Computer Assisted Methods in Engineering and Science, **20**: 207–225, 2013.

TWENTY YEARS OF THE CAMES

A model-integrated localized collocation meshless method (MIMS)

S. Gerace, K. Erhart, A. Kassab
Mechanical and Aerospace Engineering Department, University of Central Florida
4000 Central Florida Blvd., Orlando, FL, 32816, USA
e-mail: sgerace@dotdecimal.com, kerhart@dotdecimal.com, alain.kassab@ucf.edu

E. Divo
Department of Mechanical Engineering, Embry-Riddle Aeronautical University
Daytona Beach, FL, USA
e-mail: divoe@erau.edu

A model integrated meshless solver (MIMS) tailored to solve practical large-scale industrial problems is based on robust meshless methods strategies that integrate a native model-based point generation procedures. The MIMS approach fully exploits strengths of meshless methods to achieve automation, stability, and accuracy by blending meshless solution strategies based on a variety of shape functions to achieve stable and accurate iteration process that is integrated with a newly developed, highly adaptive model generation employing quaternary triangular surface discretization for the boundary, a binary-subdivision discretization for the interior, and a unique shadow layer discretization for near-boundary regions. Together, these discretization techniques provide directionally independent, automatic refinement of the underlying native problem model to generate accurate adaptive solutions without need for intermediate user intervention. By coupling the model generation with the solution process, MIMS addresses issues posed by ill-constructed geometric input and pathologies often generated from solid models in the course of CAD design.

Keywords: meshless methods, heat transfer, large-scale problems.

1. Introduction

Mesh-based finite difference, finite element, and finite volume methods all share the requirement to define connectivity of the nodal unknowns within the solution domain, and, despite efforts to automate the mesh generation process, a considerable amount of time and human effort is expended in preparing and meshing the computational model for complex problem geometries. Meshless and mesh-reduction methods [1–7] seek to replace the underlying structured connectivity with an unstructured interpolation scheme inherently affording automation in the mesh generation and refinement process. These schemes have shown considerable promise in many application areas. However, much of meshless methods research is focused on generating new meshless techniques, while failing, with some notable exceptions [8, 9] to address the underlying cause for concern in model discretization relegating this class of techniques to rather specialized endeavors. Although meshless methods have not out-competed mesh-based techniques, considerable advances within the field have led to implementations that are generally at least as efficient as unstructured mesh-based techniques.

In this paper, we present model integrated meshless solver (MIMS) that is a meshless methodology building upon current meshless research aimed at analysis of industrially-relevant problems. At its core, MIMS is a strong-form localized collocation-based meshless method [7–26] which has been developed to be both robust as well as accurate under a variety of nodal configurations through a unique blend of existing meshless shape functions and a specialized point distribution process. MIMS specifically address the underlying issues which make mesh generation such a time consuming and tedious process by close integration of the model generation and solution process in an efficient adaptive process that does not require user interaction.

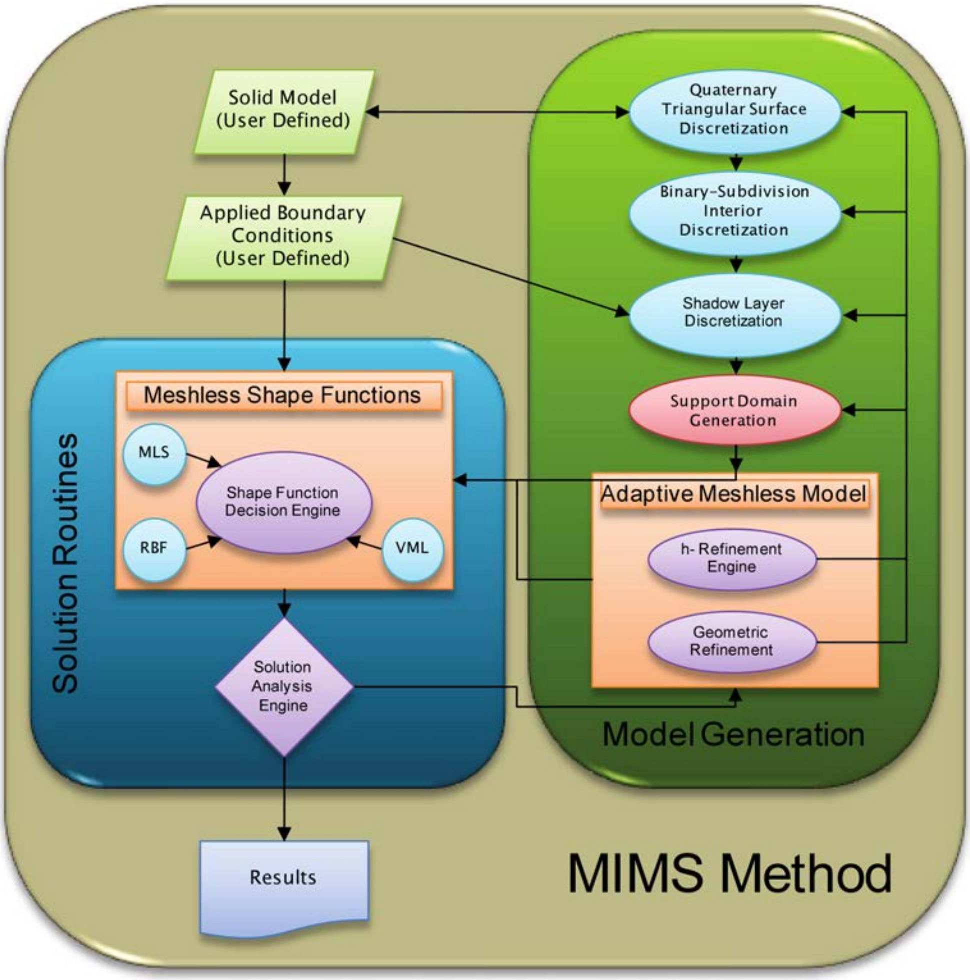

Solid Model (User Defined)
Applied Boundary Conditions (User Defined)
Quaternary Triangular Surface Discretization
Binary-Subdivision Interior Discretization
Shadow Layer Discretization
Support Domain Generation
Adaptive Meshless Model
h- Refinement Engine
Geometric Refinement
Model Generation
Solution Routines
Meshless Shape Functions
MLS
RBF
VML
Shape Function Decision Engine
Solution Analysis Engine
Results
MIMS Method

Fig. 1. The MIMS strategy.

The MIMS method utilized radial basis functions (RBF) interpolation in its discretization process. We present an overview of MIMS which consists of an engine that decides upon the underlying meshless discretization scheme with an automated adaptive model integrated point generation strategy and an automated shape function selection process. Case studies are then presented and MIMS results will be compared to a commercially available solution packages.

2. Meshless implemantation overview

We use a localized meshless approach to as the foundation of the solution mechanism [7–27] that is applied directly to a solution domain without special consideration for boundary condition application, as it is often the case when utilizing a non-interpolating approximation [1]. Localized implementation provides a minimum of computational and memory burden, and when used with collocation, permits use of a variety of interpolation schemes to develop the underlying shape functions for field and derivative evaluation. It is in the latter respect that the current method departs from current techniques in that no single interpolating method is used to construct the necessary shape functions. Instead, a blend of localized collocation meshless method based on radial basis function with polynomial reproduction (RBFP) interpolation [5–13], moving least squares (MLS) [4], and RBF-enhanced virtual finite differencing (VML) [20–26] is utilized to obtain a method that is both stable and accurate. This departure allows for a method which is not married to any particular interpolation scheme, and as such may take advantage of the relative strengths and weaknesses of each technique. We now briefly review all three methods as they apply to MIMS.

2.1. The localized meshless method framework

The meshless formulation begins by defining a set of data centers *NC* comprised of points on the boundary *NB* and points on the interior *NI*. These data centers will serve as collocation points for the localized expansion of the different field variables in the domain Ω and on the boundary Γ, see Fig. 2. The essential difference between boundary points and internal points is simply that boundary conditions will be applied at the boundary points while governing equations will be applied in strong form at the internal points. A localized expansion over a group or topology of influence points *NF* around each data center is sought such that

$$\varphi(x) = \sum_{j=1}^{NF} \alpha_j \chi_j(x) + \sum_{j=1}^{NP} \alpha_{j+NF} P_j(x). \tag{1}$$

The terms α_j are the unknown sought-after expansion coefficients while the terms $\chi_j(x)$ are expansion functions defined a-priori and we use inverse multiquadric radial basis functions [28–30] for the *NF* expansion functions $\chi_j(x)$. NP is a number of additional polynomial functions $P_j(x)$ added to the expansion to guarantee that constant and linear fields can be retrieved by the expansion exactly, thus the term radial basis function with polynomial reproduction (RBFP) is associated with this expansion. The time dependency has been dropped as a different expansion will be performed for each time level and, therefore, the expansion coefficients α_j will vary as time progresses. The selection of an influence region or localized topology of expansion around each data center is easily accomplished by a circular (spherical in 3D) search around each data center. The search is automated to guarantee that a minimum number of points will be included and additional criteria, such as including all directions around internal data centers, are met. In addition, this search must guarantee that topologies around boundary data centers do not include opposing boundaries or points around a re-entry corner.

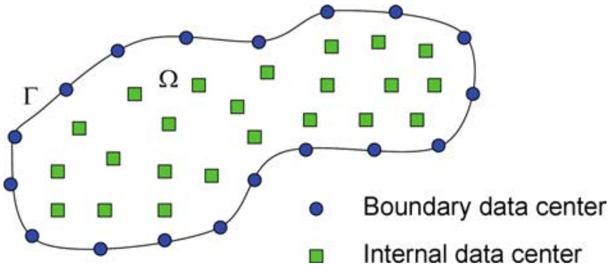


Fig. 2. Scattered point distribution in a generalized domain.

Figure 3 shows typical collocation topologies for internal and boundary data centers including re-entry corners and opposing boundaries and an example of the circular search to build the topology around an internal data center of a typical non-uniform point distribution The collocation of the known field variable φ (from previous time level or iteration step) at the points within the localized

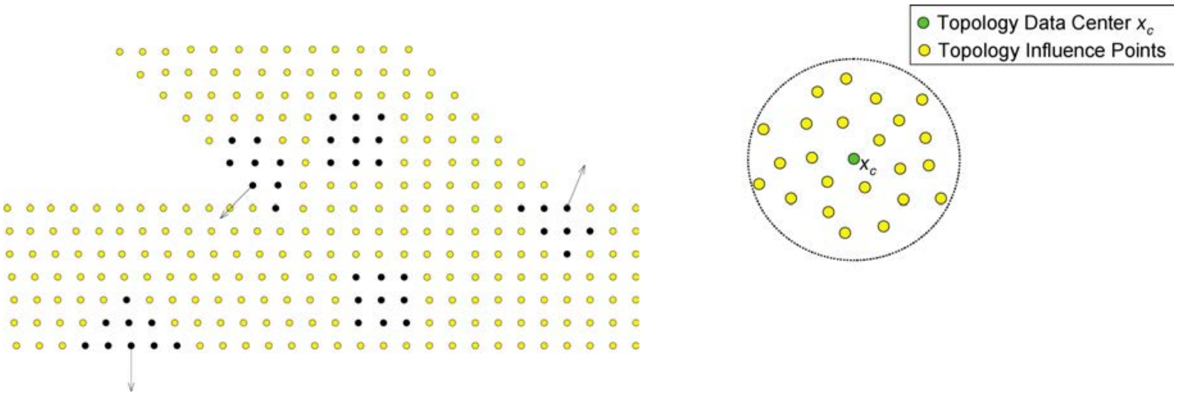


Fig. 3. Collocation topology for internal, boundary and corner data centers on uniform and non-uniform point distributions.

topology, leads to the matrix-vector form $\{\varphi\} = [C]\{\alpha\}$, so that the the expansion coefficients can be determined as $\{\alpha\} = [C]^{-1}\{\varphi\}$. The collocation matrix is

$$[C] = \begin{bmatrix} \chi_1(x_1) & \dots & \chi_{NF}(x_1) & P_1(x_1) & \cdots & P_{NP}(x_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \chi_1(x_{NF}) & \cdots & \chi_{NF}(x_{NF}) & P_1(x_{NF}) & \cdots & P_{NP}(x_{NF}) \\ P_1(x_1) & \cdots & P_1(x_{NF}) & 0 & \cdots & 0 \\ \dots & \ddots & \dots & \vdots & \ddots & \vdots \\ P_{NP}(x_1) & \cdots & P_{NP}(x_{NF}) & 0 & \cdots & 0 \end{bmatrix}_{NF+NP,NF+NP} . \tag{2}$$

A simple optimization search is employed to determine the value of the shape parameter c for every expansion over the different local topologies that cover the entire field. An initial guess for c is based on the ratio of the average distance between data centers in a topology to the number of points in the topology. A line search is performed to adjust the shape parameter c until the resulting collocation matrix $[C]$ yields a condition number in the range between 10^{11} and 10^{12} (in double-precision). This range of condition number for the collocation matrix $[C]$ has been documented to produce interpolations that render smooth derivative fields for a wide range of test functions. It is important to mention that the resulting collocation matrix $[C]$ depends only on the geometrical distribution of the points within each localized topology and therefore, the optimization of the shape parameter c is performed at a setup stage before the solution process begins. However, there may be instances when running-time optimization of the shape parameter c may be necessary as, for example, when adaptive refinement is performed or when sharp discontinuities in the solution field are found. In the latter caser the conditioning number should be reduced to capture the discontinuity. The real advantage of the localized collocation approach is capitalized in the way the derivatives of the field variable are calculated at the data center x_c of each topology. For instance, any linear differential operator L can be applied over the localized expansion equation as

$$L\phi(x_c) = \sum_{j=1}^{NF} \alpha_j L\chi_j(x_c) + \sum_{j=1}^{NP} \alpha_{j+NF} LP_j(x_c) \tag{3}$$

or

$$L\phi_c = \{L_c\}^T\{\alpha\},$$

where

$$\{L_c\}^T = \left\{ L\chi_1(x_c) \quad \dots \quad L\chi_{NF}(x_c) \quad LP_1(x_c) \quad \dots \quad LP_{NP}(x_c) \right\}_{NF+NP,1} .$$

Substitution of the expansion coefficients $\{\alpha\}$ leads to the key equation:

$$L\phi_c = \{L\}^T\{\phi\}, \tag{4}$$

where the operator vector is given by $\{L\}^T = \{L_c\}^T[C]^{-1}$. The coefficients of the vector $\{L\}$ of size $(NF, 1)$ directly interpolate the derivative of the field variable distribution $\{\phi\}$ at the data center of the topology x_c. Therefore, evaluation of the field variable derivatives at every and each of the data centers x_c is provided by a simple inner product of two small vectors: $\{L\}$ which can be pre-built and stored and $\{\phi\}$ which is the updated field variable distribution in the topology of the data center. Applying this approach to time marching, for example, using explicit first order finite differencing in time, the localized RBF interpolation is

$$\phi_c^{k+1} = \phi_c^k + \Gamma\Delta t\{L\}^T\{\phi\}^k, \tag{5}$$

where the superscript k denotes the time level, Γ is some material property, and Δt denotes the size of the time step. Given the solution $\{\phi\}$ at every point at the previous time level k can very efficiently yield the updated field variable value at each data center x_c through a simple inner product of small vectors. Imposing boundary conditions at the boundary data center x_c can be accomplished in a similar fashion. For Neuman and Robin's condition a normal derivative interpolation vector $\{\partial n\}$ is required to compute the normal derivative of the field variable $\{\phi\}$ at the boundary topology data center x_c. This normal derivative interpolation vector may be pre-built in one of two ways: (1) a simple approach is to express this vector as a combination of derivative vectors in all directions multiplied by their corresponding unit normal vector components, for instance, in 3D Cartesian coordinates, this is $\{\partial n\} = \{\partial x\}n_x + \{\partial y\}n_y + \{\partial z\}n_z$; and (2) generate additional internal points that "shadow" each boundary point in the direction of the normal vector into the domain Ω to directly approximate the normal derivatives at each boundary data center as illustrated in Fig. 4. We use the latter and more accurate "shadow point method" as it mitigates the inherent inaccuracies of the directional derivative interpolation vectors of the truncated topologies of boundary data centers especially of those around corners and highly curved boundaries. Following this approach, the normal derivative interpolation vector $\{\partial n\}$ can be computed simply as $\{\partial n\} = \left\{ \begin{matrix} 1/r_s & 0 & \ldots & 0 & -1/r_s & 0 & \ldots & 0 \end{matrix} \right\}^T$, where r_s is the distance from the boundary data center to its corresponding internal shadow point. Additionally, higher order differentiation can be accomplished by simply inserting multiple rows of internal shadow points in normal direction. In addition, this approach can be implemented to estimate not only the normal derivatives but the tangential derivatives as well.

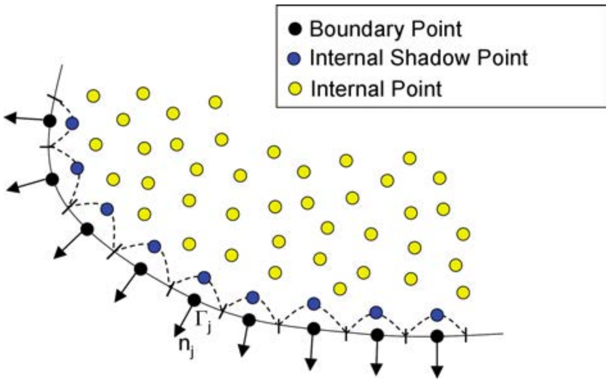


Fig. 4. Distribution of internal shadow points to compute normal derivatives.

The localized expansion approach reduces the burden of the more common global interpolation methods [31–33] by expanding the field variable locally around each data center to obtain its derivatives that can then be used if necessary in time-marching or iterative schemes. This approach yields the generation of multiple but small interpolation matrices rather than the large and fully-populated global interpolation matrix of the standard global interpolation methods. However, since the approach relies on expanding known values of the field variables, it is applicable as long as an explicit time-marching or iterative scheme is formulated and inapplicable directly to steady problems. This is not a drawback, as time marching can always be considered as a relaxation scheme for the iterative solution of steady-state problems.

Estimation of the field variables and their derivatives is accomplished by simple inner products of vectors that can be pre-built and the multiquadrics RBF only need to be evaluated at a setup stage when these vectors are being built. This reduces the CPU burden of having to evaluate fractional powers and complicated functions at every step of an iteration or time-marching scheme. In addition, the memory demands of this approach are minimal, as no global collocation matrix is allocated and only very small vectors are stored for each of the data centers. This offers tremendous advantages in terms of data preparation over global methods.

2.2. The moving least-squares (MLS) smoothing scheme

Despite the efforts in optimizing the shape parameter c of the RBF expansion for every topology, it is found that derivative fields, in particular odd one-sided derivatives, such as those found in convective terms of transport equations as well as in divergence operators, tend to oscillate specially towards the areas of large gradients such as re-circulating zones, corners, impingement zones, etc. For this reason, additional care must be taken when formulating the derivative expansion vectors within each topology. An effective method consists in the application of moving least-squares smoothing over the data center topology to approximate the derivative value at the data center. This method can be easily adapted to the localized meshless technique and, in addition, extended to be formulated in the same form where the derivative value at the data center of the topology is retrieved by a simple inner product of a vector that can be pre-built and stored, and the vector of field variable within the topology. This particular scheme retains the attractive efficiency feature of the localized meshless method. For instance consider the topology of NF influence points around the data center x_c, then, a least-squares expansion of the field variable $\phi(x)$, using NP polynomials $P_j(x)$, (where $NP < NF$) may be formulated as

$$\phi(x) = \sum_{j=1}^{NP} \alpha_j P_j(x), \tag{6}$$

with the expansion coefficients α_j found through a least-squares minimization process over all the NF influence points, leading to the standard normal equations:

$$\sum_{j=1}^{NP} \alpha_j \left[\sum_{k=1}^{NF} P_i(x_k) P_j(x_k) \right] = \sum_{k=1}^{NF} P_i(x_k)\phi(x_k) \tag{7}$$

or, in matrix-vector form: $[C]_{NP,NP}\{\alpha\}_{NP,1} = [P]_{NP,NF}\{\phi\}_{NF,1}$, where the coefficients $C_{i,j}$ of the least-squares matrix $[C]$ of size (NP, NP) are given by $C_{i,j} = \sum\limits_{k=1}^{NF} P_i(x_k)P_j(x_k)$. The expansion coefficients α_j are determined as $\{\alpha\}_{NP,1} = [C]^{-1}_{NP,NP}[P]_{NP,NF}\{\phi\}_{NF,1}$. Applying a linear differential operator L over the field variable $\phi(x)$ at the data center x_c and substituting for the MLS expansion coefficients α_j leads to

$$L\phi_c = \{L_{ls}\}^T_{1,NF}\{\phi\}_{NF,1}, \tag{8}$$

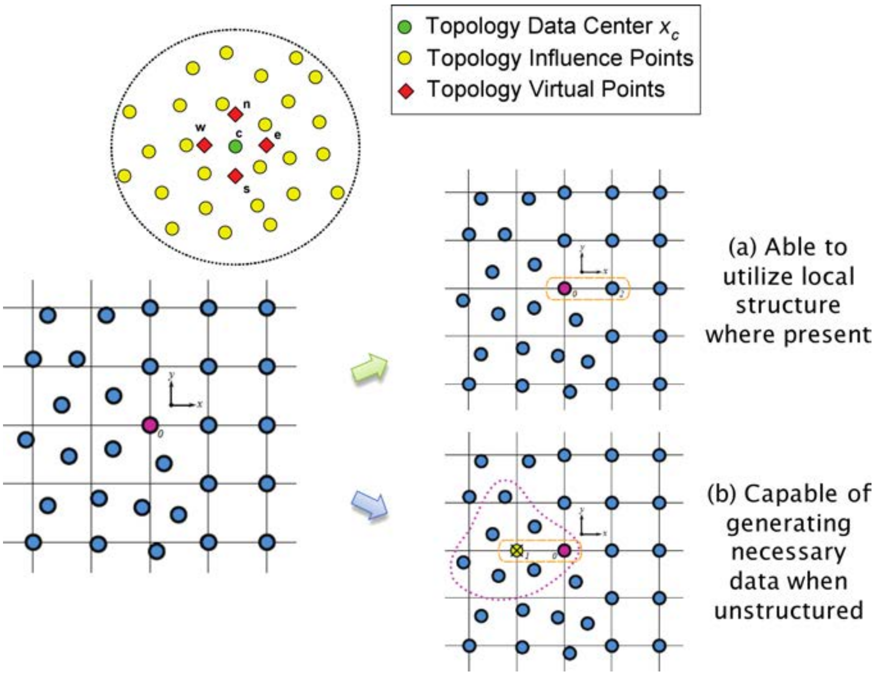
where the least-squares operator vector $\{L_{ls}\}$ is explicitly built as

$$\{L_{ls}\}^T_{1,NF} = \{LP_c\}^T_{1,NP}[C]^{-1}_{NP,NP}[P]_{NP,NF}.$$

This least-squares' smoothing scheme is performed over the same topology as the localized RBF collocation, hence moving least-squares (MLS), and this renders the methodology general in its application. The least-squares operator vectors $\{L_{ls}\}$ can be pre-built at the same pre-processing stage as the topology generation, shape parameter optimization, and RBF collocation. In addition, the least-squares expansion polynomials $P_j(x)$ are the same as those employed in the RBF collocation augmented formulation. The MLS smoothing scheme is often implemented to approximate one-sided derivatives present in transport equations to add stability to the iteration process [23–25]. It should be noted that this approach is not used in place of upwinding schemes that must be implemented in cases where the convective thresholds are surpassed as it is commonly found in forced convection problems. MLS is part of the suite of approximations utilized in MIMS as discussed in the sequel.

2.3. RBF-enhanced finite-differencing (VML)

There are difficulties encountered in the localized meshless method when dealing with steep gradients and highly convective fields and due in part to the inability of controlling how information is passed to the data center from the scattered points in the topology. A practical and general way around this problem is to take full advantage of the highly accurate field variable interpolation capabilities of the RBFs and not to directly differentiate them but, rather to utilize well-established finite difference upwinding formulations, for instance based on TVD or AUSM concepts, and RBFs to interpolate the field variable to locations required by the upwinding scheme if no data center is found at the required location. The approach is general in the sense that it can be implemented in the same localized topologies defined for the localized meshless method and formulated to yield pre-computed derivative interpolation vectors. A set of "virtual" points are distributed at locations required by finite-differencing evaluation of the derivatives, i.e., at the requisite n, s, e and w locations at which the field variable will be interpolated. The virtual spacing employed is consistent with the average spacing of the actual points within the localized topology. This is illustrated in Fig. 5. Evaluation of the derivatives by finite-differencing is performed through the RBF interpolation at the virtual locations.



Topology Data Center x_c
Topology Influence Points
Topology Virtual Points
n
w
c
e
s
(a) Able to utilize local structure where present
(b) Capable of generating necessary data when unstructured

Fig. 5. Illustration of a topology with virtual points around the data center x_c.

For example, the evaluation of the Laplace operator at the data center x_c reduces to

$$\nabla^2 \phi_c = \{L\}^T \{\phi\}, \tag{9}$$

where $\{L\}^T = \frac{1}{h^2}\left(\{\mathrm{I}_e\}^T + \{\mathrm{I}_n\}^T + \{\mathrm{I}_w\}^T + \{\mathrm{I}_s\}^T - 4\{\mathrm{I}_c\}^T\right)$ and where the interpolation vectors, $\{\mathrm{I}_e\}^T$, $\{\mathrm{I}_n\}^T$, are readily obtained from the RBFP interpolation. For instance, evaluating the field variable at the east point (real data center or virtual) results in matrix vector form $\phi_c = \{\Psi_e\}^T\{\alpha\}$, where $\{\Psi_e\} = \left\{\chi_1(x_e) \quad \ldots \quad \chi_{NF}(x_e) \quad P_1(x_e) \quad \ldots \quad P_{NP}(x_e)\right\}^T$. Substitution of the expansion coefficients α_j leads to $\phi_e = \{\Psi_e\}^T[C]^{-1}\{\phi\}$. Therefore, $\phi_e = \{\mathrm{I}_e\}^T\{\phi\}$, the east interpolation vector $\{\mathrm{I}_e\}$ is given by $\{\mathrm{I}_e\}^T = \{\Psi_e\}^T[C]^{-1}$. This can be easily extended to the other virtual points retaining the capability of rendering the derivative of the field variable $\phi(x)$ at the data center x_c through a simple inner product of two small vectors. The derivative interpolation vector $\{L\}$ can be pre-built and stored at a setup stage of the problem and implemented over the exact same topology employed for the localized RBF interpolation and moving least-squares smoothing. It is important to mention that any derivative operator can be expressed in this same form through a simple finite-differencing approximation over a set of virtual points in the topology. When a virtual point coincides with an actual meshless data point, the interpolation vector $\{\mathrm{I}\}$ at that virtual point, reduces to all zeroes except for a one at the vector entry of the coincident point. This dramatically reduces the setup time, especially around areas where the topologies are composed of uniformly-distributed points, as no shape parameter c optimization or matrix inversion is necessary to generate the derivative interpolation vectors.

3. Shape function selection

In order to obtain optimum speed, accuracy and robustness, MIMS utilizes a blend of radial basis function interpolation (RBFP), moving least squares (MLS) and virtual finite differencing (VML)

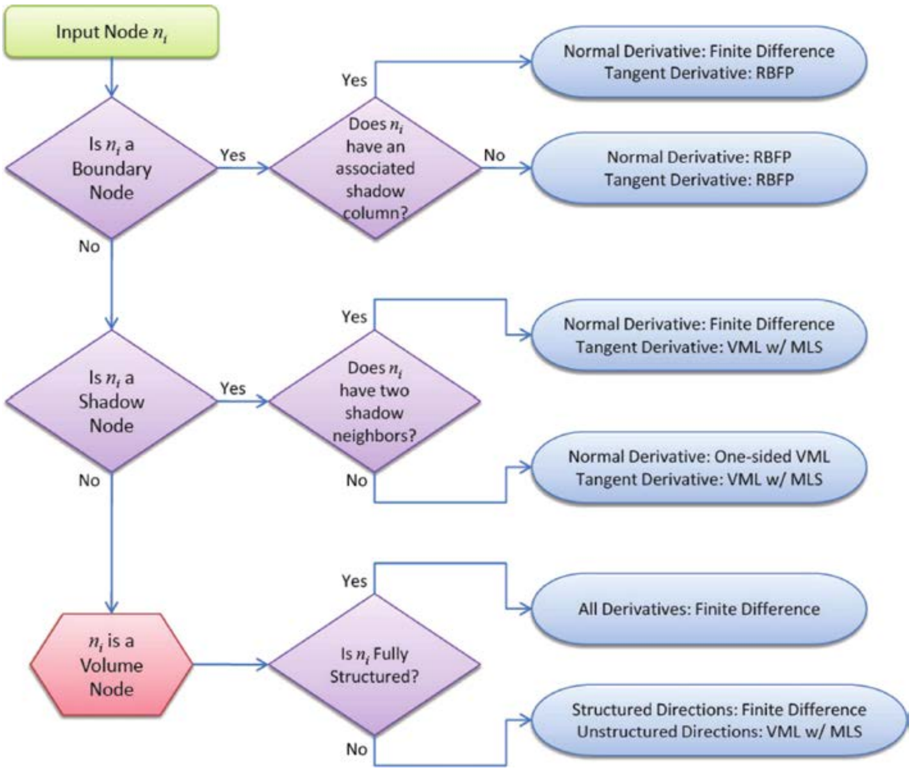

Input Node n_i
Is n_i a Boundary Node
Yes
No
Does n_i have an associated shadow column?
Yes
No
Normal Derivative: Finite Difference
Tangent Derivative: RBFP
Normal Derivative: RBFP
Tangent Derivative: RBFP
Is n_i a Shadow Node
Yes
No
Does n_i have two shadow neighbors?
Yes
No
Normal Derivative: Finite Difference
Tangent Derivative: VML w/ MLS
Normal Derivative: One-sided VML
Tangent Derivative: VML w/ MLS
n_i is a Volume Node
Is n_i Fully Structured?
Yes
No
All Derivatives: Finite Difference
Structured Directions: Finite Difference
Unstructured Directions: VML w/ MLS

Fig. 6. Shape function selection process.

to generate the necessary shape functions. Through an extensive study [37], it was decided that the process of selecting the most appropriate shape function construction technique may be summarized by the flowchart shown in Fig. 6.

4. Meshless model generation

An automated model generation technique has been developed which utilizes three fundamental techniques: quaternary triangular surface discretization, binary-subdivision interior discretization, and an adaptive boundary layer representation referred to as the shadow layer. Together, these three components make up the adaptive model generation procedure enabling an effectively automated meshless method technique [27, 37].

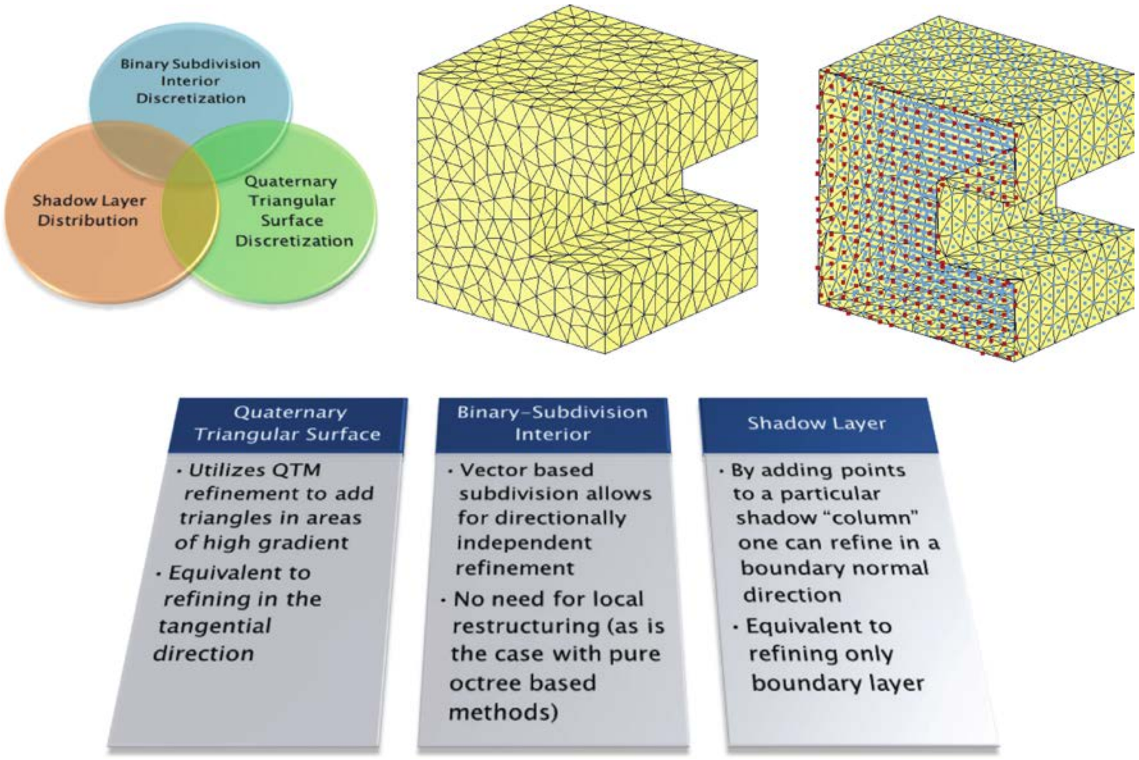

Binary Subdivision Interior Discretization
Shadow Layer Distribution
Quaternary Triangular Surface Discretization
Quaternary Triangular Surface
· Utilizes QTM refinement to add triangles in areas of high gradient
· Equivalent to refining in the tangential direction
Binary-Subdivision Interior
· Vector based subdivision allows for directionally independent refinement
· No need for local restructuring (as is the case with pure octree based methods)
Shadow Layer
· By adding points to a particular shadow "column" one can refine in a boundary normal direction
· Equivalent to refining only boundary layer

Fig. 7. The data center distribution in a u-shaped block with an interior plane highlighted.

Boundary nodes are distributed on the bounding surfaces initially using a triangulation [34–36] that is then selectively refined using a recursive quaternary triangulation structure. Quaternary triangulation is a recursive triangulation consisting of geometrically fixed splitting rules [38] and although they have been used to generate level-of-detail models for graphical applications [38, 39] as well as geophysical models for cartographic applications [40, 41] they have rarely been utilized to generate computational meshes [42–44]. A quaternary triangular mesh (QTM) is a surface meshing technique utilizing a recursive storage structure whereby each triangular element can store four child elements. Each child element is constructed by splitting each edge of the element at their respective midpoints and connecting the newly created vertices in a pair-wise fashion. The meshless nodes are placed at the vertices and also at the centroids of the elements to address the issue of narrow faces and to allow boundary nodes to further approach the edges of the geometry due to the configuration of the elements. Thus, once the QTM is constructed and refined to an appropriate initial level, computational nodes are then distributed at all non-edge vertices as well as at each leaf element centroid.

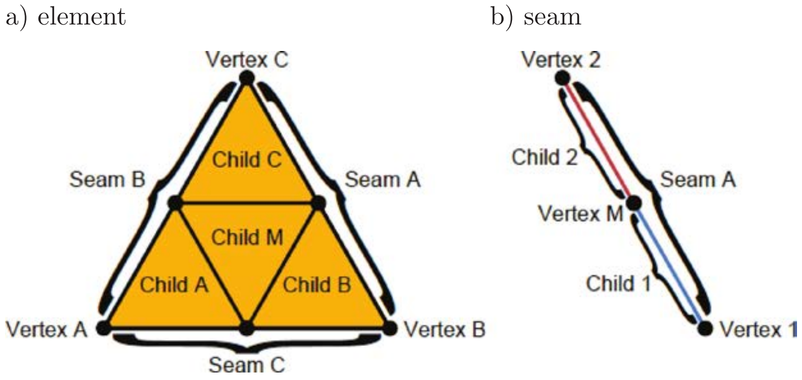


Fig. 8. Quaternary triangular constructs.

To address some issues with pure Cartesian based discretization techniques, such as the so-called stair-casing issue where irregular boundaries are present [45], and difficulties that octree and quadtree discretizations encounter in directionally independent refinement [46], we utilize a specially-tailored binary tree subdivisions for interior point distribution. The primary difference between the meshless binary tree discretization technique and previous attempts to extend the octree-based methods to non-isotropic refinement is the use of the vertex as the underlying data structure instead of the cell. Moving the point of view to the vertex allows for a more dynamic data structure capable of non-isotropic refinement as well as highly efficient topology construction [27, 37].

Finally, the meshless model generation process constructs a so-called "shadow layer" distribution that serves as an adaptive boundary layer for problems exhibiting high gradients normal to the boundary. Although handled slightly differently during model generation, it is important to note that nodes that reside in the shadow layer are, for all intents and purposes, interior nodes. No special boundary layer equations are solved, and the governing equations are collocated normally throughout this region of the domain (usually in a rotated coordinate frame).

The process of generating the shadow nodes is fairly straightforward and is illustrated on a simple two- dimensional boundary in Fig. 9. In this manner, normal derivatives may be obtained directly (both on the boundary and in the shadow layer) through finite differencing and tangential derivatives are generated in the usual meshless manner. The placement of the outer shadow layer (one farthest from boundary) is based on the distance to the nearest interior node (generally half of this value) and subsequent layers are distributed using an appropriate scaling method determined from field characteristics. It is important to realize that the process of adding shadow nodes can potentially introduce problems in highly concave boundary situations, as illustrated in Fig. 10 and, in such cases, collapsing of shadow nodes at the center of mass for the offending set of shadow nodes is necessary to eliminate instability in the underlying meshless interpolations. As the shadow layer is directly aligned with the normal and tangential directions on the boundary, it becomes trivial to produce high-aspect ratio point distributions with respect to the boundary orientation.

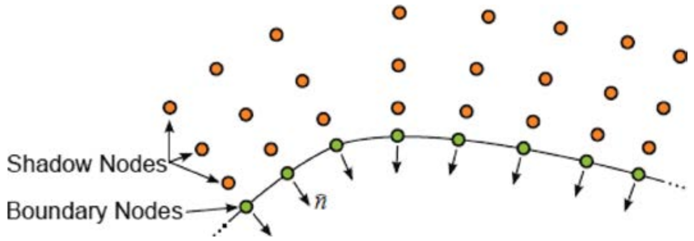


Fig. 9. Shadow layer.

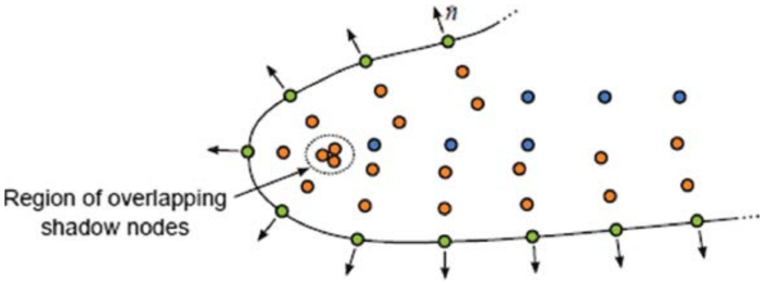
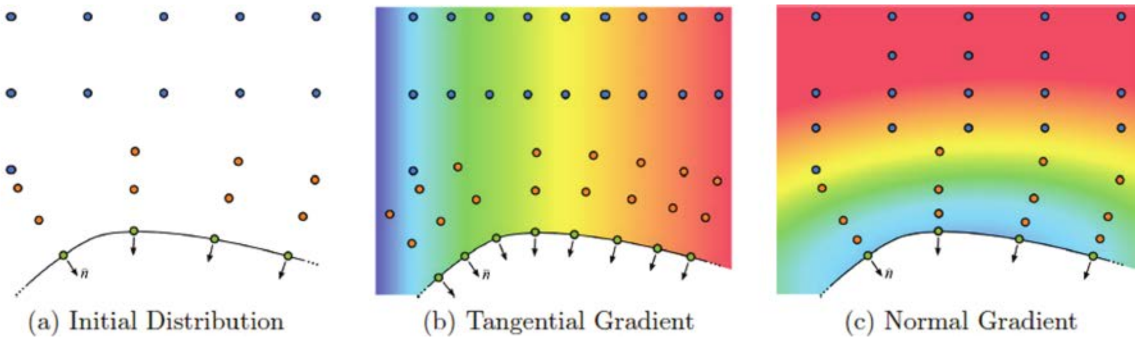


Fig. 10. Concave shadow layer region.

5. Adaptive refinement

Adaptive solution refinement is effected on two levels: (1) point distribution h-refinement strategy and (2) refinement on the underlying geometric model. Adaptive point distribution refinement strategy is performed by reducing the average nodal spacing in areas of high gradients through addition of new interior and boundary data centers as needed. The quaternary triangular surface recursive subdivision process selectively refines the point distribution on the boundary reducing spacing in the tangential direction to appropriately capture high tangential gradients. At the interior, the vertex-based structure binary-subdivision discretization provides anisotropic based on high gradients to capture flow phenomenon such as wakes and bulk flow turbulence without a need to identify these locations prior to solving the problem. At the boundary/interior interface regions, shadow layer refinement is utilized to refine in areas with large gradients normal to the boundary. Thus, all three model generation subsystems play a pivotal role in mesh refinement as illustrated in Fig. 11.

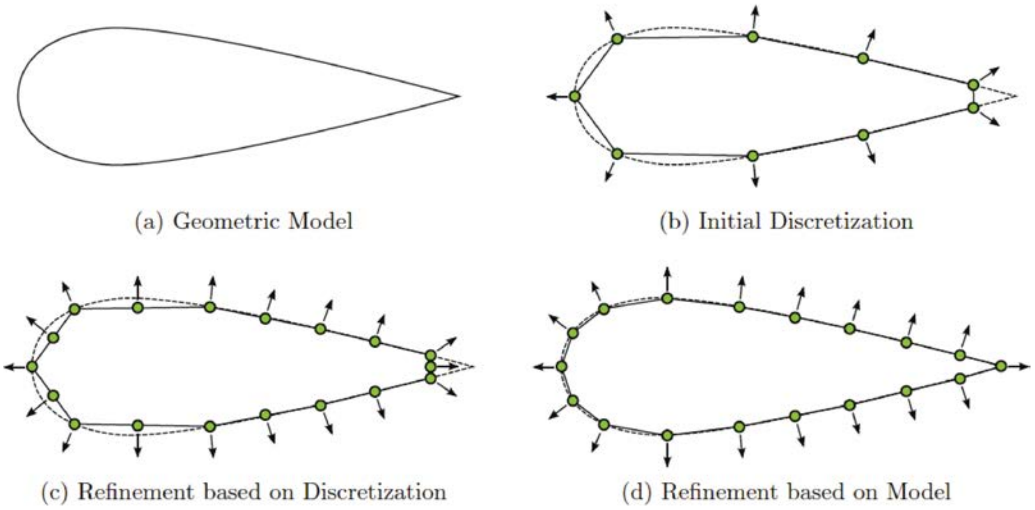


(a) Initial Distribution (b) Tangential Gradient (c) Normal Gradient

Fig. 11. Refinement examples.

The second type of refinement is that of the computational representation of the underlying geometric model, a matter often overlooked in traditional automatic refinement processes. The basic concept is that as the computational discretization (whether it will be mesh- or point-based) is refined, it should better represent the underlying geometric model. This is possible only if the point generation process has access to the underlying CAD model and the analytical surface representations (such as IGES or STEP) of the computational model. This affords discrete representation of the geometry at arbitrary resolutions as shown in Fig. 12.

The refinement process involves a three-stage approach: (1) refinement sites are selected based on their current field (gradient) value and past solution history, (2) distance-scaled gradients are calculated across the potential refinement sites (for the indicated refinement field), and (3) sites containing the highest gradients are isolated and refined. The first stage, selecting valid refinement sites, is arguably the most important as it provides a means of automatically obtaining grid convergence. However, if gradient values alone were the criteria on which refinement was based, the


(a) Geometric Model

(b) Initial Discretization

(c) Refinement based on Discretization

(d) Refinement based on Model

Fig. 12. Geometric refinement example.

same percentage of nodes would continue to be selected for refinement since they will always have the highest gradient values. As the goal of grid convergence is to achieve little if no change in solution from one discretization level to the next, the field values immediately following the previous refinement level are maintained and compared to the converged solution at the current level. If a nodal value did not change by a user-set percentage (say 1–2%) relative to the total field span, then it cannot cause a refinement site to be created (though it may still be part of a refinement site if a nearby node passes this threshold). Therefore, by controlling this refinement threshold, one can essentially enforce local grid convergence across the solution domain. Once the candidate refinement sites have been determined, the distance-scaled gradients are computed for each location. Once these gradients have been calculated, the values are sorted by magnitude and the top percentage threshold is chosen for refinement (in practice, a threshold which includes all sites with a gradient value in the top 20th percentile obtains a good balance of iteration and refinement speed). Once refinement has occurred, the solution is allowed to iterate until an acceptable convergence criteria has been met, at which point the refinement process is employed again. The benefit of this procedure is that the task of identifying when the solution has reached grid convergence is removed from the user; instead, the process continues indefinitely until there are no more remaining candidate refinement sites, indicating that no change has occurred from the previous refinement level, and that the solution has been successfully grid converged.

6. Examples

We present results from two examples involving MIMS solution of high speed compressible flow where the AUSM upwinding scheme has been adopted along with the Wilcox k-ω SST turbulence model [27]. Two well-established commercially available analysis packages were utilized throughout these case studies, denoted COMMERCIAL 1 and COMMERCIAL 2. In addition, all meshing operations were performed via a commercially available grid generation software denoted GRID GENERATOR 1. All Microsoft Windows based solutions (MIMS and COMMERCIAL 1) were performed on a system consisting of an AMD Phenom X4 3.00 GHz processor with 8 GB of RAM, while all UNIX based solutions (COMMERCIAL 2 and GRID GENERATOR 1) were performed on a Xeon 3.06 GHz processor with 4 GB of RAM. Also note that although both systems over multi-core/processor functionality, all timed operations were performed in single- threaded mode to ensure an accurate comparison. Finally, it is worth mentioning that the same person performed all necessary mesh generation and solution setup tasks for all three analysis packages (COMMERCIAL 1 and 2 as well as MIMS) in order to provide a fair comparison in skill level between each system and that person has been using GRID GENERATOR 1 to construct analysis meshes for approximately 5 years and, as such, can be considered quite knowledgeable in the area of mesh generation, and representative of an appropriately well-trained engineer.

The first problem considers a straight film cooling geometry in a rectangular channel with three-angle cooling jets, see Fig. 13.

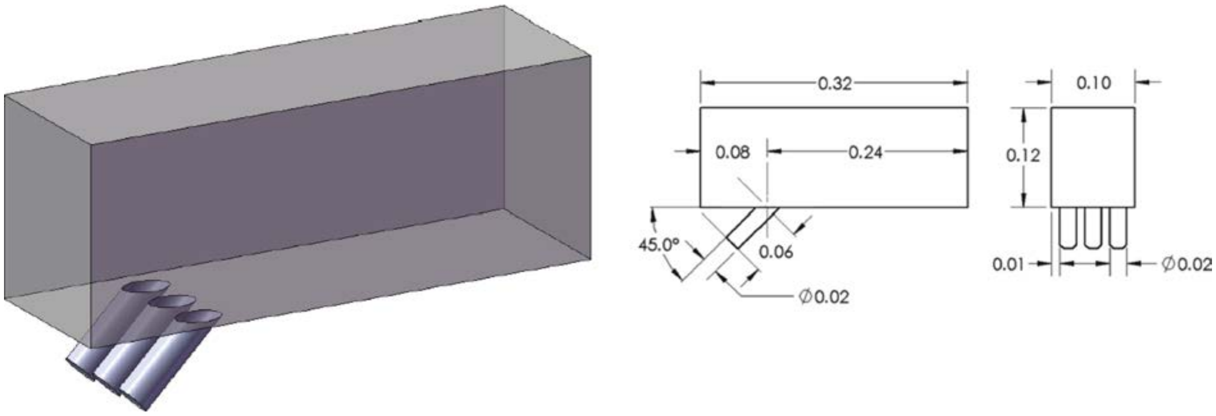


Fig. 13. Film cooling jet and chamber configuration.

To simulate film-cooling, the bulk flow inlet (located at the short end of the rectangular channel) was given conditions of $P_0 = 100\,000$ Pa and $T_0 = 800$ K, corresponding to an inlet Mach number of approximately $M = 0.4$. The outlet (located at the opposite end of the channel) was given an outlet pressure of $P_{out} = 70\,000$ Pa, and the bottom surface of the channel was set as no-slip (to produce an appropriate boundary layer). The three inlet jets were given conditions of $P_0 = 150\,000$ Pa and $T_0 = 500$ K, resulting in a slightly increased injection speed of approximately $M = 0.68$. All other walls (including the cylindrical sides of the jet injections) were given full-slip (zero shear) boundary conditions. It is worth pointing out that the jets were angled 45° into the flow, to allow for a more natural transition for the incoming, cooled fluid into the boundary layer. The structured mesh of Commercial 2 that consists of 476 889 cells and 499 350 nodes required roughly 2 hours and 15 minutes to produce while the unstructured mesh of Commercial 1 that is less refined and consists of 594 804 cells and 113 593 nodes took 27 minutes to generate, see Fig. 14. The original MIMS point distribution consists of 98 903 points as illustrated in Fig. 15.

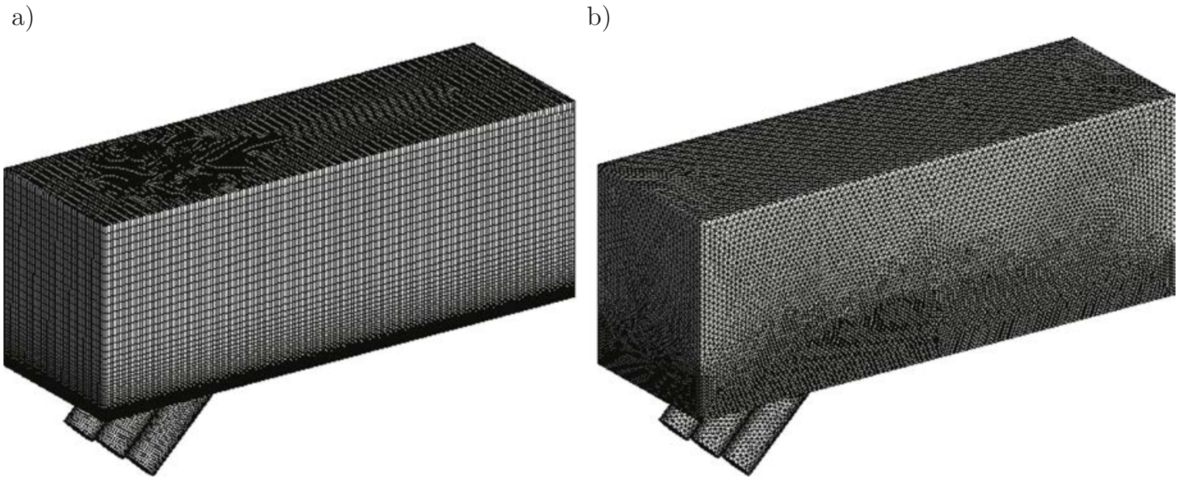


Fig. 14. Commercial code meshes of the film cooling jet configuration:
a) structure mesh (COMMERCIAL 2), b) unstructured mesh (COMMERCIAL 1).

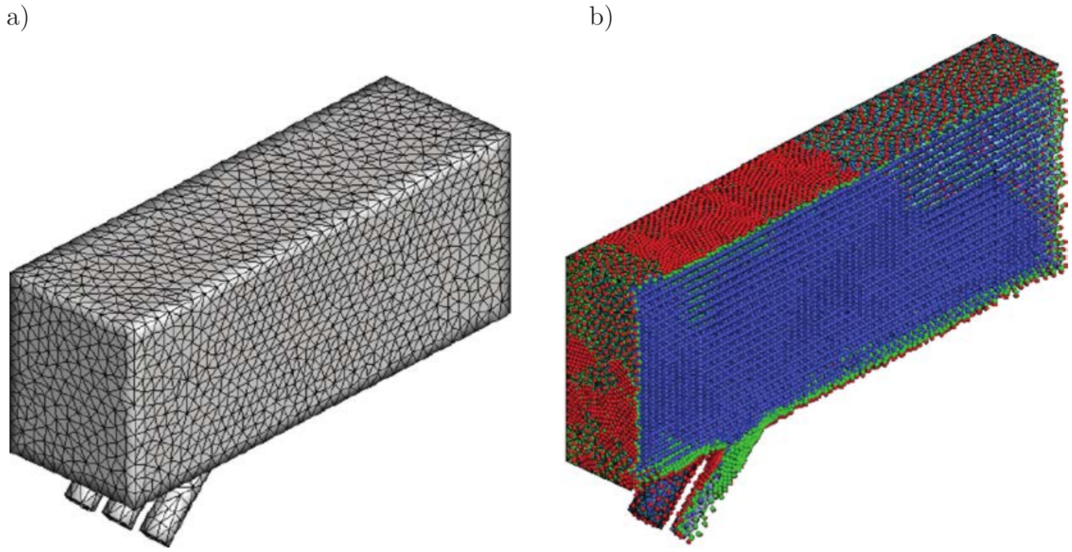
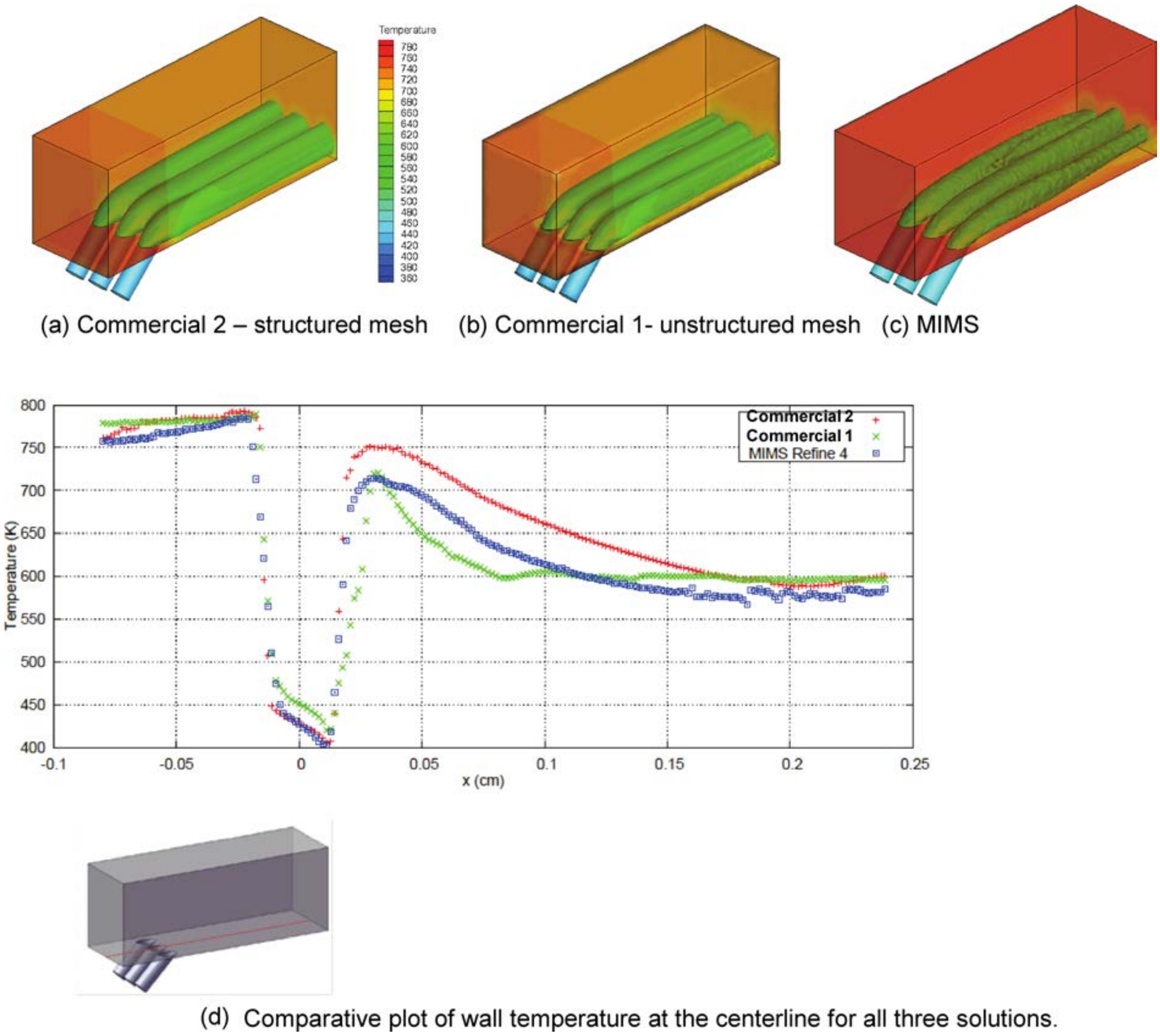


Fig. 15. Original MIMS point discretization of the film cooling jet configuration: a) initial QTM discretization, b) final point distribution.



(a) Commercial 2 – structured mesh (b) Commercial 1- unstructured mesh (c) MIMS

(d) Comparative plot of wall temperature at the centerline for all three solutions.

Fig. 16. Plot of 600 K isotherm and comparative plot of wall temperature.

To visualize the results, all three models were plotted on the same temperature scale (360 K to 780 K with 22 levels) and an isosurface was generated at the $T = 600$ K value as well as the centerline lower wall temperature was plotted as seen in Fig. 16. Good agreement is found from all three methods and meshes/point distributions. It is noted that the MIMS results are obtained after four levels of point refinement. The timings for the problem setup and solution for all three models are provided in Table 1. The MIMS process with its automatic refinement is able to complete the solution in considerably less time than either of the other two solution techniques. In addition, although the structured and unstructured mesh solutions were only 30 minutes apart, the structured mesh required almost two hours of more engineering time than the unstructured. This, coupled with the fact that all three results generate comparable solutions, provides justification for the use of a MIMS approach when requiring quick analysis of components for flow fields.

Table 1. Timings for problem setup and solution.

Total Solution Times

Task	Time
a) Commercial 1	
Mesh Setup*	02:15:00
Problem Setup*	00:05:00
Solve Time	02:16:35
Total	04:36:35
* Total Engineer Time	02:20:00
b) Commercial 2	
Mesh Setup*	00:27:00
Problem Setup*	00:05:00
Solve Time	03:32:02
Total	04:04:02
* Total Engineer Time	00:32:00
c) MIMS	
Problem Setup*	00:05:00
Initial Preprocessing	00:02:18
Tolal Solve Time	00:27:31
Total Refine Time	00:29:26
Total	01:13:15
* Total Engineer Time	00:05:00

The second example demonstrates the accuracy and adaptability of the proposed technique when presented with sharp discontinuities in an underlying flow field. To accomplish this, a nozzle presented by Hoffman is solved where the cross sectional area is given as $S(x) = 1.398 + 0.347$ tanh (0.8×-4) with the nozzle inlet and outlet located at $x = 0$ m and $x = 7$ m, respectively. The problem geometry is illustrated in Fig. 17, with the small end of the nozzle (inlet) located at $x = 0$ m and the larger end (outlet) located at $x = 7$ m. Assuming inlet conditions of $M = 1.5$, $P_0 = 100\,000$ Pa, $T_0 = 300$ K, and $P = 27\,240.3$ Pa, and an outlet pressure $P_{out} = 66\,809.6$ Pa, there is a normal shock within the nozzle located at $x = 5$ m.

The Commercial 2 software was used to solve this problem using a structured mesh with its native grid adaptation scheme leading to a final mesh of 415 450 cells or 437 346 nodes, and MIMS was used to solve the same problem with four refinement stages resulting in 88 429 nodes with an aspect ratio of approximately 10:1. The final surface point distribution and the mid-plane point

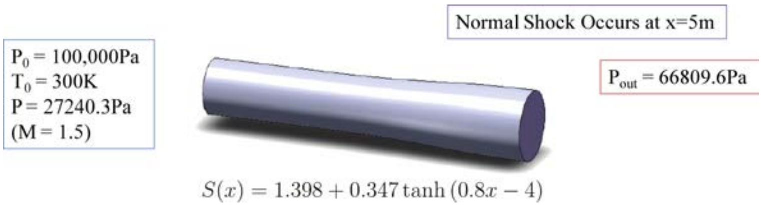
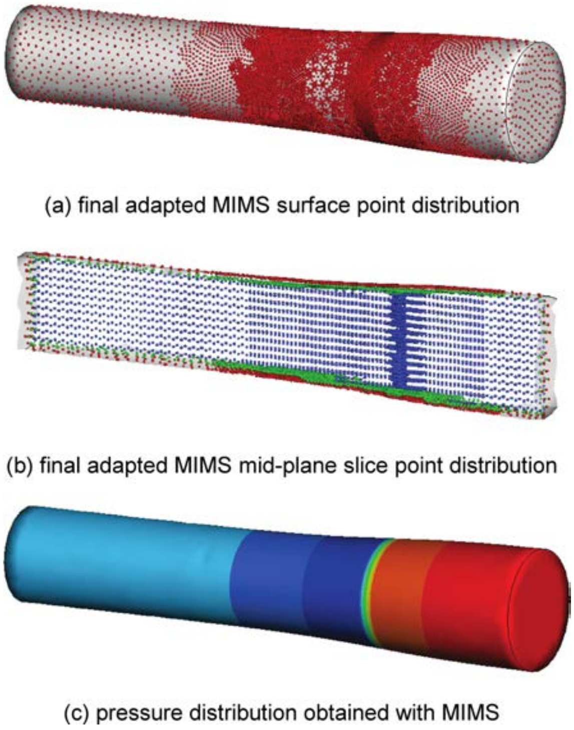


Fig. 17. Normal shock nozzle geometry.

distribution, are shown in Fig. 18, while, the midline pressure distribution is displayed in Fig. 19 comparing MIMS computed solution with Commercial 2 predictions. The shock is captured and the meshless process takes considerably less time to arrive at its converged result.


(a) final adapted MIMS surface point distribution

(b) final adapted MIMS mid-plane slice point distribution

(c) pressure distribution obtained with MIMS

Fig. 18. Surface and interior MIMS adapted point distibution and pressure solution.

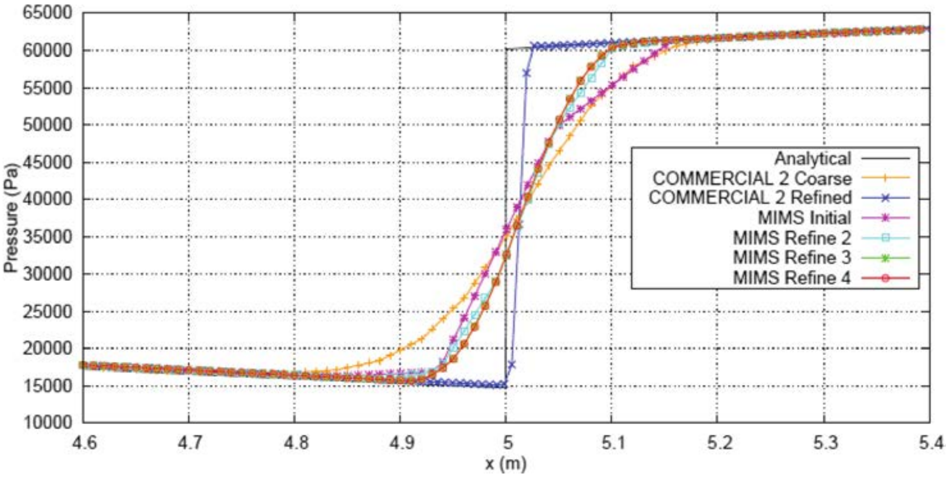


Fig. 19. Plot of pressure at midline showing shock capturing.

Table 2. Total timings for shock nozzle and computations.

Task	Time
a) Meshless	
Problem Setup*	00:02:00
Initial Preprocessing	00:01:12
Total Solve Time	00:10:24
Total Refine Time	00:07:41
Total	00:22:17
* Total Engineer Time	00:02:00
b) Commercial 2	
Mesh Setup (0.02 m)*	00:05:00
Problem Setup (0.02 cm)*	00:02:00
Solve Time (0.02 cm)	00:24:36
Mesh Setup (Clustered)*	00:12:00
Problem Setup (Clustered)*	00:02:00
Solve Time (Clustered)	02:12:27
Total	02:58:03
* Total Engineer Time	00:21:00

7. Conclusions

We presented an industrially relevant, numerical physics solution process implementing a novel meshless method. Collectively referred to as the model integrated meshless solution method, or MIMS, this methodology incorporates both a unique meshless implementation utilizing a variety of interpolation techniques as well as a novel model generation process capable of automatically generating point distributions for arbitrarily complex geometries. It is the development and fusion of these techniques which represent the primary contribution of this research. The overarching theme is that a competitive industrially relevant level, meshless method must be tightly integrated with the model generation process.

References

[1] G.R. Liu. *Mesh free methods: moving beyond the finite element method.* CRC Press, Boca Raton, FL, 2003.

[2] B. Nayroles, G. Touzot, P. Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*, **10**(5): 307–318, 1992.

[3] T. Belytschko, L. Gu, Y.Y. Lu. Fracture and crack growth by element free Galerkin methods. *Modeling and Simulation in Materials Science and Engineering*, **2**(3): 519–534, 1994.

[4] Y.Y. Lu, T. Belytschko, L. Gu. A new implementation of the element free Galerkin method. *Computer methods in applied mechanics and engineering*, **113**(3): 397–414, 1994.

[5] J.G. Wang, G.R. Liu. Radial point interpolation method for elastoplastic problems. In *Proceedings of the 1st International Conference on Structural Stability and Dynamics*, pp. 703–708, Taipei, Taiwan, 2000.

[6] J.G. Wang, G.R. Liu. Radial point interpolation method for no-yielding surface models. In *Proceedings of the First M.I.T. Conference on Computational Fluid and Solid Mechanics*, pp. 538–540, Cambridge, MA, 2001.

[7] G. Fasshauer. RBF collocation methods as pseudo-spectral methods. In *Proc. of Boundary Elements XVII*, A. Kassab, C.A. Brebbia, E. Divo, eds. WIT Press, pp. 45–57, 2005.

[8] G. Kosec. *Local meshless methods for multiphase fluid problems, Ph.D. Dissertation.* University of Nova Gorica, Slovenia, 2010.

[9] G. Kosec, B. Sarler. H-adaptive local radial basis function collocation meshless ethod. *CMC: Computers, Materials & Continua,* **26**: 227–54, 2011.

[10] E. Divo, A. Kassab. An efficient localized RBF meshless method applied to fluid flow and conjugate heat transfer. *ASME Paper IMECE – 2005-82150*, 2005.

[11] E. Divo, E.A. Kassab. A meshless method for conjugate heat transfer. Engineering Analysis, **29**: 136–149, 2005.

[12] B. Šarler. Advances in meshfree techniques, vol. 5 of *Computational Methods in Applied Sciences*, chap. from Global to local radial basis function collocation method for transport phenomena, Springer-Verlag, pp. 257–282, 2007.

[13] A. Kassab, E. Divo. An efficient localized radial basis function meshless method for fluid flow and conjugate heat transfer. *ASME Journal of Heat Transfer*, **129**: 179–183, 2007.

[14] B. Šarler, R. Vertnik. Meshfree explicit local radial basis function collocation method for diffusion problems. *Computers and Mathematics with Applications*, **51**: 1269–1282, 2006.

[15] R. Vertnik, B. Šarler. Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems. *International Journal of Numerical Methods for Heat and Fluid Flow*, **16**: 617–640, 2006.

[16] R. Vertnik, B. Šarler. Solution of incompressible turbulent flow by a mesh-free method. *CMES: Computer Modeling in Engineering & Sciences*, **44**: 65–95, 2009.

[17] G. Kosec, B. Šarler. Local RBF collocation method for Darcy flow. *CMES: Computer Modeling in Engineering & Sciences*, **25**: 197–208, 2008.

[18] G. Kosec, B. Šarler. Meshless aproach to solving freezing driven by a natural convection. *Materials Science Forum*, **649**: 205–210, 2008.

[19] G. Kosec, B. Šarler. Solution of phase change problems by collocation with local pressure correction. *CMES: Computer Modeling in Engineering & Sciences*, **47**: 191–216, 2009.

[20] S. Gerace. An interactive framework for meshless methods analysis in computational mechanics and thermofluid. *Master's thesis, University of Central Florida*, 2007.

[21] S. Gerace, K. Erhart, E. Divo, A. Kassab. Generalized finite difference meshless method in computational mechanics and thermofluids. In: *Proceedings of the Int. Conf. on Comp. Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS*, 2009.

[22] R. Vertnik, B. Sarler. Solution of incompressible turbulent flow by a mesh-free method. *Computer Modeling in Engineering and Sciences*, **44**(1): 65–95, 2009.

[23] Z. El-Zahab, E. Divo, A.J. Kassab. Minimization of the wall shear stress gradients in bypass grafts anastomoses using meshless CFD and genetic algorithms optimization. *Computer Methods in Biomechanics and Biomedical Engineering*, **13**(1): 35–47, 2010.

[24] Z. El-Zahab, E. Divo, A.J. Kassab. A localized collocation meshless method (LCMM) for incompressible flows CFD modeling with applications to transient hemodynamics. *Engineering Analysis*, **33**: 1045–1061, 2009.

[25] Z. El Zahab, E. Divo, A.J.Kassab. A meshless CFD approach for evolutionary shape optimization of bypass grafts anastomoses. *Journal Inverse Problems in Science and Engineering*, **17**(3): 411–435, 2009.

[26] K. Erhart, S. Gerace, E. Divo, A.J. Kassab. An RBF interpolated generalized finite difference meshless method for compressible turbulent flows. *AMSE Paper IMECE-2009-11452*, 2009.

[27] S. Gerace, K. Erhart, E. Divo, A.J. Kassab. Adaptively refined hybrid FDM/meshless scheme with applications to laminar and turbulent flows. *CMES: Computer Modeling in Engineering and Science*, **81**(1): 35–68, 2011.

[28] M.J.D. Powell. The Theory of Radial Basis Function Approximation. In *Advances in Numerical Analysis*, Vol. II, W. Light, ed., Oxford Science Publications, Oxford, pp. 143–167, 1992.

[29] M.D. Buhmann. *Radial Basis Functions: Theory and Implementation.* Cambridge University Press, Cambridge, 2003.

[30] R.L. Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research*, **176**: 1905–1915, 1971.

[31] E.J. Kansa. Multiquadrics- a scattered data approximation scheme with applications to computational fluid dynamics. I-surface approximations and partial derivative estimates. *Comp. Math. Appl.,* **19**: 127–145, 1990.

[32] E.J. Kansa. Multiquadrics- a scattered data approximation scheme with applications to computational fluid dynamics. II-solutions to parabolic, hyperbolic and elliptic partial differential equations, *Comp. Math. Appl.,* **19**: 147–161, 1990.

[33] E.J. Kansa, E.J.Y.C. Hon. Circumventing the Ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations. *Comp. Math. Appl.*, **39**: 123–137, 2000.

[34] M. Bern, D. Eppstein. Computing in Euclidean geometry chap. Mesh Generation and Optimal Triangulation. *Lecture Notes Series on Computing, World Scientific*, **23**(90), 1992.

[35] L.P. Chew. Guaranteed-quality mesh generation for curved surfaces. In *SCG '93: Proceedings of the ninth annual symposium on Computational geometry*, ACM, San Diego, California, USA, 1993.

[36] J. Ruppert, A. Delaunay. Refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, **18**(3): 548–585, 1995.

[37] S. Gerace. A model integrated meshless solver (MIMS) for fluid flow and heat transfer. *Ph.D. Dissertation, University of Central Florida*, 2010.

[38] L. De Floriani, E. Puppo. Hierarchical triangulation for multiresolution surface description. *ACM Trans. Graph.*, **14**(4): 363–411, 1995.

[39] X. Zhao, J. Chen, Z. Li. A QTM-based algorithm for generation of the Voronoi diagram on a sphere. In *Advances in Spatial Data Handling: Proceedings from the 10th International Symposium on Spatial Data Handling*, D. Richardson and P. van Oosterom, eds., pp. 269–285, Springer Berlin, 2002.

[40] G. Dutton. Encoding and handling geospatial data with hierarchical triangular meshes. In *Advances In GIS Research II: Proceedings of the Sixth International Symposium on Spatial Data Handling*, M.J. Kraak and M. Molenaar, eds., pp. 505–518, Taylor & Francis Publishing, 1996.

[41] G.H. Dutton. *A Hierarchical Coordinate System for Geoprocessing and Cartography*, vol. 79 of *Lecture Notes in Earth Sciences*, chap. Computational aspects of a quaternary triangular mesh, Springer Berlin Heidelberg, pp. 41–70, 1999.

[42] P. Frey, H. Borouchaki. Geometric surface mesh optimization. *Computing and Visualization in Science*, **1**(3): 113–121, 1998.

[43] D. Wang, O. Hassan, K. Morgan, N. Weatherill. EQSM: an efficient high quality surface grid generation method based on remeshing. *Computer Methods in Applied Mechanics and Engineering*, **195**(41): 5621–5633, 2006.

[44] J. Tournois, P. Alliez, O. Devillers. Interleaving Delaunay refinement and optimization for 2D triangle mesh generation. In *Proceedings of the 16th International Meshing Roundtable*, M.L. Brewer and D. Marcum, eds., Springer Berlin Heidelberg, Seattle, Washington, USA, pp. 83–101, 2007.

[45] H. Luo, J.D. Bauma, R. Lohner. A hybrid Cartesian grid and gridless method for compressible flows. *Journal of Computational Physics*, **214**(2): 618–632, 2006.

[46] Z.J. Wang, R.F. Chen, N. Hariharan, A.J. Przekwas, D. Grove. A 2n tree based automated viscous Cartesian grid methodology for feature capturing. *AIAA, Paper AIAA-99-3300*, 1999.

[47] P. Lancaster, K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, **37**(155): 141–158, 1981.