriptions, the Jacobian computation depends solely on the flux reconstruction and the Riemann solver.

D. Point Adaptation Strategies

The mesh-free modelling is particularly suitable for adaptative modelling considering its strong flexibility to accommodate points without connectivity constraints. Therefore, this work will assess various adaptation strategies for the mesh-free modelling. In general, adaptative modelling requires a refinement mechanism to insert or remove elements where necessary, and a metric to indicate where to insert or remove. This section first details the refinement mechanisms to be evaluated in this work. The first refinement mechanism is illustrated in Figure 1. In 2D, when a point is tagged to be refined, 8 additional points are inserted along all orthogonal directions to form a local Cartesian cloud (in 3D, 26 points will be added). This refinement/de-refinement approach can be recursively performed and shrinks the local collocation size by a constant factor of 2. This is analogous to the conventional adaptive Cartesian grid approach, and can be implemented using a similar tree-like structure. However, it is obvious that it is incompatible with arbitrary, irregular baseline point distributions. Still, it is important to investigate this mechanism as evenly distributed point clouds are desirable for off-body regions.

A more universal mechanism to introduce additional points is illustrated in Figure 2. When a point is tagged to be refined, this mechanism identifies its local collocation configuration and inserts the middle point of each connection. Further refinement is realised by repeating the middle point insertion, and coarsening can be achieved by removing these additional points. This approach is extremely easy to implement and is suitable for arbitrary point distributions. The drawback, however, is that the middle points may damage the orthogonality of point clouds. For a evenly distributed baseline point cloud in Figure 2, large point density differences may arise after a few refinement iterations.

In the current study, near-body or boundary layer point clouds are generated by projecting point rays along the surface norms and removing overlapping/in-body points. It is hence straightforward to implement near-body refinements by adding boundary points where necessary and repeating the projection process. Additional boundary points are added

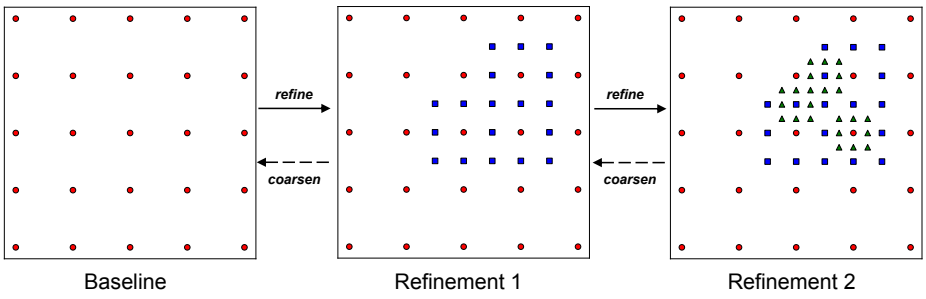


Fig. 1 Cartesian point adaptation. Different markers and colours denote points added in different refinement levels.

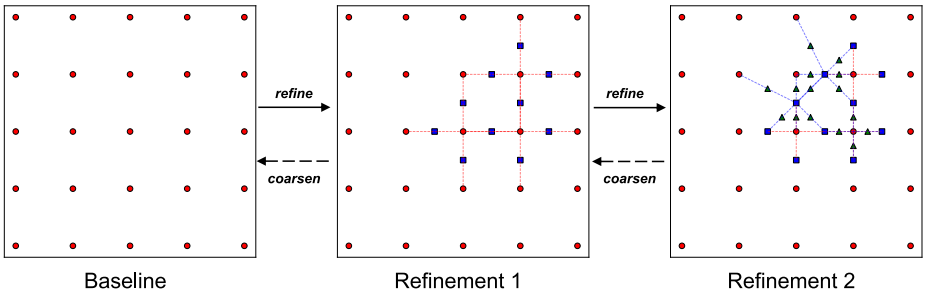


Fig. 2 Middle point adaptation. The dashed lines represent connections in a local point cloud. Different markers and colours denote points added in different refinement levels.

either through parametric definitions (e.g. splines or NURB surfaces) or through interpolation (e.g. moving-least-square or linear). The adapted near-body and off-body points clouds then blended for mesh-free simulations.

E. Collocation Search Methods

When additional points are introduced through adaptation, local point collocations must be re-established. As will be discussed in Section IV.B, the choice of collocation configurations is vital to the mesh-free modelling. Although the present least-square based mesh-free approach prefers reciprocity and orthogonality, collocations including critical members have tolerance for slight non-reciprocity. In light of this, this sections introduces different collocation search methods based on distance and local Voronoi tessellation.

1. Distance-based collocation search

The distance-based search is simple and straightforward: take all points within a certain distance about the centre point as the local collocation. This is illustrated in Figure 3. For collocations centring at points i, j, and k in Figure 3. The lines represent connections between the collocation centre and the members, with arrows pointing to the centre. A connection is reciprocal when it is double-arrowed. The search distance, which are represented by circles in Figure 3, varies according to the local point density to ensure compact support.

However, this simple approach has no guarantee on the regularity and reciprocity. For instance, the connection between points i and j in Figure 3 has only one arrow, meaning that point i will receive contributions from point j but it never contributes to point j. Increasing the search radius of point j may include i in its collocation, but may result in non-reciprocity elsewhere and damage the compactness.

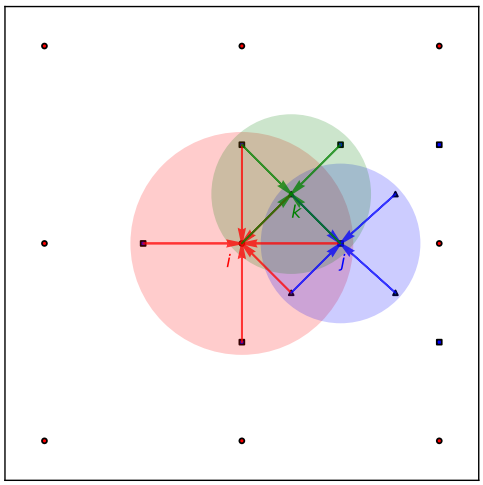


Fig. 3 Collocation search based on distance. The circles denote the search radius. The lines denote connections between the collocation centre and its members, with arrows pointing to the centre. A double-arrowed connection is reciprocal.

2. Distance-based collocation search with configuration selection

The distance-based search simply includes all points within the search radius, but the least-square approximation may be improved through biased weighting or selection. This study proposes a selection scheme that improves the orthogonality of the collocation and maintain small collocation sizes.

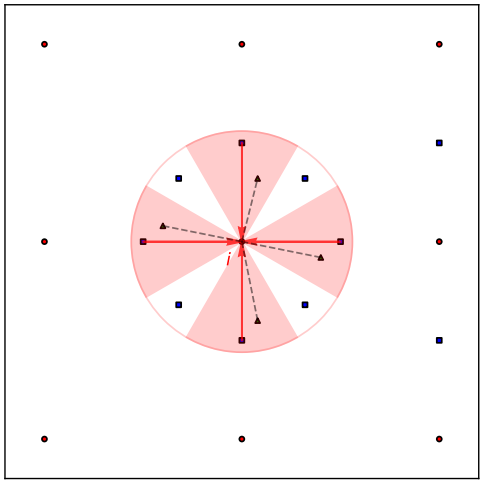


Fig. 4 Collocation search based on distance with configuration selection. The shaded area represents candidate regions, and the dashed lines denote candidate configuration. The solid lines with arrows denote the final selected configuration.

The selection process starts from preferring points closer to orthogonal directions, as shown in Figure 4. Points within certain angles about the Cartesian axes, e.g. $\pm 30°$, are marked as candidates, while all points outside these regions are excluded. The next step is to select one point in each region, together with the centre point, to comprise the local collocation. This process is realised by testing all possible combinations, and selecting the one that gives the best fit of the local field using the given basis. If no point can be found in a region, the search radius and angle will be enlarged until more points are included.

Although this approach cannot improve the reciprocity, it ensures points that are critical to the local approximation are included in the collocation, while trivial points are excluded. This process could be coupled with the flow computation to realise configuration adaptation. The drawback of this approach is associated with the large computational cost testing all possible combinations. As a first step, the current work only used this approach as a static searching algorithm to accommodate point refinements.

3. Collocation search using local Voronoi tessellation

Voronoi tessellation is a domain decomposition approach. Upon a given set of points, the tessellation decomposes the domain into Voronoi cells consisting of points closer to the cell centre than any other points in the domain. For a given collocation centre, these neighbouring points can be used as a collocation for the mesh-free modelling. For a global Voronoi tessellation using all points in the domain, the neighbouring relation for a point couple is reciprocal, thereby ensuring the reciprocity of the collocation. Moreover, the Voronoi tessellation for a given domain upon given points is unique []. This property indicates that even if the tessellation is performed locally, it is possible to have the same global results, as long as the local domain is sufficiently large. This has enabled the local Voronoi-based collocation search in this work.

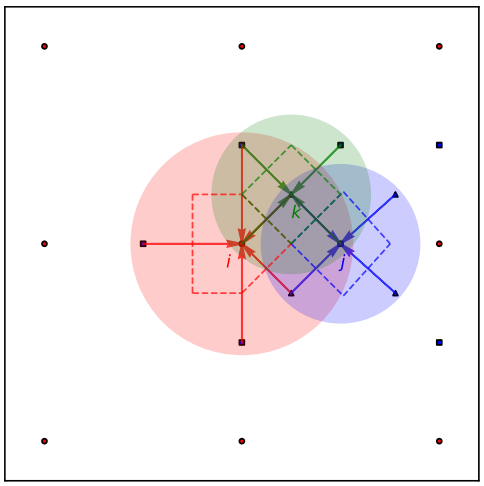


Fig. 5 Collocation search based on local Voronoi tessellation. The circles denote the search radius. The solid lines denote connections between the collocation centre and its members, with arrows pointing to the centre. The dashed lines represent the local Voronoi cells.

The local Voronoi-based collocation search is illustrated in Figure 5. Similar to the distance-based search, a group of points are first selected based on a search radius. The search radius can adopt a deliberately large value, e.g. the lowest point density in the domain, to ensure the uniqueness of the tessellation. Upon these selected points, a Voronoi tessellation is conducted. Neighbours of the Voronoi cell enclosing the centre are selected as the local collocation. As shown in Figure 5, the Voronoi cells of nearby collocations have conforming boundaries, thereby ensuring reciprocal contributions. Compared to the simple distance-based search in Figure 3, the non-reciprocal connection between i and j has been avoided.

III. Numerical Verification

A. Transonic ONERA M6 Wing

This section presents 3D mesh-free simulations of transonic flows passing the classic ONERA M6 wing. The M6 wing surface geometry is shown in Figure 6 along with point distributions and boundaries. The wing surface points were manually distributed following the sectional curvatures, while spatial points were grown along the wing surface norm until they reached the semi-spherical far-field boundary, which was about 5 times the wing span away in all directions. The computational domain consisted of 182,853 points in total, with 4,425 points on the wing surface. Note that the wing surface in Figure 6 was for illustration purposes only. In the mesh-free computation, all geometries were represented with discrete points.

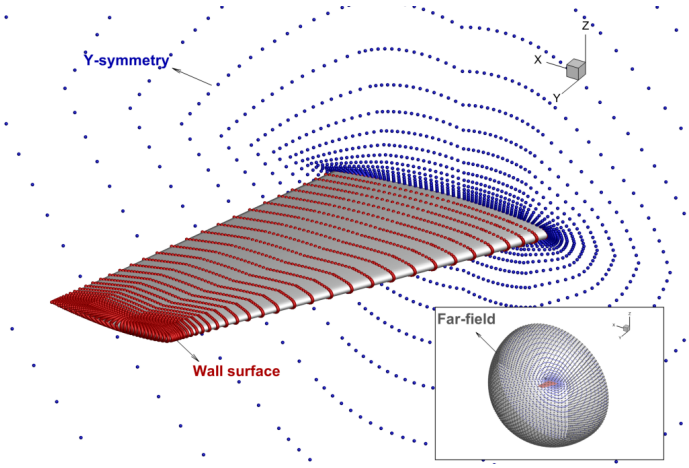


Fig. 6 Point cloud and boundaries for simulations of the ONERA M6 wing.

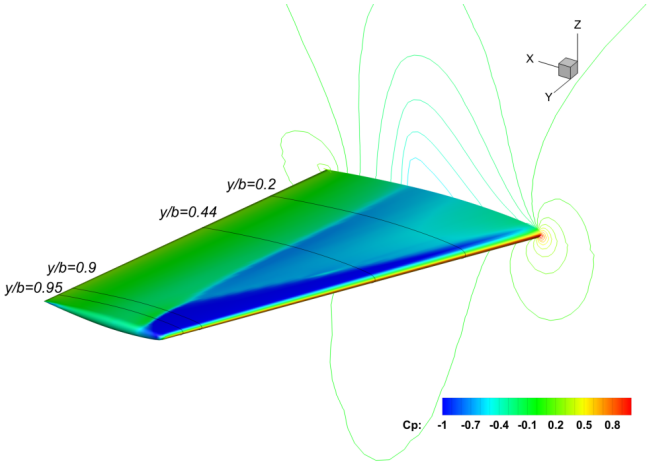
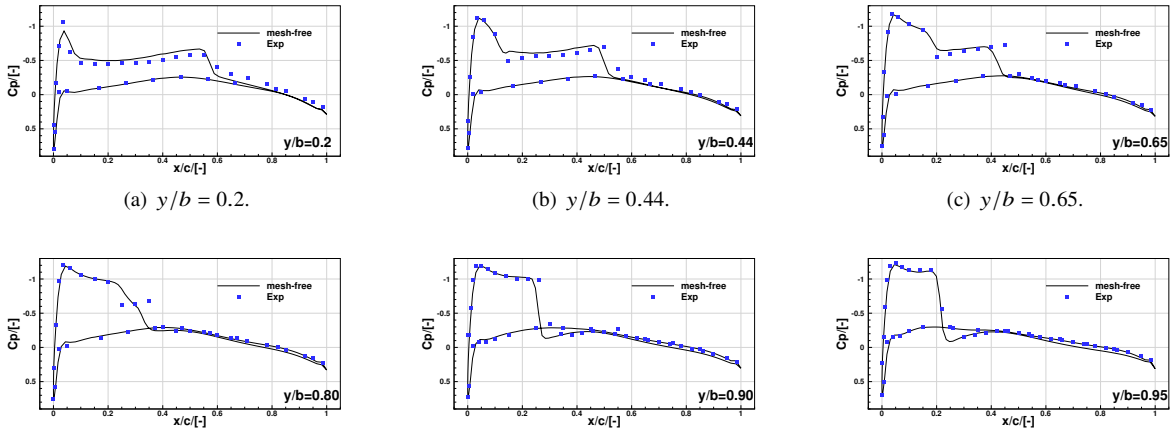


Fig. 7 Surface pressure coefficient solutions from the mesh-free simulation.

The simulations were conducted at $Ma = 0.8395$ with an $AoA = 3.06$ using Euler equations. This is a classic transonic case where multiple shock waves would form at the wing surface. This is shown in the surface pressure solution in Figure 7. The mesh-free simulation resolved two shock waves at the wing root near the leading-edge and the mid-chord sections, respectively, and the shock waves gradually merged towards the wing tip.

Sectional pressure coefficient distributions were extracted and compared with experimental measurements in Figures 8(a) to 8(f). The mesh-free results showed excellent agreement with the measurements, especially for the wing lower surface and leading-/trailing edge regions. Light discrepancies were noticed near the shock waves, particularly in Figure 8(d) at section $y/b = 0.8$ where the shock waves merge. Increasing the local point density in these regions can help

improve the shock wave resolution. Nonetheless, these results provide excellent validation for the present mesh-free solver for complex 3D simulations.



(a) $y/b = 0.2$. (b) $y/b = 0.44$. (c) $y/b = 0.65$.

(d) $y/b = 0.8$. (e) $y/b = 0.9$. (f) $y/b = 0.95$.

Fig. 8 Comparisons of wing sectional pressure coefficient distributions between the experimental and mesh-free numerical results.

B. Inviscid Flow past a Circular Cylinder

This section presents mesh-free modelling of inviscid flows passing a circular cylinder at $Ma = 0.38$. This is a classic case for validation as well as error quantification. The modelling parameters are presented in Table 1. Three sets of point clouds of increasing point density were used for the simulation to quantify the spatial accuracy, as shown in Figures 9(a) to 9(c).

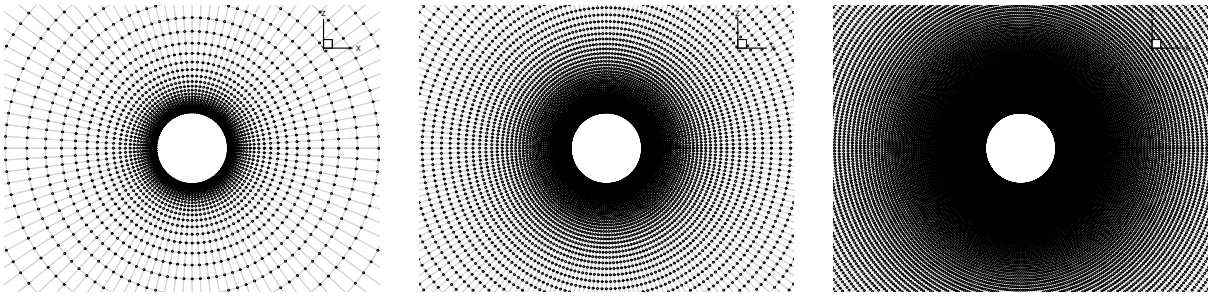
The cylinder surface was first discretised by N_a points equally distributed along the angular or azimuthal direction. N_r points were then popped out along the surface outer normal direction following an exponential growth, extended to about 20 diameters away. The overall relative entropy product, i.e. $dS = \frac{p}{p_\infty}(\frac{\rho_\infty}{\rho})^\gamma - 1$, was computed to quantify the spatial accuracy of the mesh-free scheme, because the inviscid flow remain isentropic and any increase in the entropy is associated with the numerical error.

Table 1 Modelling parameters for simulations of the flow past a circular cylinder.

Ma_∞	0.38
Flow model	Compressible Euler
Outer boundary distance	20D
First-layer point wall distance	0.001D
Radial growth rate	Exponential
Point cloud density (Angular × Radial)	100×50 (coarse) 200×100 (medium) 400×200 (fine)

Figure 10(a) presents the convergence history of the flow simulations using coarse, medium, and fine point clouds. The coarse cloud had difficulty converging, and the residual stagnated around an order of -4. This should be related to larger numerical viscosity leading to small unsteadiness in the cylinder flow. As the point density increased, the residual quickly converged to the order -8.

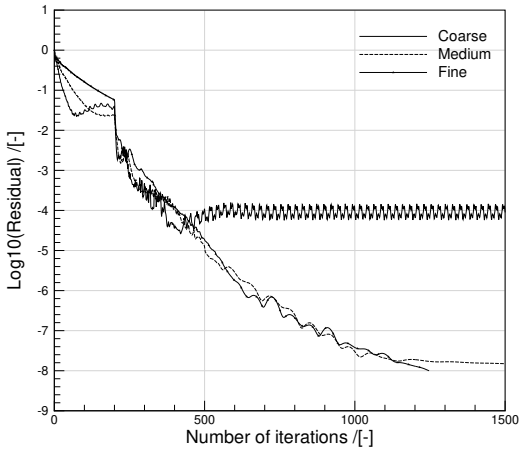
Figure 10(b) presents the cylinder surface pressure coefficient distributions for more quantitative comparisons. The pressure distribution agreed well with theoretical solutions, and the downstream stagnation pressure was only slightly



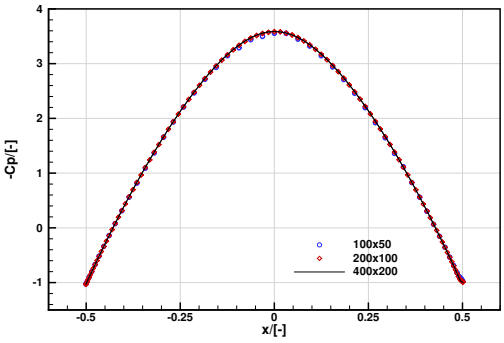
(a) Coarse: 100×50. (b) Medium: 200×100. (c) Fine: 400×200.

Fig. 9 Coarse, medium, and fine point clouds for the cylinder flow simulation. The numbers $N_a \times N_r$ denote N_a and N_r points along azimuthal and radial directions, respectively.

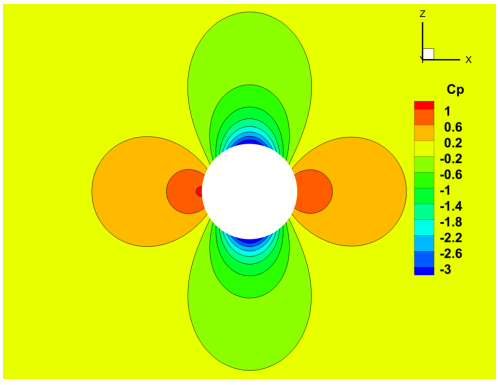
under-recovered. The coarse-point results showed slight discrepancies near $x = 0$ where the pressure is lowest. The medium- and fine-point results were identical. Figure 10(c) presents the pressure field resolved using the medium point cloud and the mesh-free modelling. The flow solution showed symmetry along the horizontal and vertical axes.



(a) Convergence history.



(b) Cylinder surface pressure coefficient distributions.



(c) Medium point cloud pressure coefficient contours.

Fig. 10 Mesh-free modelling results of inviscid flow passing a circular cylinder.

The spatial accuracy of the mesh-free scheme was measured and presented in Figure 11. The modelling error was

measured using the overall entropy product, while the spatial size h was defined as $h = 1/\sqrt{N_a \times N_r}$. The slope of the modelling error relative to the spatial size was about 2, which is consistent with the second-order least-square approximation in the mesh-free scheme. This shows that the designed spatial accuracy in the mesh-free scheme was successfully achieved.

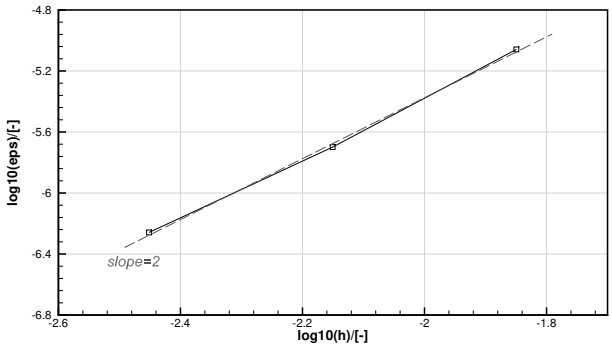


Fig. 11 Accuracy measurements using the circular cylinder flow simulations.

IV. Assessment of Point Cloud Distributions and Configurations

A. Irregular Point Distributions

For non-trivial geometries, such as wing flaps and landing gears, it is difficult, if not entirely impossible, to maintain regular spatial point distributions. This section therefore assesses the impacts of irregular point clouds on the mesh-free modelling through manually introduced disturbances. The investigation here focused on the aforementioned medium point cloud (200×100 in Table 1). Irregular point distributions were introduced by adding random offsets of about 0.3 times the local radial size (in both x and y directions), as shown in Figure 12. To study how the results change with different degrees of irregularity, the number of point layers that were disturbed off the wall the boundary was varied, as detailed in Table 2. Four different point clouds with 100, 90, 80, and 70 layers (out of 100) of disturbed points were compared here. Points on the wall boundary remained unchanged since they have to represent the boundary shape.

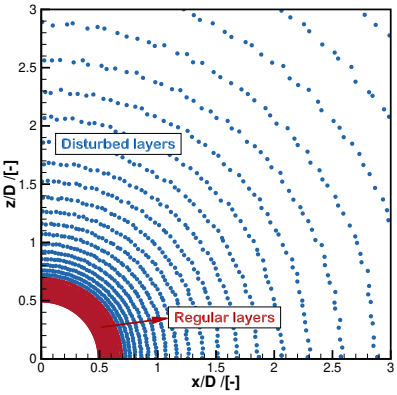


Fig. 12 Irregular point clouds with layers of random point offsets for the circular cylinder flow simulations.

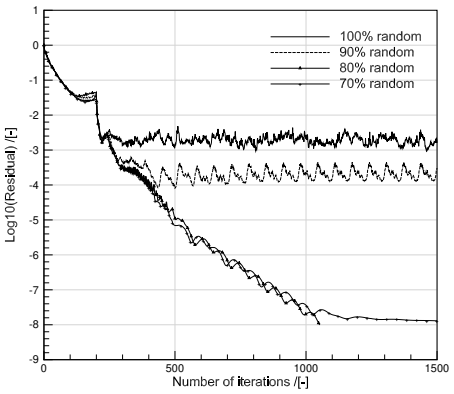
The flow convergence history using these irregular point clouds is presented in Figure 13(a). The 100% and 90% irregular point clouds had difficulty in converging, while the 80% and 70% irregular point clouds converged smoothly to the order -8. Figure 13(b) presents the surface pressure coefficients resolved on point clouds with different extents of irregularity. It shows that the fully random points led to poor results that had large discrepancies from the theoretical

Table 2 Introduction of irregular point distributions through different levels of random point offsets. δr denotes the local distance between two adjacent radial points.

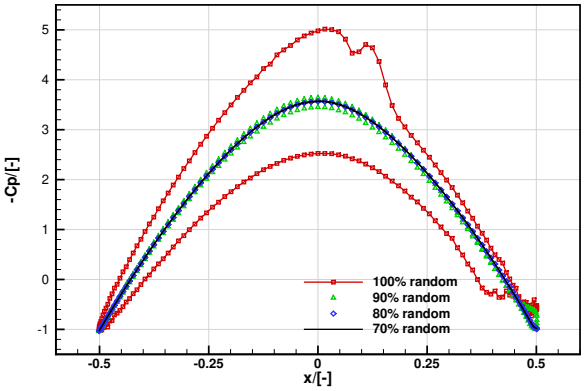
Irregularity level	Layers of un-disturbed/disturbed points off wall	Disturbance level
100%	0/100	0.3×δr
90%	10/90	
80%	20/80	
70%	30/70	

solution and failed to capture the symmetry of the solution. Flow solutions of the 100% and 90% irregular points are presented in Figure 13(c) and 13(d), respectively. The modelling results showed strong asymmetry due to irregular point distributions near the wall. As the randomness was reduced near the wall boundary, the results were significantly improved. 20 layers of regular points near the wall provided results with negligible differences.

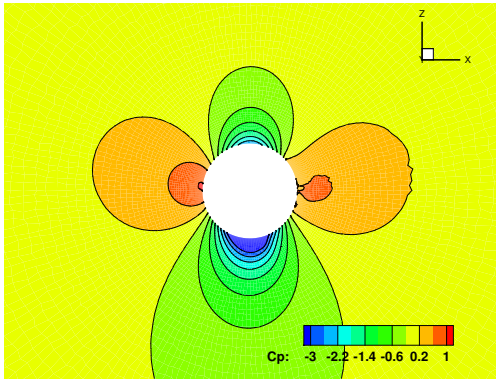
This investigation suggests that the current mesh-free method is compatible with irregular point distributions, but is still sensitive to the distributions in critical regions such as wall boundaries. This could be addressed via adapted clouds.



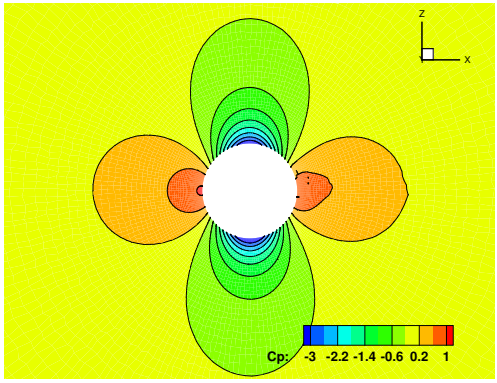
(a) Convergence history.



(b) Surface pressure coefficient distributions



(c) Cp contour with 100% randomness.

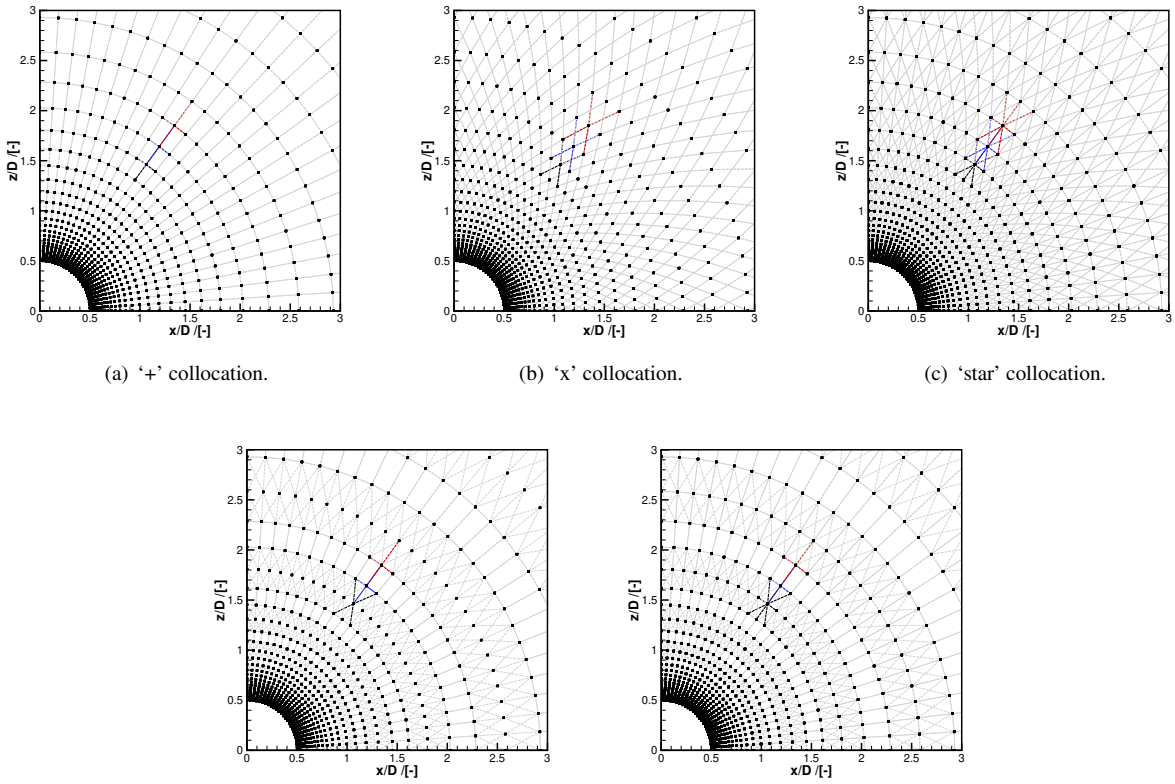


(d) Cp contour with 90% randomness.

Fig. 13 Mesh-free modelling results with different extents of irregularity.

B. Irregular Collocation Configurations

The least-square approach 1 approximates local functions and derivatives using a set of neighbouring points, i.e. the collocation. The choice of member points of the collocation, i.e. the collocation configuration, has critical impacts on the approximation accuracy. In terms of flow physics, the collocation configuration is also vital to the flow conservation as indicated by Equation 9. The global flow conservation is strictly maintained only when the connection between two points is reciprocal. This eventually causes the method edge-based and semi-mesh-free, and represents a strong restriction on the mesh-free flexibility. Therefore, this section investigates how various point collocations affect the solution upon the same point cloud (the medium point cloud 200×100 of Figure 9(b) was used). Figures 14(a) to 14(e) present five different collocations with different orthogonality and reciprocity as detailed in Table 3.



(a) '+' collocation. (b) 'x' collocation. (c) 'star' collocation.

(d) '+x' mixed collocation. (e) '+-star' mixed collocation.

Fig. 14 Various local point collocation shapes (point clouds coarsened for clarity).

Table 3 Properties of different collocation in Figures 14(a) to 14(e).

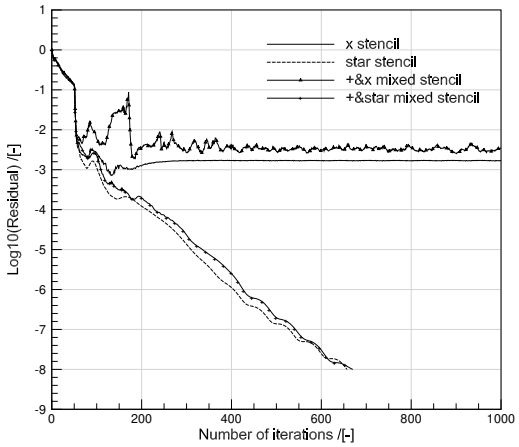
Collocation shape	'+'	'x'	'star'	'+x' mixed	'+-star' mixed
Orthogonal	Y	N	Y	N	Y
Reciprocal	Y	Y	Y	N	N

The '+' collocation in Figure 14(a) has been used in previous sections. This collocation is analogous to typical finite differences using structured grids. The 'x' collocation in 14(b) is still regular and reciprocal (i.e. the centre point is aways a member of all collocations centring at its members), but it has worse orthogonality and more distant supporting points. Figure 14(c) shows a star-shaped collocation merging the '+' and 'x' shapes, which has an extended range of support and is also reciprocal.

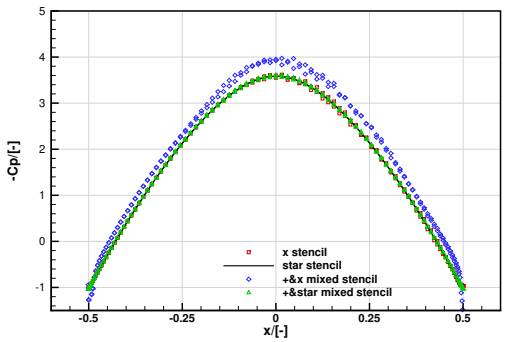
Figures 14(d) and 14(e) present mixed collocations along the radial direction. Figure 14(d) shows a mixture of 2 layers of '+' collocations and 1 layer of 'x' collocations, while Figure 14(e) shows a mixture of 2 layers of '+' collocations and 1 layer of 'star' collocations. Both configurations were not reciprocal when switch from one collocation type to another.

Figure 15(a) presents the convergence history of simulations using different collocations. Worst convergence behaviour was observed for the '+x' mixed collocation, which stayed around an order of -2.5 with small oscillations. The 'x' collocation converged slightly better to an order of -2.9 and stayed constant. As for the star-shaped and the '+-star' mixed collocations, the residual was quickly reduced to the order of -8 following similar patterns.

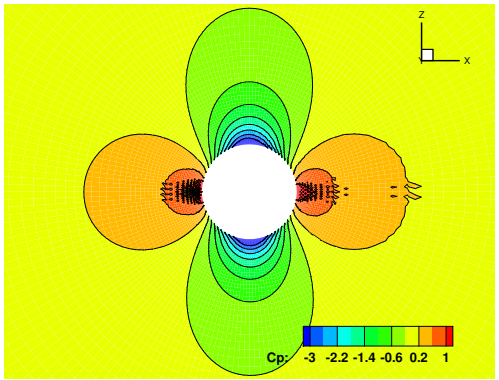
The cylinder surface pressure coefficient distributions were further extracted and presented in Figure 15(b) for more quantitative comparisons. The '+x' mixed stencil offered the worst prediction with discrepancies and oscillations all around the surface, and the symmetry was not well preserved. Figure 15(c) presents the flow solutions using the the '+x' mixed collocation. The solution suffered from strong dispersions, especially near the front and aft stagnation points. Small asymmetry and oscillations were also noted in solutions using the 'x' collocation and the '+-star' mixed collocation. The star collocations showed the best result with fine symmetry.



(a) Convergence history.



(b) Surface pressure coefficient distributions.



(c) '+x' mixed collocation.

Fig. 15 Mesh-free modelling results using different collocation shapes (inviscid flow over cylinder).

This investigation of different collocation shapes using the same point cloud demonstrates the flexibility of the present mesh-free modelling. However, it also suggests that the method is sensitive to choices of collocations for least-square computation. Collocations with poor orthogonality and reciprocity seriously deteriorate the modelling results and vice versa. The collocation should also include essential points reflecting critical properties such as up-winding.

V. Assessment of Adaptive Point Clouds

The mesh-free modelling is particularly suitable for adaptative modelling considering its strong modelling flexibility. Moreover, compared to conventional Adaptive Mesh Refinement approaches, mesh-free CFD methods have greater freedom and flexibility for adaptation. The adapting parameters could include point distributions, point numbers, collocation sizes, shape functions etc. As a first step, the present work explores the mesh-free adaptation by adding additional points when necessary.

A. Assessment of Adaptive Simulations

This section assesses the adaptive mesh-free simulation strategies using the isentropic vortex transport problem [11]. The problem has a analytical definition as follows:

$$\begin{aligned} \rho &= \left[1 - \frac{(\gamma-1)b^2}{8\gamma\pi^2} e^{1-r^2}\right]^{\frac{1}{\gamma-1}} \\ u &= u_\infty - \frac{b}{2\pi} e^{\frac{1-r^2}{2}} (y - y_c) \\ v &= v_\infty + \frac{b}{2\pi} e^{\frac{1-r^2}{2}} (x - x_c) \\ p &= \rho^\gamma \end{aligned} \tag{18}$$

where b is the dimensionless vortex strength, (x_c, y_c) is the vortex centre, $r = \sqrt{(x-x_c)^2 + (y-y_c)^2}$ is the distance from the centre, and (u_∞, v_∞) is the convection speed. $\gamma = 1.4$ is the specific heat ratio. ρ and p are the density and pressure, respectively. The vortex is placed at the centre of the computational domain with all outer boundaries set periodic. Without numerical dissipation and dispersion, the vortex should maintain the initial strength during the convection without any decay.

In this study, the convection speed was (u_∞, v_∞) set zero, so that the vortex was preserved at the initial position. The vortex would still decay due to numerical viscosity. The adaptation was expected to improve the spatial resolution and reduce the dissipation, thereby preserving the vortex shape over longer durations. The vortex strength was set as $b = 5$. The adaptation performance was examined through differences between the numerical and exact solutions after a constant 500 time steps, with the same dimensionless time step size of 0.1 large enough for the decay.

Table 4 presents the test matrix of adaptive strategies combining different adaption mechanisms and collocation search methods. The simulation was initialised on a baseline coarse point cloud consisting of 21×21 points covering the domain $[-5, 5] \times [-5, 5]$. This coarse point cloud would quickly dissipate the vortex. For Cases 1 to 3, the adaptation was successively conducted in regions where $r < 4, 3, 2, 1, r = \sqrt{(x^2 + y^2)}$, so that these cases have the same point clouds to highlight differences in collocation search methods. For Case 4, a fifth adaptation for the region $r < 0.5$ was conducted since the mid-point mechanism introduces fewer additional points than the Cartesian approach.

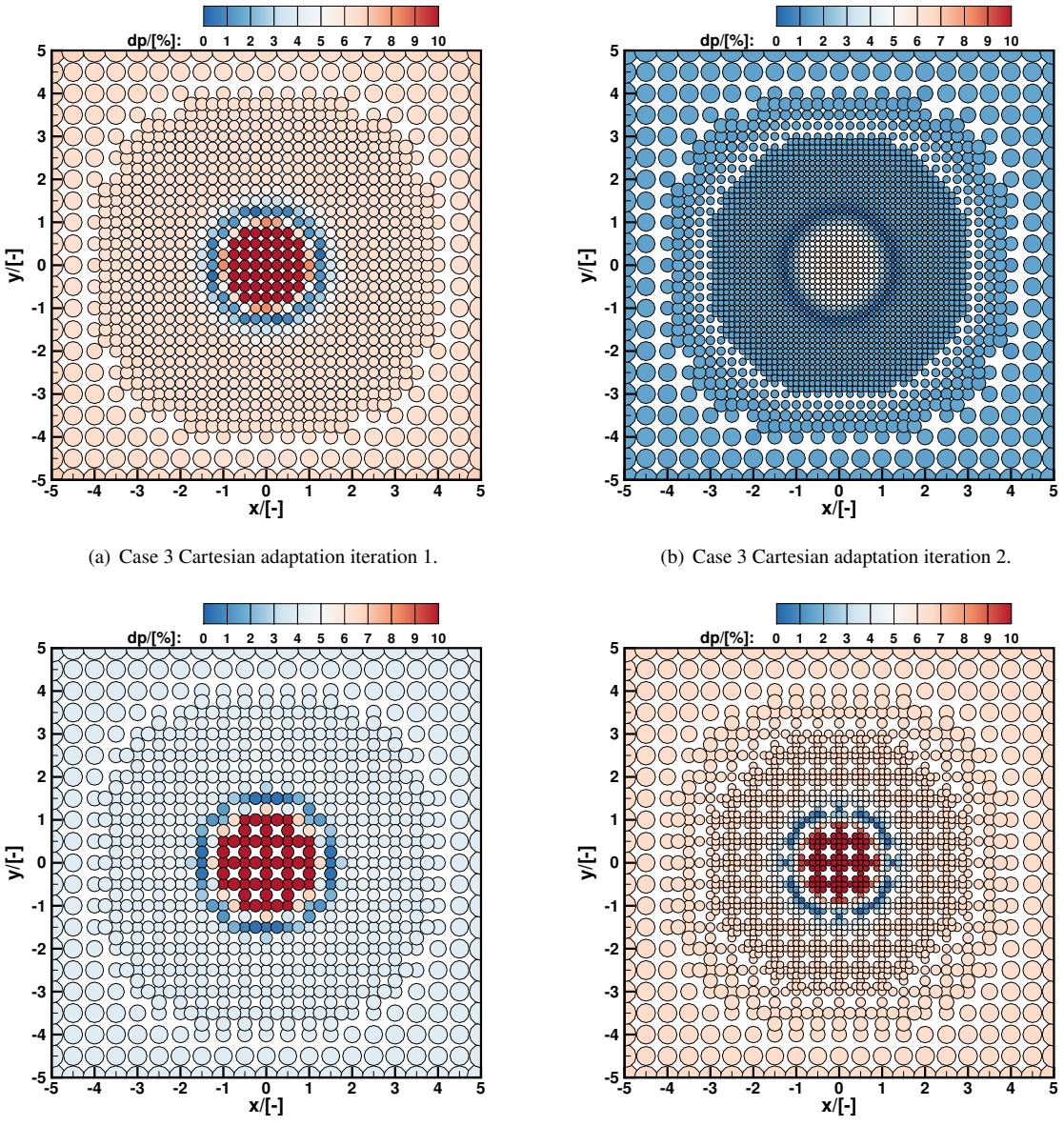
Table 4 Test matrix of mesh-free adaptive simulations.

Cases	Adaption Mechanism	Collocation Search
C1	Cartesian	Distance
C2	Cartesian	Distance with selection
C3	Cartesian	Voronoi
C4	Mid-point	Voronoi

Figures 16(a) to 16(d) illustrate the point clouds for Cases 3 and 4 in Table 4 after the first and second adaptation iterations. The contours denote the relative pressure difference $\delta p = |p_{num}/p_{exact} - 1| \times 100\%$ between the numerical solution p_{num} and the exact solution p_{exact}. The point sizes were scaled by the local point size.

Figures 16(a) and 16(b) illustrate the Cartesian point adaptation mechanism. It can be observed that adaptation preserved the initial regular point distribution with uniform density. Irregular distributions were only encountered in transitional regions where two refinement levels meet. Through the adaptation from Figure 16(a) to Figure 16(b), the pressure error was effectively reduced from about 10% maximum to 5% maximum.

Figure 16(c) and 16(d) illustrate the mid-point adaptation mechanism. In Figure 16(c), after the first adaptation iteration, the point distribution remained mostly uniform. However, with the second adaptation iteration in Figure 16(d), small local clusters formed centring the previous collocation centres, and this led to irregularity over the domain. Although the adaptation managed to shrink the high-error area near the centre, the global error was raised in larger domain.



(a) Case 3 Cartesian adaptation iteration 1.

(b) Case 3 Cartesian adaptation iteration 2.

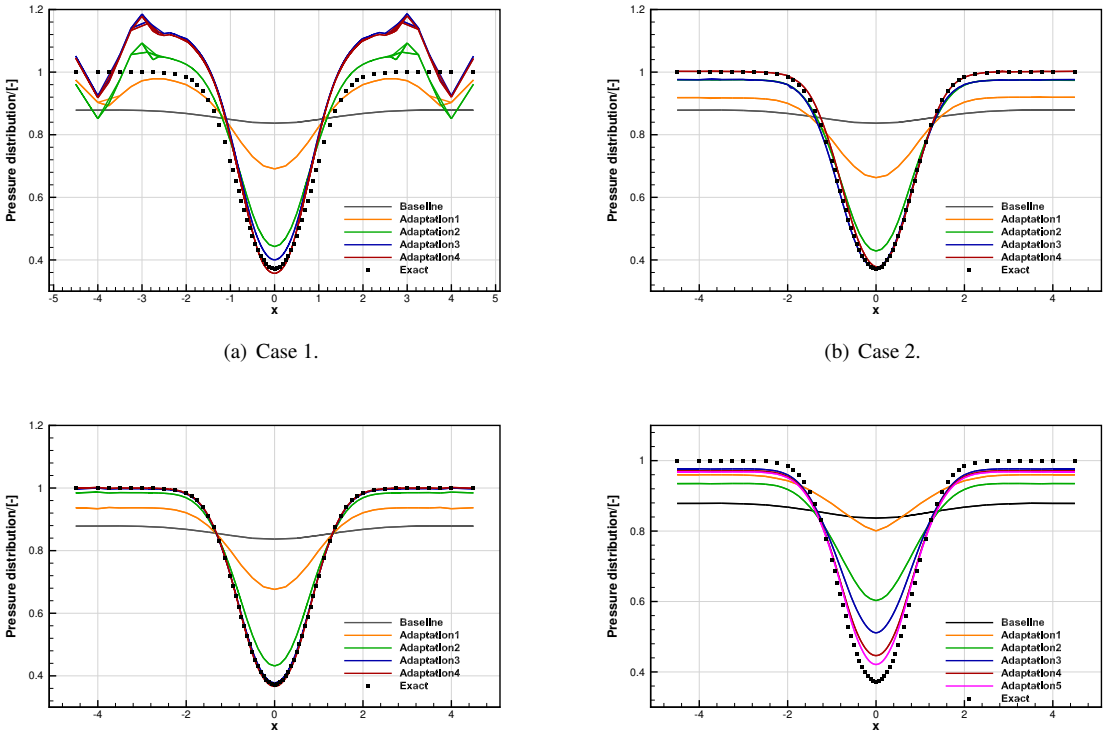
(c) Case 4 mid-point adaptation iteration1.

(d) Case 4 mid-point adaptation.

Fig. 16 Point clouds of Cases 3 and 4 in Table 4 after the first and second adaptation iterations. Point sizes were scaled with the local point size. Contours denote the relative pressure differences between the numerical and exact solutions $\delta p = |p_{num}/p_{exact} - 1| \times 100\%$**.**

The pressure profiles across the vortex centre extracted from cases in Table 4 are plotted in Figures 17(a) to 17(d). Different combinations of adaptation mechanisms and collocation search methods clearly displayed strong impacts on the adaptation solutions. Through the adaptation, Case 1 managed to improve the resolution of the central pressure valley, but solution of outer regions deviated further from the exact solution due to large point density transitions and

strong non-reciprocity in the collocations. Case 2 converged to the exact solution in the last iteration, but the solution showed very slight asymmetry. This was associated with asymmetric collocation configurations through the search. Case 3 smoothly converged to the exact solution since the third iteration using regular point distributions and reciprocal collocations. As for Case 4, the numerical solution gradually approached but was yet to conform to the exact solution through the adaptation. This was due to the poorer regularity, as well as the lower amount of additional points inserted as shown in Figure 18.



(a) Case 1.

(b) Case 2.

(c) Case 3.

(d) Case 4.

Fig. 17 Comparisons of pressure profiles across the vortex centre between numerical and exact solutions from adaptation cases in Table 4.

The overall performance of each adaptation strategy in Table 4 was measured using the overall relative pressure differences

$$\epsilon_p = \frac{\int \delta p ds}{\int ds}, \tag{19}$$

where δp is the point-wise relative pressure difference. ds is the local point size that reflects the area a point occupies. In the work, ds was defined as

$$ds = \pi r_i^2 / n_i, \tag{20}$$

where r_i is the radius of a small reference circle centring any point of interest, and n_i is the number of points covered by this circle.

Figure 18 presents the ϵp convergence history through the adaptation iterations for cases in Table 4. Case 3, combining Cartesian adaptation and local Voronoi search, showed the best convergence behaviour among all cases. This result highlights the importance of regular point distributions and reciprocal collocations. The adaptation brought the overall error ϵp to almost zero from the initial 13% on the coarse point cloud. Case 2 showed the second best convergence. Note that in this case, the reciprocity was not globally guaranteed. Nonetheless, with uniform point distribution and carefully optimised collocation configurations, the adaptation successfully reduced the final error to

about 1%. However, the computational cost of the configuration selection was much higher than that of the local Voronoi search.

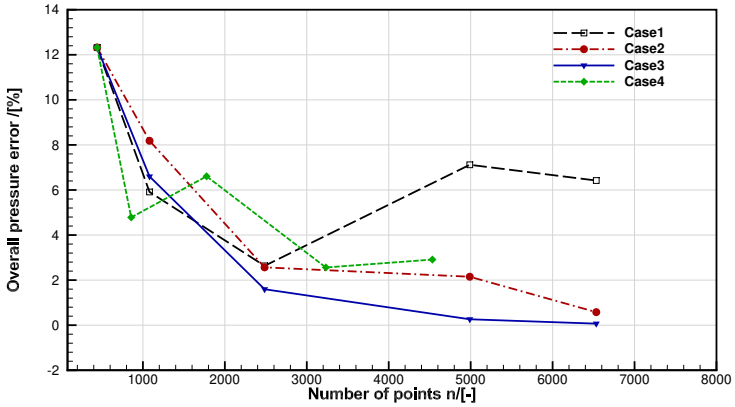


Fig. 18 Adaptation convergence of cases in Table 4. The overall relative pressure difference is defined as $\epsilon_p = \int \delta p ds / \int ds$, where $\delta p = |p_{num}/p_{exact} - 1|$ is the relative pressure difference and ds is the local point size.

Case 1, combining Cartesian adaptation and distance-based search, failed to converge using the same point clouds but with the distance-based search. The first two adaptation iterations reduced the overall error to about 3%. However, with more points added in the last two iterations, the error jumped to about 7%. The errors were mostly associated with the oscillations in regions near $r = 3$ and $r = 4$. These are transitional regions bridging areas with large point density differences, and are therefore particularly sensitive to collocation configurations. These again highlight the importance of the collocation configuration for the current mesh-free method.

Case 4, combining mid-point adaptation and local Voronoi search, also exhibited convergence behaviour, but the final overall error was about 3% and was higher than those of Cases 2 and 3. This was partially due to the lower number of additional points through the mid-point mechanism. In the last three iterations, the performance is similar to that of Case 2. It is interesting to notice that the first iteration brought the sharpest drop in the error compared to all other cases using the Cartesian adaptation. This should be associated with the more uniform global point distribution in this iteration as shown in Figure 16(c). However, as the regularity was damaged by further adaptation iterations as shown in Figure 16(d), the global error was slightly increased. As the point number increases, the solution of central pressure valley gradually approached the theoretical exact as shown in Figure 17(d), but the outer solutions improved little. These results highlight the importance of regular point distributions for the current mesh-free modelling.

B. Application to Aerofoil Simulations

This section applies the adaptive mesh-free modelling to more practical simulations of the RAE 2822 aerofoil at transonic conditions. The computational domain and initial point cloud are shown in Figure 19. Near-body points were generated by projecting point rays along the surface norm up to 0.1 c (c is the aerofoil chord), and were then blended with the successively coarsened Cartesian background points. The simulations were conducted at $Ma = 0.73$ and $AoA = 2.31^\circ$ using Euler formulations. The initial point cloud had 13,244 points with 125 points on the aerofoil surface.

For near-body points, the adaptation was realised by introducing new points to the surface through linear interpolation and re-projecting the point rays. For off-body point clouds, the adaptative simulation combined mid-point adaptation with local Voronoi search. Although the Cartesian adaptation showed the best performance in the previous assessment, it was not suitable for this case due to the irregular transition areas between near-body and multiple levels of background points. The mid-point adaptation is more universal and suitable for this case. To alleviate the point density difference issue, a point size constraint was introduced in the adaptation, i.e. if the local point size is smaller than the minimum allowed (0.012 c in the current case), no further adaptation will be conducted.

The current work used a weighted pressure gradient (WPG) metric to guide the output-based adaptation iterations. The metric ω_p is defined as

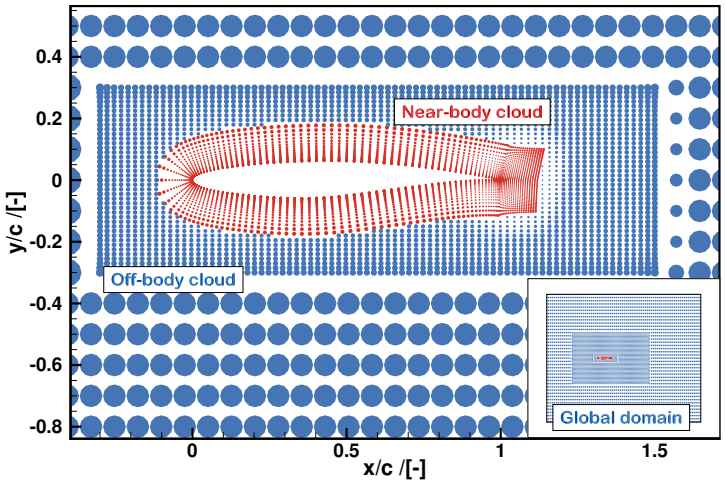


Fig. 19 RAE 2822 computational domain and point clouds. The points were scaled with the point size ds.

$$\omega_p = \frac{|\nabla p|\sqrt{ds}}{\int \sqrt{ds}}, \tag{21}$$

where $|\nabla p|$ is the pressure gradient magnitude, and ds is the local point size. This metric denotes the local pressure gradient weighted by the local point size. It prioritises high gradient regions with large point sizes (i.e. low point densities). It leads to more uniform point distributions driven by both flow and geometric features.

Figures 20(a) to 20(d) present the point cloud evolutions through the adaptation along with the weighted pressure gradient metric contours. The points were scaled by their sizes ds for illustration purposes. In each adaptation, points with $\omega_p \geq 0.05$ were tagged for refinement.

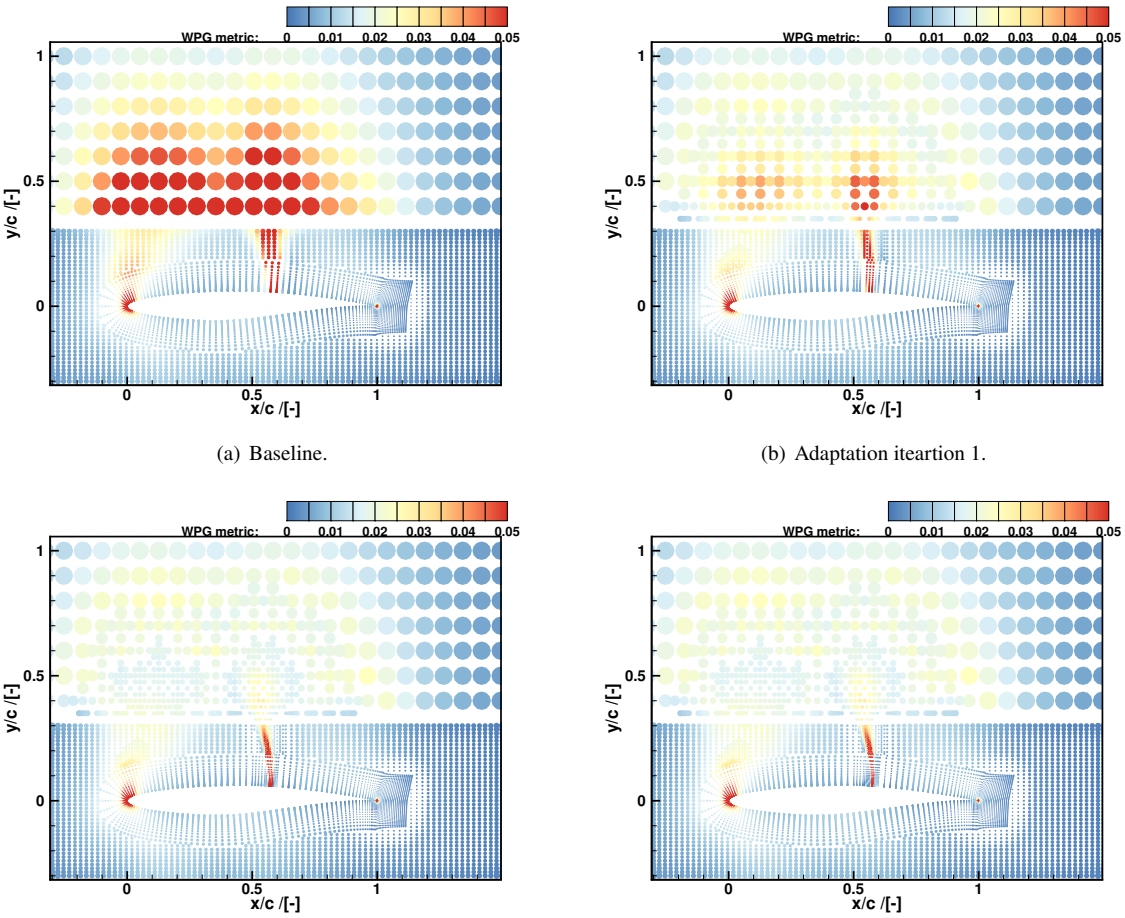
For the baseline case in Figure 20(a), points near the leading edge and the shock wave were tagged due to the high pressure gradient. Points in the coarse background region above the suction peak were also tagged due to the large point sizes. Additional points were inserted around these tagged points in the first adaptation as shown in Figure 20(b). The adaptation closely followed the flow features and also reduced the geometric differences thanks to the weighting by point sizes.

As the adaptation progressed from Figures 20(b) to 20(d), the point sizes became more uniform and smaller, and high ω_p regions shrank to focus more on the flow features. Note that from iteration 3 to iteration 4, only near-body points were refined since off-body points had reached their minimal point size limits. The final aerofoil surface points were increased to 145 from the initial 125.

Flow solutions through the adaptive mesh-free modelling are shown in Figures 21(a) to 21(d). It is clearly shown that the ω_p metric drove the adaptation to closely follow the shock wave. It also eliminated sharp point size transitions in critical regions and avoided excessive irregularity with the help of the point size constraint. The adaptation considerably improved the shock wave resolution in both near-body and off-body regions compared to the initial solution in Figure 21(a).

Figure 22 compares the aerofoil surface pressure coefficient distributions over the adaptation iterations and with experimental data []. The numerical results showed excellent agreement with the experimental data, with small discrepancies due to the inviscid modelling. The adaptation mostly influenced the shock wave resolution at the upper surface. As the adaptation progressed, it is clear that the shock resolution progressively improved. The final resolution in iteration 3 was almost a vertical line.

Figure 23 further presents the convergence history of lift and drag coefficients. The lift and drag predictions showed trends to convergence. The final Cl and Cd values were about 0.714 and 0.01467 and were close to the experimental results [] of 0.731 and 0.0121. The discrepancies were mostly due to the inviscid modelling for the current case.



(a) Baseline.

(b) Adaptation iteartion 1.

(c) Adaptation iteartion 2.

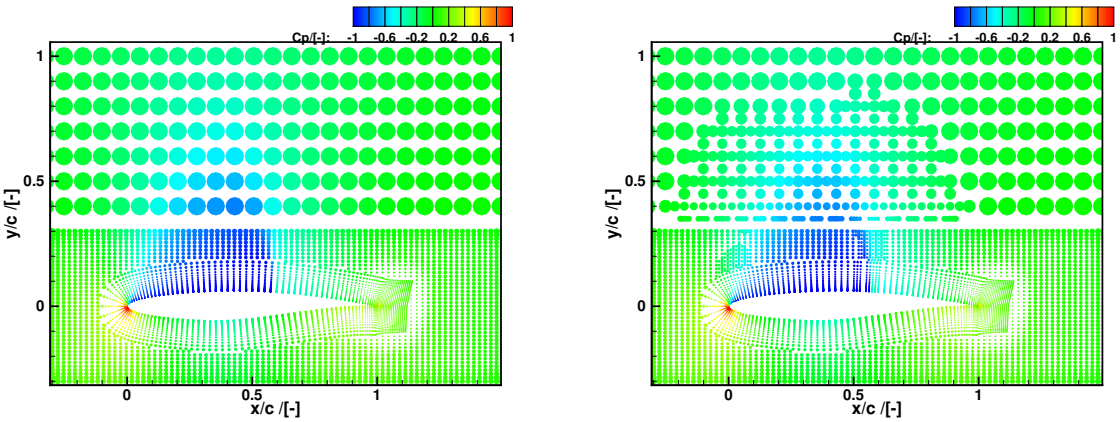
(d) Adaptation iteartion 3.

Fig. 20 The weighted pressure gradient (WPG, ω_p) metric and point distributions for each adaptation iteration. Points with $\omega_p \geq 0.05$ were tagged to be refined in each iteration. Points were scaled by the local point size.

VI. Conclusions

This work presented an investigation into implicit and adaptive mesh-free CFD modelling based on the Generalised Finite Difference Methods. The least-square-based spatial discretisation of the governing flow equations were presented in detail. The implicit temporal discretisation was also derived and presented. This work also explored various adaptation strategies and collocation search methods exploiting the mesh-free flexibility. The following conclusions can be drawn from the present study:

1) The numerical schemes and code implementations were validated and characterised using practical and benchmark test cases. For the ONERA M6 wing at transonic conditions, the mesh-free modelling resolved the multiple shock waves with excellent correlations with experimental data. Simulations of inviscid flows past the circular cylinder using point clouds of varying densities characterised the convergence and accuracy of the current realisation. The spatial accuracy was measured as second-order as intended and was consistent with the least-square basis.
2) This work also assessed the impacts of irregular point distributions and various collocation configurations. Simulations of the cylinder flow suggests that the current mesh-free modelling is sensitive to point distributions near wall boundaries. Regular and shape conforming point distributions are preferred in these regions. The current mesh-free modelling showed strong tolerance for irregular points in off-body regions. Investigations of various collocation configurations suggest that the current mesh-free modelling prefers regular and reciprocal configurations. Irregular and non-reciprocal configurations may lead to non-physical numerical results, due to poor approximations and the lack of flow conservation. However, slight non-reciprocity seemed can be tolerated



(a) Baseline.

(b) Adaptation 1.

(c) Adaptation 2.

(d) Adaptation 3.

Fig. 21 Pressure coefficient contours of the transonic RAE 2822 aerofoil in each adaptation iteration. Points are scaled by local point sizes.

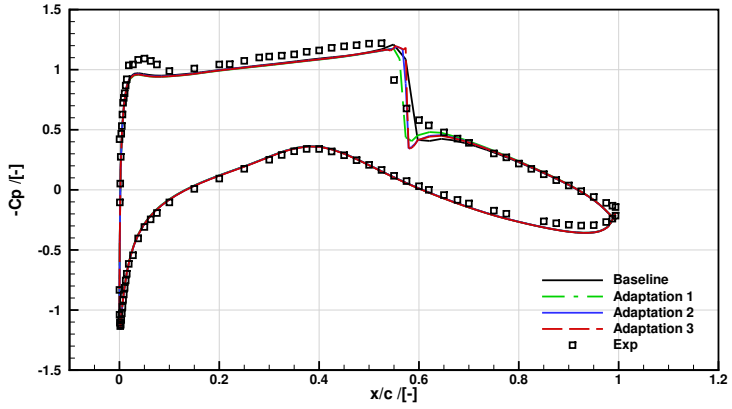


Fig. 22 RAE 2822 pressure coefficient comparisons.

as long as the collocation contains critical points.

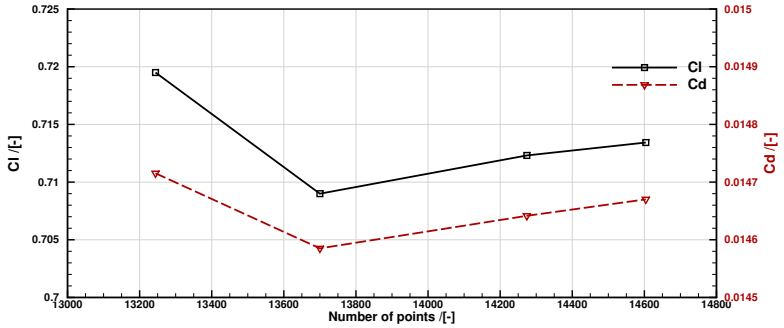


Fig. 23 RAE 2822 lift and drag convergence through the adaptative simulation.

3) This work further investigated adaptive mesh-free modelling with various adaptation strategies. The study examined point refinement mechanisms based on Cartesian points and middle point insertion, and collocation search methods based on distance, optimum selection, and local Voronoi tessellation. For the isentropic vortex simulation, the simple distance-based collocation search was unable to improve the resolution with uniform Cartesian points due to the lack of reciprocity. Selecting optimum members for the collocation managed to converge the solution with slight non-reciprocity. The combination of Cartesian-based refinement and local Voronoi collocation search offered the best performance. However, when replaced with mid-point refinement, the adaptive model showed slightly poorer performance. These results highlighted that regular point distributions and reciprocal collocations should be maintained as much as possible for successful and efficient adaptative mesh-free modelling based on GFD.
4) The mid-point refinement and local Voronoi search strategy was successfully applied to simulations of the classic RAE 2822 aerofoil at transonic conditions. A weighted pressure gradient metric was proposed to drive the adaptation iterations. This metric priorities high pressure gradient regions with large point sizes, thereby smoothing the point distributions while following the flow features during the adaptation. For the transonic aerofoil simulation, the adaptation closely followed the shock wave and reduced sharp point size transitions where necessary. The shock wave resolution was effectively improved through the adaptation.

Overall, these assessments demonstrated the accuracy, convergence, and flexibility for adaptive simulations of GFD-based implicit mesh-free methods. They also highlighted the methods' strong sensitivity to regular point distributions and reciprocal collocation configurations. The Voronoi-based collocation search maintains the reciprocity but effectively makes the methods edge-based and semi-mesh-free. Future work will continue and attempt to reduce the reciprocity dependency in the numerical schemes.

Acknowledgments

The authors would like to acknowledge the Sulis Tier 2 HPC platform hosted by the Scientific Computing Research Technology Platform at the University of Warwick, funded by EPSRC Grant EP/T022108/1 and the HPC Midlands+ consortium.

References

[1] Slotnick, J. P., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D. J., "CFD vision 2030 study: a path to revolutionary computational aerosciences," Tech. rep., 2014.

[2] Batina, J. T., "A gridless Euler/Navier-Stokes solution algorithm for complex two-dimensional applications," Tech. rep., 1992.

[3] Oñate, E., Idelsohn, S., Zienkiewicz, O., and Taylor, R., "A finite point method in computational mechanics. Applications to convective transport and fluid flow," *International journal for numerical methods in engineering*, Vol. 39, No. 22, 1996, pp. 3839–3866.

[4] Katz, A., and Jameson, A., "Edge-based meshless methods for compressible flow simulations," *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008, p. 699.

[5] Kennett, D., Timme, S., Angulo, J., and Badcock, K., "An implicit meshless method for application in computational fluid dynamics," *International Journal for Numerical Methods in Fluids*, Vol. 71, No. 8, 2013, pp. 1007–1028.

[6] Munikrishna, N., and Balakrishnan, N., "Turbulent flow computations on a hybrid cartesian point distribution using meshless solver LSFD-U," *Computers & fluids*, Vol. 40, No. 1, 2011, pp. 118–138.

[7] Osher, S., and Solomon, F., "Upwind difference schemes for hyperbolic systems of conservation laws," *Mathematics of computation*, Vol. 38, No. 158, 1982, pp. 339–374.

[8] Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of computational physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[9] Mavriplis, D. J., "Unstructured-mesh discretizations and solvers for computational aerodynamics," *AIAA journal*, Vol. 46, No. 6, 2008, pp. 1281–1298.

[10] Spalart, P., and Allmaras, S., "A one-equation turbulence model for aerodynamic flows," *30th aerospace sciences meeting and exhibit*, 1992, p. 439.

[11] Jimenez-Garcia, A., and Barakos, G., "Assessment of a high-order MUSCL method for rotor flows," *International Journal for Numerical Methods in Fluids*, Vol. 87, No. 6, 2018, pp. 292–327.