1-1-2011

Adaptively Refined Hybrid FDM-RBF Meshless Scheme with Applications to Laminar and Turbulent Viscous Fluid Flows

S. Gerace
University of Central Florida

K. Erhart
University of Central Florida

E. Divo
University of Central Florida

A. Kassab
University of Central Florida

Recommended Citation

Gerace, S.; Erhart, K.; Divo, E.; and Kassab, A., "Adaptively Refined Hybrid FDM-RBF Meshless Scheme with Applications to Laminar and Turbulent Viscous Fluid Flows" (2011). *Faculty Bibliography 2010s*. 1317.
https://stars.library.ucf.edu/facultybib2010/1317

Adaptively Refined Hybrid FDM-RBF Meshless Scheme with Applications to Laminar and Turbulent Viscous Fluid Flows

S. Gerace[1], K. Erhart[1], E. Divo[1,2] and A. Kassab[1]

Abstract: The focus of this work is to demonstrate a novel approach to true CFD automation based on an adaptive Cartesian point distribution process coupled with a Meshless flow solution algorithm. As Meshless method solutions require only an underlying nodal distribution, this approach works well even for complex flow geometries with non-aligned domain boundaries. Through the addition of a so-called shadow layer of body-fitted nodes, application of boundary conditions is simplified considerably, eliminating the stair-casing issues of typical Cartesian-based techniques. This paper describes the approach taken to automatically generate the Meshless nodal distribution, along with the details of an automatic local refinement process. Also, as the primary interest of this automated CFD solver is for aerospace applications, this work includes the development of standard two-equation turbulence models for use in this Meshless based solver. Finally, results are shown for several relevant compressible, turbulent flows example configurations, demonstrating the benefits of the automatic refinement as well as the quality of the Meshless solutions in high-speed flow applications.

Keywords: meshless methods, meshless model generation, adaptive refinement, generalized finite differencing, turbulent flow, incompressible flow, compressible flow, shocks, CFD automation

1 Introduction

Mesh-free and mesh-reduction methods are continuing to gain in popularity as increases in accuracy and efficiency have reached the point where solution robustness can be obtained over a wide range of problem domains (Gerace, Erhart, Divo,

[1] Mechanical, Materials, and Aerospace Engineering, University of Central Florida, Orlando, Florida, 32816-2450, USA

[2] School of Engineering Technology, Daytona State College, 1770 Technology Blvd., Daytona Beach, FL, 32117, USA

and Kassab, 2009; Vertnik and Sarler, 2009a,b; Stevens, Power, Lees, and Morvan, 2009; Stevens, Power, and Morvan, 2009). In addition, these so-called Meshless methods eliminate the tedious mesh-generation process by reducing the dependency of the solution algorithm on the quality of the underlying point distribution. Since the mesh-generation process is currently a major requirement of time and effort when attempting to solve problems involving complex geometries with standard solution techniques (Finite Element, Finite Volume, and Finite Difference), these Meshless techniques can offer considerable cost savings, especially in the field of computational fluid dynamics. In this paper, we will illustrate these advantages in the area of compressible fluid flow analysis through presentation of several complex example problems, using a newly developed, fully automated nodal refinement strategy coupled with a hybrid Meshless / Finite Difference solver. Several test cases with comparisons to experimental data, as well as to existing commercially available CFD technology are presented as evidence supporting the adoption of this method as a viable alternative to more conventional CFD approaches. In addition, both the underlying theory and basic algorithm structure are also presented, further illustrating the improved automation of the CFD model generation process and the highly user friendly CFD solution process that results.

This paper is organized into four sections, with conclusions following. In the first section, we will present a brief history and overview of Meshless collocation techniques and formalize the concept of virtual finite differencing, which will be used to produce several of the required solution operators, particularly the upwinding operators for convective derivatives. Following this, the procedure of the Meshless model generation will be presented, with focus on the adaptivity during the solution process. Once the Meshless method has been described in full (collocation and model generation), the governing equations will be presented, along with the specific techniques used to address common application concerns. Finally, results will be presented which seek to demonstrate the robustness this new implementation across a range of typical benchmark problems.

2 Meshless Collocation Techniques

Conventional numerical methods commonly used in most engineering applications (Finite Element, Finite Volume, and to some degree, Boundary Element methods) all require some type of well defined connectivity between nodes or volumes. The connectivity is used to specify the influence of any node to its neighbors, and to serve as an interpolation mesh over which required derivative and integral quantities may be obtained. While much progress has been made in the area of unstructured mesh (connectivity) generation, which seeks to automate the model setup process, as the complexity and size of a problem increases, it becomes exceedingly difficult to utilize such procedures. Meshless methods eliminate the need for a defined connectivity mesh and instead the influence of one node on its neighbors is defined by an interpolation technique that can be used regardless of model geometry or nodal spacing. Many interpolation techniques exist that may be used to arrive at a Meshless formulation (for a full overview, we refer the reader to the book by Liu (Liu, 2003)), however, this paper will focus on two specific techniques that have shown promising results within the area of high-speed fluid flow: local radial basis function (RBF) collocation and virtual finite difference collocation.

2.1 Local Radial Basis Function Collocation

The idea of collocating radial basis functions over local influence topologies can be largely attributed to works by Lee et al. (Lee, Liu, and Fan, 2003), Tolstykh and Shirobokov (Tolstykh and Shirobokov, 2003), and Shu et al. (Shu, Ding, and Yeo, 2003), each of whom were attempting to address the so-called "uncertainty relation" formally described by Schaback (Schaback, 1995). Stevens et al. (Stevens, Power, Lees, and Morvan, 2009) summarized the uncertainty relation succinctly as suggesting that for RBF interpolations "better [moment matrix] conditioning is actually associated with worse accuracy, and worse [moment matrix] conditioning is associated with improved accuracy". The uncertainty relationship has serious implications for global interpolation schemes (which are traditional based on spectral methods) and has largely motivated the adoption of methods which favor compact, local influence topologies over global interpolations commonly used in earlier research. In addition, it can be shown that local RBF collocation is a generalization of the compact finite differencing scheme (Wright and Fornberg, 2006), and as such, provides a solid foundation for developing accurate solution techniques based on existing formulation strategies.

Localized Radial Basis Function collocation begins with the principle that any arbitrary domain, Ω, can be interpolated over by collocating about a number of points using some basis function, R. This method divides the overall region into small subdomains, commonly referred to as topologies, in order to produce a more efficient and accurate solution method when compared to global interpolation techniques. The represented field, u, can be locally interpolated by multiplying the basis functions by a set of expansion coefficients over the local topologies, Ω_i:

$$u(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i R_i(\mathbf{x}) + \sum_{j=1}^{m} \beta_j p_j(\mathbf{x}) = \underline{R}^{\mathrm{T}}(\mathbf{x})\,\underline{\alpha} + \underline{p}^{\mathrm{T}}(\mathbf{x})\,\underline{\beta} \tag{1}$$

where $u(\mathbf{x})$ is the field value at location $\mathbf{x}$, n is the number of nodes in the local topology at position $\mathbf{x}$, α_i is the i-th expansion coefficient, and R_i is a radial basis function based on the distance between point $\mathbf{x}$ and point $\mathbf{x}_i$. In addition, in order to guarantee accurate interpolation of constant and linear fields (Kassab and Divo, 2007), m expansions are performed over an additional set of polynomial basis functions $p(\mathbf{x})$. In Eq. (1), m is the number of polynomial terms added to the approximation, β_j is the j-th polynomial expansion coefficient, and $p_j(\mathbf{x})$ is the j-th polynomial basis function. It is worth noting that the primary difference between more recent local methods and their global counterparts (Gerace, 2006; Kansa, 1990a,b) is the selection of topology size; in global techniques Eq. (1) would be applied to all nodes simultaneously, producing a single global interpolation matrix. Meanwhile, in local methods n is usually selected as the minimum number of nodes that provide an adequate sampling of the local domain (generally between 7-20 points for three-dimensional problems). The dramatic reduction in local influence is what allows local interpolation to produce a more efficient scheme without significantly sacrificing solution accuracy.

A critical component of the localized Meshless collocation technique is the determination of a suitable basis function, R, that is able to accurately interpolate between data points. There have been many suggested radial basis functions, with some of the most commonly used shown in Table 1.

Table 1: Typical Radial Basis Functions

Name	Expression	Shape Parameters
Multiquadrics	$R_i(\mathbf{x}) = \left(r_i^2 + c^2\right)^q$	c, q
Gaussian	$R_i(\mathbf{x}) = \exp\left(-cr_i^2\right)$	c
Thin Plate Spline	$R_i(\mathbf{x}) = r_i^\eta$	η
Logarithmic	$R_i(\mathbf{x}) = r_i^\eta \log(r_i)$	η

Note that for all radial basis functions shown in Table 1, r_i is defined as the radial distance between location $\mathbf{x}$ and $\mathbf{x}_i$, which in three-dimensions is given by:

$$r_i = r_i(\mathbf{x}) = |\mathbf{x} - \mathbf{x}_i| = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} \tag{2}$$

where $\mathbf{x} = \{x\ y\ z\}^{\mathrm{T}}$ and $\mathbf{x}_i = \{x_i\ y_i\ z_i\}^{\mathrm{T}}$. Although all of the basis functions listed in Table 1 have been used in Meshless method implementations to some extent, many agree that the family of so-called Inverse Hardy Multiquadrics (MQ) (Hardy, 1971) (a multiquadric with $q = -0.5$) produce the most stable and accurate results across the largest subset of problem domains (Kansa, 1990b,a; Divo and Kassab, 2005; Sarler, Tran-Cong, and Chen, 2005). As such, the specific radial basis function employed will be the inverse Hardy MQ of the form:

$$R_i(\mathbf{x}) = \frac{1}{\sqrt{r_i^2 + c^2}} \tag{3}$$

where c is known as the shape parameter of the interpolation. The choice of value for the shape parameter is critical for obtaining accurate results (Driscoll and Fornberg, 2002; Chandhini and Sanyasiraju, 2007; Bayona, Moscoso, and Kindelan, 2011), and as such, a method of arriving at an optimal shape parameter value on a topology-by-topology basis utilizing Singular Value Decomposition has previously been presented by the authors (Kassab and Divo, 2007; Gerace, 2007). Additionally, some authors (Sarler and Vertnik, 2006; Stevens, Power, Lees, and Morvan, 2009) have supplemented the standard MQ-RBF by use of additional interpolating functions which has been shown to further improve the techniques.

To arrive at the necessary interpolation weights for a given topology, Eq. (1) may be applied to all of the nodes in the topology, resulting in the following set of expressions:

$$u(\mathbf{x}_k) = \sum_{i=1}^{n} \alpha_i R_i(\mathbf{x}_k) + \sum_{j=1}^{m} \beta_j p_j(\mathbf{x}_k) \qquad k = 1, 2, \ldots, n \tag{4}$$

or in matrix form:

$$\underline{u} = \underline{\underline{G}}\,\underline{\alpha} + \underline{\underline{F}}\,\underline{\beta} \tag{5}$$

Notice that Eq. (4) only provides n equations, however, the introduction of polynomial basis terms has increased the number of unknowns to $n+m$. The remaining equations are provided by requiring the polynomials to satisfy an extra requirement which guarantees a unique approximation (Golberg, Chen, and Bowman, 1999). This requirement is usually imposed as the following constraint equations:

$$\sum_{i=1}^{n} p_j(\mathbf{x}_i)\,\alpha_i = 0 \qquad j = 1, 2, \ldots, m \tag{6}$$

or in matrix form:

$$\underline{\underline{F}}^{\mathrm{T}}\underline{\alpha} = 0 \tag{7}$$

Combining Eqs. (5) and (7) gives:

$$\begin{bmatrix} \underline{\underline{G}} & \underline{\underline{F}} \\ \underline{\underline{F}} & \underline{\underline{0}} \end{bmatrix} \begin{Bmatrix} \underline{\alpha} \\ \underline{\beta} \end{Bmatrix} = \begin{Bmatrix} \underline{u} \\ \underline{0} \end{Bmatrix} \tag{8}$$

where $\underline{\underline{G}}$ is the pure RBF moment matrix and $\underline{\underline{F}}$ is the polynomial augmentation matrix. Thus Eq. (8) may be more succinctly represented as:

$$\underline{\underline{H}}\left\{\begin{array}{c}\underline{\alpha}\\ \underline{\beta}\end{array}\right\}=\left\{\begin{array}{c}\underline{u}\\ \underline{0}\end{array}\right\} \tag{9}$$

with $\underline{\underline{H}}$ representing the fully augmented moment matrix.

At this point, the derivation procedure presented by Liu (Liu, 2003) may be followed which reduces overall matrix multiplications and decouples the pure RBF moment matrix from the polynomial components. This is an important consideration for proper optimization of the RBF shape parameter. For the sake of brevity, we present the final resulting nodal shape function in Eq. (10), and refer the interested reader to the referenced text for full details:

$$\underline{\Phi}(\mathbf{x})=\underline{R}^{\mathrm{T}}\underline{\underline{S_{\alpha}}}+\underline{p}^{\mathrm{T}}\underline{\underline{S_{\beta}}} \tag{10}$$

It is important to note that the shape functions provided by Eq. (10) are strictly geometrically dependent, and as such, may be precomputed during preprocessing stages of the algorithm.

A major benefit of the formulation shown in Eq. (10) is that the direct derivatives of the underlying RBF interpolators may be easily obtained as:

$$\partial\underline{\Phi}=\partial\underline{R}^{\mathrm{T}}\underline{\underline{S_{\alpha}}}+\partial\underline{p}^{\mathrm{T}}\underline{\underline{S_{\beta}}} \tag{11}$$

where ∂ may represent any derivative operator (such as $\partial/\partial x$, $\partial^2/\partial x^2$, ∇^2, etc.) for which analytical derivatives exist for both $R_i(\mathbf{x})$ and $p_j(\mathbf{x})$.

Our previous research has shown that this Localized MQ-RBF collocation process provides adequate numerical results in most cases, however, there a few concerns which prevent it from being the optimal choice for all situations: (1) due to the radially symmetric nature of the basis functions, optimal results are only obtained for symmetric derivative operators and one-sided operators (such as upwinded derivatives) are often not well captured; (2) it has been shown that the weights obtained through this process are identical to those obtained through traditional finite differencing (Chandhini and Sanyasiraju, 2007) in areas consisting of locally structured nodes (this fact should be exploited in order to increase computational efficiency); and (3) areas of very high gradients (shocks, boundary layers, etc.) may cause oscillations in the RBF interpolator, resulting in poor numerical stability. In response to these issues, special upwinding strategies have typically been developed which utilize upwind distorted topologies and NVD type-control of convective terms for various steady and transient incompressible flow applications (Zahab, Divo, and

Kassab, 2009a,b; Vertnik, Zaloznik, and Sarler, 2006; Vertnik and Sarler, 2006). However these approaches do not address the issue of efficiency. Therefore another collocation technique, described in the next section, has been developed and is applied in areas where local structure is present or can be simulated.

2.2 Virtual Finite Difference Collocation

Virtual finite differencing, unlike RBF, is not an approximation or interpolation method, but rather a formulation technique which utilizes interpolating shape functions to form a generalized finite difference approach. There are several benefits of formulating the problem in a traditional finite difference fashion. First, the finite difference method is one of the oldest PDE solution techniques and as such, it has a large knowledge base pertaining to optimizations in terms of solution accuracy and speed. Second, because the formulations are derived from the Taylor series representation, they have predictable error and, as such, many techniques have been developed to utilize this error within the solution process. Third, because of the prevalence of finite differencing methods, there have been many proposed approaches for handling convective derivatives where upwinding is necessary, a common source of concern for more traditional Meshless techniques (such as the local RBF collocation scheme previously described). In this respect, virtual finite differencing is a useful tool for developing a robust Meshless methodology. However, it should not to be mistaken for a simple generalized finite differencing approach. Instead, virtual finite differencing encompasses the idea that in order to obtain appropriate levels of solution efficiency in Meshless methods, one must take advantage of as much local structure as possible.

As the foundation of virtual finite differencing is structured finite differencing, we first note that all finite difference operators may be placed into the standard form of:

$$\partial u(\mathbf{x}) = \partial \underline{\Phi}(\mathbf{x})\, \underline{u} \tag{12}$$

where the derivative shape function vector $\partial \underline{\Phi}$ consists of the leading coefficients of each influence node, multiplying each respective nodal value (we direct the interested reader to Tannehill, Anderson, and Pletcher (Tannehill, Anderson, and Pletcher, 1997) for full derivation of common structured finite difference stencils). If the underlying point distribution is regular, such as the one shown in Figure 1a, then finite differencing may be applied without any concerns; however, as soon as the point distribution becomes irregular, such as in Figure 1b (note the missing node at the red x for the orange topology) then the derivative operators which rely on a structured set of data can no longer be applied. This limitation prevents pure finite differencing from being applied to arbitrary geometries since almost any real-world model will have boundary surfaces that are not grid aligned (which introduces irregularities into the point distribution). However, by utilizing point distribution techniques based on Cartesian alignment, even a highly irregular geometry will consist of regions exhibiting local structure (such as the right side of the point distribution in Figure 1b). For this reason, it is beneficial to take advantage of those areas with local structure and directly apply finite differencing. Further justifying the use of pure finite differencing in areas of local structure is the fact that local RBF interpolation, when applied to a structured topology with the proper configuration, will generate the same shape functions that are obtained via finite differencing, though at a much higher computational expense (due to the additional operations necessary to build and invert the respective moment matrices).

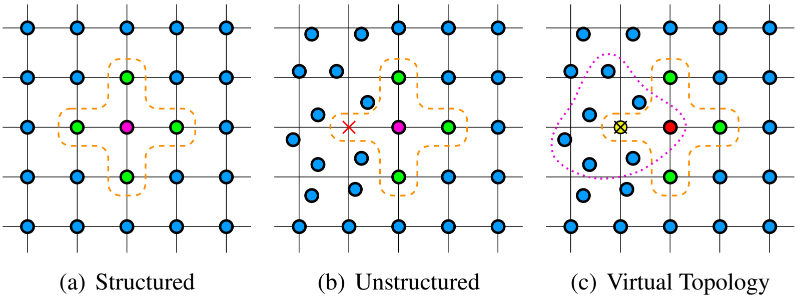

(a) Structured (b) Unstructured (c) Virtual Topology

Figure 1: Sample Point Distributions

Thus, to apply finite differencing equations to an unstructured region such as the one shown in Figure 1b, some means of approximating the missing node values must be developed. Fortunately, such a method has already been described in the previous section and these local radial basis function interpolations may be applied to determine the missing nodal values. Once these have been obtained, it becomes trivial to directly apply the finite difference equations to represent the underlying derivatives. Thus, the process of obtaining the shape functions via this procedure begins by constructing a virtual node at the required locations of missing structure and building a local topology about that position (as shown in Figure 1c as the yellow node and the purple dotted region, respectively). Once the virtual topology has been determined, local RBF interpolation may be utilized to determine the value at the desired location. This interpolation will be of the form:

$$u(\mathbf{x}) = \underline{\Phi}(\mathbf{x})\,\underline{u} \tag{13}$$

where $u(\mathbf{x})$ represents the unknown value at the virtual node.

Once the value of the virtual node has been determined, the finite difference stencil may be directly applied over the given domain. However, rather than simply use the interpolators to determine the value at each virtual node, they may be integrated into the shape functions provided by finite differencing to produce a compact form which adds no additional overhead to the solution process (other than the increase in local topology size). For example, to develop a second order, second derivative operator for the red node (located at coordinates x, y, z) shown in Figure 1c, the unknown quantity at the virtual node located at $(x-\Delta x, y, z)$ is required. Thus, an interpolation is used to obtain the value as:

$$u\left(x-\Delta x, y, z\right) = \underline{\Phi}\left(x-\Delta x, y, z\right)\underline{u} \tag{14}$$

which may be substituted into the appropriate finite difference equation, resulting in:

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x,y,z} = \frac{u\left(x+\Delta x, y, z\right) - 2u\left(x, y, z\right) + \underline{\Phi}\left(x-\Delta x, y, z\right)\underline{u}}{\Delta x^2} \tag{15}$$

It is important to note that although a second order approximation is shown here for illustrative purposes, the order of approximation used for each finite difference term in the resulting governing equations have been chosen following conventional guidelines (Tannehill, Anderson, and Pletcher, 1997; Hoffmann and Chiang, 2004; Liou and Steffen, 1993). By realizing that the existing weights can be combined with the finite difference weights, and that the virtual topology nodal vectors can be appended to one another (making sure to combine duplicate nodes and their associated weights), Eq. (15) may be rewritten in terms of shape functions as:

$$\frac{\partial^2 u\left(\mathbf{x}\right)}{\partial x^2} = \left\{\begin{array}{ccc} \frac{1}{\Delta x^2}\underline{\Phi}\left(x-\Delta x, y, z\right) & -\frac{2}{\Delta x^2} & \frac{1}{\Delta x^2} \end{array}\right\} \left\{\begin{array}{c} \underline{u} \\ u\left(x, y, z\right) \\ u\left(x+\Delta x, y, z\right) \end{array}\right\} = \underline{\Phi}'\underline{u} \tag{16}$$

thus resulting in a similar form which may be integrated with the direct differentiation schemes shown in the previous section. It is important to draw attention to the fact that in Eqs. (15) and (16) only one term is expressed with the interpolation function Φ, this comes from the fact that in the particular stencil shown in Figure 1c, only the neighboring node in the negative x direction requires a local interpolation (since no physical node exists in the required location, $x-\Delta x$). Thus, although Eq. (16) describes the final form for a central differenced second order derivative, the resultant shape functions must be constructed on a case-by-case basis given the local topology at each node in the domain.

In summary, virtual finite differencing encapsulates two principles regarding the use of finite difference within the context of Meshless solvers. The first principle states that, where possible, pure finite differencing should be utilized to obtain the necessary derivative operators. The second principle states that in areas with some local structure, it is beneficial to use the existing information that is present (such as the values at $u(x+\Delta x,y,z)$, $u(x,y\pm\Delta y,z)$, and $u(x,y,z\pm\Delta z)$ in Figure 1c) to provide greater accuracy than would be possible through direct differentiation of a pure interpolation based approach (especially when the complete set of structured data is present for a particular direction, such as y or z derivatives in Figure 1c and for upwinded derivatives). This technique also differs from other generalized finite differencing schemes in that we may apply local RBF interpolation for some derivative operators (such as the Laplacian operator) and virtual finite differencing for other operators (upwinded convective derivatives) within the same solver implementation.

3 Domain Discretization and Refinement

One of the critical advantages of Meshless methods is that they allow for the capability of completely automated point and topology generation in arbitrarily complex three-dimensional models. However, few works have been presented which provide a thorough description of complete model generation processes tailored to Meshless point distribution. Interestingly enough, it is these techniques which allow a solution process to fully utilize the advantages inherent in Meshless methods, and as such, we have devoted considerable efforts toward the development of techniques which automate the nodal distribution process and allow for solution-based adaptive refinement of the problem domain. This section will present an overview of our model generation process, with focus on its adaptive refinement capabilities. It should be noted that the goal of these developments are to ensure that accurate solutions are obtained even when there is no prior knowledge regarding the expected solution behavior.

Our models are first constructed from a triangulated surface which defines the boundary of the problem. Triangulated, or tessellated, surfaces were chosen for several reasons. First, because the tessellated surface is a very common entity in computer geometry and graphics, extensive literature exists describing how best to control refinement on the surface (such as Delaunay refinement and mesh optimization (Shewchuk, 2002; Tournois, Alliez, and Devillers, 2007)). Second, because the triangulated surfaces are relatively simple compared to other analytical surfaces, calculations required for surface integrations, volume calculations (volume, centroid, containment tests), and other necessary components are relatively straightforward to implement and fast in calculation. Third, even though solutions may be obtained using Meshless methods without a defined boundary connectivity, in order to postprocess terms such as surface forces, stresses, moments, and other area based values, each boundary node must have a defined area and normal direction. By using a triangulated surface representation, the boundary nodes inherit their parent facet's area and normal properties, facilitating easy translation from the Meshless solution domain into the geometric domain of the problem.

Despite being based on a structured boundary representation, it is important to keep in mind that the collocation based Meshless techniques utilized herein require only a distribution of points to solve the governing equations. Therefore, once the boundary representation has been developed, a process of placing nodes on the surface in a consistent manner must be established. When placing computational nodes on the surface of a triangulated volume, the most logical location to place nodes is either at the vertices or centroids of the elements. Either of these placement locations should be sufficient, however, both locations have special considerations that must be discussed. Using the element vertices seems like the most logical choice for nodal placement, however, there is the issue of how to handle those vertices that lie on geometric edges where the normal vector is not easily defined. Although averaging the two normal vectors may seem like a viable solution, for the case of general face intersections, this is not always appropriate. Thus, to place nodes at element vertices, one solution is to utilize only vertices that lie on the interior of geometric faces. If this technique is utilized, however, there may be situations where no nodes are placed on the entire face. In this situation, because the geometric face is very narrow, no vertices lie on the interior of the region, and thus, no nodes will be placed on this face. It is therefore advantageous to not only place nodes at the vertices, but also at the centroids of the elements. By doing so, not only is the issue of narrow faces addressed, but it also allows boundary nodes to further approach the edges of the geometry due to the configuration of the elements. Therefore, once the surface triangulation is constructed and refined to an appropriate initial level, computational nodes are then distributed at all non-edge vertices as well as at each element's centroid. Once an appropriate initial surface is defined, the model is ready for interior discretization and then for running of the actual solution. However, in order to obtain good solution accuracy, discretization convergence (referred to as grid convergence in mesh-based techniques) must be ensured; therefore an appropriate boundary refinement scheme must be implemented as well.

Boundary refinement is accomplished by first representing the triangulated surface discretization as a quaternary triangular mesh (QTM), a surface structure commonly used in level-of-detail models for graphical applications (De Floriani and Puppo, 1995; Zhao, Chen, and Li, 2002) as well as geophysical models for cartographic applications (Dutton, 1996, 1999). However, these structures are rarely utilized to generate computational meshes, primarily due to the fact that when dealing with three-dimensional models, mesh-based techniques require a three-dimensional volumetric mesh (tetrahedrons). However, with Meshless methods, only a simple point distribution is needed on the boundary, and as such, the quaternary triangulation structure is perfectly suited for refinement tasks. The concept of QTM refinement involves developing a recursive storage structure whereby each triangular element stores 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 pairwise fashion. It is important to note that the actual data structure is stored in a recursive manner, which allows for fast indexing as well as efficient geometric operations through the implicit level-of-detail representation.

At this point a complete procedure for distributing the nodes on the boundary has been described, however, it is important to note that despite being based on a structured triangular distribution, ultimately it is only a distribution of nodes which is required for this Meshless technique. As such, there is no reason why nodes which are deemed inappropriate (because they are too close to their neighbors or they exist on a highly-skewed element) can't simply be omitted from the computational domain. Thus, prior to solving the problem it is important to apply an appropriate distance filter which eliminates nodes on the boundary whose closest neighbor is within some threshold distance γ. Although the filtering criteria may be any appropriate metric for the given problem domain, we have found good success using a simple distance filter based on a percentage of the local element edge length such that:

$$\gamma_i = \frac{1}{3n}\sum_{k=1}^{n} ave\left(\mathscr{L}_k^{\langle 1\rangle},\mathscr{L}_k^{\langle 2\rangle},\mathscr{L}_k^{\langle 3\rangle}\right) \tag{17}$$

where n is the number of coincident elements to node i, and $\mathscr{L}_k^{\langle q\rangle}$ is the q-th edge length of element k. Note that Eq. (17) essentially sets the filter at one-third of the average edge length for all coincident elements to a particular node. The ability to simply omit a problematic node is a prime example of a computational liberty that is afforded by utilizing a Meshless method over more conventional mesh-based techniques.

Having defined the boundary distribution process, the next step in the point distribution process is to create the so-called "shadow" nodes, which serve as a boundary layer distribution in the problem solution. The process of generating the shadow layer begins by analyzing the boundary conditions applied to each surface of the model. This is done such that only surfaces with high normal-gradient conditions will generate a shadow layer. In particular, for applications in fluid flow, this technique is of primary interest where no-slip wall conditions are applied. The process of actually generating the shadow nodes is fairly straightforward and is illustrated on a simple two dimensional boundary in Figure 2. Notice that in this figure, the shadow layer has a depth of 3 (the depth is generally initialized at 2 and allowed to refine as the solution progresses), and each boundary node (shown in green) has a single associated column of shadow nodes. 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 via direct local RBF differentiation. The placement of the outer shadow layer (one furthest 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. By adjusting the scaling method, one can obtain faster (or slower) growth in the shadow layer, analogous to the stretching of the grid in a typical structured boundary layer mesh. However, in the case of Meshless methods, this scaling factor may be automatically controlled by the field behavior (through examination of the normal-to-tangential gradient ratio) and updated accordingly during refinement phases.

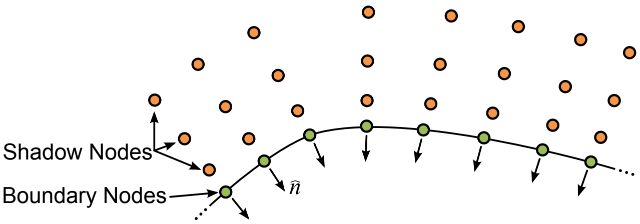


Figure 2: Shadow Layer

It is important to realize that the process of adding shadow nodes can potentially introduce problems in highly concave boundary situations, as illustrated in Figure 3. To address this issue, a technique of collapsing shadow nodes which are very close to one another is necessary in order to eliminate instability in the underlying Meshless interpolations. This process begins by creating a collapsed node at the center of mass for the offending set of shadow nodes. This collapsed node utilizes an interpolation operator which simply averages the values from the nearby lumped nodes. In this manner, any node which is not part of the collapsed set simply utilizes the collapsed node instead of the set of overlapping shadow nodes which may lead to numerical instability. Since the overlapping nodes are, by definition, very close to one another, a simple average value usually provides sufficient accuracy to represent the nodes as a single point of influence. Similarly, when solving the governing equation at the offending nodes, all overlapping nodes are invisible to each other and they each individually satisfy the governing equation at their respective location in space. By lumping their effect for external influences and eliminating their effect on one another, the nodes do not need to be directly removed and the respective boundary nodes do not need to apply special finite difference equations.

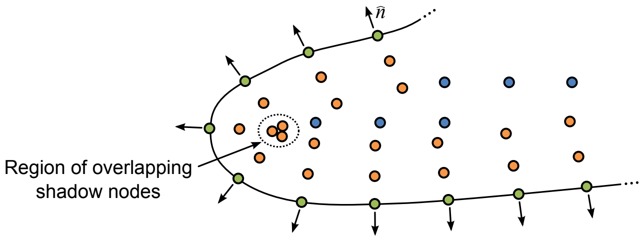
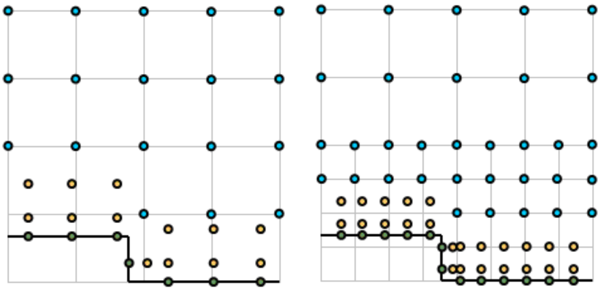


Figure 3: Concave Shadow Layer Region

By utilizing a discretization that can provide an independent boundary layer representation, the model generation procedure can develop point distributions capable of capturing high boundary gradients without having to resort to extreme interior refinement in order to correspond with non-aligned boundaries. In addition, since 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.

The final development in the point distribution process is to construct the recursive octree structure used throughout the bulk of the domain interior. An octree structure is used largely due to its ability to generate and refine itself in a very straightforward and automated fashion. It is important to note that the refinement of the interior, shadow, and boundary nodes is a disconnected process, therefore compatibility conditions must be included in order properly marry the refinements. The compatibility condition, which is generally implemented as a simple check on local feature size, ensures that as one local region is refined, all other regions which exist in this nearby area are refined as well (Figure 4 shows an example of one such situation). In Figure 4a, we see an initial discretization for part of a two dimensional boundary. After some number of iterations, it is determined that the current discretization at the boundary layer (yellow shadow nodes) is not sufficient to accurately capture the high gradients that were calculated. Rather than refine only the boundary nodes (green nodes) and their associated shadow nodes, the interior octree distribution (blue nodes) is also refined to an appropriate level to match the nearby boundary discretization, as shown in Figure 4b (this process is accomplished automatically via nodal spacing compatibility conditions). It is important to note that refinement can be initiated anywhere within the region (boundary, shadow, or interior nodes) and these compatibility conditions ensure that the refinement will propagate to all other nearby regions as needed.



(a) Before Refinement (b) After Refinement

Figure 4: Sample Boundary-Interior Refinement

4 Governing Equations

The current Meshless formulation is an expansion of previous work (Gerace, Erhart, Divo, and Kassab, 2009; Erhart, Gerace, Divo, and Kassab, 2009b,a) which begins with the three-dimensional Navier-Stokes equations in conservative variable form, given as:

$$\frac{\partial Q}{\partial t}+\frac{\partial E_c}{\partial x}+\frac{\partial F_c}{\partial y}+\frac{\partial G_c}{\partial z}=\frac{\partial E_v}{\partial x}+\frac{\partial F_v}{\partial y}+\frac{\partial G_v}{\partial z} \tag{18}$$

where:

$$Q=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho e_t\end{bmatrix}\quad E_c=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ (\rho e_t+p)u\end{bmatrix}\quad F_c=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vw\\ (\rho e_t+p)v\end{bmatrix}\quad G_c=\begin{bmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ (\rho e_t+p)w\end{bmatrix}$$

$$E_v=\begin{bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \tau_{xz}\\ E_v^{'}\end{bmatrix}\quad F_v=\begin{bmatrix}0\\ \tau_{yx}\\ \tau_{yy}\\ \tau_{yz}\\ F_v^{'}\end{bmatrix}\quad G_v=\begin{bmatrix}0\\ \tau_{zx}\\ \tau_{zy}\\ \tau_{zz}\\ G_v^{'}\end{bmatrix}$$

and $E_v^{'}=u\tau_{xx}+v\tau_{xy}+w\tau_{xz}-q_x$, $F_v^{'}=u\tau_{yx}+v\tau_{yy}+w\tau_{yz}-q_y$, and $G_v^{'}=u\tau_{zx}+v\tau_{zy}+w\tau_{zz}-q_z$. Note that Eq. (18) represents the full Navier-Stokes equations, where E_c, F_c, and G_c represent the convective terms of the governing equations and E_v, F_v, and G_v represent the viscous terms. In addition to this set of equations, an equation of state is required; this paper assumes an ideal gas with constant specific heat ratio, γ, allowing the pressure field to be calculated from the density and temperature values as:

$$p=\rho C_v(\gamma-1)T \tag{19}$$

Additionally, the shear stress τ_{ij} from Eq. (18) may be expressed in vector form as:

$$\tau_{ij} = \left[\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\delta_{ij}\frac{2}{3}\mu\frac{\partial u_k}{\partial x_k}\right]+\mu_T\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right)-\frac{2}{3}\rho k\delta_{ij} \tag{20}$$

where δ_{ij} is the Kronecker Delta, k is the turbulent kinetic energy, and μ_T is the turbulent or eddy viscosity. The first bracketed term is considered the laminar shear stress, while the second is the Reynolds Stress (turbulent shear stress).

Two-equation turbulence models have come to be the standard CFD turbulence techniques for Finite Volume and Finite Difference techniques and it is anticipated that this will be the case for Meshless methods as well. Two-equation models are the lowest order, physically-based models that obtain full closure of the Reynolds Stress terms without the need for algebraic specification of turbulence parameters (Wilcox, 2006). These models employ detailed transport equations which reasonably account for the convection, diffusion, production, and dissipation of various turbulence parameters. We have developed two common models, namely a k-ω model and a combined k-ω/k-ε model. The transport equations and relationships amongst the various turbulence parameters are given by the equations shown below:

$$\rho\left(\frac{\partial k}{\partial t}+u\frac{\partial k}{\partial x}+v\frac{\partial k}{\partial y}\right)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_T}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]+P_k-\rho\varepsilon \tag{21}$$

$$\rho\left(\frac{\partial \varepsilon}{\partial t}+u\frac{\partial \varepsilon}{\partial x}+v\frac{\partial \varepsilon}{\partial y}\right)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_T}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right]+C_{\varepsilon 1}P_k\frac{\varepsilon}{k}-C_{\varepsilon 2}\rho\frac{\varepsilon^2}{k} \tag{22}$$

$$\rho\left(\frac{\partial \omega}{\partial t}+u\frac{\partial \omega}{\partial x}+v\frac{\partial \omega}{\partial y}\right)=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_T}{\sigma_\varepsilon}\right)\frac{\partial \omega}{\partial x_j}\right]+\alpha P_k\frac{\omega}{k}-\beta\rho\omega^2 \tag{23}$$

These transport equations along with the following relations define the two-equation models:

$$\mu_T=\rho C_\mu\tfrac{k^2}{\varepsilon}\quad or\quad \mu_T=\rho\tfrac{k}{\omega}\quad with\quad \varepsilon=k\,\omega\beta^* \tag{24}$$

Our current research has focused on the standard k-ω model, as this model allows solutions of the turbulence equations all the way to the domain walls. This property of the k-ω model makes it desirable for an automatically generated grid technique like our Meshless approach, since other common turbulence models, such as the k-ε model, typically require the use of special wall functions to prevent the numerical instabilities that result from both k andε approaching zero at the wall. Incorporation of such wall functions is not a major difficulty, especially for structured grid approaches, however incorrect application of wall functions can lead to undesirable model predictions, and our experience has shown that k-ω models are less sensitive to these issues. It is also understood that k-ω models are typically best suited to wall bounded flows, which may limit our accuracy in the intended focus area of separated, external aerospace flows, for which the k-ε models typically perform better. We have addressed this concern by implementing a combined k-ε/k-ω model, which utilizes the desirable features of each model by incorporation of a single transport equation for ε/ω (written in terms of ω only). The necessary equations are easily derived from the relations given above, and a simple blending function is used to switch from the k-ω model in the near wall region to the k-ε model in the outer layer regions (see Hoffmann (Hoffmann and Chiang, 2004) for further details).

These turbulent compressible flow equations are solved using a fully explicit time-marching scheme to reach steady-state solutions as:

$$Q^{(k+1)} = Q^{(k)} - \Delta t \left[\frac{\partial E_c}{\partial x} + \frac{\partial F_c}{\partial y} + \frac{\partial G_c}{\partial z}\right]^{(k)} + \Delta t \left[\frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} + \frac{\partial G_v}{\partial z}\right]^{(k)} \tag{25}$$

It is worth noting that this approach is also applicable to unsteady flows as long as a suitable initial condition is provided. Since steady-state solutions are the current focus, a local time-stepping procedure has been implemented to expedite convergence. Local time stepping is well described in many resources (such as Hoffman (Hoffmann and Chiang, 2004)) and is based on maintaining the local CFL number below the stability threshold. The standard explicit time marching scheme allows all unknown field derivatives to be evaluated at the previous time step, thereby creating a very simple update or advancement equation (it was decided to utilize an explicit, local time-marching scheme over more efficient implicit schemes for ease of implementation, however, future efforts may include efficiency and stability comparisons with an implicit solution scheme). The required field derivatives however, must be captured in an accurate fashion in order to produce a reliable CFD approach. It is well known that standard central type differencing produces accurate and stable results for the diffusion/stress terms as these terms generally promote changes nearly uniformly in all directions. As such, for these terms direct differentiation of the local RBF interpolations is most appropriate. The convective terms however, carry much of the flow information with them so that symmetry in the derivatives is typically never seen. It is this lack of symmetry that produces the need to upwind the convective derivatives in order to ensure the proper direction of travel for flow field information. The importance of the upwinding procedure is widely known and our Meshless method approach has been found to have similar behavior to that of Finite Differencing in terms of the effects of improper upwinding. For this reason we have implemented a form of the Advection Upstream

Splitting Method (AUSM) proposed by Liou and Steffen (Liou and Steffen, 1993), which seeks to combine the accuracy of the Roe splitting method with the speed and simplicity of Van Leer Splitting schemes.

We refer to the original paper by Liou and Steffen for full details, however, it is worth presenting a brief overview of the method as it is applied to this work. AUSM begins by recognizing that each of the convective flux vectors, E_c, F_c, and G_c are comprised of both convective and pressure terms such that any of these vectors may be decomposed into true convective terms, $\Psi_i^{(c)}$, which are convected by some "suitably defined velocity", and pressure terms which are "governed by the acoustic wave speeds" (where $\Psi_i^{(c)}$ can refer to any of the three convective flux vectors). Thus, AUSM provides a means to properly separate the components and determine the correct direction of information propagation for each. Separated in this manner, each component is functionally upwinded using a "Mach-number-weighted average", as well as an appropriate pressure splitting weighting technique. In addition, because this formulation technique employs its own blended finite difference operators, the virtual finite differencing procedure is utilized to obtain necessary quantities at the necessary left and right neighbors of the presented half nodes.

The implementation procedure is not complete without also including a discussion of the application of pertinent boundary conditions. Since our Meshless approach utilizes interpolation operations, the specification of Dirichlet boundary conditions is trivial. Neumann or normal flux type boundaries are specified through the use of special normal direction virtual finite difference operators. These operators are constructed by placing one or more virtual nodes into the domain in the local boundary normal direction. It should be noted that for boundaries with one or more shadow nodes, the virtual nodes are not necessary and the shadow nodes can be used directly. A finite difference is then applied across these virtual nodes to complete the operator (the number of virtual nodes is based on the order of the FD approximation). The field values at the virtual nodes are obtained through the above described interpolation techniques. Far field, inlet, and outlet type boundaries are handled via a special extrapolation operator. It is well known that RBF interpolations are not stable for extrapolation, even at short distances, thus a finite difference approach has been implemented in a fashion very similar to the normal flux operator. Virtual points are placed in the normal boundary direction and their values are appropriately interpolated. A linear or quadratic extrapolation is then used to compute the value on the boundary based on the virtual node values, providing a scheme that is much more stable and robust then extrapolation via the RBF interpolators. A typical method based on the sign of the eigenvalues of the Flux Jacobian matrix, see Hoffman (Hoffmann and Chiang, 2004) for details, is used to determine the appropriate variables for extrapolation at the various types of boundaries.

5 Results

To illustrate the concepts and developments presented in this paper, several test cases will be presented below. These test cases were designed either to highlight the new application to compressible fluid flow with turbulence, or to illustrate the overall effectiveness and applicability of our Meshless CFD techniques (in particular, the importance of an adaptive refinement strategy). Additionally, the often questioned issue of local mass conservation is addressed to provide additional confidence in the soundness of the approach. This verification of mass conservation was completed by using volumetric cells which come directly from the underlying octree grid which is present throughout a large portion of the domain interior. A standard mass flux balance was completed on each cell and the results are shown to provide evidence that our Meshless method satisfies the conservation properties of the governing equations.

5.1 Supersonic Expanding Nozzle

The first test case involves supersonic, laminar flow through a simple smooth walled expanding nozzle (note that while the fluid is considered viscous the wall friction is neglected in this example). A two-dimensional depiction of the problem geometry is given in Figure 5, noting that the geometry is constant in the z direction with a thickness of 0.05m.

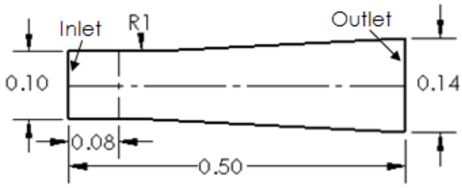
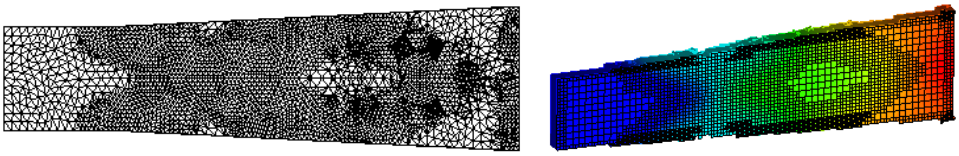


Figure 5: Geometry of Smooth Walled Expanding Nozzle (all units in meters)

To generate the supersonic flow field, an inlet Mach number of $M = 2.0$ and a stagnation pressure and temperature of $P_0 = 100,000$Pa and $T_0 = 300$K, were imposed, respectively, and as mentioned, all walls were assumed to be friction free (slip walls).

This case exhibits a series of interacting compression and expansion waves within the nozzle due to the linear expansion of the nozzle geometry. Although this indicates a poor nozzle design, it serves as an interesting test problem due to the complex flow field that exists within the computational domain. In addition it serves as a prime example of the importance of adaptive point distribution refinement, as the solution behavior for this test case is highly irregular, it would be difficult to predict an appropriate mesh size when discretizing the problem for solution with more conventional CFD techniques. However, with automatic refinement, the Meshless solution was able to be initialized with a relatively coarse discretization of approximately 45,000 nodes (placed nearly uniformly) and then allowed to refine three times (on mach and pressure gradients) after intervals of 4,000 iterations, resulting in a final discretization of approximately 160,000 nodes. Compare this to the two-dimensional solution generated within the commercial CFD package FLUENT, whose grid converged computational mesh consisted of approximately 70,000 nodes (the equivalent 3D mesh would contain more than 500,000 nodes), and the substantial savings in computational costs becomes obvious. To illustrate the ability of the refinement process to only add nodes where appropriate, Figure 6a shows the final surface distribution generated and Figure 6b shows a slice exposing the final octree distribution.



(a) Surface Distribution (b) Octree Distribution

Figure 6: Point Distributions for Supersonic Expanding Nozzle

For a quantitative comparison, the pressure levels along the mid-line ($y = 0$, $z = 0.025$) were compared to those obtained via FLUENT and are shown in Figure 7. Several Meshless solutions are shown, representing the different stages of adaptive refinement. By examining the Meshless results as the refinement progresses, we see that that the solution quality is improving as the point distribution is adaptively refined. There is excellent agreement (average error of $1.02\% \pm 0.13\%$) between the solutions obtained using FLUENT and the final refined point distribution solved using the described Meshless method technique.

This problem illustrates a major advantage of the Meshless technique over other methods in that an initially poor discretization does not prevent the user from obtaining a good final solution. Furthermore, the user does not need to be aware of any characteristic flow phenomenon prior to obtaining results, an important consideration when approaching complex flow problems.

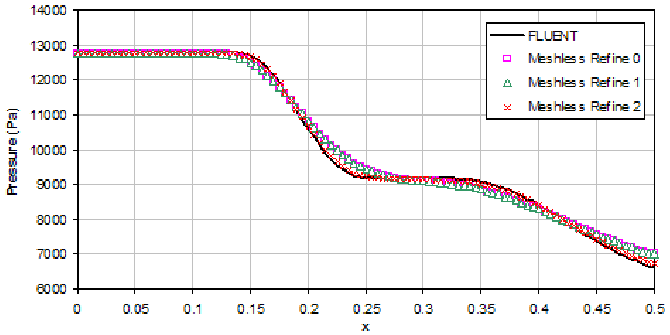


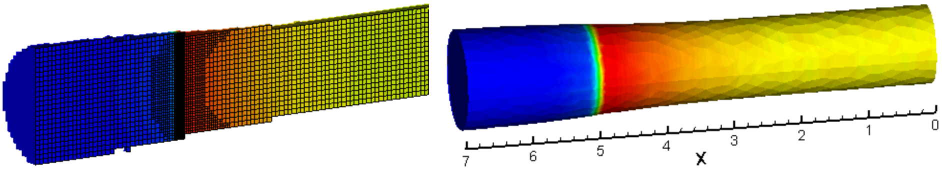
Figure 7: Mid-line Pressure Comparison

5.2 *Nozzle with Normal Shock*

The difficulty of capturing a standing shock is well known in the CFD community and for classic techniques such as finite volumes and finite differencing the necessary tools are well established. However, since little literature exists regarding the use of Meshless methods for such applications it is important to determine their behavior when faced with these types of flow discontinuities. To test this behavior, a nozzle developed from the Method of Characteristics, presented by Hoffman (Hoffmann and Chiang, 2004), is used to verify both the adaptive solution technique and the ability of the proposed method to capture a normal flow discontinuity (note that this problem was solved in the laminar regime to agree with analytical data). Specifying inlet conditions of $M = 1.5$, $P_0 = 100,000$Pa, $T_0 = 300$K, and $P = 27240.31$Pa, and choosing the shock at a location $x/L = 5/7$, provides the outlet pressure as $P_{out} = 66809.64$Pa according to one-dimensional flow theory (assuming $\gamma = 1.4$).

The solution was started on an initially coarse, uniform grid consisting of roughly 60,000 nodes and was allowed to adaptively refine three times throughout the solution process (on mach and pressure gradients), resulting in a final point distribution of just over 175,000 nodes. Figure 8a illustrates the final octree distribution by exposing a slice of the domain near the shock location. Additionally, the final Mach profile is plotted in Figure 8b, clearly illustrating the shock occurring at the expected location.

To quantitatively assess the results obtained using our Meshless method, the Mach and pressure fields were plotted down the mid-line ($y = 0$, $z = 0$) of the nozzle, and compared to the analytical solution found using one-dimensional approximations. Figures 9 and 10 plot the Mach and pressure comparisons, respectively, of a zoomed in portion of the solution near the discontinuity. These results demon-



(a) Final Octree Distribution (b) Mach Contours

Figure 8: Characteristic Nozzle with Stationary Normal Shock

strate that the proposed method accurately captures the discontinuity occurring at the shock location without excessive dissipation or solution oscillation, and shows that the results are consistently improved as the solution is refined (note that the final solution had an average error of $0.62\% \pm 0.02\%$, with the bulk of the error obviously occurring at the shock location). By utilizing the adaptive refinement process and the aforementioned Advection Upwind Splitting Method, the discontinuity (strong shock) has been captured with excellent clarity, providing a crisp representation for both Mach number and static pressure fields on the final, refined point distribution.

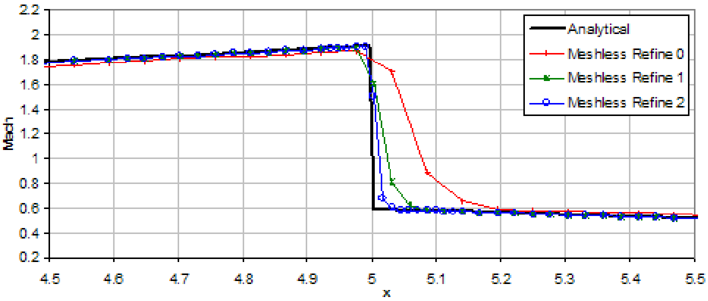


Figure 9: Mach Comparison for Axisymmetric Nozzle

5.3 *NACA-0012 Airfoil Flow*

The final laminar test case involved a standard NACA 0012 airfoil placed at an angle of attack of $\alpha = 10^o$, and analyzed at both subsonic flow (free-stream conditions of $M_\infty = 0.8$ and Reynolds number of $Re_\infty = 500$) as well as supersonic flow (free-stream conditions of conditions of $M_\infty = 2.0$, and Reynolds number of $Re_\infty = 1000$). Although this particular case is highly unphysical due to the extremely low Reynolds number with corresponding Mach number, it does provide

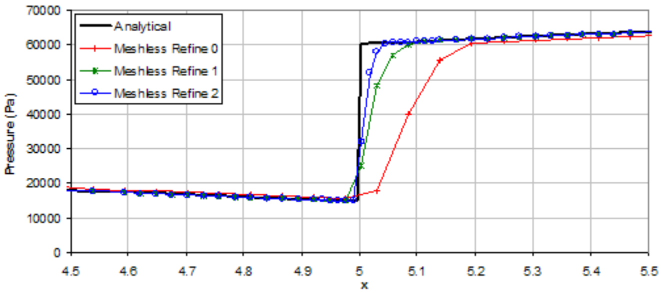
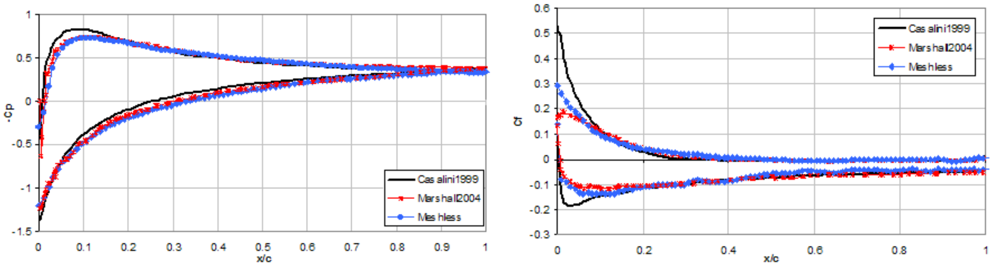


Figure 10: Pressure Comparison for Axisymmetric Nozzle

a non-turbulent flow field and is a common test case used to demonstrate proper flow characteristics. The results obtained are compared to data presented by Marshall and Ruffin (Marshall and Ruffin, 2004), who in turn compares their data to Casalini and Dadone (Casalini and Dadone, 1999) with good correlation. Note that the Meshless computational geometry extends 2 chord lengths in front of the airfoil, 8 chord lengths behind the airfoil, and 6 chord lengths on the top and bottom of the airfoil (one chord is $0.2m$). Additionally, the initial point distribution consisted of approximately 160,000 nodes, and after two levels of refinement (on mach, pressure, and mass imbalance), the final point distribution consisted of just over 525,000 nodes. The computed surface pressure and skin friction distributions for these problems are shown in Figures 11 and 12. It is clearly seen that the subsonic pressure results compare exceptionally well to those of Marshall and nearly the entire solution is within 2% of these results (everywhere except a very small area near the stagnation point), and neither computational results are able to match the peak value better than within about 7%. Additionally, the Meshless skin friction results are shown to be as good or better than those of Marshall, when compared to the results of Casalini, at all chord positions for both the upper and lower airfoil surfaces, with the stagnation region behavior being much more closely matched by our Meshless results. The agreement with Marshall for the supersonic case is not as close, however, the general trends are seen to be very similar and without a reliable third source for comparison, it cannot be stated for certain which results are in better agreement with the true behavior.

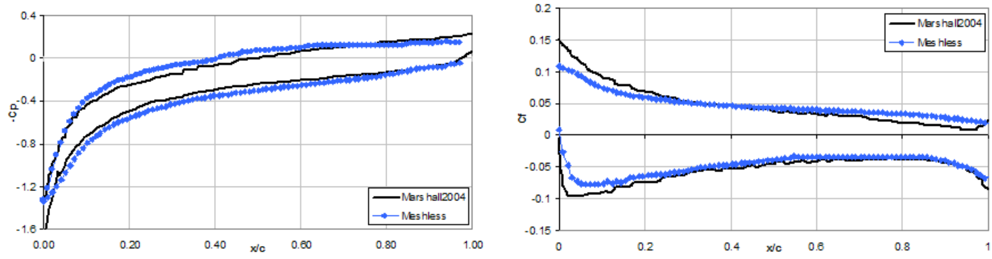
In addition, an analysis of local mass conservation was also performed to address common concerns that are raised against collocation based Meshless approaches. Since the interior octree distribution provides a fully structured grid over much of the domain, the grid can easily allow the development of interior cells or volumes so that cell based mass and energy fluxes can be computed (for this problem these values were used as additional criterion for the refinement process). Figures 13 and

14 shows the contours of the cell mass imbalance for both the subsonic and supersonic inlet conditions. These figures provide a good qualitative view of the small mass imbalances, but a detailed quantitative domain analysis showed that more than 98% of the field cells had mass imbalances of less than 1% of the incoming local cell mass for the subsonic and supersonic flows, respectively (full results are tabulated in Table 2 and are reported as the percentage of cells containing less than a specified mass imbalance).



(a) Centerline Surface Pressure Coefficient (b) Centerline Skin Friction Coefficient

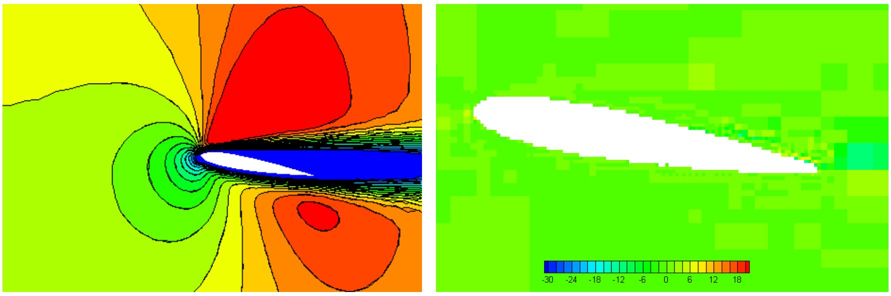
Figure 11: Subsonic NACA 0012 ($M_\infty = 0.8$; $Re_\infty = 500$)



(a) Centerline Surface Pressure Coefficient (b) Centerline Skin Friction Coefficient

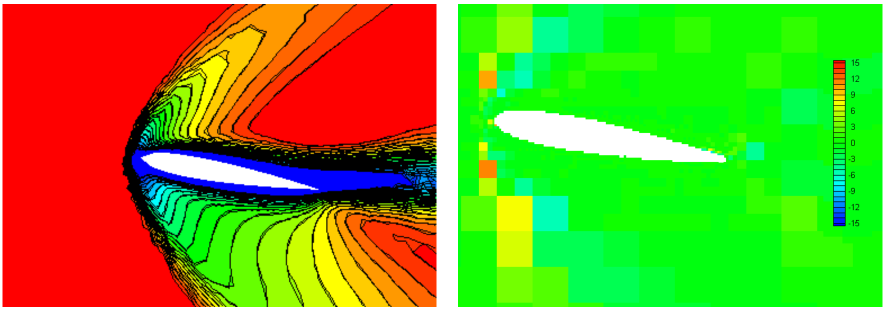
Figure 12: Supersonic NACA 0012 ($M_\infty = 2.0$; $Re_\infty = 1000$)

As a final illustration, Figure 15 shows a slice of the final, refined octree distribution along the central plane of the model, with nodes occurring at the intersections of the cell boundaries. This figure demonstrates the refinement capabilities of the model generation process, and illustrates that even complex models are able to be automatically solved without need for human interaction. In addition, it shows the importance of basing refinement on actual field characteristics, as the refinement



(a) Mach Contours (b) Central Plane Mass Flux Imbalance

Figure 13: Subsonic NACA 0012 ($M_\infty = 0.8$; $Re_\infty = 500$)



(a) Mach Contours (b) Central Plane Mass Flux Imbalance

Figure 14: Supersonic NACA 0012 ($M_\infty = 2.0$; $Re_\infty = 1000$)

Table 2: NACA 0012 Flow Mass Conservation Results

Subsonic (Initial)	Subsonic (Refined)	Supersonic (Initial)	Supersonic (Refined)
$94.5\% \leq 1.0\%$	$98.7\% \leq 1.0\%$	$96.4\% \leq 1.0\%$	$98.9\% \leq 1.0\%$
$89.2\% \leq 0.5\%$	$92.1\% \leq 0.5\%$	$90.9\% \leq 0.5\%$	$93.2\% \leq 0.5\%$
$65.3\% \leq 0.1\%$	$76.0\% \leq 0.1\%$	$58.0\% \leq 0.1\%$	$64.0\% \leq 0.1\%$

process occurs most around the airfoil geometry, where the highest gradient and mass imbalance values are encountered.

5.4 Turbulent Flat Plate

Developing flow over an infinitely thin flat plate is a classic test case for turbulence model verification. This simple test case is useful as the desired turbulent solutions are well known and understood. This case consists of a rectangular flow domain

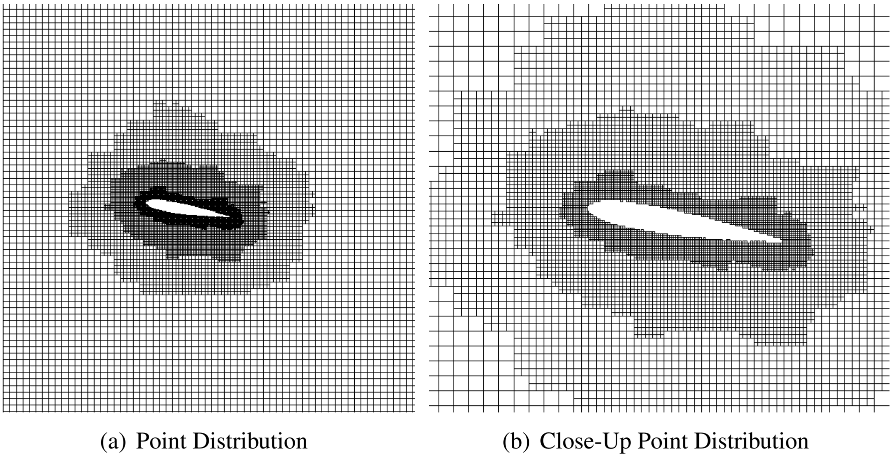
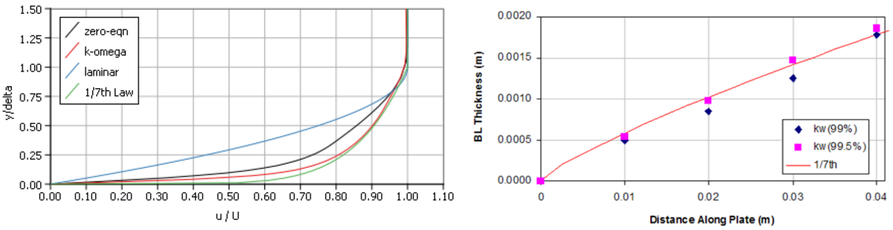

(a) Point Distribution (b) Close-Up Point Distribution

Figure 15: Slice of Final Octree Distribution for Subsonic Airfoil

with uniform fixed inlet conditions, an initial short length of zero shear wall, followed by a no-slip flat plate and standard outflow boundary, with inlet conditions of $P_0 = 100,000$Pa, $T_0 = 300$K. Since this problem focuses on the leading edge of the plate, the Reynolds number is quite small, but none-the-less turbulent results are sought.

Traditionally, RANS type turbulence models themselves are incapable of predicting the transition from laminar to turbulent flow and are simply an on/off type of model. Therefore, allowing the model to remain on at the leading edge is equivalent to enforcing a turbulent boundary layer from the onset of the plate surface. Therefore the results of this study should correlate well to empirical power-law solutions assuming turbulent flow from the leading edge, and indeed we have found such correlations to be quite good. Figure 16a shows the normalized velocity profiles at a location of $x = 40$mm from the leading edge for two distinct turbulence models (empirical boundary layer profiles for laminar and 1/7th law turbulent flows are included for comparison purposes). There is reasonable agreement between the computed and theoretical turbulent profiles with deviations mainly present in the viscous sublayer where it is well known the power-law profiles do not hold well. Figure 16b shows the overall boundary layer growth using the k-ω model, including a comparison to the classic power law results. 16 shows that the BL thickness is predicted within 5% of the theoretical distance at all locations along the wall using a definition 99.5% of the free-stream velocity for the BL edge. Figure 17 shows the wall shear stress for the computed and theoretical results, where again the laminar values are shown for comparison purposes. The matching here is again reasonable and the asymptotic behavior as the flow becomes removed from the leading edge is captured quite well.



(a) Normalized Velocity Profiles ($x = 40$mm) (b) Streamwise Boundary Layer Growth

Figure 16: Turbulent Flat Plate Boundary Layer

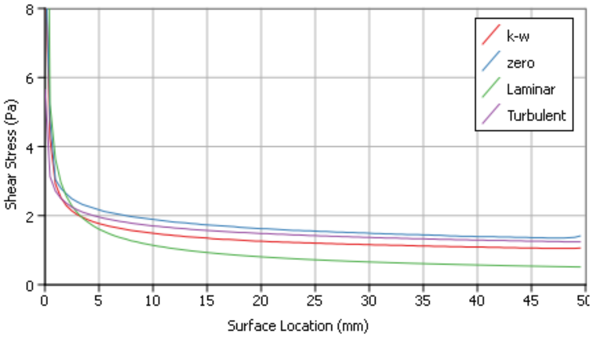


Figure 17: Wall Shear Stress Distribution

5.5 *Turbulent Flow Over a Backward Facing Step*

The following test case is widely used throughout the literature as a benchmark for turbulent solution quality and involves the flow over a step or sudden-expansion in an internal channel flow (classically referred to as the backward facing step). Figure 18 shows the specific geometric and flow conditions utilized for this test case (this problem has a Reynolds number of approximately 250,000). This problem is widely used because it tests turbulence model performance for both near wall and separated or wake-like flows. Many publications, such as the DNS simulations of Le (Le, Moin, and Kim, 1997), show that for a wide range of Reynolds number the flow reattachment location should fall near to 6.28 step heights beyond the expansion. Our Meshless based solutions utilizing the combined k-ε/k-ω model match this value quite well, as evidenced by the results provided below which show 6.5 step heights as the attachment length. Figure 19 shows the x-velocity flow contours, which are typical for this type of flow. Additionally, the profiles of velocity are shown for several locations following the step in Figure 20 along with those given by Kim (Kim, 1978) from his experimental work. Good agreement is visible between the two sets of data, demonstrating the well captured recirculation zone. Finally, Figure 21 shows the shear stress coefficient along the lower wall, which allows for precise determination of the location of the reattachment point ($x/H = 6.5$), which is easily within the acceptable range based on published experimental data.

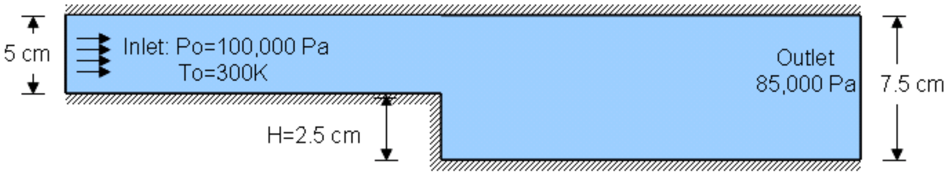


Figure 18: Geometry and Boundary Conditions for Backward Step

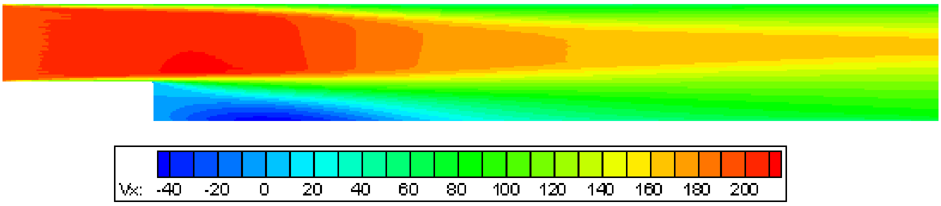


Figure 19: X-direction Velocity Contours Behind a Step (k-ε/k-ω model)

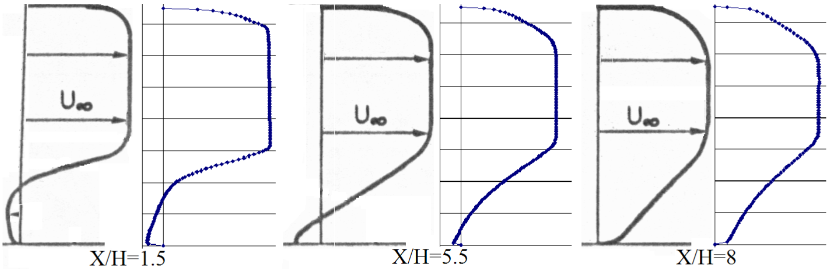


Figure 20: Down-Stream Velocity Profiles (Kim (Kim, 1978) vs. Meshless)

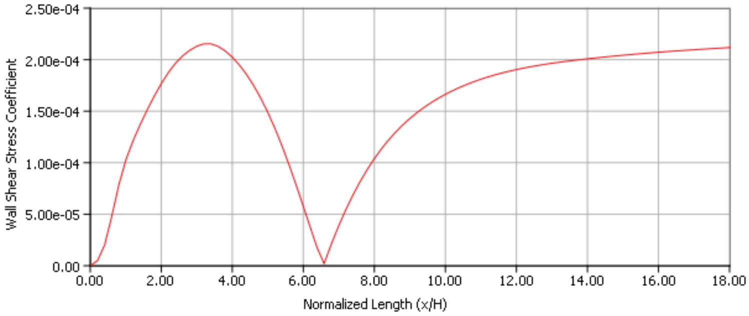


Figure 21: Wall Shear Stress Distribution Along a Flat Plate

6 Conclusions

Current publications demonstrating fluid mechanics applications of Meshless methods have focused largely on incompressible, laminar flows with fixed nodal distributions. We have presented an application of a localized Meshless method for the solution of compressible, viscous flows with turbulence utilizing an innovative blend of localized radial basis function collocation and virtual finite differencing strategies, which is a significant step forward for Meshless methods in flow field analysis. In addition, the highly automated nature of the tailored model generation process has allowed us to implement an adaptive nodal distribution method which is capable of both boundary and interior refinement. We have found promising results for several high-speed flow problems and have presented a few of these herein. We have tried to also illustrate a key advantage of this technique over more traditional methods; mainly that the robustness of the Meshless and refinement algorithms allows arrival at high quality solutions even when very rough initial point distributions are used. Thus, by coupling our Meshless routines with an automatic refinement procedure we have developed an approach whereby the analyst need not have any prior understanding of the expected flow characteristics in order to arrive at accurate results. These results provide confidence that important flow characteristics (such as boundary layers, shocks, etc.) that were not acceptably reproduced in the original point distribution will be automatically detected, properly refined, and well captured by the end of the solution analysis.

Acknowledgement: The authors would like to acknowledge the funding received in support of this project from Convergent Modeling Inc. through the Naval Air Systems Command STTR N08-T008.

References

Bayona, V.; Moscoso, M.; Kindelan, M. (2011): Optimal constant shape parameter for multiquadric based rbf-fd method. *Journal of Computational Physics*, vol. 230, pp. 7384–7399.

Casalini, F.; Dadone, A. (1999): Computations of viscous flows using a multigrid finite volume lambda formulation. *Engineering Computations*, vol. 16, no. 7, pp. 767 – 786.

Chandhini, G.; Sanyasiraju, Y. (2007): Local RBF-FD solutions for steady convection-diffusion problems. *International Journal for Numerical Methods in Engineering*, vol. 72, no. 3, pp. 352–378.

De Floriani, L.; Puppo, E. (1995): Hierarchical triangulation for multiresolution surface description. *ACM Trans. Graph.*, vol. 14, no. 4, pp. 363–411.

Divo, E.; Kassab, A. (2005): An efficient localized rbf meshless method applied to fluid flow and conjugate heat transfer. In *ASME IMECE*.

Driscoll, T.; Fornberg, B. (2002): Interpolation in the limit of increasingly flat radial basis functions. *Computers & Mathematics with Applications*, vol. 43, no. 3-5, pp. 413–422.

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

Dutton, G. H. (1999): *A Hierarchical Coordinate System for Geoprocessing and Cartography*, volume 79 of *Lecture Notes in Earth Sciences*, chapter Computational aspects of a quaternary triangular mesh, pp. 41–70. Springer Berlin Heidelberg, 1999.

Erhart, K.; Gerace, S.; Divo, E.; Kassab, A. (2009): An rbf interpolated generalized finite difference meshless method for compressible turbulent flows. In *Proceedings of the ASME 2009 International Mechanical Engineering Congres & Exposition*. ASME.

Erhart, K.; Gerace, S.; Divo, E.; Kassab, A. (2009): Turbulent compressible flow analysis with meshless methods. In *Proceedings of the Int. Conf. on Comp. Methods for Coupled Problems in Science and Engineering (COUPLED PROBLEMS 2009)*.

Gerace, S. (2006): *A Meshless Method Approach for Solving Coupled Thermoelasticity Problems*. Honors in the Major Thesis, University of Central Florida, 2006.

Gerace, S. (2007): An interactive framework for meshless methods analysis in computational mechanics and thermofluids. Master's thesis, University of Central Florida, 2007.

Gerace, S.; Erhart, K.; Divo, E.; Kassab, A. (2009): 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)*.

Golberg, M. A.; Chen, C. S.; Bowman, H. (1999): Some recent results and proposals for the use of radial basis functions in the bem. *Engineering Analysis with Boundary Elements*, vol. 23, no. 4, pp. 285–296.

Hardy, R. L. (1971): Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research*, vol. 76, pp. 1905–1915.

Hoffmann, K. A.; Chiang, S. T. (2004): *Computational Fluid Dynamics*, volume 2. Engineering Education System, 4 edition.

Kansa, E. J. (1990): Multiquadrics: A scattered data approximation scheme with applications to computational fluid dynamics i-surface approximations and partial derivative estimates. *Computers and Mathematics with Applications*, vol. 19, pp. 127–145.

Kansa, E. J. (1990): Multiquadrics: A scattered data approximation scheme with applications to computational fluid dynamics ii-solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Applications*, vol. 19, pp. 147–161.

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

Kim, J. J. (1978): *Investigation of Separation and Reattchment of a Turbulent Shear Layer: Flow Over a Backward-Facing Step*. PhD thesis, Stanford University, 1978.

Le, H.; Moin, P.; Kim, J. (1997): Direct numerical simulation of turbulent flow over a backward-facing step. *Journal of Fluid Mechanics*, vol. 330, pp. 349–374.

Lee, C. K.; Liu, X.; Fan, S. C. (2003): Local multiquadric approximation for solving boundary value problems. *Computational Mechanics*, vol. 30, no. 5-6, pp. 396–409.

Liou, M.-S.; Steffen, C. J. (1993): A new flux splitting scheme. *Journal of Computational Physics*, vol. 107, pp. 23–39.

Liu, G. R. (2003): *Mesh Free Methods: Moving beyond the Finite Element Method.* CRC Press, Boca Raton, FL.

Marshall, D. D.; Ruffin, S. M. (2004): An embedded boundary cartesian grid scheme for viscous flows using a new viscous wall boundary condition treatment. In *Proceedings of the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, no. AIAA 2004-0581, Reno, NV.

Sarler, B.; Tran-Cong, T.; Chen, C. (2005): Meshfree direct and indirect local radial basis function collocation formulations for transport phenomena. In Kassab, A.; Brebbia, C.; Divo, E.(Eds): *Boundary Elements XVII*, pp. 417–428. WIT Press.

Sarler, B.; Vertnik, R. (2006): Local explicit radial basis function collocation method for diffusion problems. *Computers and Mathematics with Applications*, vol. 51, pp. 1269–1282.

Schaback, R. (1995): Multivariate interpolation and approximation by translates of a basis function. *Series In Approximations and Decompositions*, vol. 6, no. 1, pp. 491–514.

Shewchuk, J. (2002): Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry*, vol. 22, no. 1-3, pp. 21–74.

Shu, C.; Ding, H.; Yeo, K. (2003): Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 941–954.

Stevens, D.; Power, H.; Lees, M.; Morvan, H. (2009): The use of pde centres in the local rbf hermitian method for 3d convective-diffusion problems. *Journal of Computational Physics*, vol. 228, no. 12, pp. 4606–4624.

Stevens, D.; Power, H.; Morvan, H. (2009): An order-n complexity meshless algorithm for transport-type pdes, based on local hermitian interpolation. *Engineering Analysis with Boundary Elements*, vol. 33, no. 4, pp. 425–441.

Tannehill, J. C.; Anderson, D. A.; Pletcher, R. H. (1997): *Computational Fluid Mechanics and Heat Transfer.* Hemisphere Publishing Corporation.

Tolstykh, A.; Shirobokov, D. (2003): On using radial basis functions in a ’finite difference mode’ with applications to elasticity problems. *Computational Mechanics*, vol. 33, pp. 68–79.

Tournois, J.; Alliez, P.; Devillers, O. (2007): Interleaving delaunay refinement and optimization for 2d triangle mesh generation. In Brewer, M. L.; Marcum, D.(Eds): *Proceedings of the 16th International Meshing Roundtable*, pp. 83–101, Seattle, Washington, USA. Springer Berlin Heidelberg.

Vertnik, R.; Sarler, B. (2006): Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems. *International Journal for Numerical Methods in Heat and Fluid Flow*, vol. 16, pp. 617–640.

Vertnik, R.; Sarler, B. (2009): Simulation of continuous casting of steel by a meshless technique. *International Journal of Cast Metals Research*, vol. 22, no. 1-4, pp. 311–313.

Vertnik, R.; Sarler, B. (2009): Solution of incompressible turbulent flow by a mesh-free method. *CMES: Computer Modeling in Engineering and Sciences*, vol. 44, no. 1, pp. 65–95.

Vertnik, R.; Zaloznik, M.; Sarler, B. (2006): Solution of transient direct-chill aluminium billet casting problem with simultaneous material and interphase moving boundaries by a meshless method. *Engineering Analysis with Boundary Elements*, vol. 30, pp. 847–855.

Wilcox, D. C. (2006): *Tubulence Modeling for CFD.* DCW Industries.

Wright, G. B.; Fornberg, B. (2006): Scattered node compact finite difference-type formulas generated from radial basis functions. *Computational Methods*, vol. 9810751, pp. 1391–1395.

Zahab, E. Z.; Divo, E.; Kassab, A. J. (2009): A localized collocation meshless method (lcmm) for incompressible flows cfd modeling with applications to transient hemodynamics. *Engineering Analysis with Boundary Elements*, vol. 33, no. 8–9, pp. 1045–1061.

Zahab, E. Z.; Divo, E.; Kassab, A. J. (2009): A meshless cfd approach for evolutionary shape optimization of bypass grafts anastomoses. *Inverse Problems in Science and Engineering*, vol. 17, no. 3, pp. 411–435.

Zhao, X.; Chen, J.; Li, Z. (2002): A qtm-based algorithm for generation of the voronoi diagram on a sphere. In Richardson, D.; van Oosterom, P.(Eds): *Advances in Spatial Data Handling: Proceedings from the 10th International Symposium on Spatial Data Handling*, pp. 269–285. Springer Berlin.