Received: 20 July 2023 | Revised: 11 January 2024 | Accepted: 13 January 2024

DOI: 10.1002/fld.5266

RESEARCH ARTICLE

WILEY

Assessment of implicit adaptive mesh-free CFD modelling

Tao Zhang[1] | George N. Barakos[2]

[1]School of Engineering, University of Leicester, Leicester, UK

[2]CFD Laboratory, School of Engineering, University of Glasgow, Glasgow, UK

Correspondence
George N. Barakos, CFD Laboratory, School of Engineering, University of Glasgow, Glasgow, UK.
Email: george.barakos@glasgow.ac.uk

Funding information
EPSRC, Grant/Award Number: EP/T022108/1

Summary

This work presents details and assesses implicit and adaptive mesh-free CFD modelling approaches, to alleviate laborious mesh generation in modern CFD processes. A weighted-least-squares-based, mesh-free, discretisation scheme was first derived for the compressible RANS equations, and the implicit dual-time stepping was adopted for improved stability and convergence. A novel weight balancing concept was introduced to improve the mesh-free modelling on highly irregular point clouds. Automatic point cloud generations based on strand and level-set points were also discussed. A novel, polar selection approach, was also introduced to establish high-quality point collocations. The spatial accuracy and convergence properties were validated using 2D and 3D benchmark cases. The impact of irregular point clouds and various point collocation search methods were evaluated in detail. The proposed weight balancing and the polar selection approaches were found capable of improving the mesh-free modelling on highly irregular point clouds. The mesh-free flexibility was then exploited for adaptive modelling. Various adaptation strategies were assessed using simulations of an isentropic vortex, combining different point refinement mechanisms and collocation search methods. The mesh-free modelling was then successfully applied to transonic aerofoil simulations with automated point generation. A weighted pressure gradient metric prioritising high gradient regions with large point sizes was introduced to drive the adaptation. The mesh-free adaptation was found to effectively improve the shock resolution. The results highlight the potential of mesh-free methods in alleviating the meshing bottleneck in modern CFD.

KEYWORDS

adaptivity, aerodynamics, implicit, meshfree, Navier-Stokes, validation

1 | INTRODUCTION

The generation and handling of high-quality CFD (Computational Fluid Dynamics) grid is recognised as a major challenge in the CFD 2030 Vision.[1] Meshes provide proper spatial decompositions for the underlying numerical computation in modern CFD workflows. They have direct and substantial impact on the obtained results. Despite recent advancements in algorithms and software, the meshing process is still difficult, laborious, and time-consuming for non-trivial geometries. The process is often iterative, and requires intensive user interactions due to the lack of foreknowledge of the flows to be resolved. The user-dependent meshing also represents a significant source of uncertainty in CFD. Apart from generation, meshes also pose strong restrictions on modelling flexibility. For problems involving large domain motions and/or deformations, meshes must be designed cautiously, and complementary techniques such as sliding or overset methods are necessary.[2] For h-adaptive modelling, adaptation algorithms must pay substantial attention to mesh connectivity and quality.

To alleviate these mesh restrictions, various mesh-free CFD methods have been developed over the years. Mesh-free CFD methods discretise the governing flow equations on discrete points without global connectivity or volumetric elements. This simplifies the meshing process as it is reduced to point generation and distribution. This substantially improves the modelling flexibility for moving/deforming domains thanks to the point-based discretisation. Particularly, h- and p-adaptive modelling is made extremely easy without connectivity and quality restrictions, either through re-distributing or inserting point clouds.

Current mesh-free CFD methods can be roughly categorised according to the underlying governing equation forms. One class of mesh-free methods was derived from the weak form of the governing equations. These include the Diffusive Element (DE) method,[3] the element-free Galerkin (EFG) method,[4] and the meshless local Petrov-Galerkin (MLPG) method.[5] The principle of these methods is to construct test/trial functions through approaches, for example, moving least-squares or Partition of Unity[6] interpolation, thereby alleviating the need for mesh connectivity. Since these methods are derived from the integral weak form, they have better properties preserving flow conservation. However, because of the underlying weak form, many of these methods also need a global, or local background mesh, for integration. Another factor that hinders their wider applications in CFD is their complexity, which requires considerable effort to adapt from existing algorithms.

Another class of mesh-free methods focuses on the strong form of the governing equations. The principle of this class of methods is to construct differential operators upon a local point collocation or stencil. The collocation-based mesh-free methods are often termed Finite Point Methods (FPM)[7] or Differential Quadrature (DQ) methods,[8] and often coincide with the Generalised Finite Difference Methods (GFDM). The FPM has various choices for basis functions (e.g., polynomial basis[9] and Taylor expansions[10]) and approximation approaches (e.g., weighted least squares[11,12] and moving least squares). The approximation can also be realised through radial basis functions (RBF),[8,13] leading to the RBF-FD approaches. These methods are straightforward and share remarkable similarities with well-established, finite difference methods in CFD. In many cases, the differential operators can be grouped as virtual volume and edge-based terms,[14] for the final schemes to resemble popular finite volume methods. This is important, as it reduces the effort to incorporate such methods in well-established CFD algorithms. In recent years, attempts have also been made to bridge mesh-free FEM and FPM, leading to the Mesh-free Finite Volume concept.[15] However, these collocation-based methods share the challenge of preserving flow conservation when applied to CFD because of the underlying strong form. In this respect, the Smoothed Particle Hydrodynamics (SPH)[16,17] method is a special alternative that preserves flow conservation. It adopts a Lagrangian description and allows point motions and interactions. It treats points as particles with mass and momentum, but it often suffers from lower modelling accuracy and difficulties in boundary representations.

For aerodynamic modelling, earlier studies[9–11,18,19] have demonstrated the potential of collocation-based mesh-free methods in handling complex geometries and flow conditions. However, to promote these methods within the CFD community and fulfil their potential, significant research and development on algorithms, and on complementary techniques, for example, point cloud generation and adaptation, collocation/stencil search, implicit time marching, and especially the handling of flow conservation, are still necessary. The lack of flow conservation may significantly deteriorate the modelling fidelity and lead to unpredictable errors.

This paper presents and verifies novel methodology developments for collocation-based CFD modelling with implicit time marching, and adaptive modelling. A least-squares-based, mesh-free scheme, with implicit temporal discretisation was first derived and verified using benchmark cases. Novel concepts in automatic point cloud generation, collocation search, and weight balancing were proposed. The study then investigated the impacts of irregular point distributions. Various adaptive modelling strategies combining different point refinement mechanisms and collocation search methods were then evaluated. The adaptation performance was assessed through simulations of the isentropic vortex. The adaptation strategy was then applied to transonic aerofoil simulations and was found to effectively improve the solution resolution.

2 | NUMERICAL METHODOLOGIES

2.1 | Collocation-based mesh-free discretisation

The principle of many collocation-based, mesh-free discretisation schemes[7,20] is to construct differential operators using shape functions upon discrete points. Such approach is often termed as the Differential Quadrature (DQ) method,[8] and shares substantial similarities with the Generalised Finite Difference Method (GFDM). Given an arbitrary and sufficiently smooth function $\phi(\mathbf{x})$ with $\mathbf{x} = (x, y, z)$ in a Cartesian system, known at a set of n_i discrete points centring at point i, $\mathbf{\Phi}_i = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), ..., \phi(\mathbf{x}_{n_i})]^T$ in a local collocation ($j = 1, 2, ...n$), a local approximation can be constructed using shape functions $\mathbf{N_i}(\mathbf{x}) = [N_{i,1}, N_{i,2}, ..., N_{i,n_i}]$, such as:

$$\hat{\phi}(\mathbf{x}) = \mathbf{N_i}(\mathbf{x})_{(1\times n_i)}\mathbf{\Phi}_{i(n_i\times 1)}. \tag{1}$$

The spatial derivatives of ϕ can be estimated from the approximation as:

$$\frac{\partial\phi(\mathbf{x})}{\partial\mathbf{x}} \approx \frac{\partial\hat{\phi}(\mathbf{x})}{\partial\mathbf{x}} = \frac{\partial\mathbf{N_i}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{\Phi}_i, \tag{2}$$

where $\frac{\partial\mathbf{N_i}(\mathbf{x})}{\partial\mathbf{x}}$ is the spatial derivative of the shape functions. This collocation-based differential operation can be used for the spatial discretisation of PDEs in a manner similar to the classic finite differences scheme.

This work focuses on the compressible Reynolds Averaged Navier-Stokes (RANS) equations for aerodynamic simulations. The inviscid part of the governing flow equations can be written in strong form in a Cartesian frame as:

$$\frac{\partial\mathbf{U}}{\partial t} + \nabla\cdot\mathbf{F} = 0, \tag{3}$$

where $\mathbf{U}$ is the vector of conservative flow variables, and $\mathbf{F}$ is the convective flux term. Assuming a local point collocation centring at point i, with the local support (i.e., all points but the collocation centre i) denoted by the set $\mathbf{S}_i$ ($i \notin \mathbf{S}_i$), the flow equation can be discretised at the centre point i by approximating the flux derivatives:

$$\frac{d\mathbf{U}_i}{dt} = -\sum_{j\in\mathbf{S}_i}\sum_{\nu=1}^{d}\frac{\partial N_{i,j}}{\partial x^\nu}(\mathbf{x}_j)\mathbf{F}_j^\nu, \tag{4}$$

where the superscript ν denotes the number of spatial dimensions from 1 up to $d = 3$. $j \in \mathbf{S}_i$ are called supporting points for the collocation centre point i, and their Cartesian coordinates are $\mathbf{x}_j$. Due to the hyperbolic nature of the equations, the discretisation in Equation (4) needs further numerical dissipation or upwind schemes for stability. The upwind schemes can be incorporated by evaluating the flux terms at the middle point of i and j, which are denoted as $j - \frac{1}{2}$:

$$\frac{d\mathbf{U}_i}{dt} = -\sum_{j\in\mathbf{S}_i}\sum_{\nu=1}^{d}\frac{\partial N_{i,j}}{\partial x^\nu}(\mathbf{x}_{j-\frac{1}{2}})\mathbf{F}_{j-\frac{1}{2}}^\nu, \tag{5}$$

where the flux term $\mathbf{F}_{j-\frac{1}{2}}$ are reconstructed at the middle point, and Riemann solvers can be used to account for the upwind properties. In this work, Osher's scheme[21] was adopted, where the mid-point flux was reconstructed as:

$$\mathbf{F}_{j-\frac{1}{2}} = \frac{1}{2}(\mathbf{f}(\mathbf{p}_L) + \mathbf{f}(\mathbf{p}_R)) - \frac{1}{2}|\tilde{\mathbf{A}}(\mathbf{p}_L, \mathbf{p}_R)|(\mathbf{p}_R - \mathbf{p}_L), \tag{6}$$

where $\tilde{\mathbf{A}}$ is the Roe-averaged Jacobian matrix.[22] $\mathbf{p}_L$ and $\mathbf{p}_R$ are the primitive variable vectors on the left- and right-hand sides of the middle point, respectively. There are various ways to determine the middle point values. A first-order reconstruction would simply take the values of the two neighbouring points i and j, while higher-order limited, or unlimited reconstruction can also be incorporated here.

Note that in Equation (5), the flux term between points i and j is not always reciprocal, as point i may not necessarily be in the point cloud centring at point j. This reciprocity is a necessary condition for the flow conservation.[20] To alleviate this problem, this work has enforced reciprocal contributions between all points, that is, adding i to j's support if it was not in. This additional restriction drives the present mesh-free scheme towards 'edge-based' schemes. Nevertheless, this deteriorates little the mesh-free flexibility, as the collocation or stencil edges are highly flexible and diverse (including crossover connections[20]), and these connections are associated with weights that can adjust the contributions.

It should also be noted that Equation (5) resembles the finite volume scheme (FVM) in the local point cloud. If we take the term $\frac{\partial N_{i,j}}{\partial x^\nu}\left(x^\nu_{j-\frac{1}{2}}\right)$ as a virtual edge term, Equation (5) can be regarded as a virtual FVM discretisation in a cell centring at point i and with $j \in \mathbf{S}_i$ virtual edges or faces. This similarity with FVM schemes is an important advantage for numerical implementations, as established FVM methods and codes can be easily transformed to mesh-free versions with minor modifications. The realisation of hybrid mesh-based/mesh-free methods is also feasible.

Viscous terms and turbulence closures are also included following the same approximation. For the viscous terms, the gradient variables are first constructed using the following modifications[23] to avoid the odd-even node decoupling:

$$\nabla\phi_{ij} = \frac{1}{2}(\nabla\phi_i + \nabla\phi_j) - \left[\frac{1}{2}(\nabla\phi_i + \nabla\phi_j)\cdot\frac{\delta\mathbf{x}}{|\delta\mathbf{x}|} - \frac{\phi_j - \phi_i}{|\delta\mathbf{x}|}\right]\cdot\frac{\delta\mathbf{x}}{|\delta\mathbf{x}|}, \tag{7}$$

where $\delta\mathbf{x} = (\mathbf{x}_j - \mathbf{x}_i)$ is the distance between the two points. i is the collocation centre point, and $j \in \mathbf{S}_i$ are the support points. Note that the least-square approach can also be used to compute gradients and higher-order derivatives, but this requires higher-order basis functions. The current work adopted the gradient computation in Equation (7) which is popular in unstructured FVM, and the derivatives of the viscous flux are then approximated using the least-square approach. This choice is compatible with low-order basis functions. The resulted viscous flux is computed by averaging at the midpoint of i and j.

RANS turbulence closures are also included in the discretisation as additional source terms. The one-equation Spalart-Allmaras (SA) model[24] was used in the present work. The convective and diffusive terms were evaluated following the same procedures described for the inviscid and viscous terms.

2.2 | Weighted least-squares approximations

Collocation-based, mesh-free, discretisation has various options for approximation and basis functions. Nevertheless, different realisations face similar challenges regarding collocation selection, flow conservation, implicit schemes, adaptation strategies, and so forth. The current work focuses on the least-squares-based scheme with polynomial basis as a first step, in assessing mesh-free performance and adaptive simulations.

For the arbitrary field function $\phi(\mathbf{x})$ in Equation (1), the approximation function $\hat{\phi}(\mathbf{x})$ can be constructed using polynomials:

$$\hat{\phi}(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T\alpha, \tag{8}$$

where $\mathbf{b}(\mathbf{x})$ is an $m \times 1$ vector of the polynomial basis with m terms and α is an $m \times 1$ vector of coefficients. For a local collocation centring at point i with in total n_i points (including the central point), the coefficient vector α can be determined through the following linear combination:

$$\mathbf{\Phi}_{i(n_i\times1)} = \mathbf{B}_{(n_i\times m)}\alpha_{(m\times1)} \tag{9}$$

where $\mathbf{\Phi}_i = \{\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_{n_i})\}^T$ is an $n_i \times 1$ vector of known function values at all n_i collocation points. α is an $m \times 1$ coefficient vector corresponding to polynomial basis with m terms. The basis matrix $\mathbf{B}$ is defined as:

$$\mathbf{B} = \left\{\mathbf{b}(\mathbf{x}_1), \mathbf{b}(\mathbf{x}_2), \dots, \mathbf{b}(\mathbf{x}_{n_i})\right\}^T, \tag{10}$$

where $\mathbf{b}(\mathbf{x}_j)$ $(j = 1, 2, \dots, n_i)$ is an $m \times 1$ vector of the m-term polynomial basis values at point $\mathbf{x}_j$. The current work adopted the complete first-order polynomials as the basis. The linear system of Equation (9) is of the size $n_i \times m$ and is not necessarily square. We assume that $n_i \geq m$ so that the system is not under-determined and has at least one solution.

Assuming the point collocation has been carefully designed to avoid singularities, we can use the weighted least-squares approach to determine the best fit with the minimum L^2 norm, and solve for the coefficients α:

$$\alpha = (\mathbf{B}^{\mathbf{T}}\mathbf{W}\mathbf{B})^{-\mathbf{1}}\mathbf{B}^{\mathbf{T}}\mathbf{W}\mathbf{\Phi}_i = \mathbf{C}\mathbf{\Phi}_i, \tag{11}$$

where $\mathbf{C} = (\mathbf{B}^{\mathbf{T}}\mathbf{W}\mathbf{B})^{-\mathbf{1}}\mathbf{B}^{\mathbf{T}}\mathbf{W}$ is an $m \times n_i$ coefficient matrix. $\phi_i = \hat{\phi} = \{\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_{n_i})\}^T$ is the $n_i \times 1$ vector of known values at all n_i points in this local point collocation centring at i. $\mathbf{W}$ is an $n_i \times n_i$ diagonal weighting matrix adjusting contributions from each cloud point.

The computational cost of the matrix inversion is negligible due to its small size, and the computation is parallel and static if the point collocations remain the same during the modelling. However, the point cloud topology and the weighting should be designed with caution, to guarantee the inversion operation, and good approximation accuracy. Collocation search approaches and weighting functions are hence presented and investigated in later sections. Inserting Equations (11) into (8), the function $\phi(x)$ is approximated by $\hat{\phi}(x)$ as

$$\hat{\phi}(\mathbf{x}) = \mathbf{b}^T(\mathbf{x})\alpha = b^T(\mathbf{x})\mathbf{C}\mathbf{\Phi}_i = \mathbf{N}_i(\mathbf{x})\mathbf{\Phi}_i, \tag{12}$$

where $\mathbf{N}_i(\mathbf{x}) = \mathbf{b}^T(\mathbf{x})\mathbf{C}$ is regarded as the $1 \times n_i$ local shape function vector. The partial derivatives of $\phi(\mathbf{x})$ can be determined through the approximation:

$$\frac{\partial\phi(\mathbf{x})}{\partial\mathbf{x}} \approx \frac{\partial\hat{\phi}(\mathbf{x})}{\partial\mathbf{x}} = \frac{\partial\mathbf{N}_i(\mathbf{x})}{\partial\mathbf{x}}\mathbf{\Phi}_i. \tag{13}$$

Partial derivatives of $\phi(\mathbf{x})$ now depend on the partial derivatives of the shape function $\mathbf{N_i}(\mathbf{x})$. The partial derivatives of the shape functions are written as follows:

$$\frac{\partial\mathbf{N}_i(\mathbf{x})}{\partial\mathbf{x}} = \frac{\partial\mathbf{b}^T(\mathbf{x})}{\partial\mathbf{x}}\mathbf{C}, \tag{14}$$

and $\frac{\partial\mathbf{b}^T(\mathbf{x})}{\partial\mathbf{x}}$ are the partial derivatives of the basis polynomial. Note that the matrix $\mathbf{C}$ is kept constant while evaluating the derivatives. The term $\frac{\partial\mathbf{N}_i(\mathbf{x})}{\partial\mathbf{x}}$ can be regarded as a difference operator, constructed using the least-square principle, based on the basis functions and geometric information of the local point collocation.

2.3 | Weighting functions and weight balancing

For the WLS approximation on a point collocation, the weighting function w adjusts the contribution of each point (while the weighting matrix $\mathbf{W}$ contains the weight values at collocation points.). Typically, w is a positive and radially decaying function depending on the distance to the collocation centre, and w is equal to zero outside the local collocation. This is to maintain the compactness of the support and the locality of the approximation. The weighting function affects the conditioning and accuracy of the WLS approximation as studied by Ortega et al.[25] A rapidly decaying weighting function may improve approximation accuracy but compromise the conditioning and vice versa. Typical choices for the weighting function include Gaussian, inverse-distance, and polynomial functions. The present work adopted a simple second-order inverse-distance weighting. The collocation central point was given a weight of 1.0, while all support points share a total weight of 1.0 according to their inverse squared Euclidean distance to the centre.

However, for irregular point collocations, the least-square approach gives biased approximation towards wherever sample points are clustered, since they contribute more to the overall error. Typical radial-based weighting functions are insufficient to alleviate this situation. A 2D example is presented in Figure 1A, where all support points are of equal distance to the centre, but are unevenly distributed along the angular directions. For a symmetric function such as $g(x, y) = e^{-(x^2+y^2)}$ (the dashed contour line in Figure 1A and the grey wireframe in Figure 1B), a first-order WLS approximation using this collocation and radial-based weighting leads to biased approximation towards the $+x$ direction, as shown by the red plane and projection in Figure 1B. This leads to unintended biased differencing operators for the flow modelling, induced by the point clustering.

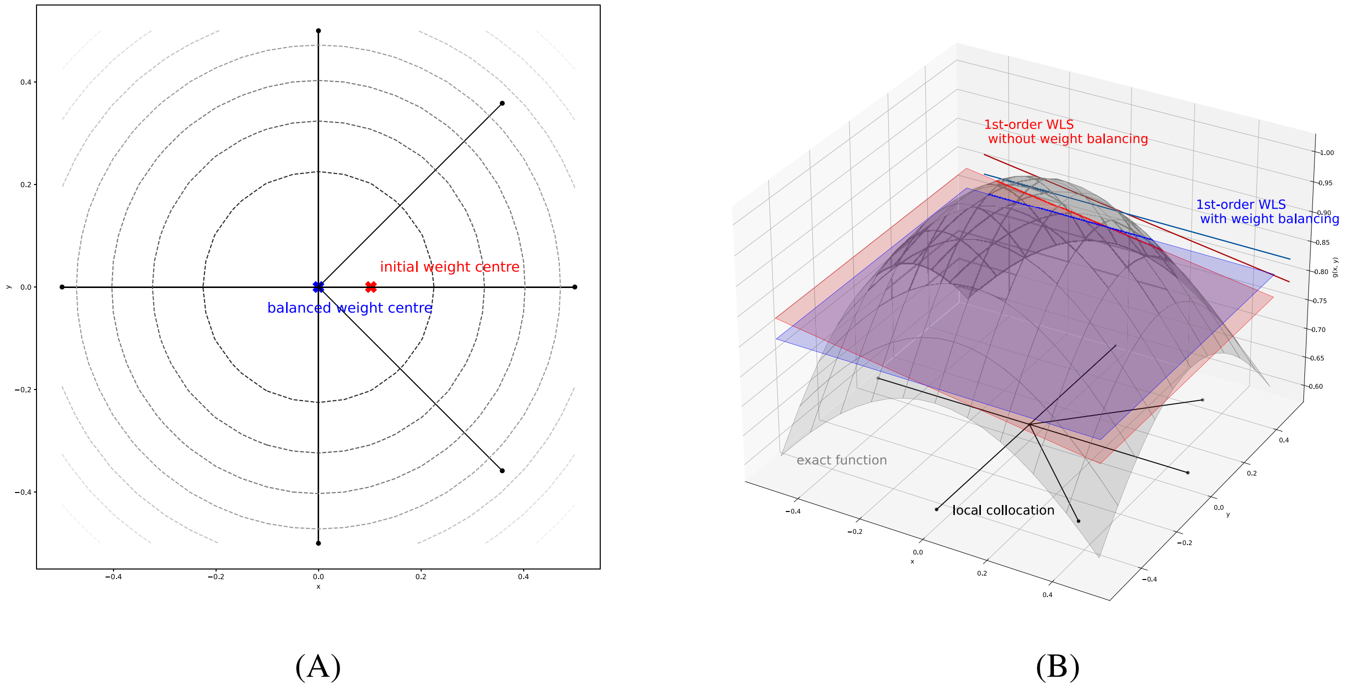


FIGURE 1 WLS approximations using irregular point distributions and the correction effect of the weight balancing. (A) A collocation with irregular angular point distributions. The dashed lines denote the example symmetric function $g(x,y) = e^{-(x^2+y^2)}$ to be approximated using WLS and this collocation. (B) First-order WLS approximations of a symmetric function with and without weight balancing.

This study proposes a weight balancing approach to alleviate this problem. The idea is to adjust the weighting of each support point, so that their weight centre falls upon the central point (i.e., the local coordinate origin), as shown in Figure 1A. The initial weight centre $\mathbf{x}_{c0}$ is defined as:

$$\mathbf{x}_{c0} = \frac{\sum(w_j \mathbf{x}_j)}{\sum w_j}, \tag{15}$$

where $j \in \mathbf{S}_i$ is the support points of the local collocation $\mathbf{S}_i$ centring at point i ($i \notin \mathbf{S}_i$). $\mathbf{x}_j$ is the local Cartesian coordinate (origin at point i) of the support point j and w_j is its weight.

By introducing a small change δw_j to each support point j, we aim to balance the weight centre at the central point i:

$$\mathbf{0} = \frac{\sum((w_j + \delta w_j)\mathbf{x}_j)}{\sum(w_j + \delta w_j)}, \tag{16}$$

where $(w_j + \delta w_j) > 0$ and $\sum \delta w_j = 0$. If the weight is balanced at the centre, clustered points must have their weights reduced, and added to other points. This reduces their contributions in the overall squared error and alleviates the biased approximation. This is illustrated in Figure 1B with the blue plane and projection. The weight-balanced approximation now recovers the symmetry despite the irregular and clustered point distributions. This avoids the unintended biased differencing in the flow modelling induced by point distributions.

Equation (16) represent a convex combination of all support points, and it requires the centre point i to be within the convex hull composed by all support points j to ensure at least one solution. This can be guaranteed while generating the point clouds and establishing collocations, and this is also beneficial for the quality of least-square approximations. Equation (16) can then be solved as an optimisation problem with equality and inequality constraints, but this can be expensive as it has to be done for every collocation.

A simplified solution is to relax the constraint $\sum \delta w_j = 0$ to $\delta w_j \geq 0$, as this ensures that $(w_j + \delta w_j) > 0$ and the total weight can be scaled back to 1. Substituting Equations (15) into (16), we have:

$$\sum(\delta w_j \mathbf{x}_j) = -\mathbf{x}_{c0} \sum w_j, \tag{17}$$

where $\sum w_j$ is constant, and nominally $\sum w_j = 1$. Equation (17) represents a linear decomposition of the vector $-\mathbf{x}_{c0}$ using vectors $\mathbf{x}_j$ in an d-dimensional Euclidean space. A simple solution is to find up to d linearly independent $\mathbf{x}_{j1}$ to $\mathbf{x}_{jd}$ that lead to positive δw_{j1} to δw_{jd} values, and set the rest δw_j as zero.

In 2D, this means decomposing $-\mathbf{x}_{c0}$ as

$$-\mathbf{x}_{c0} = \delta w_{j1} \mathbf{x}_{j1} + \delta w_{j2} \mathbf{x}_{j2}, \tag{18}$$

where vectors $\mathbf{x}_{j1}$ and $\mathbf{x}_{j2}$ bracket $-\mathbf{x}_{c0}$ with an angle less than 180 degrees. The extra positive weights δw_{j1} and δw_{j2} are then added to points $j1$ and $j2$. In 3D, three vectors that are not co-planar, and contain $-\mathbf{x}_{c0}$ within the cone they form are needed. These are feasible as long as the collocation centre i is within the convex hull of its support points, and the computations are simple using a polar or spherical coordinate system. If no such vectors can be found, the point collocation will be revised.

2.4 | Implicit temporal discretisation

The temporal discretisation in this work adopted implicit schemes which are more efficient and stable, but are rarely used for mesh-free schemes due to complexity. The classic dual time-stepping concept[26] was used here. Taking the RHS of Equation (5) as a residual vector $-\mathbf{R}$, and applying an implicit temporal discretisation to the ODE, we have:

$$D_t(\mathbf{U}) = -\mathbf{R}^{n+1}, \tag{19}$$

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

$$\mathbf{R}^* = D_t(\mathbf{U}) + \mathbf{R}^{n+1} = 0, \tag{20}$$

where $\mathbf{R}^*$ is a pseudo residual term. By adding a temporal term, and applying another temporal discretisation D_τ about the pseudo time τ, a steady flow problem can be formulated:

$$D_\tau(\mathbf{U}) = -\mathbf{R}^{*(l+1)}, \tag{21}$$

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

$$\mathbf{R}^{*(l+1)} \approx \mathbf{R}^{*l} + \frac{\partial \mathbf{R}^*}{\partial \mathbf{p}}(\mathbf{p}^l)\Delta\mathbf{p}, \tag{22}$$

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

$$D_\tau(\mathbf{U}) = -\mathbf{R}^{*l} - \frac{\partial \mathbf{R}^*}{\partial \mathbf{p}}(\mathbf{p}^l)\Delta\mathbf{p}, \tag{23}$$

where the operator D_τ can take simple forms, such as the backward Euler, since this is essentially a steady problem, and the term $\Delta\mathbf{p}$ can be easily solved from a large linear system. A particular advantage here is that acceleration techniques can be adopted to accelerate the solution process. In this work, the local time step approach is used. When handling essentially steady problems, the temporal discretisation can directly follow Equation (20) by dropping the first level of discretisation in Equation (19).

For the local collocation centring at point i, the major difficulty here is the determination of the Jacobian matrix $\frac{\partial \mathbf{R}_i}{\partial \mathbf{p}_j}$ at each supporting point $j \in \mathbf{S}_i$, which can be expanded as:

$$\frac{\partial \mathbf{R}_i}{\partial \mathbf{p}_j} = \sum_{j \in \mathbf{S}_i} \sum_{\nu=1}^{d} \frac{\partial N_{i,j}}{\partial x^\nu} \left(x^\nu_{j-\frac{1}{2}} \right) \frac{\partial \mathbf{F}^\nu_{j-\frac{1}{2}}}{\partial \mathbf{p}_j}. \tag{24}$$

We note 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 function, the reconstruction, and the Riemann solver. In this work, the eventual large linear systems were solved using conjugate gradient approaches with OpenMP parallel computing.

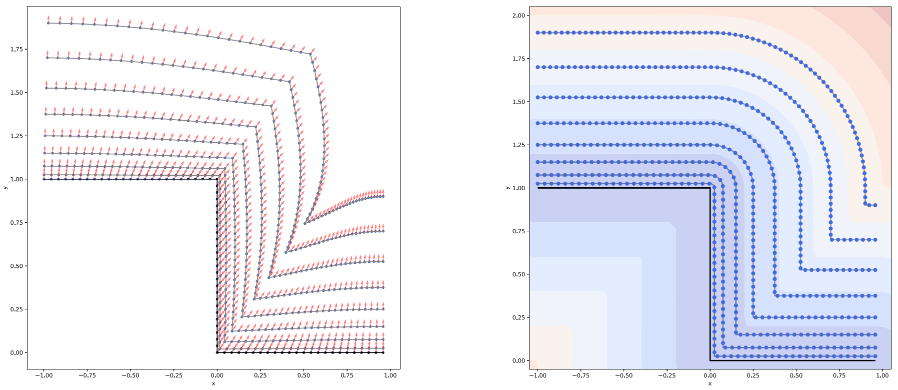
2.5 | Near-body point cloud generation and boundary conditions

The generation of suitable point clouds for the mesh-free modelling is a non-trivial task, as the point distributions should respect flow physics. Nevertheless, due to the flexibility of mesh-free schemes, the point cloud generation can be considerably easier than manual mesh generation. A detailed review of point generation methods can be found in Reference 27.

The present work has adopted a simple and automatic approach for the point cloud generation. In the near-body or boundary layer region, points were popped out of wall boundaries following the surface normal vectors. This is to ensure normal point distributions for boundary layer flows. The near-body points are then blended in off-body background point clouds that have regular point distributions. The blending is naturally compatible with the mesh-free scheme, as it has much relaxed requirement on connectivity and supports highly irregular points distributions.

The near-body point cloud generation is similar to the strand grid generation concept,[28] as illustrated in Figures 2A. However, since the mesh-free scheme has no strict requirement for structured mesh connectivity, it is hence much more flexible, and does not suffer from problems such as distorted cells or cell intersection around corners.

This approach requires the input of the boundary geometry, represented by a set of surface points and the surface normal vectors at each point. Such information can be easily extracted by most modern CAD system, through conversions of geometries to for example, STL format. At each surface point, an array of points are then generated along the surface normal direction. The points along this ray may follow certain distribution laws according to the boundary type and flow physics, for example, exponential distributions near wall boundaries and even/hyperbolic distributions in the far field. The current work also introduced a multi-layer smoothing technique, to alleviate abrupt geometry transitions, as illustrated in Figure 2A for a square shape. Normal vectors in each layer are smoothed by averaging with nearby points, and new point layers are then grown following the smoothed vectors. For convex corners, the point distribution becomes less ideal in outer layers as shown in Figure 2A, but these are still compatible with the mesh-free modelling.



(A) (B)

FIGURE 2 Demonstration of near-body point cloud generation. (A) Near-body strand point generation for a square with multi-layer normal vector smoothing. (B) Near-body point generation for a square based on minimal wall distance (or level-set) functions.

This work also adopted an additional approach to improve the point distribution in the outer layers, as illustrated in Figure 2B. This approach computes the signed-distance field (or the level-set function), as shown by the contours in Figure 2B, and places points on contour lines or iso-surfaces. It gives improved point distributions near corners. Since the mesh-free scheme has no strict restriction on structured connectivity, the point numbers and distributions, on each layer are flexible. This is similar to the advancing front approach for unstructured mesh generation, but without restrictions on triangular elements.

Further examples for more complex shapes, for example, the slat and main elements of the 30P30N aerofoil, are shown in Figure 3A,B. Note that signed-distance fields are also recommended inside boundaries to allow for the generation of ghost points, which will be used to enforce boundary conditions. The computation of the signed-distance distribution is not a waste as it is required by many RANS closures as wall distances. The drawback, of this approach is that it does not always guarantee wall-normal point distributions in the boundary layer.

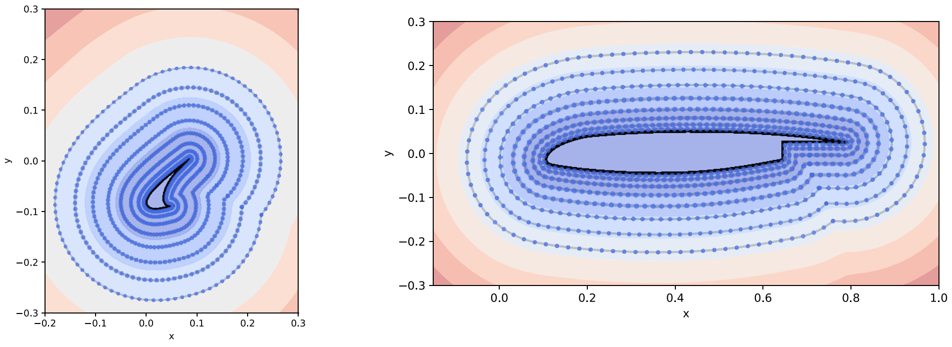
It is hence recommended to combine the strand point generation in near-wall regions, and the level-set point distribution in the outer region. Once the near-body points are generated, the off-body background point generation and blending are easy. The point generation process is fully automatic, given the input geometry and parameters, for example, distribution laws and boundary types.

The near-body point generation process, is also compatible with adaptation. For near-body adaptation, for example, in the boundary layer, new surface points are interpolated through local geometry approximations, and new point arrays are introduced where necessary. For off-body adaption, this approach simply re-blends the new point clouds.

Boundary conditions in the present work, are implemented through halo points, as illustrated in Figure 4. Note that in Figure 4 the surface or boundary outer norm points inside the computational domain, while the ghost points are outside the computational domain. These points can be generated in many ways, for example, by mirroring from the first or more layers of points outside the boundary, or by placing point inside the boundary according to wall distance fields. These points have prescribed or reflected values for implementations of Dirichlet or Neumann boundary conditions, respectively. WLS approximations were also performed for these boundary point collocations with halo points, hence the accuracy was not compromised at boundaries. Note that the halo points also participated in the collocation search process to ensure high-quality point collocations at boundaries.

2.6 | Collocation search approaches

The collocation search is a process to choose suitable support points, or stencil, for a given point centre. The choice has direct impact on the local WLS approximation accuracy, and should respect the flow physics. Poorly selected points may result in singular matrices in Equation (11). One may argue that this process is the same as establishing mesh connectivity upon a given cloud of points. Indeed, the current collocation-based, mesh-free method can adopt existing mesh connectivity as collocations, but this is not necessary. For the current mesh-free scheme, the collocations are for the WLS approximations and the connectivity is for flux exchange. It allows for more flexible choices, for example, cross-over



(A) (B)

FIGURE 3 Near-body point cloud generation for the 30P30N aerofoil using level-set approaches. Contours represent the minimal wall distance field. (A) 30P30N aerofoil slat element. (B) 30P30N aerofoil main element.

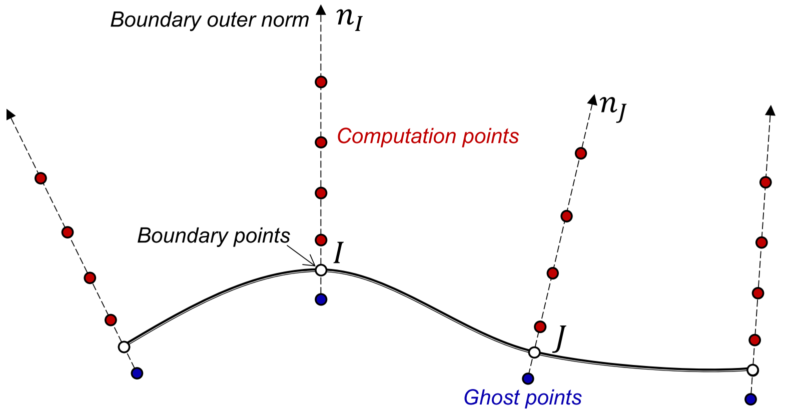


FIGURE 4 Boundary condition implementation and point cloud generation.

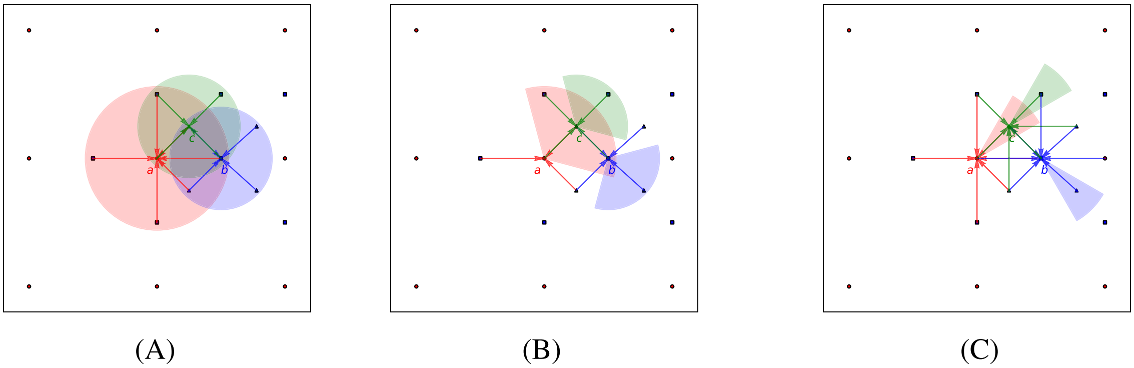


FIGURE 5 Different collocation search methods. The circles denote the search radius. The lines denote connections between the collocation centre and its members, with arrows pointing to the centre. (A) Distance-based search. (B) Distance-based search with polar selection using a large polar interval of $\delta\theta = \pm\pi/3$. (C) Distance-based search with polar selection using a small polar interval of $\delta\theta = \pm\pi/12$.

connections,[20] which would not be tolerated in meshes. Moreover, all points within a collocation are associated with weights, as described in Section 2.3, which reflect and adjust their contributions. These weights provide extra flexibility for mesh-free modelling.

To establish a local point collocation, the easiest approach is to simply include all points within a certain distance, that is, the distance-based search as shown in Figure 5A. However, this raises two important problems for the local WLS approximation. The first problem is related to the compactness, as it is difficult to determine the search radius. The second problem is that the collocation quality cannot be guaranteed and may lead to ill-conditioned WLS approximation. Additional criteria and mechanisms are necessary to select certain points with the distance for a high-quality collocation. This section presents two selection approaches, based on polar selection and approximate/accurate Voronoi tessellation, respectively.

2.6.1 | Polar selection

Good local WLS approximations prefer comprehensive information from all directions within a compact region. For a group of points within a certain distance, this can be achieved by selecting points that are close to the centre and are evenly distributed along the polar/spherical directions. This is the basic idea of the polar selection approach illustrated in Figure 5B,C.

To establish the point collocation centring at point a in Figure 5B, all points within the search radius are initially included as candidates and a polar system is built centring at a. Then, starting from one candidate, for example, point c, a polar region of $\theta_c \pm \delta\theta$ is constructed as denoted by the red wedge in Figure 5B (θ_c is the angular coordinate of point c

and $\delta\theta$ is a pre-defined polar interval). All points within this polar region are then examined, and the closest point to a is selected as a collocation member, while others are removed from the candidate pool. In the red polar region of Figure 5B, point c is selected while the other two points are removed. This process is repeated until no candidate is left. In 3D, this process is largely similar, but the 2D polar region is replaced by a 3D cone region in a spherical system.

This polar selection process ensures that only the closest point will be selected within a polar interval, thereby promoting uniform point distributions along the angular directions. The number of collocation members can be controlled by adjusting the polar interval $\delta\theta$ and search radius. For instance, in Figure 5B, with $\delta\theta = \pi/3$, collocation a has three support members from all directions. While in Figure 5C, when the search radius is increased and the polar interval is reduced to $\delta\theta = \pi/6$, collocation a has six members from all directions. This is an important property as higher-order WLS approximations will require an increasing number of points. Note that the crossover of several edges in Figure 5C is allowed and does not always lead to conservation issues.[20] A final correction is to ensure that all collocation connections are reciprocal, which is vital for the flow conservation through flux exchange. This is easily done by amending any single-ended connections.

2.6.2 | Approximate/accurate Voronoi selection

Voronoi tessellation[29] is a domain decomposition method that gives the dual graph of Delaunay triangulation, which has been widely used in mesh generation. 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, neighbouring Voronoi cells can be used as a collocation for mesh-free modelling. An accurate Voronoi tessellation ensures the compactness and reciprocity of the collocation results, and this property has been used by Suchde et al.[30] to construct conservative mesh-free operators.

Note that for the present mesh-free scheme, the Voronoi tessellation is only used as a tool to identify neighbouring points, as shown in Figure 6A, while extra information such as cell volume or edge/face sizes is unnecessary. This suggests that approximate Voronoi tessellation can be adopted to reduce the computational complexity and costs.

This work has adopted a simple pixel- or lattice-based approximate Voronoi tessellation, as illustrated in Figure 6B. For a given initial local point cloud based on distance, the space is first divided into a finite amount of lattices. Each lattice is then assigned a 'colour' according to the nearest cloud point. An approximate Voronoi tessellation, consisting of discrete pixels or lattices, is obtained once all lattices have been coloured. The neighbouring information for the collocation centre is obtained by inspecting the central Voronoi cell. Compared to the typical accurate Voronoi approach, this method provides straightforward neighbouring information for the mesh-free collocation. Using the pre-defined coarse lattices (typically 10-by-10 in 2D), this approach reduces the computational complexity from the typical $O(nlogn)$ of an accurate

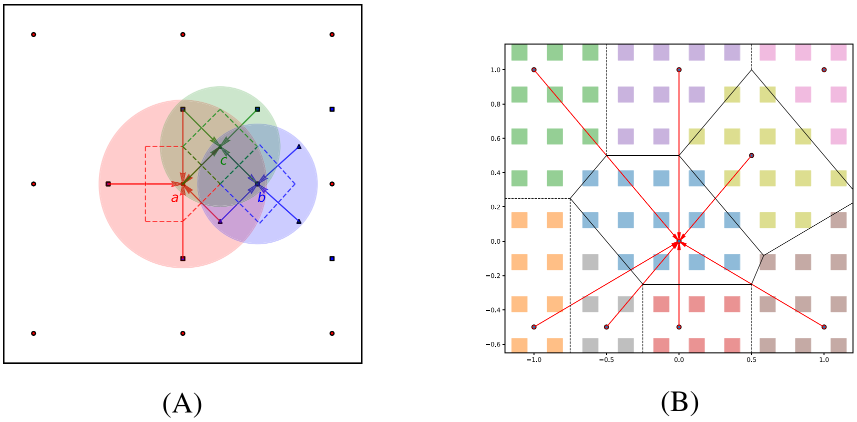


FIGURE 6 Collocation selection approach based on accurate and approximate Voronoi tessellations. (A) Collocation selection based on accurate or approximate Voronoi tessellation. The dashed lines represent the local Voronoi cells. (B) Selecting collocation members using approximate Voronoi based on pixels or lattices. The coloured square marks denote the discrete pixels, and the black lines represent a reference accurate Voronoi tessellation for comparisons.

Voronoi to $O(n)$, where n is the number of points in the domain. Note that this approach may compromise the reciprocal property of Voronoi tessellation due to the approximation, but the reciprocity can always be enforced afterwards.

Compared to the polar selection, this approach based on accurate/approximate Voronoi provides more compact and reciprocal collocations, and the results tends to resemble mesh connectivity. However, as will be shown in later sections, this does not necessarily lead to improved modelling results, because it is difficult to control the number and quality of collocation members for WLS approximations through the Voronoi approach.

2.7 | Adaptation strategies and mechanisms

The mesh-free method is particularly suitable for adaptive modelling considering its flexibility to accommodate points without connectivity constraints. Compared to conventional Adaptive Mesh Refinement (AMR) approaches, mesh-free CFD methods have greater freedom and flexibility for adaptation.[7] The adapted parameters could include point distributions, point numbers, collocation sizes, shape functions, and so forth. As a first step, the present work explores mesh-free adaptation by adding cloud points where necessary.

Figure 7 shows the design structure of the mesh-free adaptive modelling framework in this work. The process is similar to conventional, a posteriori AMR approaches. The adaptive modelling requires three modules to drive the workflow, that is, the adaptation metric, the point adaptation mechanism, and the collocation search approach.

The adaptation metric is computed upon intermediate flow solutions, and indicates whether adaptation is necessary and where it should happen. This metric is often feature-based or error-based and can be case-specific. The point adaptation mechanism inserts or removes as indicated by the metric. It could be extended to move point positions as well. The collocation search module establishes the local collocations for regions affected by the adaptation. Collocation adaptation could be implemented in this module too. These modules are largely decoupled and can be connected with simple file or memory exchanges. The adaptation iterates with the new point clouds and collocations, and eventually outputs the final flow and point cloud solutions. This section details the refinement mechanisms. The adaptation metric can be case-specific and will be detailed in later evaluations.

The first refinement mechanism is illustrated in Figure 8. In 2D, when a point is tagged to be refined, eight additional points are inserted along all orthogonal directions to form a local Cartesian cloud (in 3D, 26 points will be added). This refinement/coarsening 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 this 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.

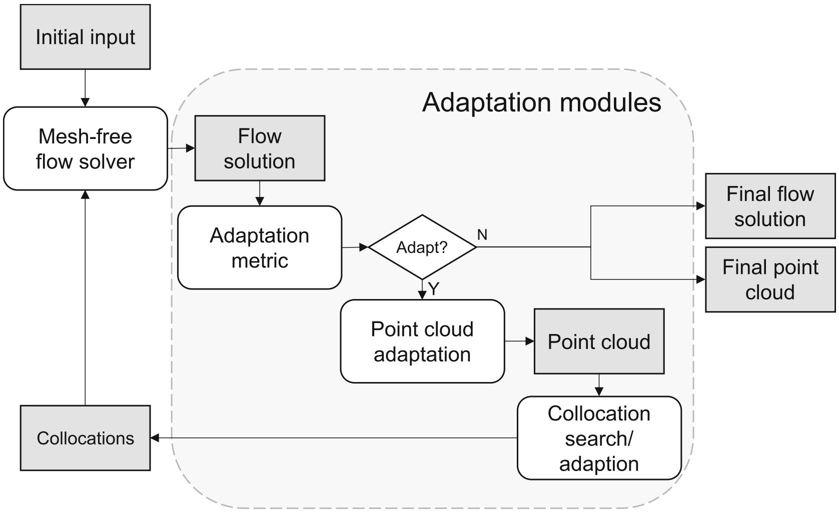


FIGURE 7 Design structure of the adaptive mesh-free modelling work flow. Shaded blocks represent data.

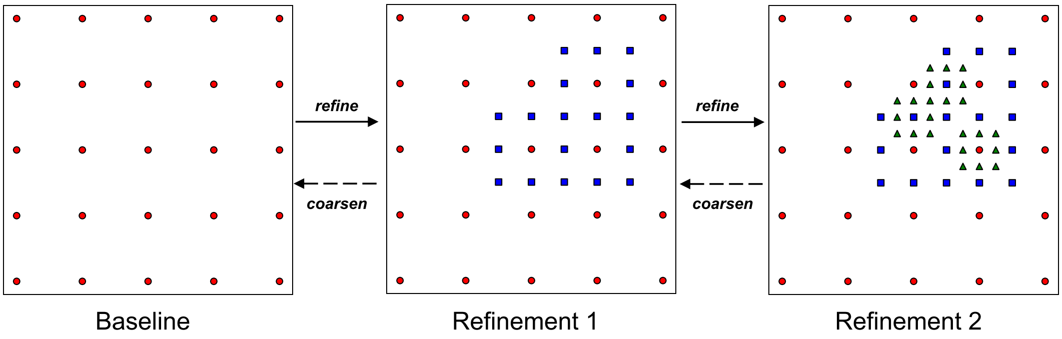


FIGURE 8 Cartesian point adaptation. Different markers and colours denote points added on different refinement levels.

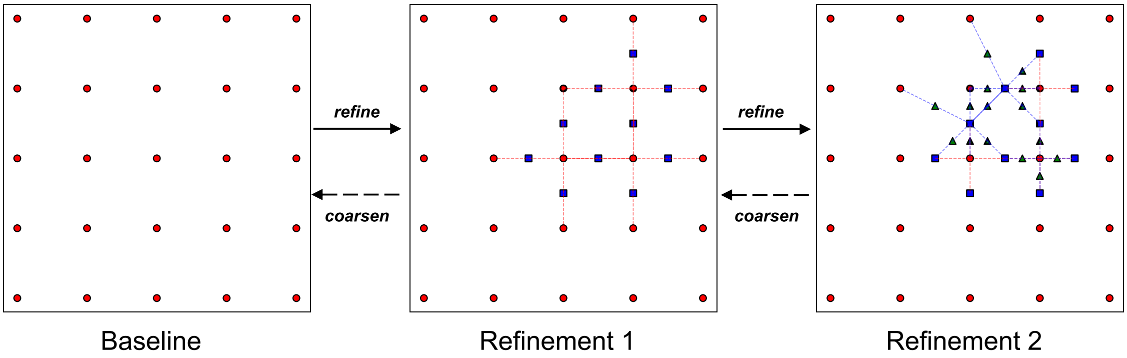


FIGURE 9 Middle point adaptation. The dashed lines represent connections in a local point cloud. Different markers and colours denote points added on different refinement levels.

A more universal mechanism to introduce additional points is illustrated in Figure 9. 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 easy to implement, and is suitable for arbitrary point distributions. The drawback, however, is that the middle points may damage the regularity of point clouds. For an evenly distributed, baseline point cloud in Figure 9, 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 as described in previous sections. 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 either through parametric definitions (e.g., splines or NURB surfaces) or through interpolation (e.g., moving-least-squares or linear). The adapted near-body and off-body points clouds were then blended for mesh-free simulations.

3 | NUMERICAL VALIDATION

3.1 | Inviscid flow past a circular cylinder

This section presents mesh-free modelling of inviscid flow passing a circular cylinder at $Ma = 0.38$.[31] This is a classic case for validation, as well as, error quantification. 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 D away, where D is the cylinder diameter (taken here as the reference length). The first layer of points were 0.001 D above the surface. For the mesh-free modelling, the first-order polynomial was used as the basis function for flow approximation, so the error should be of second-order. The collocations were manually selected with reciprocity and regularity. Three sets of point clouds were used to measure the spatial accuracy of the modelling. The coarse, medium, and fine clouds had $(n_a \times n_r)$ 100×50, 200×100, and 400×200 points respectively. The overall relative entropy product, that is, $\epsilon = \frac{p}{p_\infty}\left(\frac{\rho_\infty}{\rho}\right)^\gamma - 1$, was computed to quantify the spatial error of the modelling.

The spatial accuracy of the mesh-free scheme was measured and presented in Figure 10. 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 intended first-order least-squares approximation in the mesh-free scheme. This shows that the designed spatial accuracy in the mesh-free scheme was successfully achieved.

Figure 11B presents the pressure field resolved using the medium point cloud and the mesh-free modelling. The flow solution showed excellent symmetry along the horizontal and vertical axes. Figure 11A presents the surface pressure coefficient distributions for more quantitative comparisons. The pressure distribution agreed well with reference results,[31] and the downstream stagnation pressure was only slightly 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.

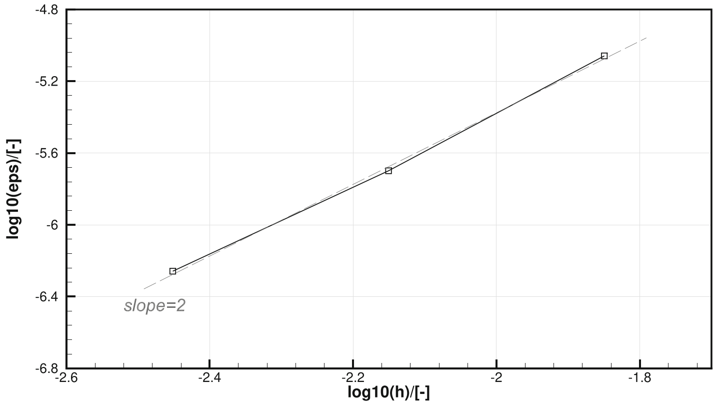
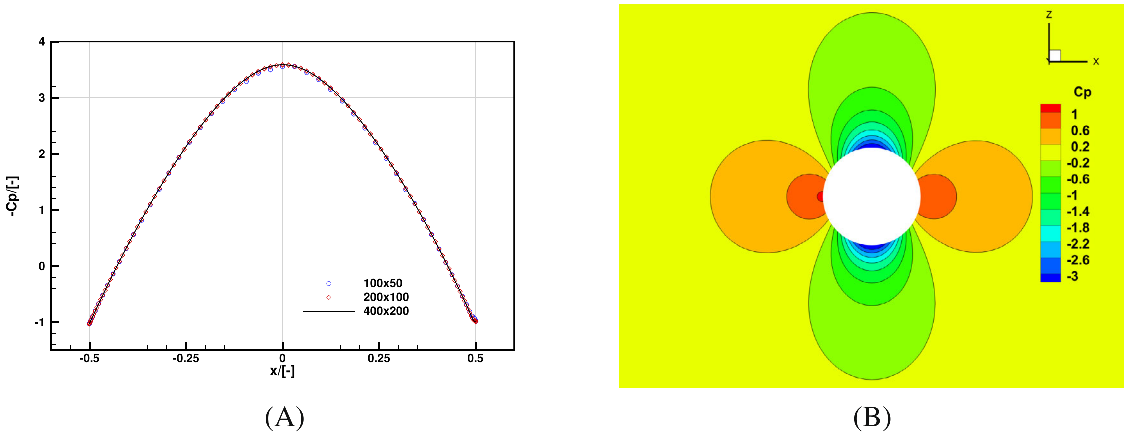


FIGURE 10 Accuracy measurements using the circular cylinder flow simulations on three different point clouds.



(A) (B)

FIGURE 11 Flow solutions of inviscid flow passing a circular cylinder. (A) Cylinder surface pressure coefficient distributions. (B) Medium point cloud pressure coefficient contours.

3.2 | The 30P30N multi-element aerofoil

RANS simulations of the 30P30N mutli-element aerofoil[32] were also carried out. The simulations were performed at $AoA = 5.5\circ$ and $8.5\circ$ with $Ma = 0.17$. The Reynolds number based on the chord length is 1.71×10^6. The SA model was used as the turbulence closure.

Figure 12 presents the flow solutions along with the point clouds for the computation. In total 30,418 points were involved in this RANS simulation. The distance from the first point layer to the wall surface corresponded to a y^+ value of about 1. The points were scaled by the local point densities for visualisation. The contours were obtained through interpolation from the points to a background plane for visualisation as well. For the near-body region, the points were automatically generated following the strand and level-set approaches for the slat, main, and flap elements. Close-up views of point distributions near the slat-main and main-flap gaps are presented in Figure 13A,B, respectively. The near-body points were blended together with regular background points for the modelling. The point collocations were established using the polar selection approach. The entire point generation and pre-processing were largely automated with few manual inputs, for example, geometries, boundary layer growth rates, and background point density.

Comparisons with experimental data at $AoA = 5.5\circ$ and $AoA = 8.5\circ$ are presented in Figure 14A,B, respectively. At both angles of attack, surface pressure solutions on the slat, main, and flap elements agreed well with the experimental data.[32] Note that the current results were obtained from steady RANS modelling. The slight discrepancies can be reduced

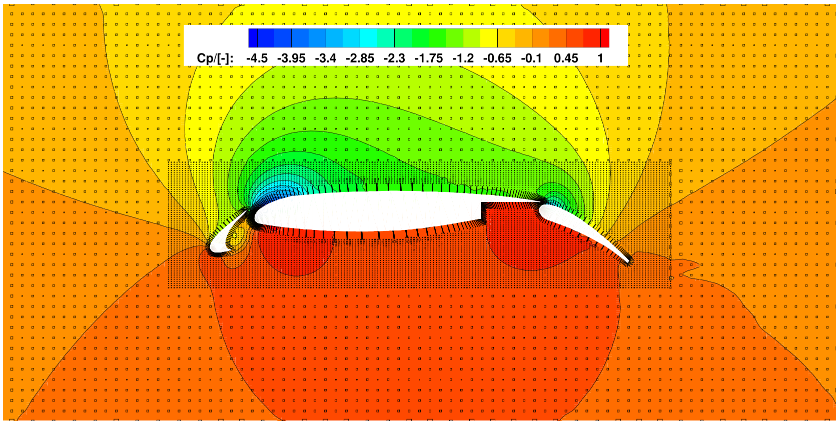


FIGURE 12 Pressure coefficient contours of the 30P30N multi-element aerofoil at $AoA = 5.5\circ$. The mesh-free solutions were interpolated to a background Cartesian plane for visualisation. The scatters denote the point cloud for the computation. The points were scaled with the local point size.

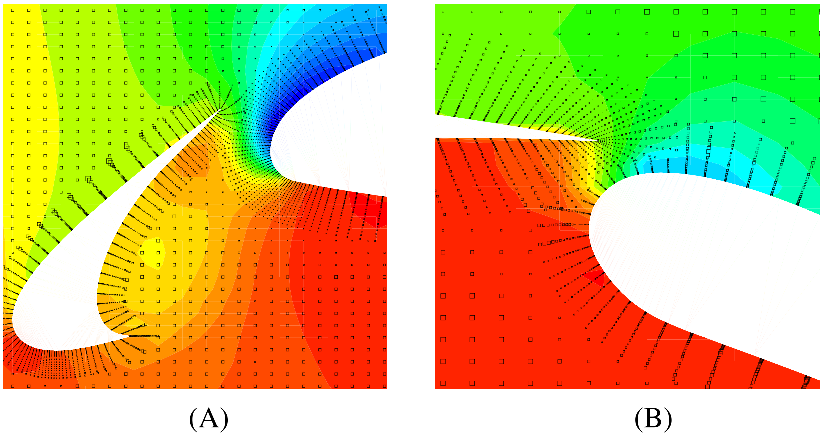
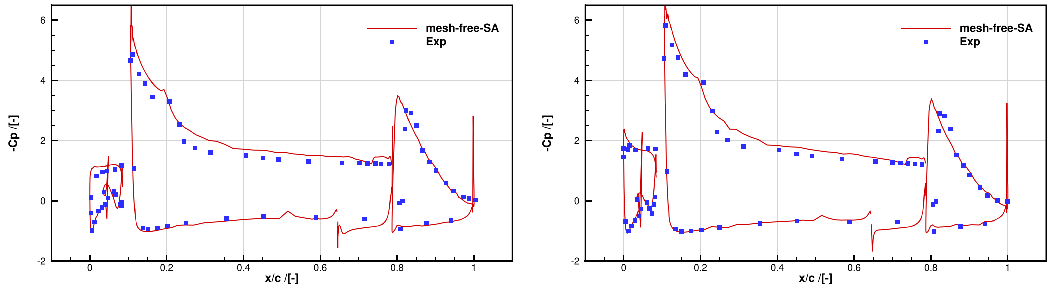


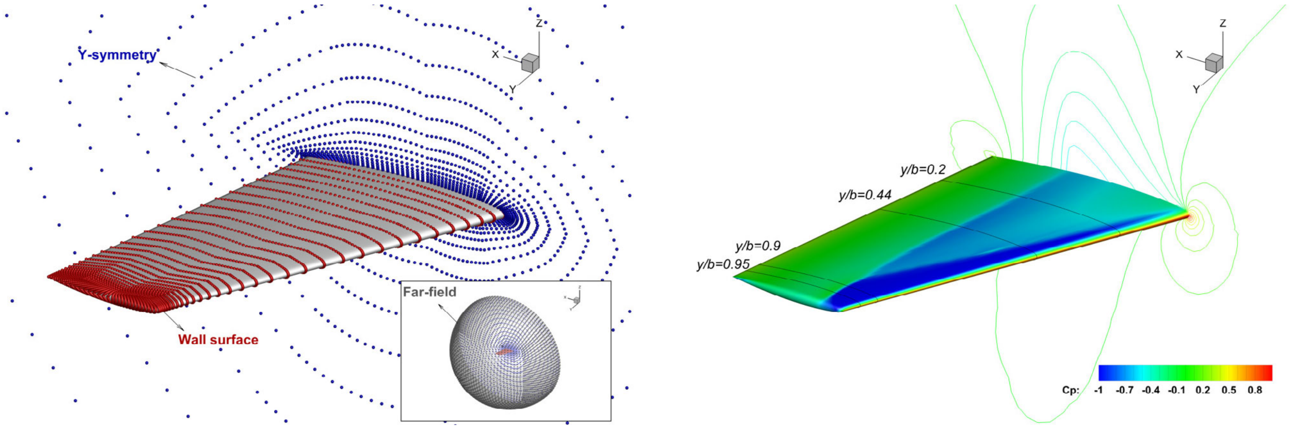
FIGURE 13 Pressure coefficient contours and point clouds near the slat and flap. (A) Flow solution and point distributions near the slat. (B) Flow solution and point distributions near the flap.



(A)

(B)

FIGURE 14 Surface pressure coefficient comparisons between the mesh-free computation and experimental results.[32] (A) $AoA = 5.5\circ$. (B) $AoA = 8.5\circ$.



(A)

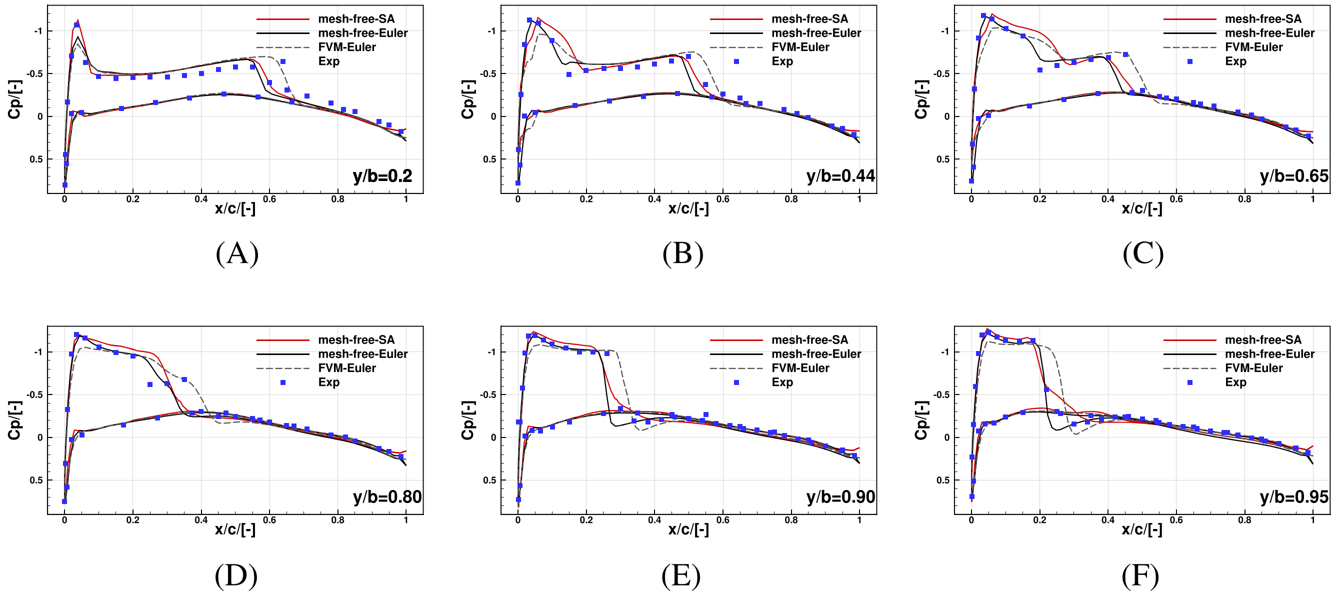
(B)

FIGURE 15 Mesh-free simulations of the ONERA M6 wing at transonic conditions. (A) Point cloud and boundaries for simulations of the ONERA M6 wing. (B) Surface pressure coefficient solutions from the mesh-free simulation.

using finer point clouds, unsteady time stepping, and improved turbulence modelling. The spikes were due to sharp geometric transitions in the numerical modelling. In terms of integrated forces, the predicted lift coefficients at $AoA = 5.5\circ$ and $AoA = 8.5\circ$ are 2.91 and 3.16, respectively, which are close to the corresponding experimental results of 2.81 and 3.17[33] at the same conditions. Overall, the present results demonstrate and validate the effectiveness and flexibility, of the proposed mesh-free tool chain in handling complex geometries.

3.3 | Transonic ONERA M6 wing

This section presents 3D inviscid and RANS mesh-free simulations of transonic flows passing the classic ONERA M6 wing. The M6 wing surface geometry is shown in Figure 15A 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, about five times the wing span away in all directions. For inviscid modelling, the computational domain consisted of 182,853 points in total, with 4425 points on the wing surface. For RANS simulations using the SA model, the point cloud was increased to 278,841 in total, with more points added for the boundary layer resolution. Note this represents a coarse point cloud for the benefit of light computational costs for this work. As a result, the distance from the first point layer to the wall surface led to y^+ values of about 3 to 5. Note that the wing surface in Figure 15A is for illustration purposes only. In the mesh-free computation, all geometries were represented with discrete points. The collocations were established using accurate Voronoi search.



(A)

(B)

(C)

(D)

(E)

(F)

FIGURE 16 Comparisons of wing sectional pressure coefficient distributions between the experimental,[34] FVM, and mesh-free numerical results. (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) $y/b = 0.95$.

The simulations were conducted at $Ma = 0.8395$ and $AoA = 3.06$ degrees using Euler equations. This is a classic transonic case where multiple shock waves would form on the wing surface, as Figure 15B shows. 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[34] in Figure 16A–F. The inviscid and RANS mesh-free results agreed very well with the measurements, especially on the wing's lower surface and leading-/trailing edge regions. A further comparison was made against the classic inviscid FVM approach using a mesh converted from the same point cloud for the mesh-free Euler simulation. Note that despite the same point distributions, the two spatial decompositions (the point cloud and the mesh) still have important differences in terms of where unknowns are stored and how local functions/derivatives are reconstructed etc. Nonetheless, Figure 16A–F show that the two methods have excellent agreement in regions where no shock wave presents, especially at the wing lower surface. The main discrepancy is associated with the shock resolution due to differences in for example, numerical schemes and spatial decompositions. Improving the cloud/mesh resolution near the shock may diminish the differences, but this needs further investigations. The mesh-free RANS simulations offered more realistic shock predictions due to the dissipation, and slightly improved the surface pressure solutions near the leading edge and at the lower surface. Small discrepancies were still noticed near the shock waves, particularly in Figure 16D at $y/b = 0.8$ where the shock waves merge. Increasing the local point density in these regions can help improve the shock wave resolution. Overall, these results provide excellent validation for the present mesh-free solver for complex 3D simulations.

4 | ASSESSMENT OF POINT DISTRIBUTION AND COLLOCATION SEARCH

4.1 | Assessment of weight balancing for irregular point clouds

The effects of weight balancing (WB) of Equation (16) in improving modelling performance when using irregular point clouds are evaluated in this section. The cylinder case with 100 × 50 points in Section 3.1 was used here but with irregular point clouds. As shown in Figure 17, for this study, the first 10 layers of points off the wall surface were kept regular for boundary resolutions, while random perturbations were introduced to the rest 40 layers. For each perturbated layer, the

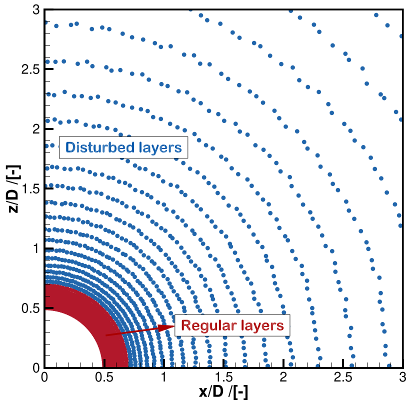


FIGURE 17 Irregular point clouds with random perturbations for the circular cylinder flow simulations.

random offset $(\delta x, \delta z)$ to the points was introduced as:

$$(\delta x, \delta z) = a_R \delta r(E_x, E_z), \tag{25}$$

where $E_x, E_z \in [-0.5, 0.5]$ are random numbers at each point. δr is the radial distance to the previous layer (before disturbance). δr grows larger as the point layer is further away from the wall. $a_R \in [0{,}100\%]$ is a constant controlling the randomness level. This section investigated $a_R = 30\%, 35\%, 40\%$, with and without the weight balancing. For this investigation, the point collocations remained the same despite the different randomness levels: each point took adjacent radial and angular points to form a structured local collocation.

Figure 18A–C present the flow convergence history with different randomness levels and with/without weight balancing, in terms of the L^2 norm of the flow residuals. As the randomness level a_R increased, the convergence became worse due to the irregular point distributions. Nevertheless, it is clearly shown that while the weight balancing was activated, the convergence was faster and smoother. At the higher randomness level of $a_R = 40\%$, weight balancing improved the final convergence level by about three orders of magnitude.

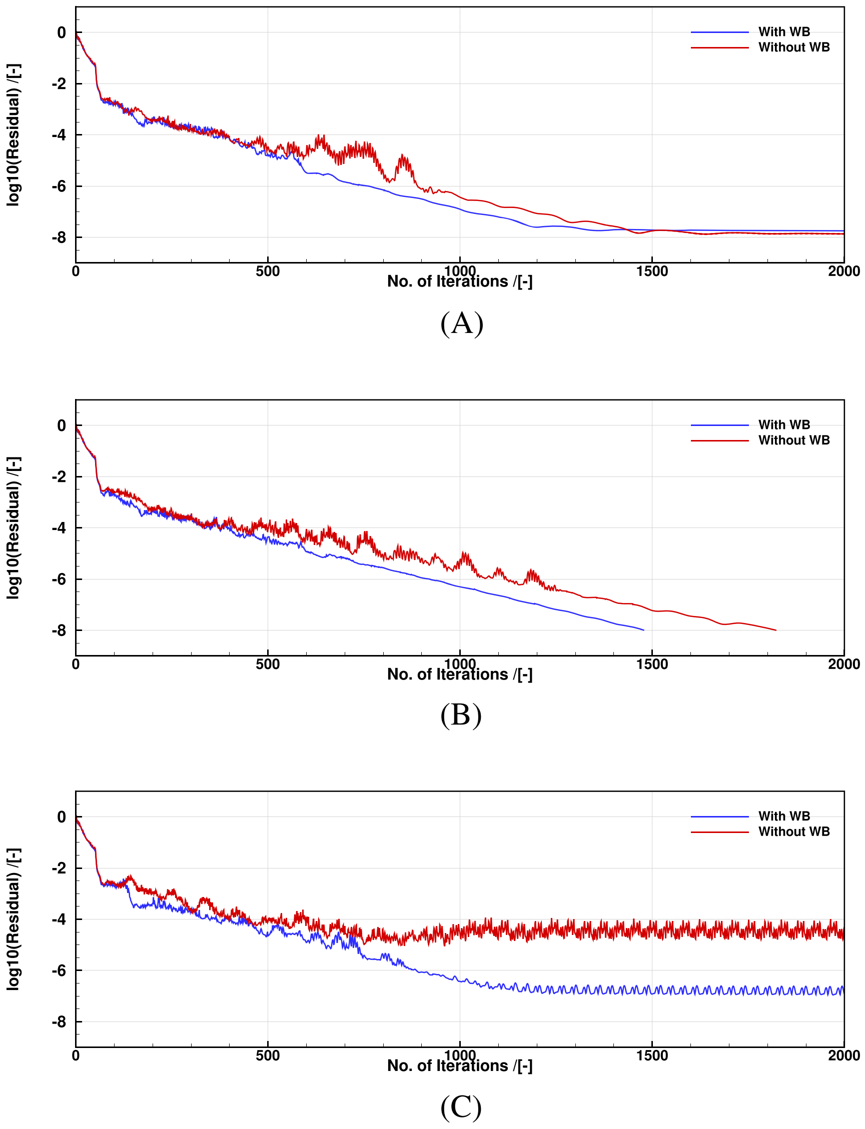
Surface pressure solutions at $a_R = 40\%$ with and without the weight balancing are compared in Figure 19. The reference data is fine point cloud 400×200 results from Figure 11A with regular point distributions. Without the weight balancing, the mesh-free scheme could give fairly good flow solutions, although the suction pressures were slightly under-resolved with small asymmetry, due to the random points. When the weight balancing was activated, the pressure solution agreed very well with the reference data and the symmetry was recovered despite the irregular points. These highlight the effects the weight balancing in improving the modelling convergence and accuracy, when handling irregular point clouds.

4.2 | Assessment of collocation search approaches

This section compares different collocation search approaches on regular and irregular point clouds. The evaluations also focused on the cylinder case with 100×50 points using inviscid modelling.

Figure 20A–D presents the collocation search results on regularly distributed point clouds around the cylinder. The grey lines around each point represent the local point collocation. Figure 20A is a reference result, where the collocations were manually assigned according to the point radial and angular structures. Figure 20B presents the collocations searched using the polar search method, with a polar interval of $\delta\theta = \pi/4$ aiming at securing four support points for a point centre. It is notable that the polar search results are almost identical to the structured collocations in Figure 20A, with minor differences in the boundary layers, which will be shown later.

Figure 20C used the approximate Voronoi search with 10-by-10 lattices for each local domain, while Figure 20D adopted the accurate Voronoi tessellation. Figure 20D offers identical results to the structured collocations in Figure 20(a). The approximate Voronoi in Figure 20C, however, tends to include extra diagonal connections in addition to the accurate Voronoi results. This was due to the coarse lattices in the approximate Voronoi process, and finer lattices would push the



(A)

(B)

(C)

FIGURE 18 Convergence histories of circular cylinder simulations with irregular points, with and without the weight balancing (WB). (A) Outer 40 layers disturbed with randomness $a_R = 30\%$. (B) Outer 40 layers disturbed with randomness $a_R = 35\%$. (C) Outer 40 layers disturbed with randomness $a_R = 40\%$.

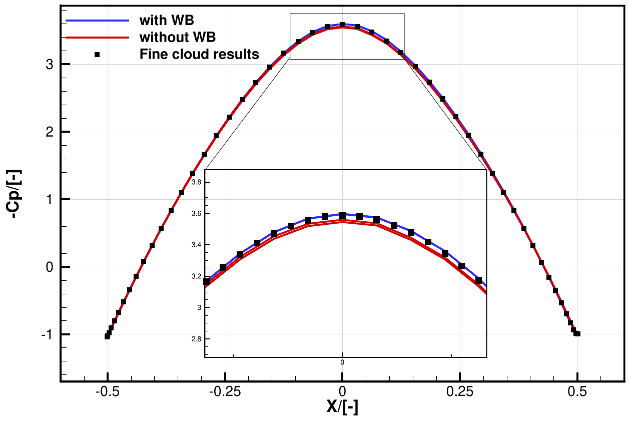


FIGURE 19 Surface pressure solutions with and without weight balancing (WB) with randomness $a_R = 40\%$. Reference data is the fine point cloud results in Figure 11A.

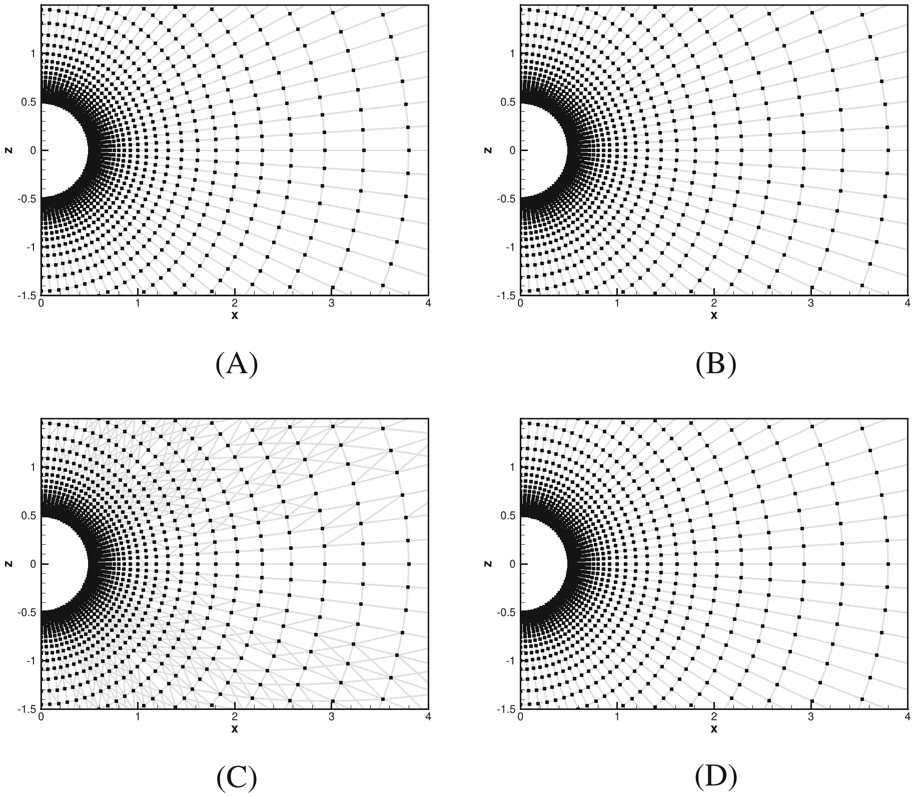
results towards the accurate Voronoi. Note that the crossover of diagonal connections can be tolerated in the present WLS scheme.

Figure 21A–D present the collocation results on irregular point clouds. Random offsets were also introduced to the outer 40 radial layers, with a randomness level of $a_R = 35\%$ as defined in the previous section. Figure 21A is again the reference collocation result, using manually assigned structured collocations. Upon this irregular point distribution, the structured collocation does not ensure good WLS approximations.

Figure 21B presents the collocation results using the polar selection method. It can be observed that each point in Figure 21B has compact and comprehensive support from all directions. This ensures good local WLS approximations. Many crossovers of connections can be noted, but these are compatible with the present mesh-free scheme. Figure 21C,D present the collocation results using approximate and accurate Voronoi tessellations, respectively. The results between these two are largely similar, with the approximate approach in Figure 21C tends to give additional connections. The results are also similar to the polar selection approach, but with fewer crossover connections.

The ability to establish good collocations in boundary layer region is critical for the collocation search, and this is compared in Figure 22A–D. These are the 10 regular point layers off the wall blended with outer irregular points. Figure 22A is again the structured collocation as the reference. Figure 22B shows the boundary layer collocations using polar selection. It is notable that, near the wall boundary, the polar approach included more nearby off-wall points. This was because only one layer of ghost points were included in the search process near this boundary, and the polar approach automatically reduced polar intervals to ensure information from all directions. However, later results will show that this did not compromise the modelling quality.

Figure 22C,D present results near the boundary using approximate and accurate Voronoi, respectively. The accurate Voronoi offers identical results compared to the structured case in Figure 22A, while the approximate approach included a few more diagonal connections.



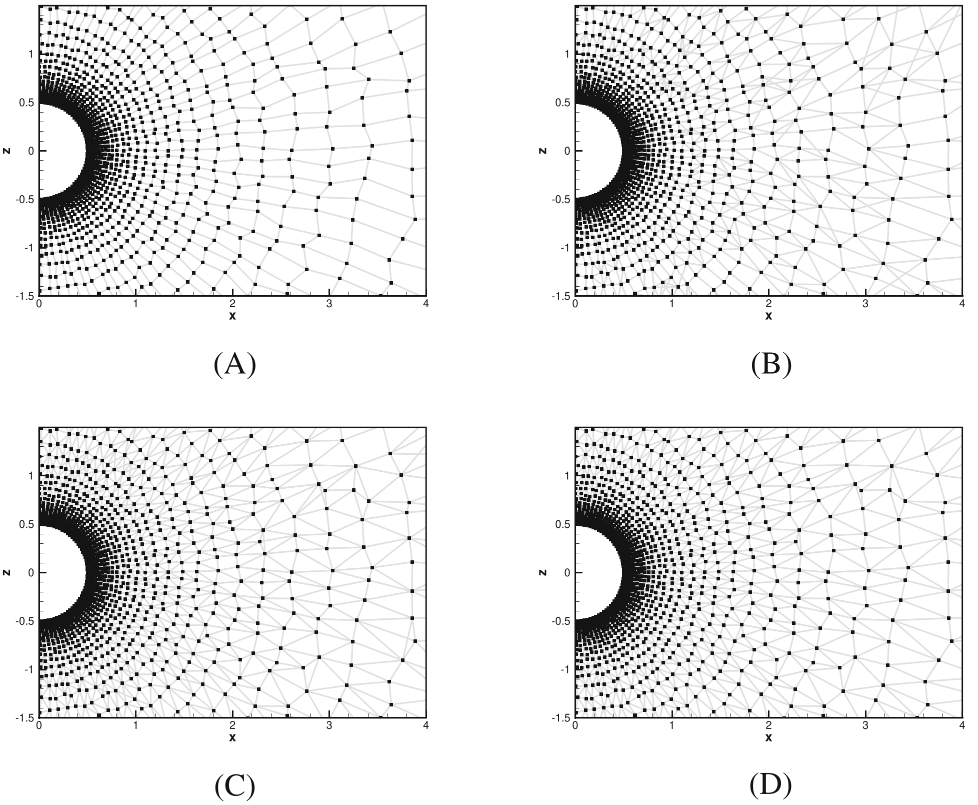
(A)

(B)

(C)

(D)

FIGURE 20 Established point collocations (grey lines) on regular point clouds using various collocation search approaches. (A) Structured point collocations. (B) Point collocations based on distance with polar selection. (C) Point collocations based on approximate Voronoi tessellation. (D) Point collocations based on accurate Voronoi tessellation.



(A)

(B)

(C)

(D)

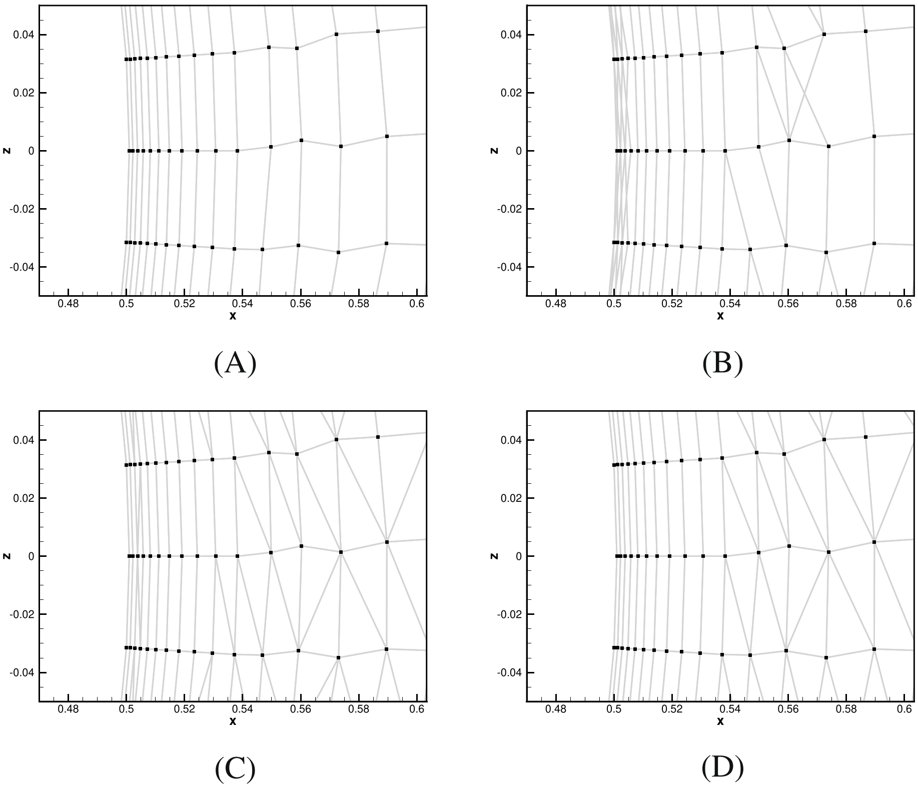
FIGURE 21 Established point collocations (grey lines) on irregular point clouds (outer 40 layers perturbed with randomness $a_R = 35\%$) using various collocation search approaches. (A) Structured point collocations. (B) Point collocations based on distance with polar selection. (C) Point collocations based on approximate Voronoi tessellation. (D) Point collocations based on accurate Voronoi tessellation.

The convergence behaviours of different collocation search approaches are compared in Figure 23A,B, for regular and irregular point clouds, respectively. For the regular point cloud, the structured and accurate Voronoi approach showed the same convergence history, as the collocation results are identical. The approximate Voronoi showed the slowest convergence. It is notable that the polar selection approach showed the fastest convergence. The differences between the polar selection and the structured collocations are mainly in the boundary layer, as described earlier. The results here suggest that the additional connections from nearby points in the boundary layer are beneficial for the numerical process.

For irregular point clouds, as shown in Figure 23B, the convergence was all slower by about 200 steps in general using the same CFL of 50. The polar selection again showed the best convergence. The accurate Voronoi approach converged better than the structured collocations, but the residual stagnated around the order −7 eventually. The approximate Voronoi approach converged slightly slower than the structured collocations.

Surface pressure solutions are presented in Figure 24 for irregular points using various collocation search approaches. The black dots are the fine cloud solutions in Figure 11B as the reference. It can be noted that all collocation search approaches offered generally good predictions on this coarse point cloud. This highlights the flexibility of the present mesh-free scheme.

Small discrepancies are mostly near the lateral regions of the cylinder, where the pressure is the lowest. Despite the irregular point distributions, the structured collocation and the polar selection approach offered the best results with good symmetry and agreement with the reference. The accurate Voronoi approach led to slightly under-predicted pressures and small asymmetry. The asymmetry was enhanced when using the approximate Voronoi approach. These results highlight that the proposed polar selection approach is more suitable for the present mesh-free scheme when establishing point collocations. The Voronoi approach, which is popular in mesh generation, does not necessarily generate better collocations for the mesh-free modelling. The polar selection approach ensures compact support and promotes information from all directions, which are vital for improved local WLS approximations.



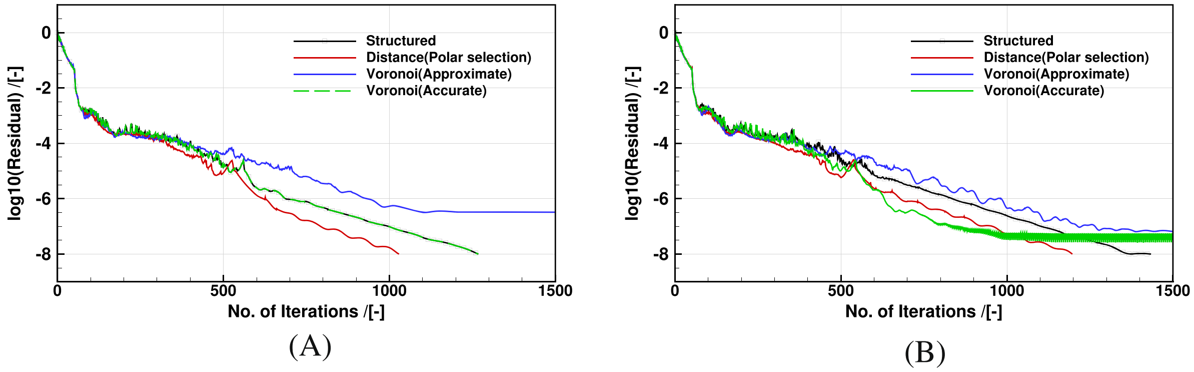
(A)

(B)

(C)

(D)

FIGURE 22 Established point collocations (grey lines) in the boundary layer on irregular point clouds (outer 40 layers perturbed with randomness $a_R = 35\%$) using various collocation search approaches. (A) Structured point collocations. (B) Point collocations based on distance with polar selection. (C) Point collocations based on approximate Voronoi tessellation. (D) Point collocations based on accurate Voronoi tessellation.



(A)

(B)

FIGURE 23 Comparisons of convergence behaviours of different collocation search approaches using the same point cloud. (A) Convergence history on regular point clouds using different collocation searches. (B) Convergence history on irregular point clouds using different collocation searches.

5 | ASSESSMENT OF ADAPTIVE MESH-FREE MODELLING

5.1 | Assessment of adaptation strategies

This section assesses the adaptive mesh-free simulation strategies using the isentropic vortex transport problem.[35] This model problem has the theoretical solution:

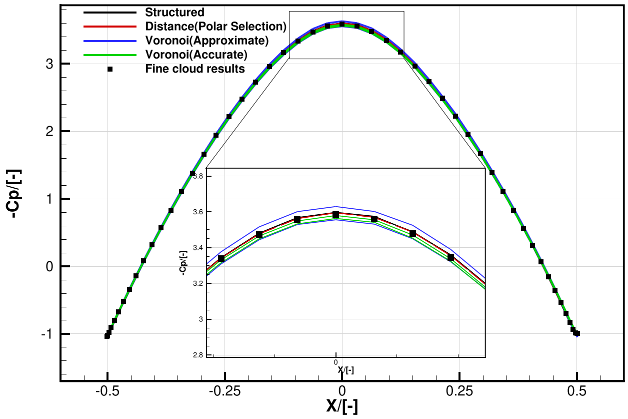


FIGURE 24 Comparisons of surface pressure solutions on irregular point clouds using different collocation search approaches.

TABLE 1 Test matrix of mesh-free adaptive simulations.

Cases	Adaption mechanism	Collocation search
C1	Cartesian	Distance
C2	Cartesian	Distance (Polar Selection)
C3	Cartesian	Voronoi (Accurate)
C4	Mid-point	Voronoi (Accurate)
C5	Mid-point	Distance (Polar Selection)

$$\begin{aligned}\rho &= \left[1 - \frac{(\gamma-1)b^2}{8\gamma\pi^2}e^{1-r^2}\right]^{\frac{1}{\gamma-1}},\\ u &= u_\infty - \frac{b}{2\pi}e^{\frac{1-r^2}{2}}(y-y_c),\\ v &= v_\infty + \frac{b}{2\pi}e^{\frac{1-r^2}{2}}(x-x_c),\\ p &= \rho^\gamma,\end{aligned} \tag{26}$$

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, (u_∞, v_∞) is the convection speed, $\gamma = 1.4$ is the specific heat ratio, and ρ and p are the density and pressure, respectively. The vortex is initialised at the centre of the computational domain with all outer boundaries set to be periodic. Without numerical dissipation and dispersion, the vortex should maintain its initial strength and shape during the convection without any decay. The vortex strength was set as $b = 5$.

The convection speed (u_∞, v_∞) was set to zero, so that the vortex stays at its initial position. However, the modelling was initialised using a very coarse point cloud consisting of 21×21 points covering the domain $[-5,5] \times [-5,5]$. The large spatial size would quickly decay the vortex with numerical viscosity. The adaptation was expected to improve the spatial resolution and reduce the dissipation, thereby preserving the vortex shape over longer durations. The adaptation performance was examined using the differences between the numerical and exact solutions after 500 time steps, with the same dimensionless step size of 0.1, chosen large enough for the decay to occur.

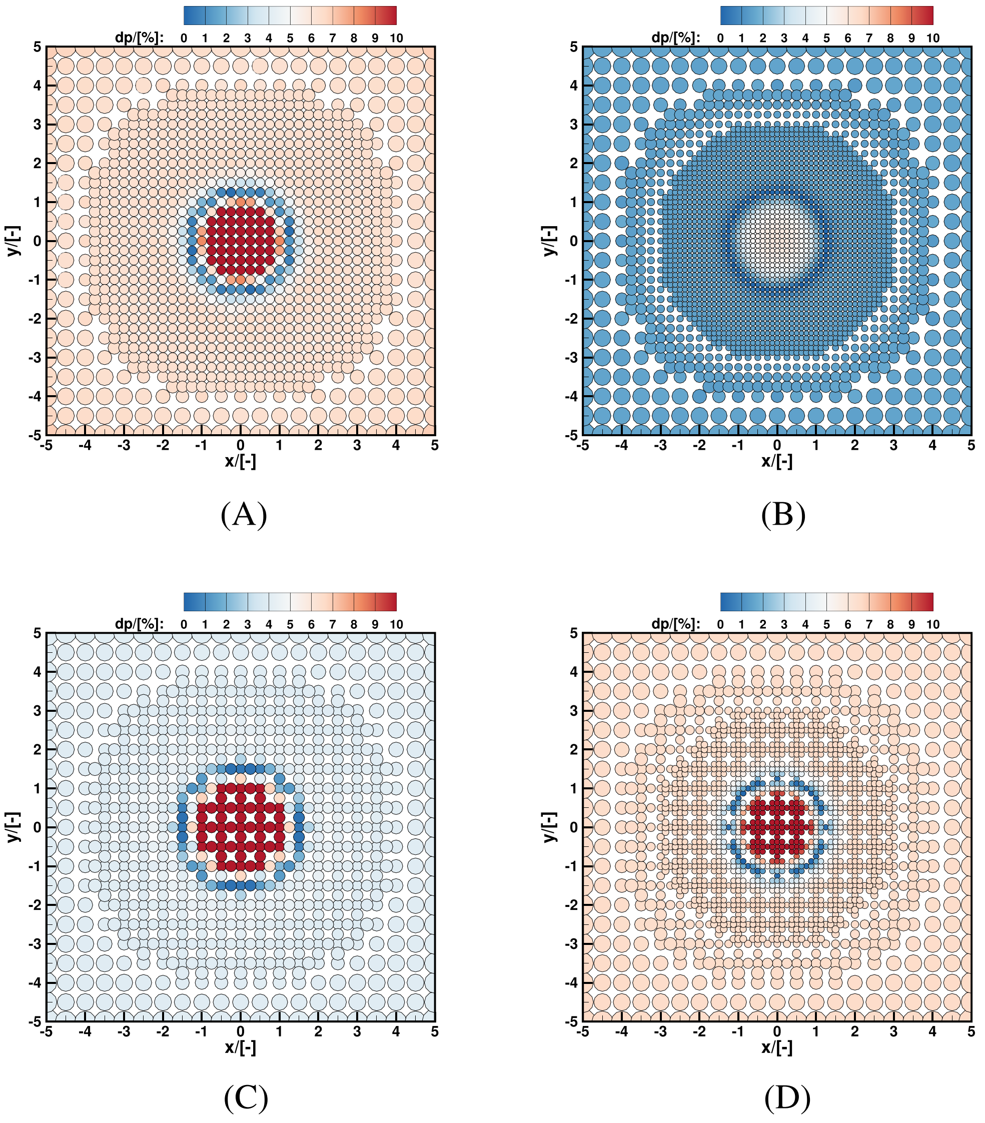
Table 1 presents the test matrix of adaptive strategies combining different adaptation mechanisms and collocation search methods. Case 1 used Cartesian refinement with the simple distance-based collocation search. The search radius was set as 1.5 times the local point size. Case 2 was similar to Case 1 but adopted the polar selection with a polar interval of $\pm\pi/4$. Case 3 combined the Cartesian refinement and the accurate Voronoi selection. Case 4 was similar to Case 3 but replace the refinement with the mid-point mechanism. Case 5 also used the mid-point mechanism but with polar selections. 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 Cases 4 and 5, further iterations were carried out in regions where $r < 0.5$ since the mid-point mechanism introduces fewer additional points with poorer regularity than the Cartesian approach.

Figure 25A–D show the clouds for Cases 3 and 4 of Table 1, 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.

Figure 25A,B illustrate the Cartesian point adaptation mechanism. The points were scaled by their local point size ds, which is defined for a point collocation centring at i as

$$ds = \pi r_i^2 / n_i, \tag{27}$$

where r_i is the mean distance to the collcoation centre, and n_i is the number of points in this collocation. It can be observed that the adaptation preserved the initial regular point distribution with uniform density. Irregular distributions were only



(A)

(B)

(C)

(D)

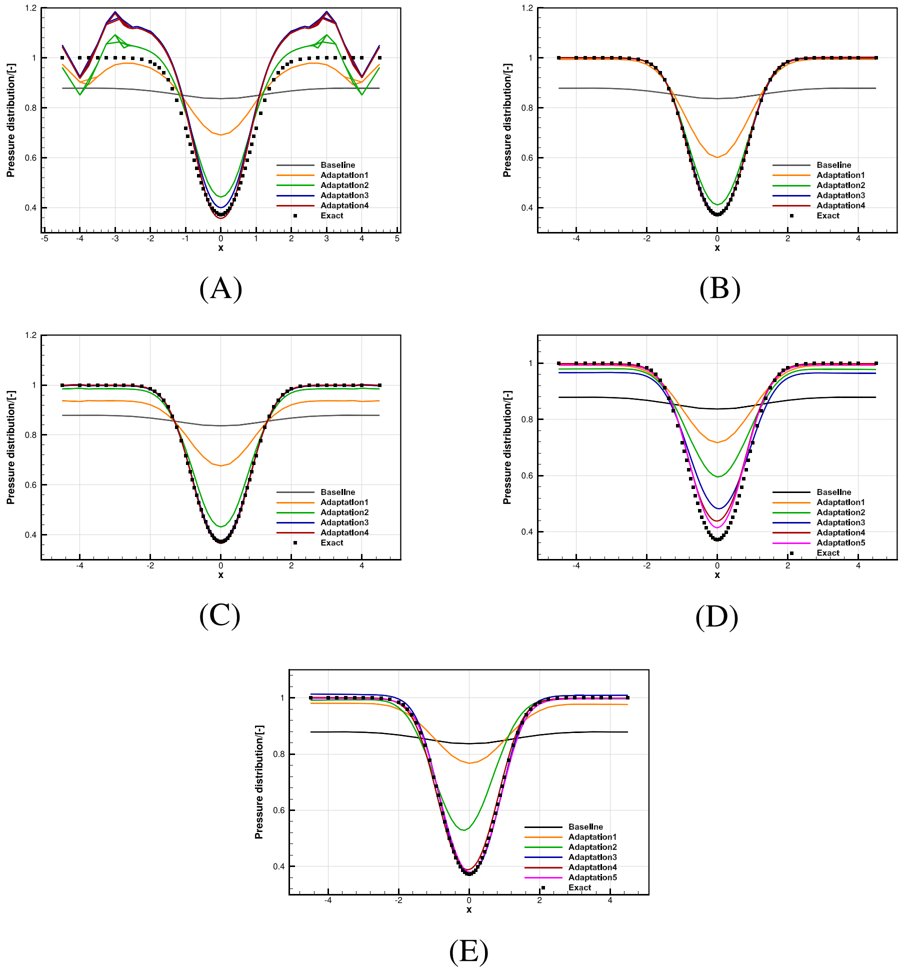
FIGURE 25 Point clouds of cases 3 and 4 in Table 1 after the first and second adaptation iterations. Point sizes were scaled with the local point size. Contours denote the relative pressure differences between the numerical and exact solutions $\delta p = |p_{num}/p_{exact} - 1| \times 100\%$. (A) Case 3 Cartesian adaptation iteration 1. (B) Case 3 Cartesian adaptation iteration 2. (C) Case 4 mid-point adaptation iteration 1. (D) Case 4 mid-point adaptation iteration 2.

encountered in transitional regions where two refinement levels meet. Through the adaptation from Figure 25A,B, the pressure error was effectively reduced from about 10% maximum to a max of 5%.

Figure 25C,D illustrate the mid-point adaptation mechanism. In Figure 25C, after the first adaptation iteration, the point distribution remained mostly uniform. However, with the second adaptation iteration, in Figure 25D, small local clusters were 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 the larger domain.

The pressure profiles across the vortex centre extracted from the cases of Table 1 are plotted in Figure 26A–E. Different combinations of adaptation mechanisms and collocation search methods showed diverse impacts on the obtained solutions. Through the adaptation, Case 1 improved the resolution of the central pressure valley, but the solution of outer regions deviated further from the exact solution. This was mainly due to the inadequate choice of support sizes on the irregular points in these transitional regions, which compromised the WLS approximation. However, with the polar selection, Case 2 managed to quickly converge the modelling to the exact solution. Case 3 smoothly converged to the exact solution by the third iteration, using regular point distributions and reciprocal collocations.

As for Case 4, the numerical solution gradually approached the analytical one but was yet to conform at the final adaptation iteration. Small discrepancies remained near the centre of the pressure profile. However, when switched to Case 5 with the polar selection, the flow solution quickly converged since the third adaptation. There was very slight asymmetry in the results, which was due to asymmetric collocations from the polar selection approach.



(A)

(B)

(C)

(D)

(E)

FIGURE 26 Comparisons of pressure profiles across the vortex centre between numerical and exact solutions from adaptation cases in Table 1. (A) Case 1: Cartesian adaptation + distance search. (B) Case 2: Cartesian adaptation + distance search (with polar selection). (C) Case 3: Cartesian adaption + Voronoi search (accurate). (D) Case 4: Mid-point adaption + Voronoi search (accurate). (E) Case 5: Mid-point adaption + Distance search (polar selection).

The overall performance of each adaptation strategy of Table 1 was quantified using the overall relative pressure differences:

$$\epsilon_p = \frac{\int \delta p ds}{\int ds}, \tag{28}$$

where δp is the point-wise relative pressure difference, and ds is the local point size defined in Equation (27) reflecting the area a point nominally occupies.

Figure 27 presents the ϵ_p convergence history through the adaptation iterations for the cases of Table 1. Apart from Case 1 with the simple distance-based search, all other cases managed to reduce the overall error ϵ_p below 1% through adaptation, from an initial 13% on the coarse point cloud.

Case 1, combining Cartesian adaptation and distance-based search, struggled to converge using the same point clouds. 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 with irregular point distributions and large point size differences. The inaccuracy was mostly due to the inadequate choice of the support sizes and collocation members.

Case 2, combining Cartesian adaptation and polar selection, showed the best convergence. The error was almost zero since the third adaptation iteration. For Case 3 with the same Cartesian adaptation but using accurate Voronoi selection, the convergence was slower, but eventually reached zero. Case 5 with the mid-point adaptation and polar selection, showed the second-best convergence and the final error was about 0.4%. When switched to accurate Voronoi selection in Case 4, the convergence was worse with higher oscillations, but eventually reached a similar error of about 0.8%.

These results highlight the importance of regular point distributions and suitable collocation search methods for adaptive modelling. Both polar selection and accurate Voronoi selection approaches were shown able to handle the adaptive modelling with irregular points, and the polar selection showed slightly faster convergence.

5.2 Application to aerofoil simulations

This section applies the adaptive mesh-free modelling to simulations of the RAE 2822 aerofoil at transonic conditions. The computational domain and initial point cloud are shown in Figure 28. 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-growing the strand points. For off-body point clouds, the adaptive simulation combined mid-point adaptation with the accurate 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

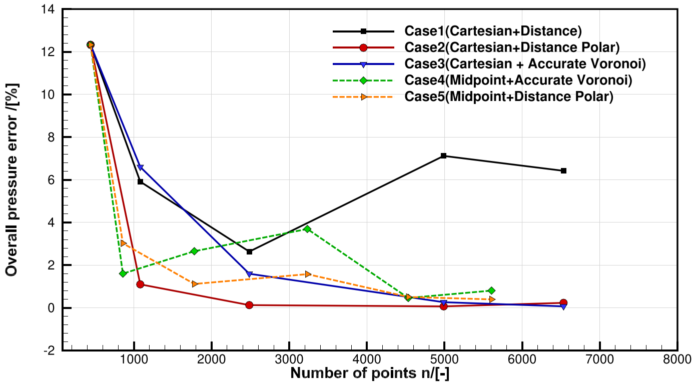


FIGURE 27 Adaptation convergence of cases in Table 1 represented using the overall relative pressure difference ϵ_p in Equation (28).

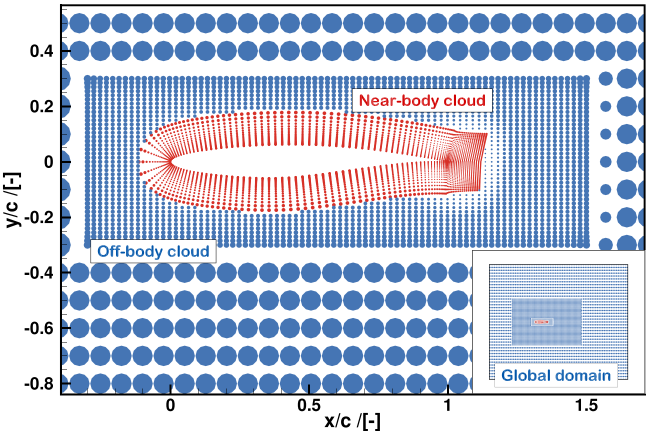
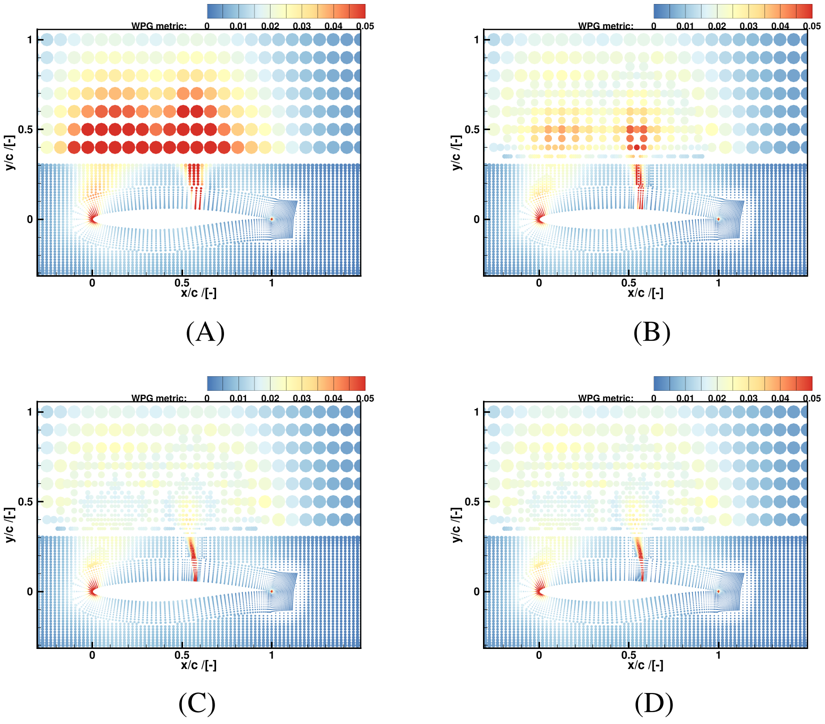


FIGURE 28 RAE 2822 computational domain and point clouds. The points were scaled with the point size *ds*.



(A)

(B)

(C)

(D)

FIGURE 29 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. (A) Baseline. (B) Adaptation iteartion 1. (C) Adaptation iteartion 2. (D) Adaptation iteartion 3.

background points. To alleviate the point clustering issue during the adaptation, a point size constraint was introduced in the adaptation, that is, local point size smaller than the minimum allowed (0.012 *c* in the current case) led to no further local adaptation.

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

$$\omega_p = \frac{|\nabla p| ds}{\int ds}, \tag{29}$$

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.

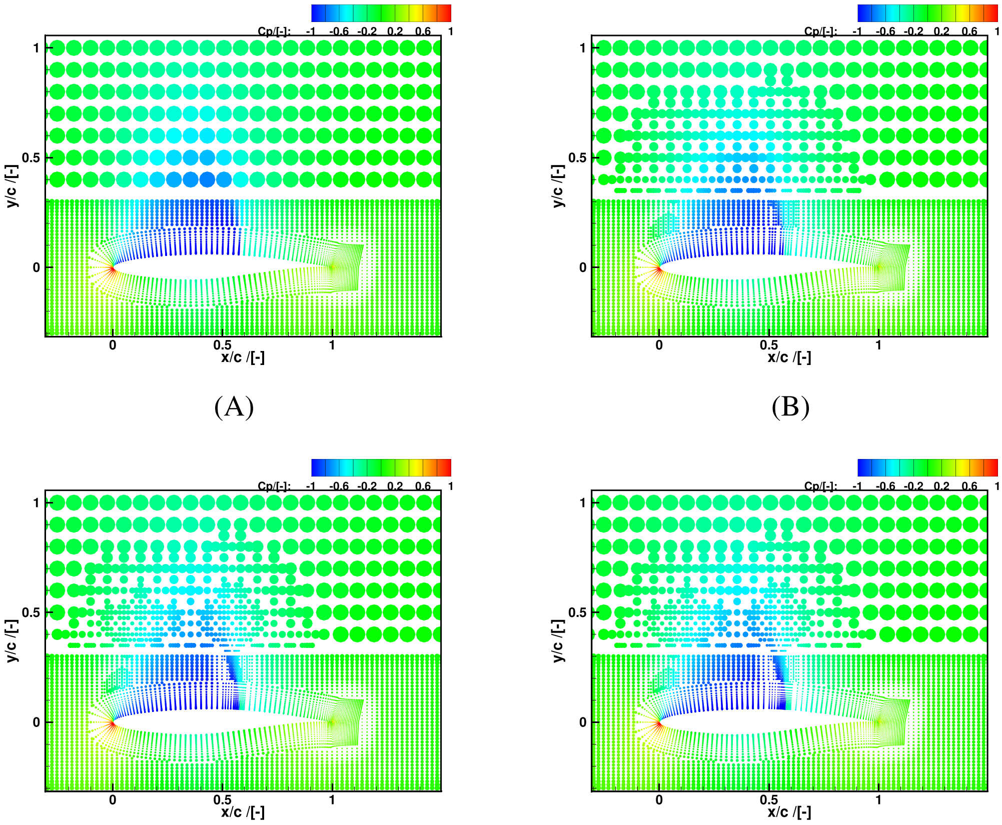
Figure 29A–D present the point cloud evolutions through the adaptation along with the weighted pressure gradient metric contours. The points were scaled by their sizes ds for illustration purposes. In each adaptation, points with $\omega_p \geq 0.05$ were tagged for refinement.

For the baseline case in Figure 29A, 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 29B. 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 Figure 29B–D, the point sizes became more uniform and smaller, and high ω_p regions shrank, and concentrated near flow features (shock, stagnation, etc.). 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 Figure 30A–D. It is clearly shown that the ω_p metric clustered the adaptation near the shock. 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 resolution in both near-body and off-body regions compared to the initial solution in Figure 30A.

Figure 31 compares the aerofoil surface pressure coefficient distributions over the adaptation iterations and with experimental data.[36] 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.



(A)

(B)

(C)

(D)

FIGURE 30 Pressure coefficient contours of the transonic RAE 2822 aerofoil in each adaptation iteration. Points are scaled by local point sizes. (A) Baseline. (B) Adaptation 1. (C) Adaptation 2. (D) Adaptation 3.

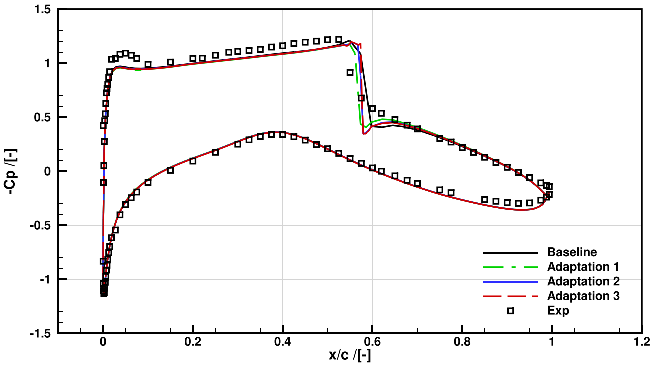


FIGURE 31 RAE 2822 pressure coefficient comparisons.

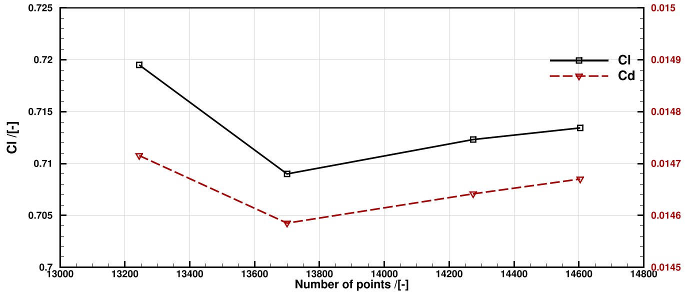


FIGURE 32 RAE 2822 lift and drag convergence through the adaptative simulation.

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 32 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[36] of 0.731 and 0.0121. The discrepancies were mostly due to the inviscid modelling for the current case.

6 | CONCLUSIONS

This work investigated implicit and adaptive mesh-free CFD modelling using point collocations. The weighted least-squares-based spatial discretisation of the governing flow equations, and the implicit temporal discretisation were presented in detail. A new weight balancing concept proposed to improve mesh-free modelling on highly irregular point clouds was assessed. Automatic point cloud generation techniques for complex geometries were also proposed. To establish the local point collocations, a novel distance-based search approach with polar selection was proposed and compared with collocation search methods based on accurate and approximate Voronoi tessellations. This work further explored various adaptation strategies exploiting the mesh-free flexibility. The following conclusions can be drawn from the present study:

1. The proposed mesh-free numerical schemes, including weight balancing, boundary conditions, collocation search methods, point cloud generation, and code implementations were assessed using several benchmark 2D/3D test cases, with both inviscid and RANS results. Simulations of inviscid flow past the circular cylinder using point clouds of varying densities characterised the second-order spatial accuracy of the current implementations, as intended. RANS simulations of the 2D 30P30N multi-element aerofoil, with automatically generated and blended clouds, offered good agreement with experimental data despite the highly complex geometry. Inviscid and RANS simulations of the 3D ONERA M6 wing at transonic conditions also provided good resolutions of the shock waves and excellent agreement with experimental data.

2. The proposed weight balancing concept was found capable of improving the mesh-free convergence and accuracy on highly irregular point clouds. The weight balancing adjusts the weights to avoid biased local WLS approximations induced by irregular point distributions. For the inviscid cylinder flow case with highly irregular points, the weight balancing not only accelerated the convergence, but also improved the final solution accuracy, despite the highly irregular point distributions.
3. The proposed distance-based collocation search approach with polar selection showed superior performance, compared to structured or Voronoi-based approaches, for the present WLS-based mesh-free scheme on both regular and irregular point clouds. For the WLS approximation, the polar selection process ensures information from all directions while maintaining the compactness. For the inviscid cylinder flow case, the polar selection offered faster convergence than structured collocations, and the accurate/approximate Voronoi approaches, on both regular and irregular points. The polar selection also offered slightly better surface pressure solutions compared to Voronoi-based approaches.
4. This work further investigated adaptive, mesh-free, modelling with different adaptation strategies. The study examined point refinement mechanisms based on Cartesian points and middle point insertion and various collocation search methods. For the isentropic vortex transport problem, the simple distance-based collocation search could not improve the resolution with uniform Cartesian points due to inadequate support sizes and member selection. Enforcing polar and Voronoi selections managed to converge the adaptive modelling. The combination of Cartesian-based refinement and polar selection, offered the fastest convergence and best accuracy. When replaced with mid-point refinement, the adaptive modelling with polar and Voronoi selection both showed good convergence behaviour. These results exposed the importance of point distributions and suitable collocation search approaches for successful and efficient adaptive mesh-free modelling.
5. Adaptive mesh-free modelling was then 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 prioritises 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. The results highlighted the flexibility of mesh-free adaptive modelling when handling complex geometries and flow physics.

Overall, this work demonstrated the accuracy, convergence, and flexibility of adaptive simulations using collocation-based implicit mesh-free methods. New concepts, including the weight balancing, automatic point cloud generation using strand or level-set points, and the polar selection approach, were introduced and assessed on highly irregular points and geometries. Future work will improve the overall automation, flexibility, conservation properties, and apply the methods to more complex geometries and problems. More adaptation parameters for example, polynomial orders and weighting functions are to be evaluated.

ACKNOWLEDGEMENTS

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.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Tao Zhang https://orcid.org/0000-0003-2514-8901
George N. Barakos https://orcid.org/0000-0002-5029-8453

REFERENCES

1. Slotnick JP, Khodadoust A, Alonso J, et al. CFD Vision 2030 Study: a Path to Revolutionary Computational Aerosciences. Tech. rep. 2014.
2. Zhang T, Barakos GN. Development of simulation tools for high Fidelity analysis of compound rotorcraft. AIAA Scitech 2020 Forum. 2020 1258.
3. Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. *Comput Mech*. 1992;10(5):307-318.

4. Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. *Int J Numer Methods Eng*. 1994;37(2):229-256.
5. Atluri SN, Zhu T. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Comput Mech*. 1998;22(2):117-127.
6. Melenk J, Babuška I. The partition of unity finite element method: basic theory and applications. *Comput Methods Appl Mech Eng*. 1996;139(1):289-314.
7. Oñate E, Idelsohn S, Zienkiewicz O, Taylor R. A finite point method in computational mechanics. Applications to convective transport and fluid flow. *Int J Numer Methods Eng*. 1996;39(22):3839-3866.
8. Shu C, Ding H, Chen H, Wang T. An upwind local RBF-DQ method for simulation of inviscid compressible flows. *Comput Methods Appl Mech Eng*. 2005;194(18-20):2001-2017.
9. Batina JT. A Gridless Euler/Navier-Stokes Solution Algorithm for Complex Two-Dimensional Applications. Tech. rep. 1992.
10. Katz A, Jameson A. Edge-based meshless methods for compressible flow simulations. Paper presented at: 46th AIAA Aerospace Sciences Meeting and Exhibit. 2008 699.
11. Kennett D, Timme S, Angulo J, Badcock K. An implicit meshless method for application in computational fluid dynamics. *Int J Numer Methods Fluids*. 2013;71(8):1007-1028.
12. Munikrishna N, Balakrishnan N. Turbulent flow computations on a hybrid cartesian point distribution using meshless solver LSFD-U. *Comput Fluids*. 2011;40(1):118-138.
13. Kansa EJ. Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics—I surface approximations and partial derivative estimates. *Comput Math Appl*. 1990;19(8-9):127-145.
14. Jiang Y. Algebraic-volume meshfree method for application in finite volume solver. *Comput Methods Appl Mech Eng*. 2019;355:44-66.
15. Hopkins PF. A new class of accurate, mesh-free hydrodynamic simulation methods. *Monthly Notices R Astronom Soc*. 2015;450(1):53-110.
16. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices R Astronom Soc*. 1977;181(3):375-389.
17. Lucy LB. A numerical approach to the testing of the fission hypothesis. *Astronom J*. 1977;82:1013-1024.
18. Yousuf M, Balakrishnan N. Residual-based grid adaptation for meshless LSFD-U solver. *AIAA J*. 2019;57(4):1649-1658.
19. Ortega E, Flores R, Oñate E, Idelsohn S. A-posteriori error estimation for the finite point method with applications to compressible flow. *Comput Mech*. 2017;60:219-233.
20. Kwan-yu Chiu E, Wang Q, Hu R, Jameson A. A conservative mesh-free scheme and generalized framework for conservation laws. *SIAM J Sci Comput*. 2012;34(6):A2896-A2916.
21. Osher S, Solomon F. Upwind difference schemes for hyperbolic systems of conservation laws. *Math Comput*. 1982;38(158):339-374.
22. Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *J Comput Phys*. 1981;43(2):357-372.
23. Mavriplis DJ. Unstructured-mesh discretizations and solvers for computational aerodynamics. *AIAA J*. 2008;46(6):1281-1298.
24. Spalart P, Allmaras S. A one-equation turbulence model for aerodynamic flows. Paper presented at: 30th Aerospace Sciences Meeting and Exhibit. 1992 439.
25. Ortega E, Oñate E, Idelsohn S. An improved finite point method for tridimensional potential flows. *Comput Mech*. 2007;40(11):949-963.
26. Jameson A. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. Paper presented at: 10th Computational Fluid Dynamics Conference. 1991 1596.
27. Suchde P, Jacquemin T, Davydov O. Point cloud generation for meshfree methods: an overview. *Arch Comput Methods Eng*. 2023;30(2):889-915.
28. Katz A, Wissink AM, Sankaran V, Meakin RL, Chan WM. Application of strand meshes to complex aerodynamic flow fields. *J Comput Phys*. 2011;230(17):6512-6530.
29. Aurenhammer F, Klein R, Lee D-T. *Voronoi Diagrams and Delaunay Triangulations*. World Scientific Publishing Company; 2013.
30. Suchde P, Kuhnert J, Schröder S, Klar A. A flux conserving meshfree method for conservation laws. *Int J Numer Methods Eng*. 2017;112(3):238-256.
31. Bassi F, Rebay S. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J Comput Phys*. 1997;138(2):251-285.
32. Pascioni K, Cattafesta LN, Choudhari MM. *An Experimental Investigation of the 30P30N*. Multi-Element High-Lift Airfoil; 2014.
33. Murayama M, Yokokawa Y, Ura H, et al. Experimental study of slat noise from 30P30N three-element high-lift airfoil in JAXA Kevlar-wall low-speed wind tunnel. Paper presented at: 2018 AIAA/CEAS Aeroacoustics Conference. 2018 3460.
34. Schmitt V. Pressure Distributions on the ONERA M6-Wing at Transonic Mach Numbers, Experimental Data Base for Computer Program Assessment. AGARD AR-138. 1979.
35. Jimenez-Garcia A, Barakos G. Assessment of a high-order MUSCL method for rotor flows. *Int J Numer Methods Fluids*. 2018;87(6):292-327.
36. Cook P, McDonald M, Firmin M. Aerofoil rae 2822-pressure distributions, and boundary layer and wake measurements. Experimental data base for computer program assessment. *AGARD Report AR*. 1979;138:47.

How to cite this article: Zhang T, Barakos GN. Assessment of implicit adaptive mesh-free CFD modelling. *Int J Numer Meth Fluids*. 2024;1-31. doi: 10.1002/fld.5266