

STARS


University of Central Florida
STARS

Electronic Theses and Dissertations, 2004-2019

2006

Automated Adaptive Data Center Generation For Meshless Methods

Eric Mitteff
University of Central Florida

 Part of the [Mechanical Engineering Commons](#)
Find similar works at: <https://stars.library.ucf.edu/etd>
University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Mitteff, Eric, "Automated Adaptive Data Center Generation For Meshless Methods" (2006). *Electronic Theses and Dissertations, 2004-2019*. 761.
<https://stars.library.ucf.edu/etd/761>



AUTOMATED ADAPTIVE DATA
CENTER GENERATION FOR
MESHLESS METHODS

by

ERIC ALAN MITTEFF
B. S. University of Central Florida, 2004

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Mechanical, Materials, and Aerospace Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2006

ABSTRACT

Meshless methods have recently received much attention but are yet to reach their full potential as the required problem setup (i.e. collocation point distribution) is still significant and far from automated. The distribution of points still closely resembles the nodes of finite volume-type meshes and the free parameter, c , of the radial-basis expansion functions (RBF) still must be tailored specifically to a problem. The localized meshless collocation method investigated requires a local influence region, or topology, used as the expansion medium to produce the required field derivatives. Tests have shown a regular cartesian point distribution produces optimal results, however, in order to maintain a locally cartesian point distribution a recursive quadtree scheme is herein proposed. The quadtree method allows modeling of irregular geometries and refinement of regions of interest and it lends itself for full automation, thus, reducing problem setup efforts. Furthermore, the construction of the localized expansion regions is closely tied up to the point distribution process and, hence, incorporated into the automated sequence. This also allows for the optimization of the RBF free parameter on a local basis to achieve a desired level of accuracy in the expansion. In addition, an optimized auto-segmentation process is adopted to distribute and balance the problem loads throughout a parallel computational environment while minimizing communication requirements.

TABLE OF CONTENTS

LIST OF FIGURES.....	iv
CHAPTER 1: INTRODUCTION.....	1
CHAPTER 2: MESHLESS METHODS.....	3
2.1 Global Meshless Methods.....	3
2.2 Local Meshless Methods.....	6
CHAPTER 3: QUADTREE.....	7
CHAPTER 4: LOCAL EXPANSION.....	10
CHAPTER 5: FREE PARAMETER.....	12
CHAPTER 6: PARALLEL SEGMENTATION.....	21
CHAPTER 7: RESULTS.....	24
7.1 Quadtree.....	25
7.2 Local Expansion.....	30
7.3 Segmentation.....	30
CHAPTER 8: CONCLUSION.....	34
LIST OF REFERENCES.....	35

LIST OF FIGURES

Figure 1 - Arbitrary Global Domain.....	4
Figure 2 - Example of dependency of the shape parameter for Multiquadric RBFs.....	5
Figure 3 - Arbitrary Local Domains.....	6
Figure 4 - Application of quadtree to an arbitrary geometry.....	8
Figure 5 - Radial inflation about an interior, boundary and clustered data centers.....	11
Figure 6 - Sample domain of influence.....	12
Figure 7 - Field variable, T , over sample domain.....	14
Figure 8 - First derivative, $\frac{\partial T}{\partial x}$, over sample domain.....	14
Figure 9 - Second derivative, $\frac{\partial^2 T}{\partial x^2}$, over sample domain.....	15
Figure 10 - Cross derivative, $\frac{\partial^2 T}{\partial x \partial y}$, over sample domain.....	15
Figure 11 - Percent error vs. free parameter, c	16
Figure 12 - Conditioning number of the collocation matrix, Ψ vs. free parameter, c	17
Figure 13 - Percent error vs. conditioning number of the collocation matrix, Ψ	18
Figure 14 - Typical topology nodal distribution.....	19
Figure 15 - Iterative procedure for finding the free parameter, c	20
Figure 16 - Auto-segmentation through Voronoï cells.....	22
Figure 17 - Sample segmentation individuals.....	23
Figure 18 - Point generation over a cavity.....	26
Figure 19 - Adaptive data center insertion for a moving wave.....	27
Figure 20 - Adaptive data center insertion for heat conduction.....	28
Figure 21 - Solution times for dynamic nodal point generation.....	29

Figure 22 - Collocation topology for internal, boundary, and corner data centers.....	30
Figure 23 - Auto-segmentation applied to a meshless model.....	31
Figure 24 - Auto-segmentation evolution for a square.....	32
Figure 25 - Auto-segmentation evolution for flow over a cylinder.....	33
Figure 26 - Evolution of the objective function of the genetic algorithm.....	33

CHAPTER 1

INTRODUCTION

Traditional computational techniques such as finite difference and finite volume methods (FDM and FVM) [1,2], the finite element method (FEM) [3] and the boundary element method (BEM) [4-6] have been efficiently and routinely applied to an assortment of science and engineering field problems. Despite the great success of these numerical methods they all have a common drawback, for most cases the time consumed in mesh generation or preprocessing is greater than that of the time consumed for the actual computation. For example, the generation of a typical 3D mesh over a film-cooled turbine blade may take several weeks, while the computation itself may take only a few hours on a high performance computer. Recently, meshless or mesh-free methods have become a topic of much research in an attempt to mitigate or reduce the effort and time needed for modeling of a solution domain. The great benefit of the meshless methods is that they do not require a nodal connectivity or mesh which is the source of much burden in a mesh method such as FEM, FDM or FVM. Without the need for a mesh nodal points must still be generated in a somewhat uniform manner. One such approach can be obtained through quadtree subdivision. Here, a fully autonomous procedure is developed to employ quadtree subdivision to meshless models for the purpose modeling regular and irregular geometries.

Parallelizing computational solvers have made efficient use of multiple processors for traditional mesh methods as well as global meshless methods. The burden of parallelizing a domain lies with the model preprocessing stage which requires manual decomposition techniques. Traditionally, decomposed meshless models required the use of artificial interior boundaries, or interfaces, in order to iteratively solve independent well posed problems [7].

With the introduction of a localized meshless collocation method, fully bounded sub domains are not a necessity due the nature of the explicit solution schemes. This allows the solution of a partial domain for each individual iteration given that the information from the previous iteration is made available. An automated parallel segmentation will be developed through the use of Voronoï cells, or Dirichlet tessellation to break up the domain into segments for distribution to multiple processors.

This thesis will begin by reviewing the fundamental developments of both the global meshless methods and the local meshless methods and the controlling parameters that affect their behavior. This will introduce the problems encountered and set the stage for the motivation of this thesis to address these issues. Next, the automated data center distribution techniques will be discussed along with an implementation of a dynamic adaptive algorithm as well. Following this, a method of gathering neighboring points, called a local expansion, will be described which will is used by the local meshless methods to declare the scope of the influence of the domain upon each data center in the field. Once the local expansions are defined, the free parameter, or the shape parameter c , will be optimized by analyzing various aspects of the dependent functions. Specifically, this section will address the issues corresponding to accuracy and conditioning of the collocation matrices. Then, a method will be developed to decrease the total computation time through the use of parallel machines. This will be achieved by segmenting the problem into smaller pieces, as described earlier, by respecting the relative computing power of each processor while minimizing total communication effort between processors. Finally, verification with examples will show that dynamic models can be made effortlessly and total computation time can be significantly reduced with an automated segmentation routine.

CHAPTER 2

MESHLESS METHODS

Traditionally, solving engineering problems involved the use of a mesh method such as finite elements, finite volume, finite differencing or boundary elements. Each of these methods all require some form of meshing or nodal connectivity during the preprocessing stage of any analysis. With the continuous advancement of affordable computing power analysts are solving much larger and more complex problems which, in turn, leads to extended preprocessing time. It is this preprocessing, or meshing, that is quickly becoming the most time consuming part of model development for a complex system. The goal of the meshless methods is to completely remove this preprocessing stage by removing the need for meshing all together, hence, the term meshless methods. The meshless methods are based on radial basis functions that interpolate field data from neighboring data centers regardless of their spatial distribution. It is this freedom from structured node distribution that make this method so appealing to researchers.

Two basic types of meshless methods were recently developed; global and local. The global meshless methods relied upon interpolating over an entire field domain for evaluation at any given point in space, whereas the local meshless methods interpolated a field variable over only a small region, or locally.

2.1 Global Meshless Methods

Global meshless methods rely in the fact that a given domain, Ω , can be interpolated by collocating about the data centers with some radial basis function, $\chi(x)$. For the arbitrary region shown in figure 0,

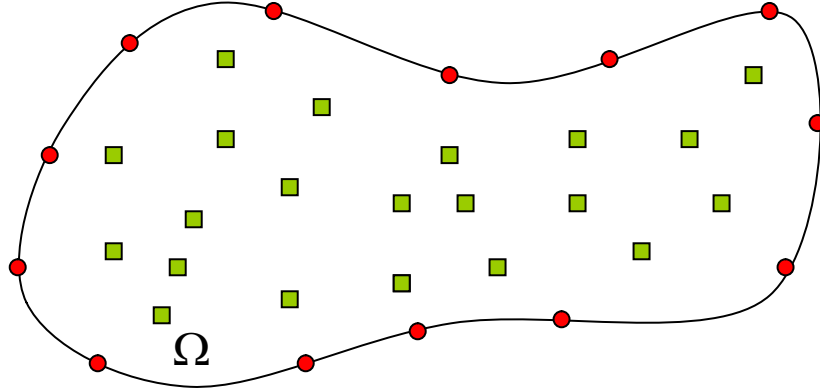


Figure 1 - Arbitrary Global Domain

which can be described by some function $f(x)$, the global meshless collocation is given by equation 1.

$$f(x) = \sum_{i=1}^N \alpha_i \chi_i(x) \quad (1)$$

where N is the total number of points in the domain, α_i is the expansion coefficients for f , and χ_i is the basis functions. If the field variable, f , is known at the collocation points then the basis function χ , can be evaluated at the collocation points. Given the values that are known, the only unknown in this equation is the expansion coefficient vector, α .

The radial basis functions, χ , consist of algebraic expressions defined in terms of the Euclidian distance from an 'expansion point' or data center to a general field point. Several radial basis functions that have been investigated include:

i) Polyharmonic Splines:

$$\chi_j(x) = r_j^{2n}(x) \ln(r_j(x)) \quad \text{in } 2D \quad ()$$

$$\chi_j(x) = r_j^{2n-1}(x) \quad \text{in } 3D \quad ()$$

ii) Multiquadrics:

$$\chi_j(x) = [r_j^2(x) + c^2]^{n-\frac{3}{2}} \quad ()$$

iii) Gaussian:

$$\chi_j(x) = \exp\left[-\frac{r_j^2(x)}{c^2}\right] \quad ()$$

where:

$$r_j(x) = \sqrt{(x - x_j)^2 + (y - y_j)^2} \quad \text{in } 2D$$

In each of the proposed radial basis functions the parameter n is some positive integer and the free parameter c , which is sometimes referred to as the shape parameter, governs the accuracy of the interpolation. For example, figure 2 shows the response of the Hardy Multiquadric RBF, where $n = 1$.

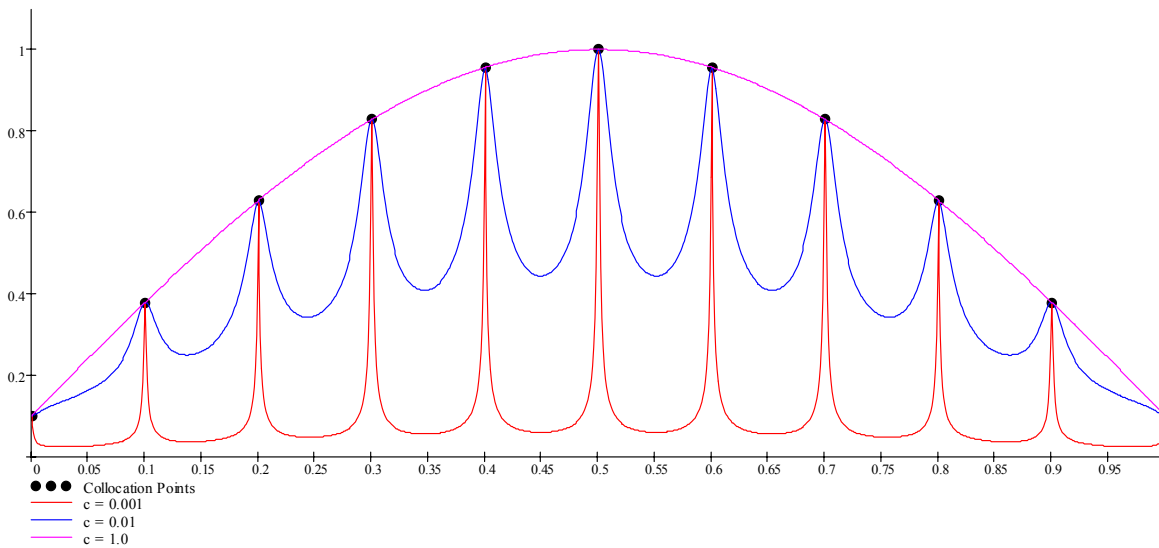


Figure 2 - Example of dependency of the shape parameter for Multiquadric RBF's

The resulting interpolation is heavily dependant upon the free parameter, c , which is left for optimization later in this paper.

2.2 Local Meshless Methods

The local meshless technique is based on the global meshless method in that it still uses the idea of solving for expansion coefficients to interpolate the field variable, however, instead of building a single large set of expansion coefficients for the entire domain, a small set of expansion coefficients are built for each data center.

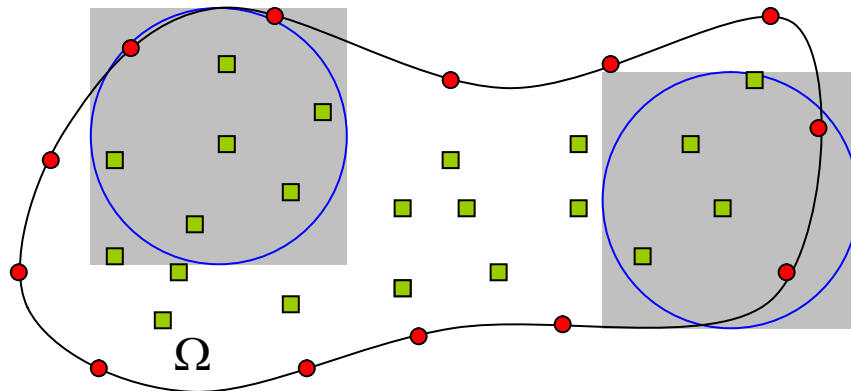


Figure 3 - Arbitrary Local Domains

These small sets of expansion coefficients are based on the influence of neighboring points that create smaller sub-domains, or local expansions, as shown in figure 0. This allows for an explicit iterative solution process while the global meshless methods are implicit in nature which are based on the direct solution of a large fully populated matrix. The local meshless methods involve many small vector multiplications similar to an explicit finite difference method which helps to alleviate much of the computational burden imposed by solving large, ill-conditioned matrices which leads to more consistent reliable solutions.

CHAPTER 3

QUADTREE

Although meshless methods do not require a mesh, they do, however, require a point distribution throughout the domain. The point distribution can be accomplished through an array of automated techniques such as Delaunay, advancing front or a recursive quadtree method.

Past research[7] has shown that although a non-uniform point distribution can be used for solving problems in meshless methods, uniform distributions have proven to produce superior results for both linear and non-linear problems. Also, distributing points in a uniform fashion is a trivial task relative to other unstructured methods. It is for these reasons that a cartesian based point distribution method was chosen as the method of choice. Quadtree methods, based on the cartesian coordinate system, lends itself for full automation to produce uniformly distributed data centers as well as the intrinsic ability to recursively cluster areas of interest such as boundary layers or vortices. This frees the analyst from spending undue effort meshing a given geometry.

A quadtree is a tree data structure whose construction is based on the recursive decomposition of the cartesian plane [8]. Specifically, a region based quadtree is defined by the recursive partitioning of the cartesian plane into four equally sized quadrants which lie parallel to the coordinate axis. This recursion can be repeated until a model resolution criterion or partitioning limit is achieved.

The quadtree method has the ability to model irregular geometries and refinement of areas of interest and it lends itself the ability for complete automation. Complete automation leads to a reduced amount of effort and time by the user. The use of a quadtree method has the capability to trace the boundary by clustering quadtrees to gain resolution for good model geometry approximation. Figure 4 shows an example of the procedure that is undergone to bring

an initial quadtree discretization to the final quadtree discretization followed by the boundary and interior point insertion. The corners of the quadtree squares that lie on the boundary become a boundary point whereas any other quadtree corner that lies within the enclosed boundary becomes an interior point.

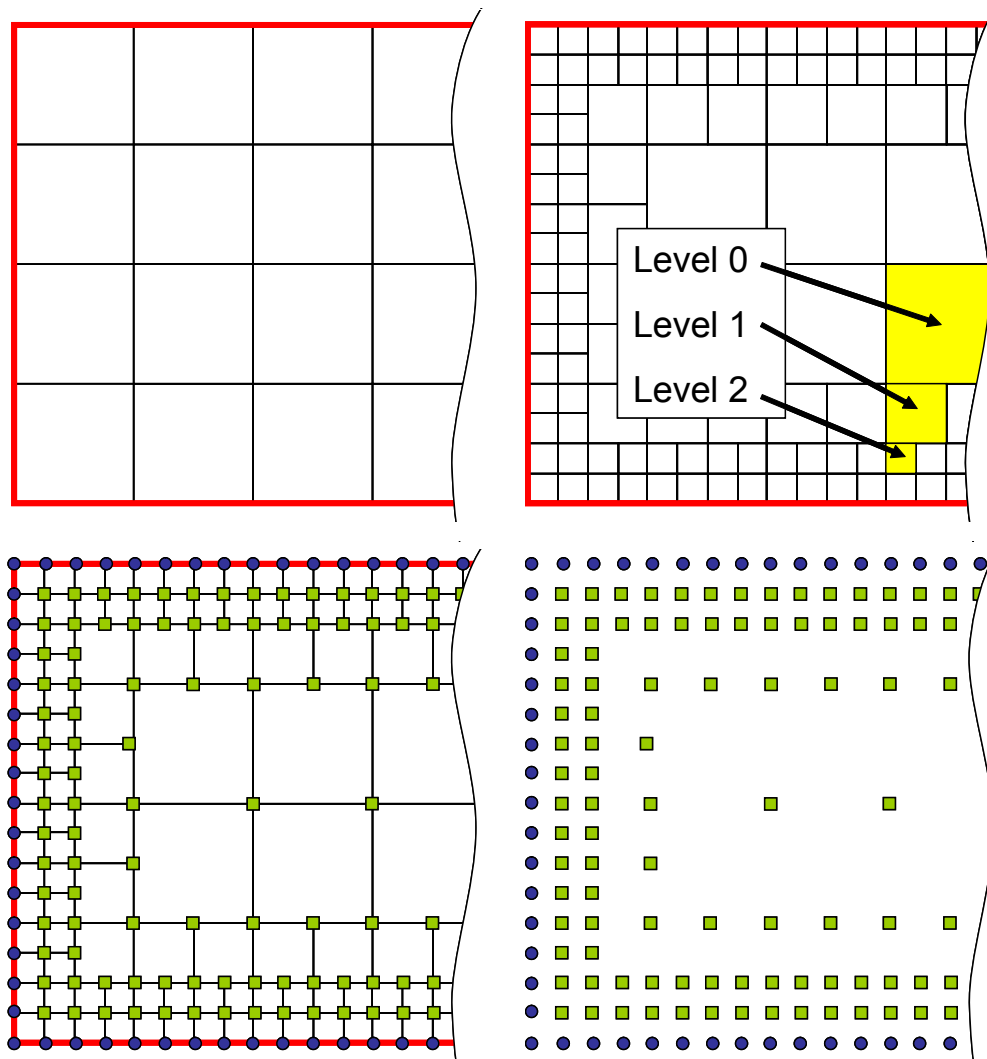


Figure 4 - Application of quadtree to an arbitrary geometry; (a) initial quadtree distribution, (b) quadtree blocks refined near boundaries, (c) boundary and interior nodes placed at quadtree block corners, (d) final data center distribution

Due to the autonomous nature of the nodal point generation it is conceivable that the generation of the data centers can be in response to a field variable, such as a large gradient, during the solution process. Areas that have a large gradient will generally require a higher resolution than other areas to accurately capture any behavior such as a fluid boundary layer with no-slip conditions imposed.

CHAPTER 4

LOCAL EXPANSION

The localized meshless collocation method investigated requires a local influence region, or topology, used as the expansion medium to produce the required field derivatives. The creation of the local expansion regions is accomplished by incrementally inflating a circle to encompass other data centers until some criteria has been met, typically a fixed number of points of influence. Although including only a small number of points in a local expansion will result in less computational effort, they may not always produce an accurate interpolation of the field variable or any differential of the field variable. On the other hand, an excessive amount of points may produce a smoother field interpolation or differential but radial basis functions are notorious for their oscillative behavior between data centers. Also, expansions with a large number of points results in large expansion matrices which, in turn, leads to additional roundoff errors and extended computation time requirements. A balance between the two must be achieved for a given governing equation through trial and error methods or other optimization techniques.

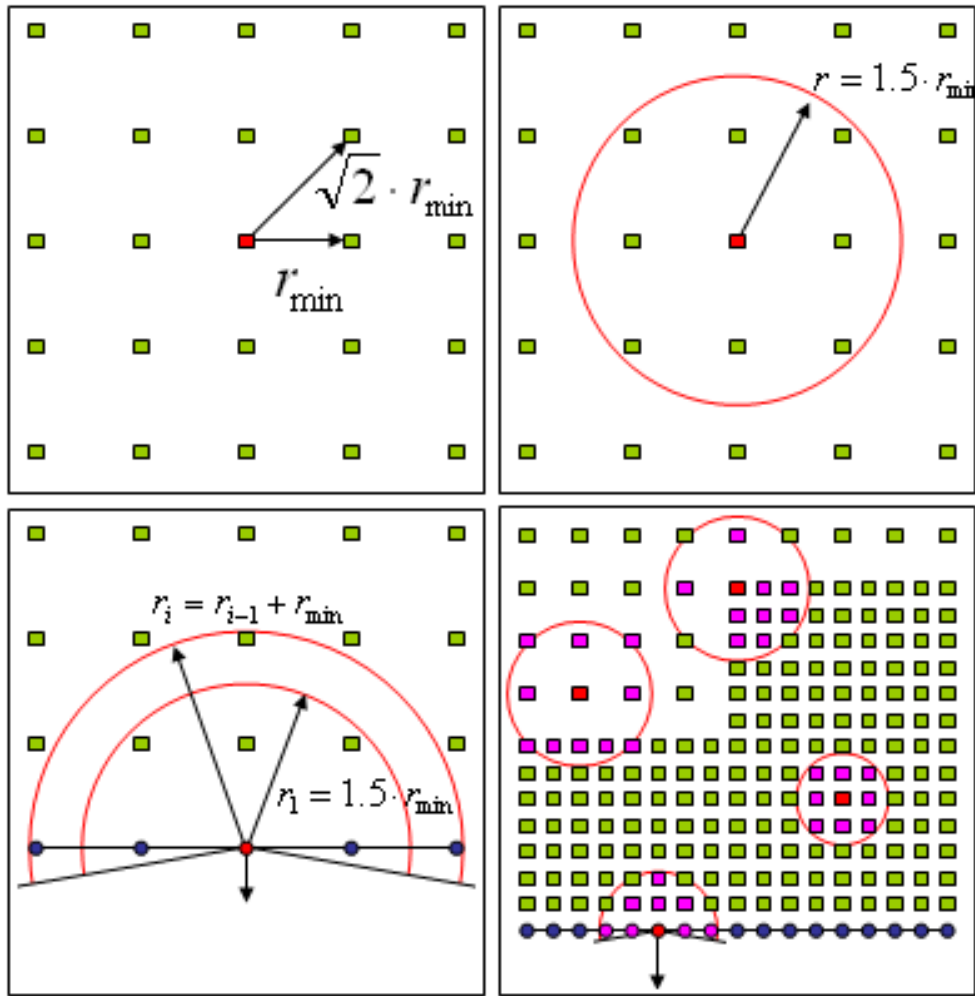


Figure 5 - Radial inflation about an interior, boundary and clustered data centers

The inflating bubble around a data center is initially set to one and a half times the distance to the nearest point, r_{min} . The initial bubble radius is sufficient enough to capture the required points if they are distributed across a perfectly cartesian grid. Otherwise, the bubble is incrementally increased in size by a factor of r_{min} until all the necessary number of points are captured by the local expansion as shown in Figure 5. In the case of a local expansion about a boundary point care is taken not to accidentally acquire points that exist on the outside of the local area of interest, such as a point around the corner of a reentrant region.

CHAPTER 5

FREE PARAMETER

Increasing the free parameter c , which is commonly referred to as the shape parameter, controls the flattening characteristic of the expansion functions, such as Equation (3). The flat character of the expansion functions is favorable in the sense that a smoother and more accurate representation of the field variable can be achieved [9,10]. However, as the expansion functions are flattened by an increase in the free parameter c , the simultaneous system of linear equations becomes more ill-conditioned.

As an example of the influence that the free parameter imposes a sample problem was studied. Figure 6 shows a domain the size of the unit square containing 25 points for an RBF collocation.

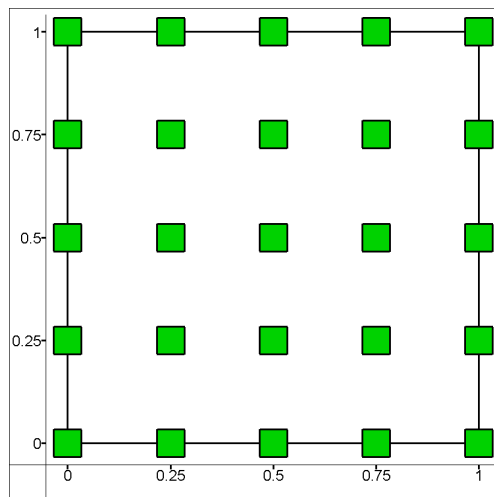


Figure 6 - Sample domain of influence

A simple function was applied in order to give each point in the domain a field value. This particular function was chosen to have a Laplacian of 0 and a non-zero cross derivative as shown in equations 2.

$$T(x, y) = x^3y - xy^3 \quad (2)$$

$$\frac{\partial T}{\partial x} = 3x^2y - y^3$$

$$\frac{\partial^2 T}{\partial x^2} = 6xy$$

$$\frac{\partial^2 T}{\partial x \partial y} = 3x^2 - 3y^2$$

$$\nabla^2 T = 0$$

The collocation matrix, Ψ , can be built using the inverse multiquadric radial basis function as demonstrated in equations 3 and 4.

$$\psi_j(x) = \frac{1}{\sqrt{r_j^2(x) + c^2}} \quad (3)$$

$$\Psi_{ij}(c) = \psi_j(x_i) \quad (4)$$

Figure 7 shows the exact field variable, T , and resulting RBF interpolation for various values for the free parameter, c . For small values of c the field is not approximated very well, but as c increases the field behaves better.

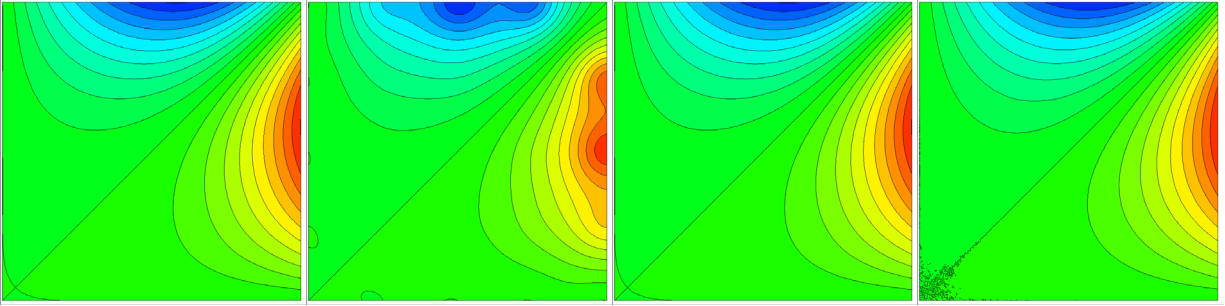


Figure 7 - Field variable, T , over sample domain (a) exact, (b) with $c = 0.1$, (c) $c = 1.2$ and (d) $c = 20$

The first derivative of the field variable, $\frac{\partial T}{\partial x}$, shown in figure 8 is again well behaved for larger values of c .

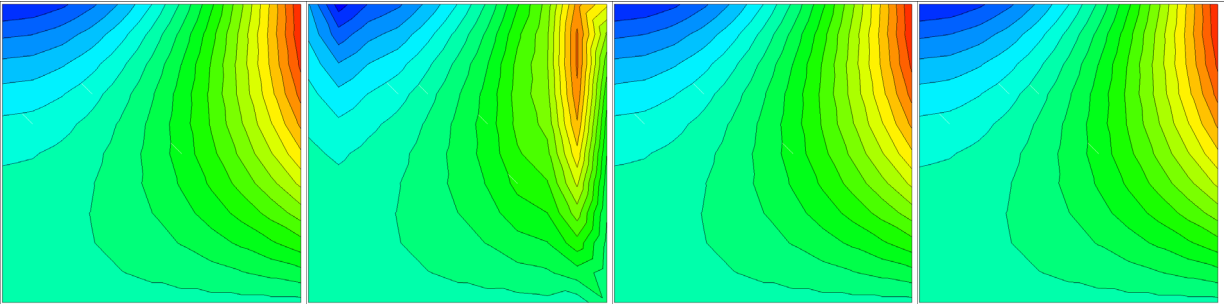


Figure 8 - First derivative, $\frac{\partial T}{\partial x}$, over sample domain (a) exact, (b) with $c = 0.1$, (c) $c = 1.2$ and (d) $c = 20$

The behavior of the second derivative, $\frac{\partial^2 T}{\partial x^2}$, produces unacceptable results for small values of c and once again appears to be stable for larger c values as shown in figure 9.

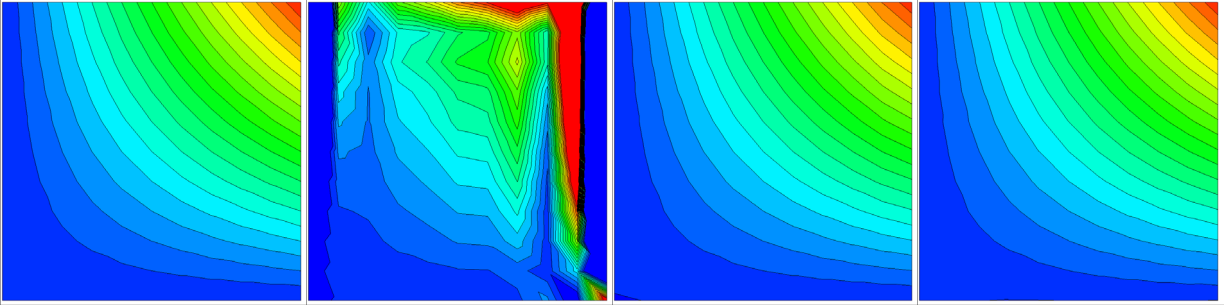


Figure 9 - Second derivative, $\frac{\partial^2 T}{\partial x^2}$, over sample domain (a) exact, (b) with $c = 0.1$, (c) $c = 1.2$ and (d) $c = 20$

Once again, the cross derivative of the field variable, T , is ill-behaved for small c values and produces seemingly well behaved results for larger c values.

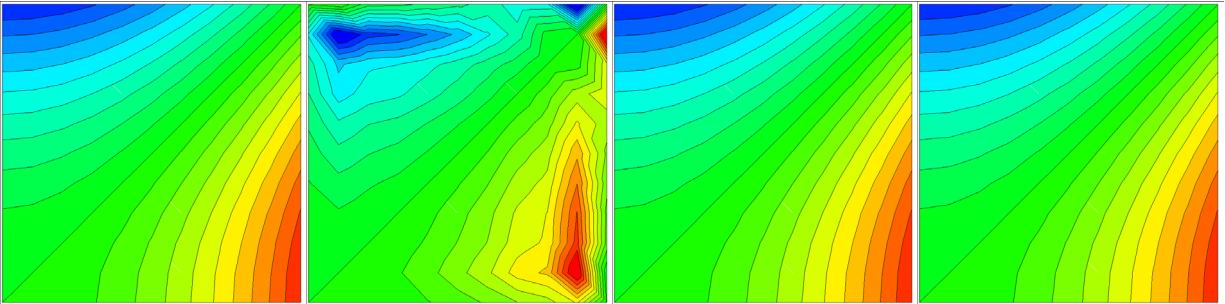


Figure 10 - Cross derivative, $\frac{\partial^2 T}{\partial x \partial y}$, over sample domain (a) exact, (b) with $c = 0.1$, (c) $c = 1.2$ and (d) $c = 20$

The question could be asked of which value of c should be chosen. A graph was produced to show the relationship between the free parameter, c , and the relative error of the field derivatives at the collocation points which can be seen in figure 11. The error clearly decreases as c increases, however, a point is reached in which the error sharply increases. The goal is to find the optimum c that yields the most accurate results.

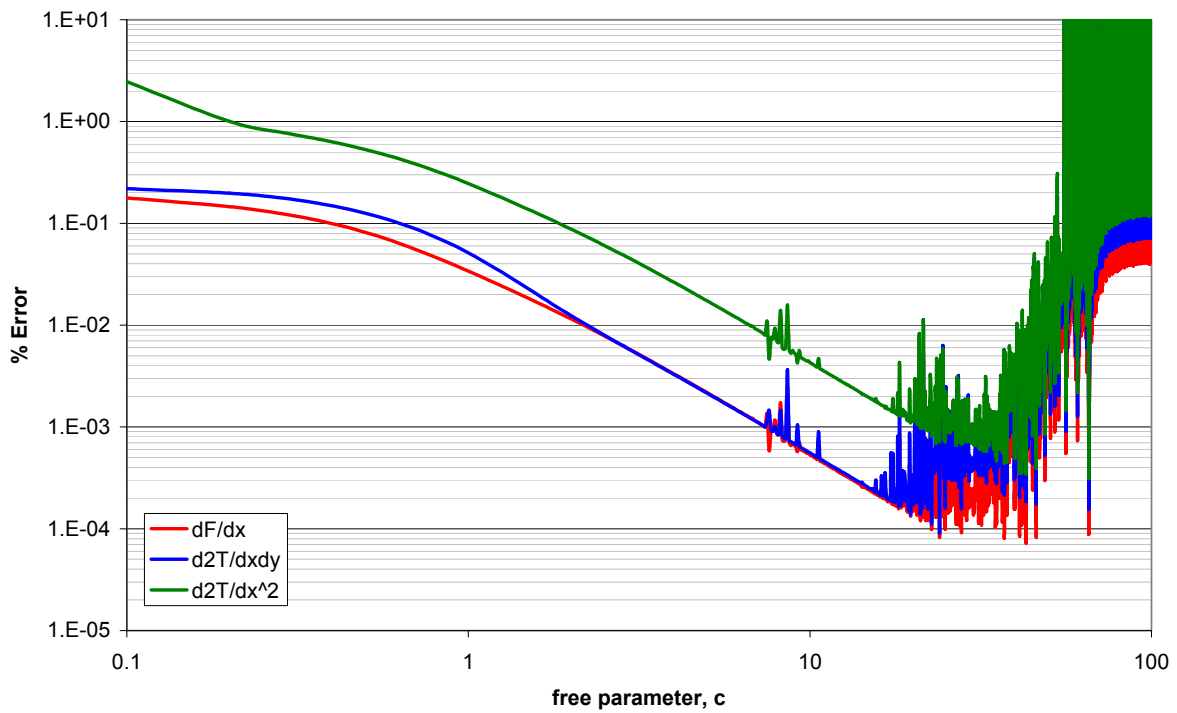


Figure 11 - Percent error vs. free parameter, c .

The reason for the sharp increase in the error as the free parameter, c , increases is due to the fact that the collocation matrix, Ψ , becomes linearly dependent. The linear dependency, or level of singularity, can be quantified by finding the conditioning number of the collocation matrix itself. The conditioning number is found by dividing the largest element by the smallest element in the resulting matrix produced by singular value decomposition of the collocation matrix which is shown below in equation 5.

$$K(c) = \frac{\text{Max}[SVD(\Psi)]}{\text{Min}[SVD(\Psi)]} \quad (5)$$

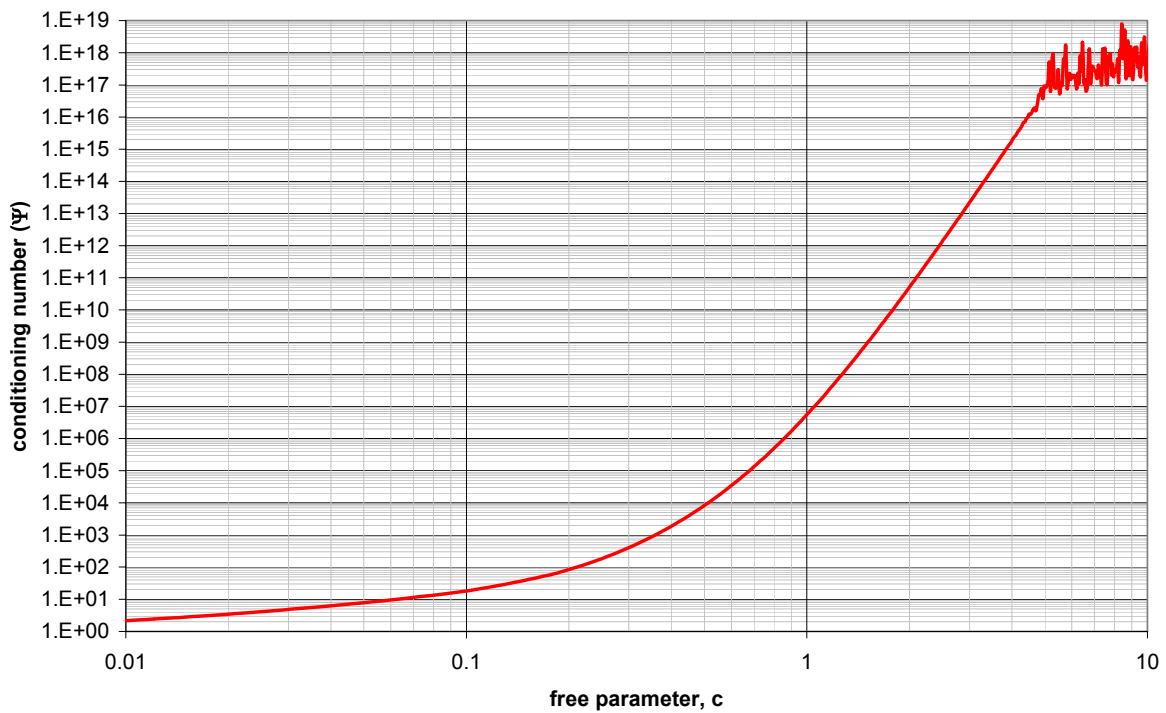


Figure 12 - Conditioning number of the collocation matrix, Ψ vs. free parameter, c

Now, viewing the error as a function of the conditioning number of the collocation matrix in figure 13, instead of as a function of the free parameter as in figure 11, gives a range of acceptable conditioning numbers to target. The free parameter, c , can be backed out given a conditioning number through a series of iterations such as the procedure proposed in figure 15.

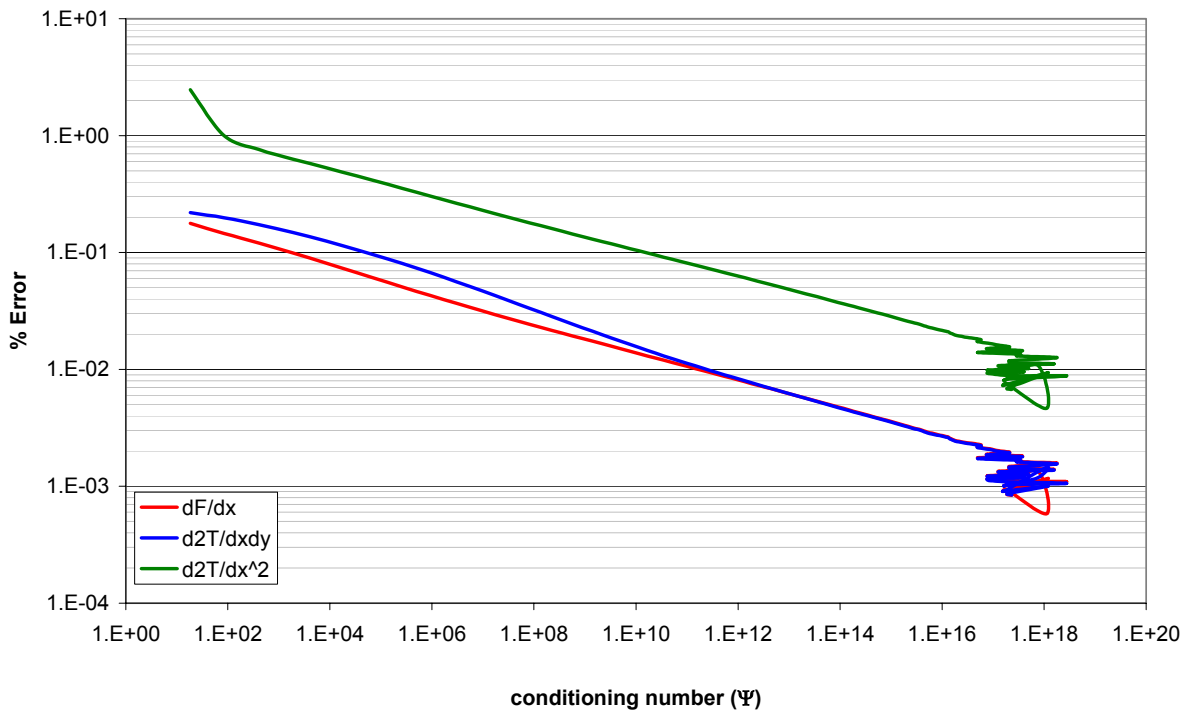


Figure 13 - Percent error vs. conditioning number of the collocation matrix, Ψ

In an effort to maximize accuracy, the free parameter c , will be selected such that the conditioning number of the resulting collocation matrix will be forced to a value of 10^{12} through a series of linear searches with an initial guess of about forty times the average distance between nodes of a given topology where:

$$r_{avg} = \frac{1}{N} \sum_{i=1}^N r_i \quad (6)$$

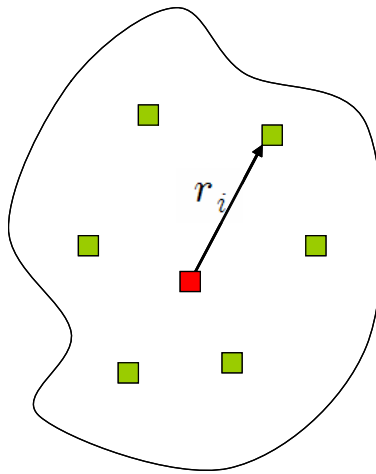


Figure 14 - Typical topology nodal distribution

Once an initial c value is selected the collocation matrix is built and the conditioning number then found. If the conditioning number is too low then c is increased by a factor of 1.1. If the conditioning number is too high then c is multiplied by a factor of 0.95. This procedure is repeated until the target conditioning number is produced.

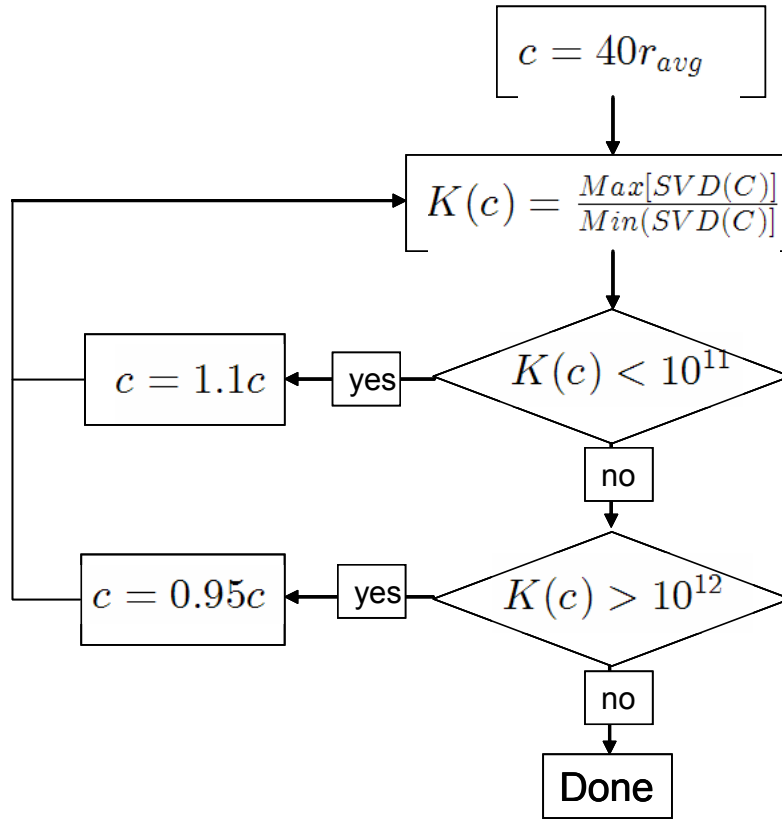


Figure 15 - Iterative procedure for finding the free parameter, c

CHAPTER 6

PARALLEL SEGMENTATION

One of the easiest ways to minimize the computational time necessary to solve a given problem is to implement a parallel approach. In order to maintain autonomy a procedure must be devised to methodically decompose, or segment, a model of interest. Dirichlet [11] first proposed a method whereby given a set of points P_i , in arbitrary space, could be systematically decomposed into a set of convex regions R_i such that the region R_i is the space closer to point P_i than any other point. This geometrical construction, known as Dirichlet tessellation, results in a set of non overlapping convex regions called Voronoï regions. Each Voronoï region is then assigned to a processor and each node within that region is assigned to that processor. With the nodes grouped into regions only the nodes that require information from nodes in other regions need to be communicated thereby creating a need to minimize the amount of nodes that need their data passed between processors. Also, the number of nodes in each Voronoï region needs to be optimized to fit the processor in a given cluster. The geometric placement of the processor points, P_i , governs both the size of each region and the communication effort between processors. Therefore, the geometric placement of the processor nodes can be optimized by employing a discrete genetic algorithm.

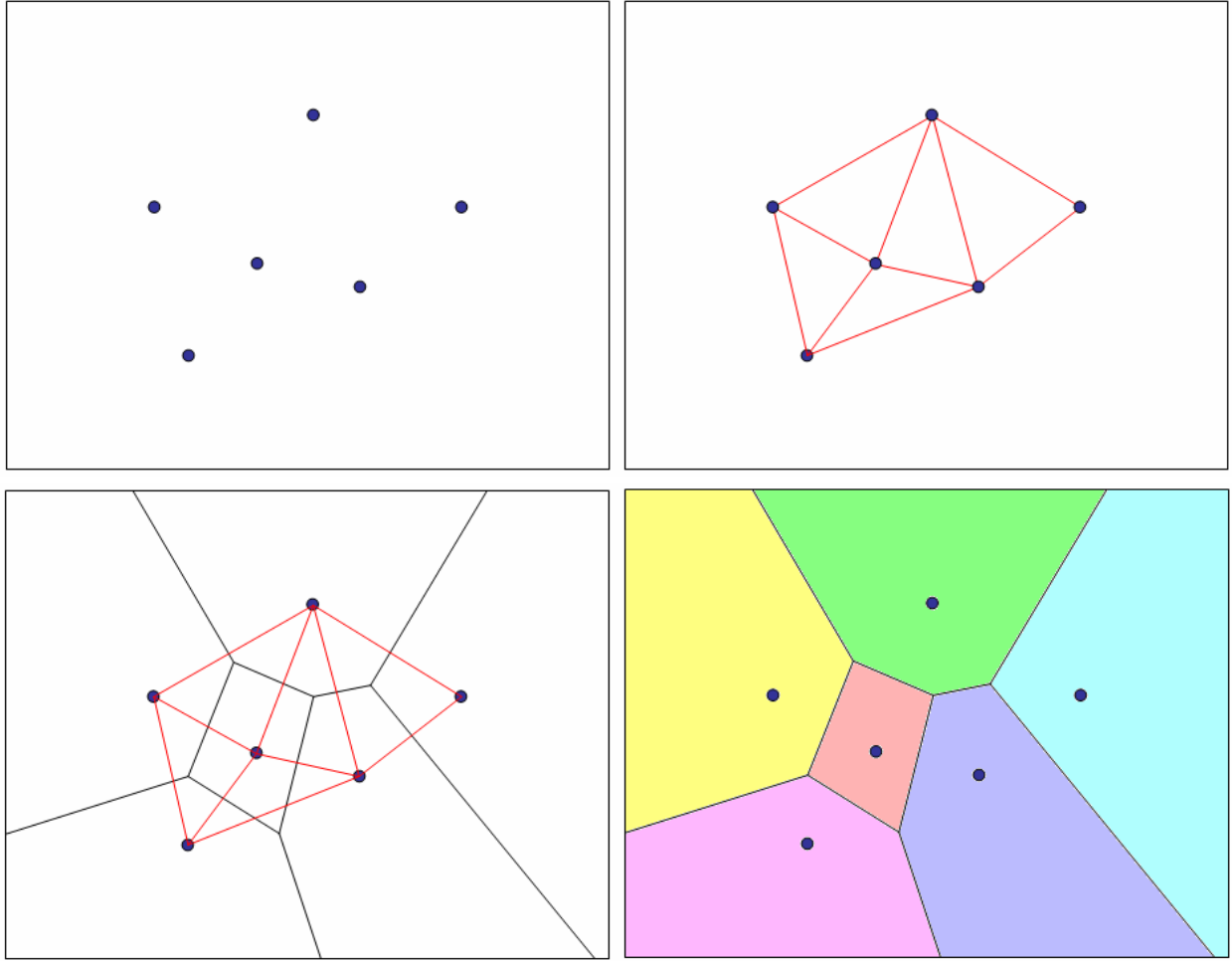


Figure 16 - Auto-segmentation through Voronoi cells; (a) Point distribution, (b) Delaney triangulation, (c) triangulation and Voronoi diagram, (d) Voronoi cells

$$Fitness = \sqrt{\frac{1}{N} \sum [NP_{i,optimal} - NP_{i,actual}]^2} + \beta \cdot NC_{actual} \quad (7)$$

Here, N is the number of Voronoi processor regions, NP_i is the number of nodes within R_i , NC is the number of nodes that are required to be communicated between processors and β is a tuning parameter dependent upon the performance and communication abilities of a given cluster. The objective is to minimize the fitness function such that the nodes are distributed

proportionally based on individual machine performance and communication of nodal data is minimized.

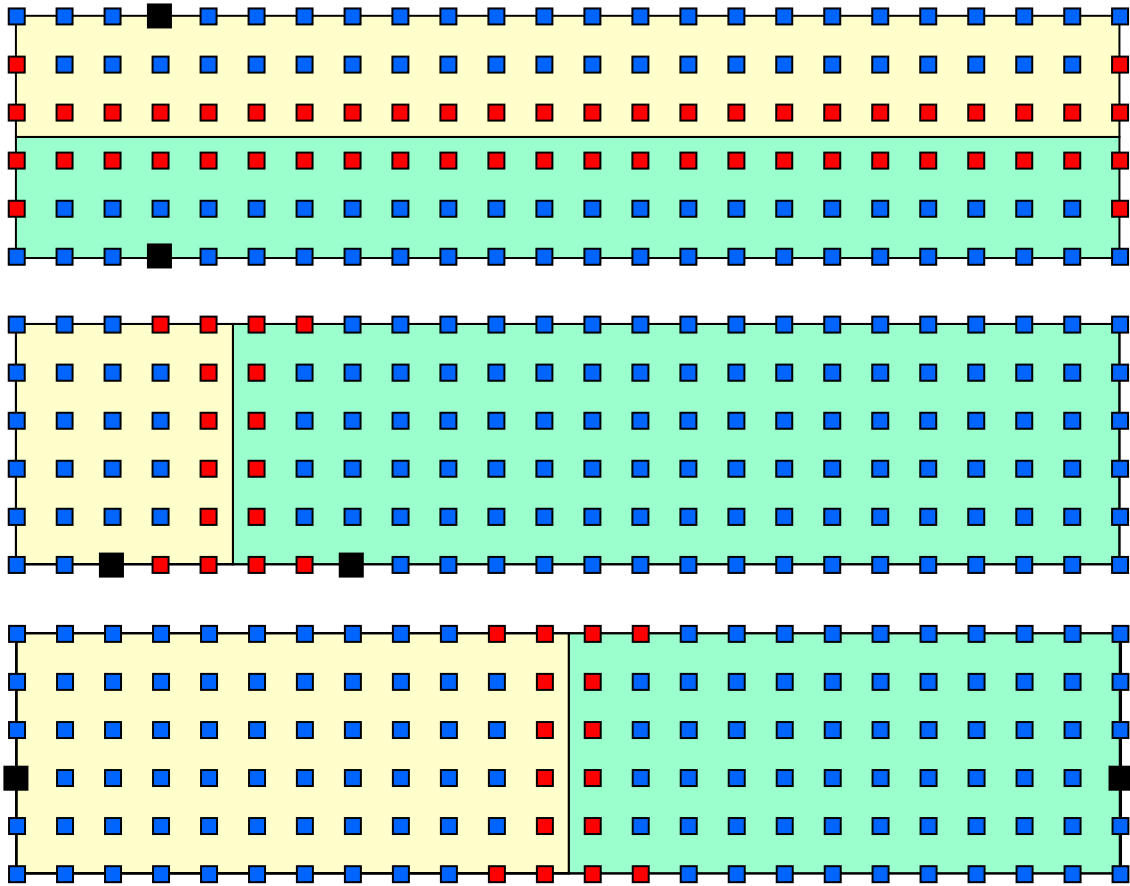


Figure 17 - Sample segmentation individuals; (a) node distribution is optimal but communication is not (b) communication is optimal but node distribution is not (c) both node distribution and communication are optimal

Figure 17 shows an example of three potential cases produced by the genetic algorithm. The first case is proportionally well distributed but communication may be prohibitively slow. On the other hand, the second case shows a good example of optimal communication requirements however, they are not proportionally well fit for similar processors. The third case

represents the optimal configuration; minimal communication and proportionally well distributed.

CHAPTER 7

RESULTS

The quadtree nodal distribution routine was employed for several cases to test the capabilities over an irregularly shaped geometry. Also, the ability to repetitively generate nodal distributions for moving geometries was tested for a moving wave. The local expansion method, which is used to define local topologies, was tested on a discretized artery with a bypass graft. Finally, the auto-segmentation routine was employed to test the ability to break domains into smaller groups for efficient parallel computation.

7.1 Quadtree

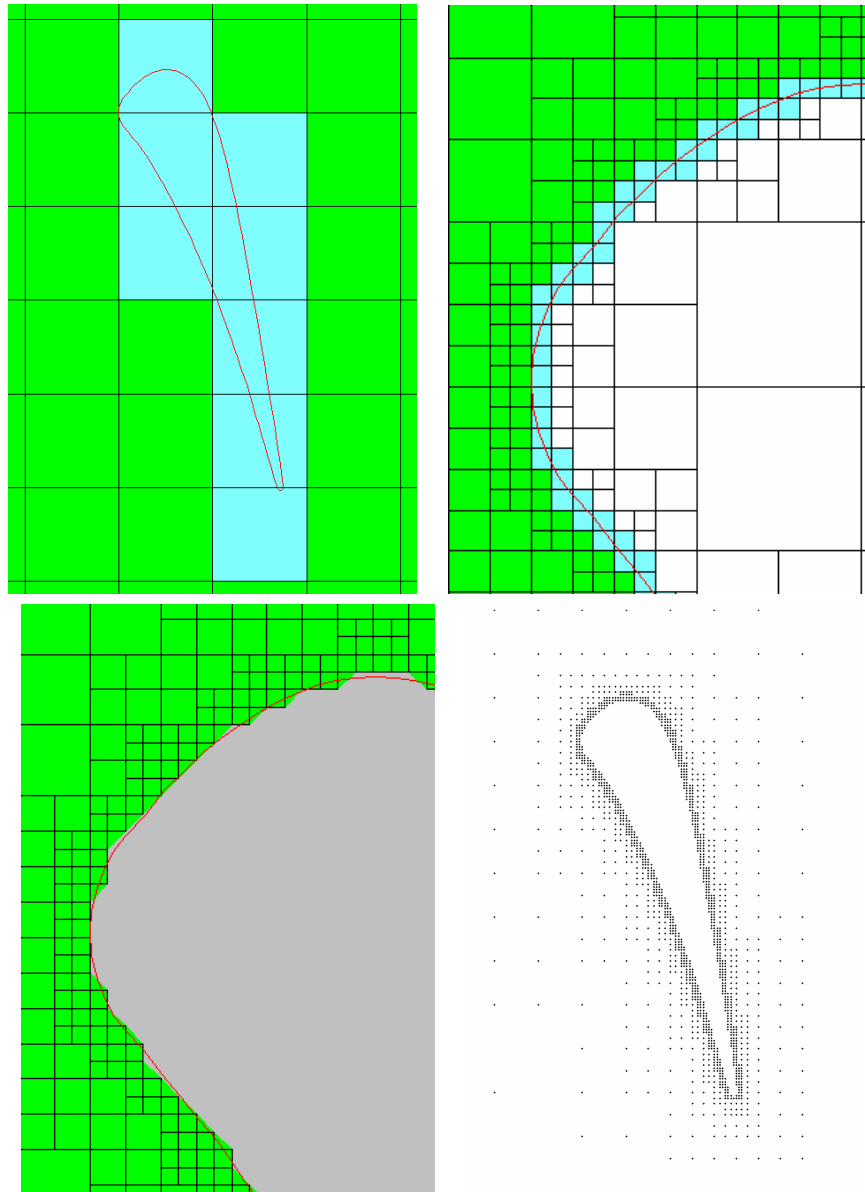


Figure 18 - Point generation over a cavity. Green cells represent regions within the domain, white cells are outside the domain and blue cells lie on the geometric boundary; (a) Initial quadtree distribution, (b) close up to edge, (c) final quadtree distribution, (d) final nodal distribution

A cavity is subjected to a quadtree discretization as shown in Fig. 18. As the cavity boundary is discretized the desired level of model resolution is achieved. After five levels of discretization beyond the initial quadtree distribution the cavity boundary is well approximated while maintaining a perfect Cartesian nodal distribution.

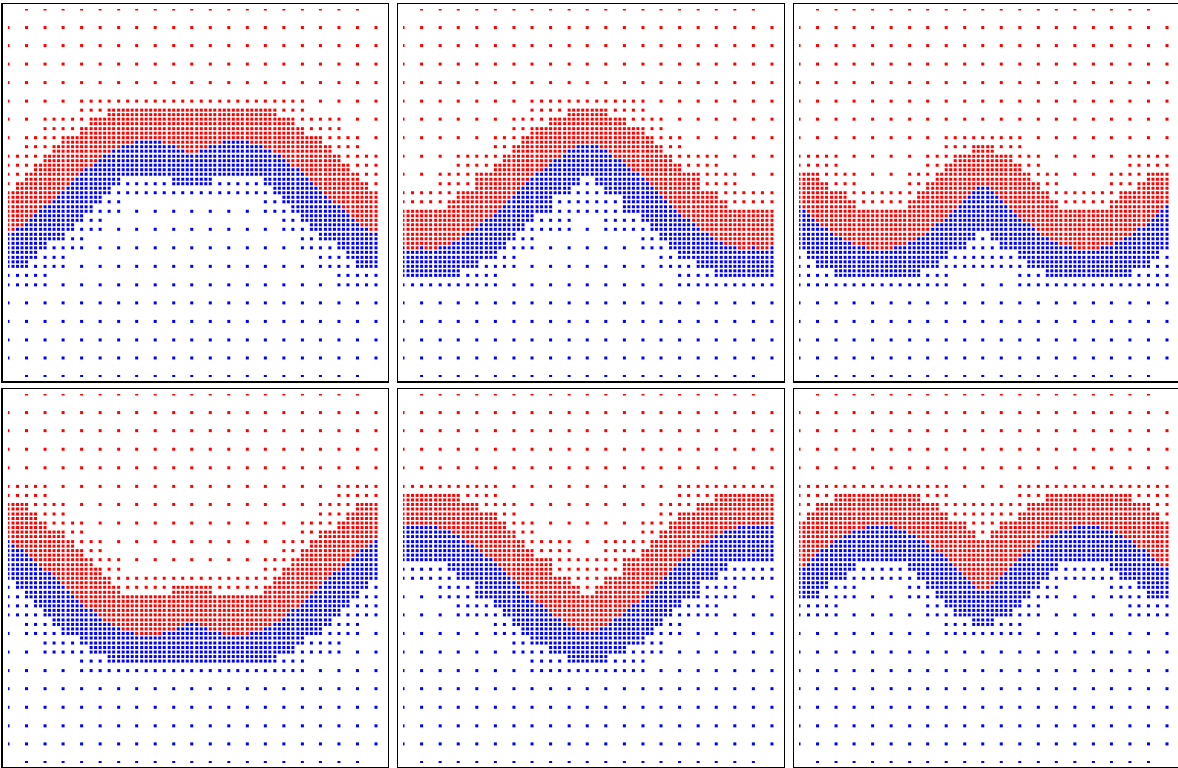


Figure 19 - Adaptive data center insertion for a moving wave

A liquid-gas interface is remodeled at every time step to capture the behavior of a moving wave in figure 19. The clustering of the points near the interface assure accurate modeling for high resolution computations.

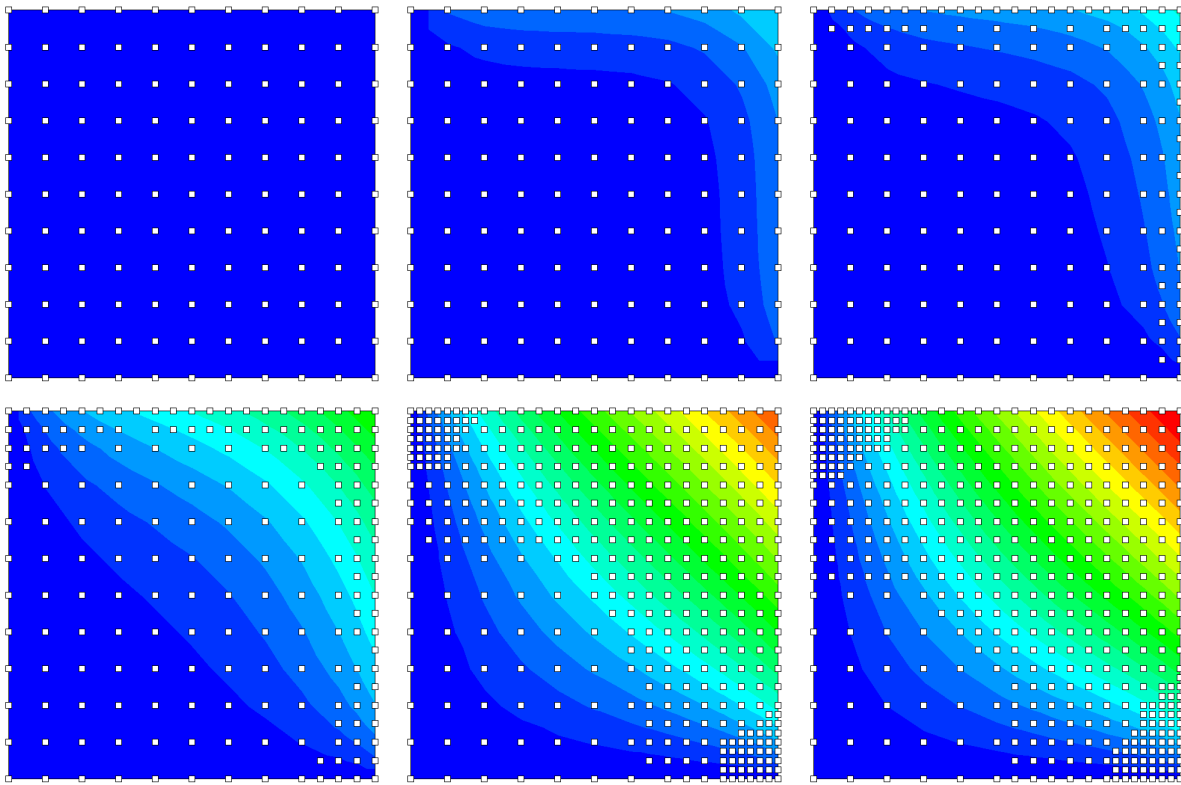


Figure 20 - Adaptive data center insertion for heat conduction

Another example involving the adaptive data center insertion capabilities is that of reacting to a field gradient to cluster in areas of high activity. Figure 20 shows the initial point distribution followed by later time steps. Even early on during the solution the adaptive data center insertion algorithm was called upon to examine the field for large gradients for modification to the current nodal point distribution. This was effectively carried out and maintained an appropriate point distribution for maximum accuracy while still minimizing solution time. Also, due to the autonomous nature of the algorithm, absolutely no user interaction was required to make modifications to the data center distribution during the solution process, only requiring a few governing parameters to control the behavior and limits.

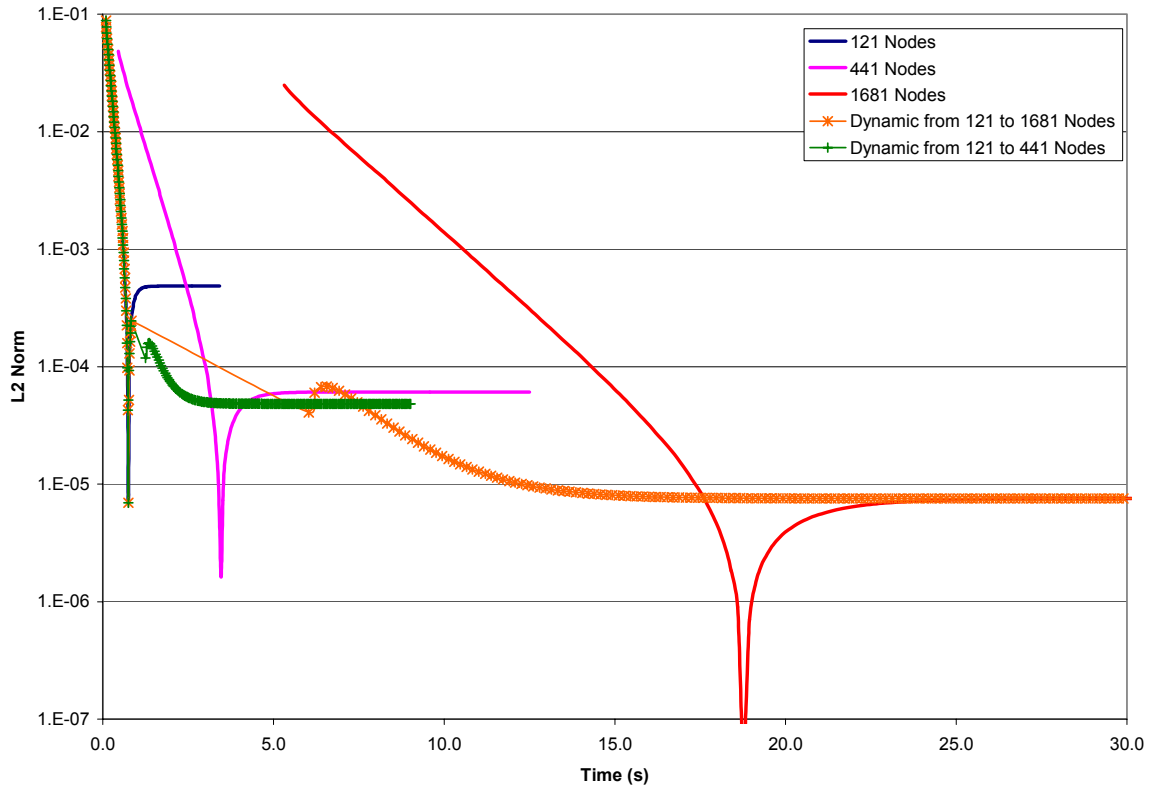


Figure 21 - Solution times for dynamic nodal point generation

Looking at the solution times required for the different geometries, the time was nearly cut in half while still maintaining good accuracy.

7.2 Local expansion

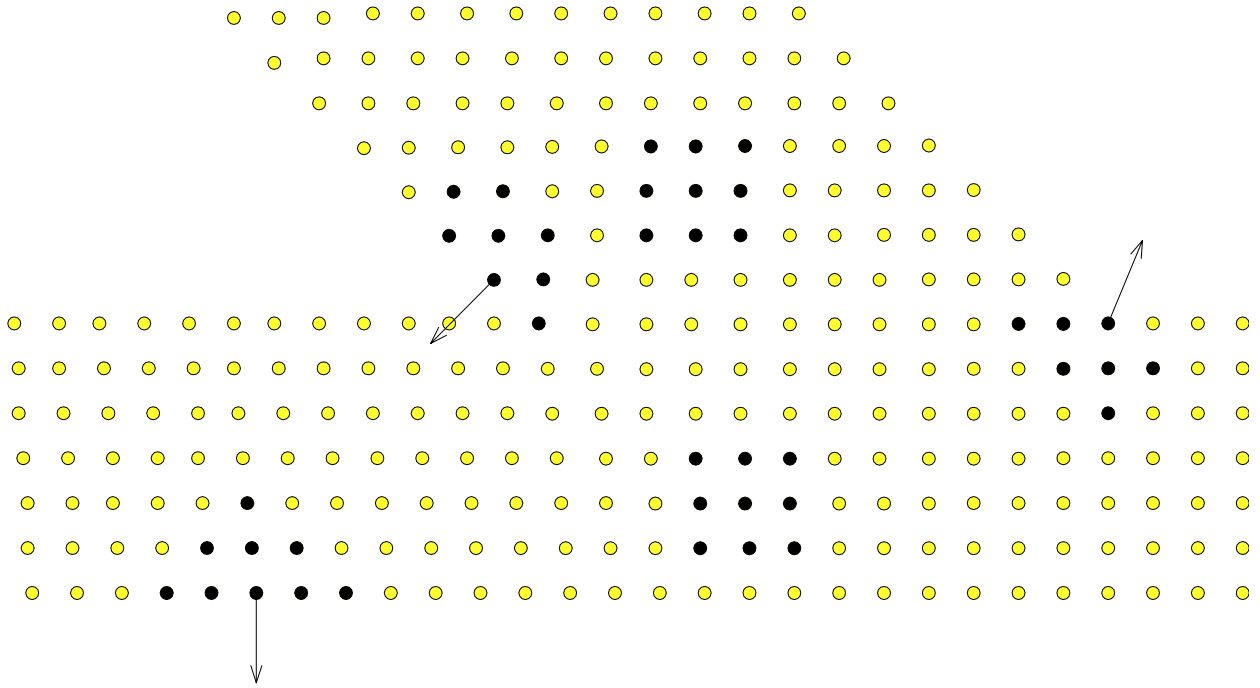


Figure 22 - Collocation topology for internal, boundary, and corner data centers

The re-entrant region of an artery with a bypass graft was studied for the sake of verifying the ability of the local expansion to properly select nodal neighbors and exclude nodes that lie on or within an opposing boundary, such as near the re-entrant region.

7.3 Auto segmentation

Below is an example of the resulting quadtree point distribution with automated auto-segmentation produced for the artery bypass graft problem as generated by the adaptive genetic algorithm described in chapter 5. Each region to be assigned to a specific processor is color-coded, here we illustrate the use of 5 equally capable processors in a parallel computation. Clearly, the objectives of proportional node distribution based on individual processor performance and minimization of nodal data communication are achieved.

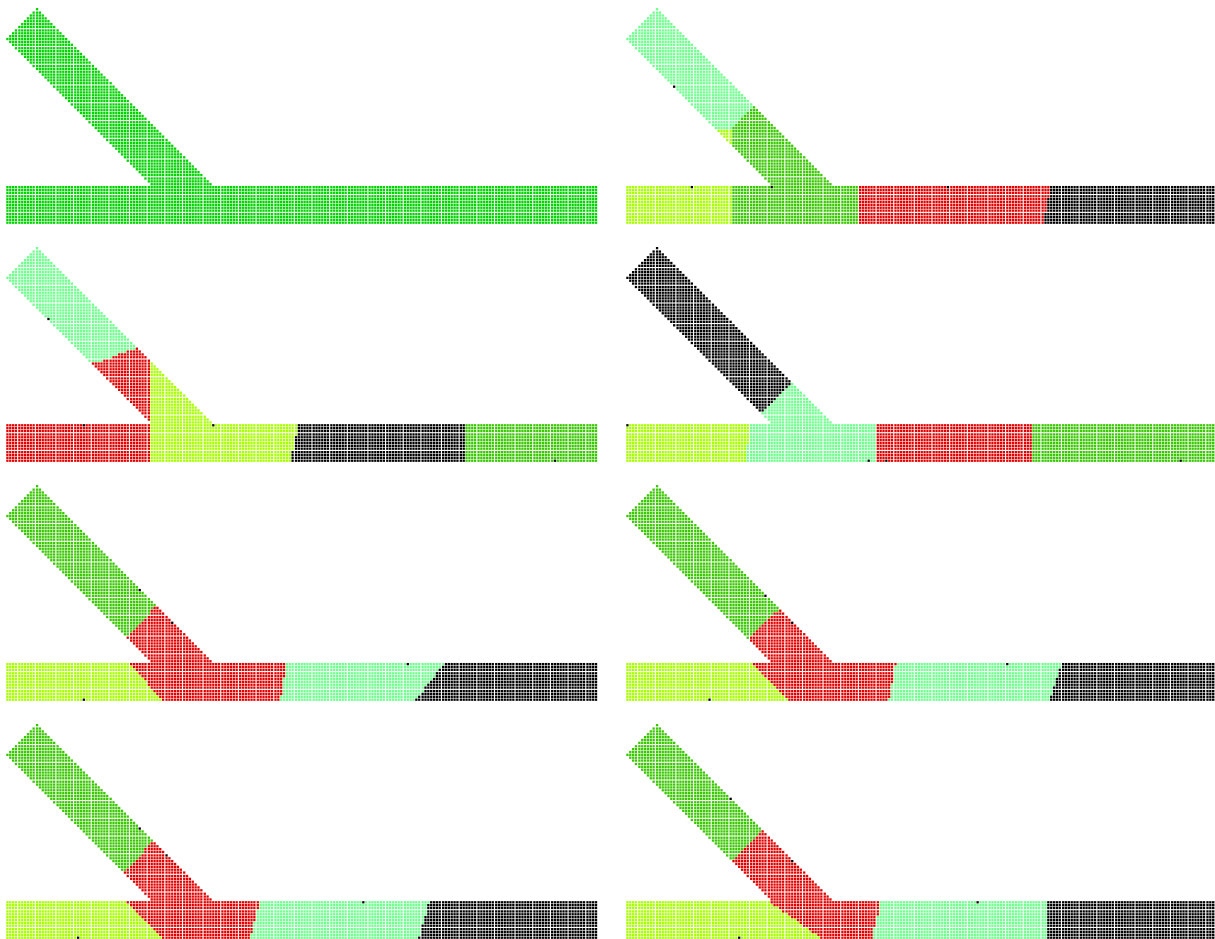


Figure 23 - Auto-segmentation evolution for a bypass graft

Although trivial in nature, the square in figure 24 effectively demonstrated the ability of the automated segmentation algorithm to choose between using two processors and three processors. Ultimately, the additional communication cost of adding the third processor was outweighed by the reduced computational time.

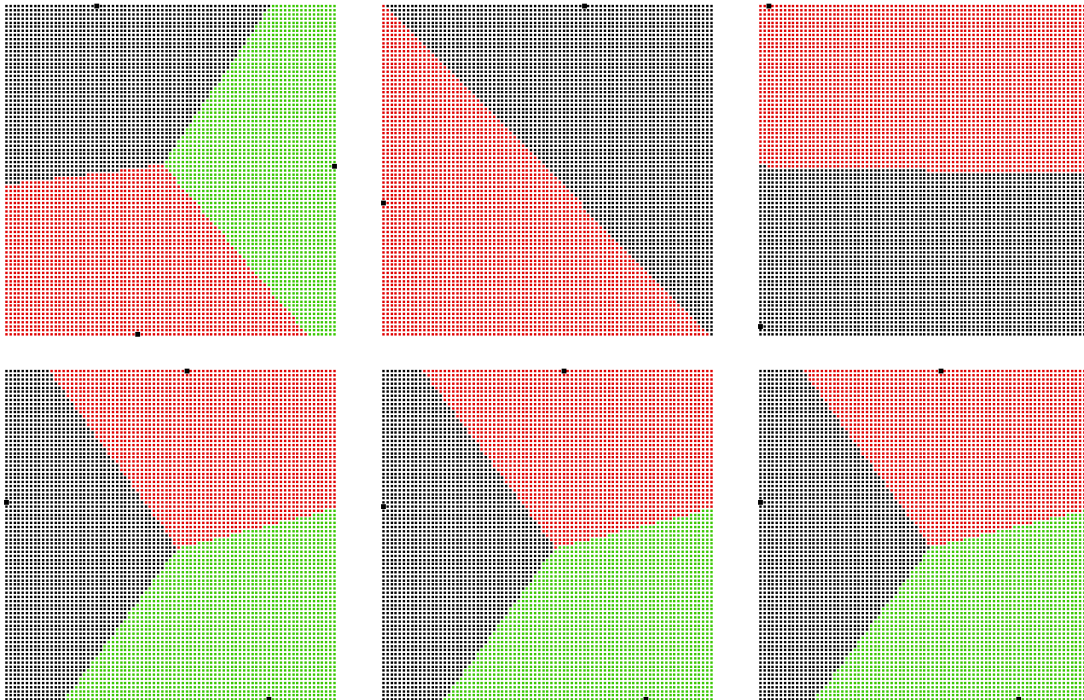


Figure 24 - Auto-segmentation evolution for a square

All of the automated segmentation examples up to this point did not involve clustering of any kind. Figure 25 is a model of flow over a cylinder with clustering downstream from the cylinder to the exit. Having allowed the model to be spread across seven or more processors, the auto segmentation found that using more than five processors would introduce unnecessary communication effort, and thus, reduced the number of processors to only five.

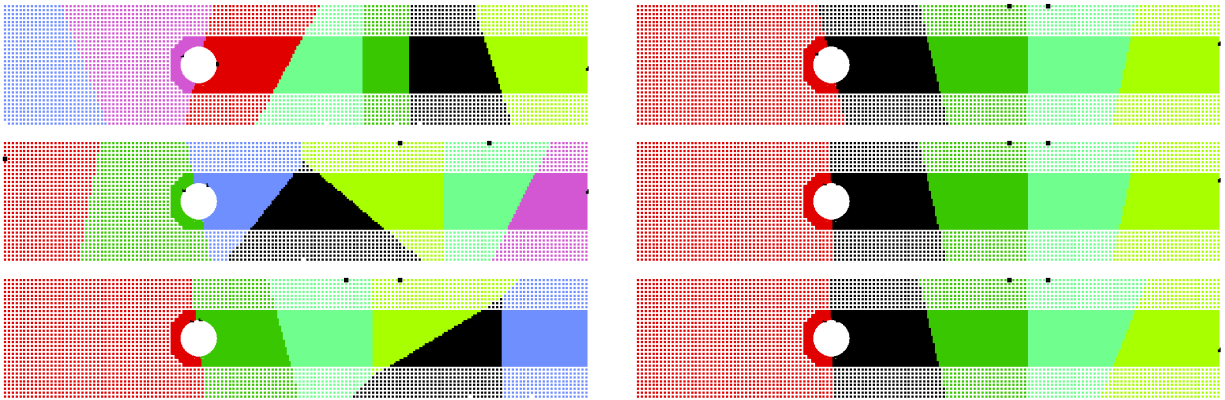


Figure 25 - Auto-segmentation evolution for flow over a cylinder

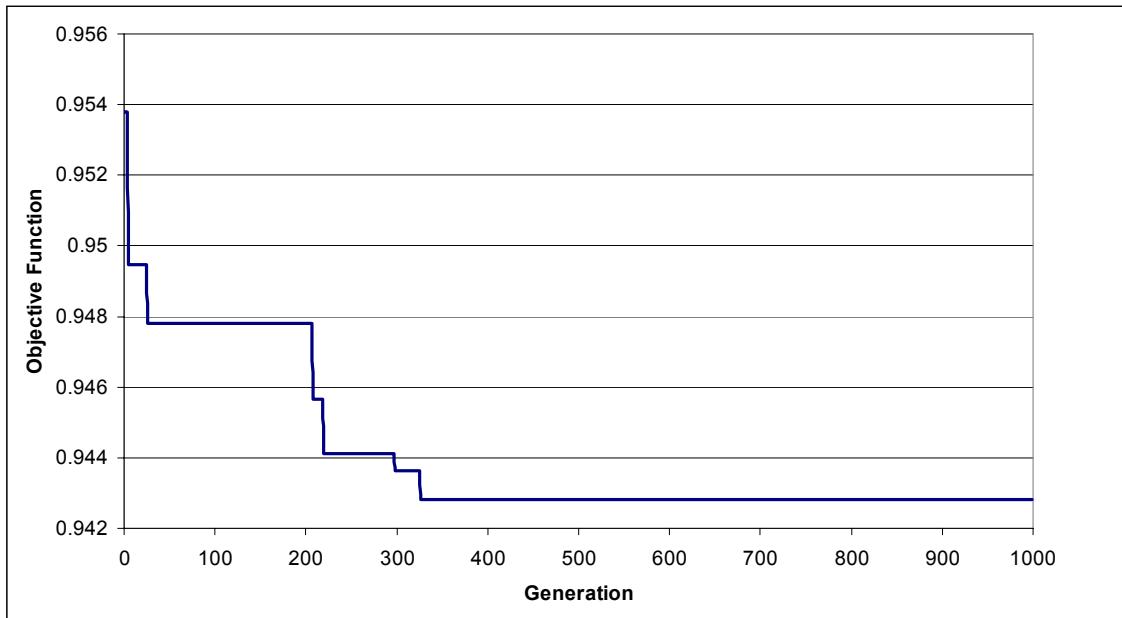


Figure 26 - Evolution of the objective function of the genetic algorithm

CHAPTER 8

CONCLUSIONS

In this paper, a recursive quadtree scheme was developed for the point distribution of a localized meshless collocation method in order to maintain a locally cartesian grid at each of the regions of influence or topologies. The quadtree method allows modeling of irregular geometries and refinement of regions of interest and it lends itself for full automation, thus, reducing problem setup efforts. Furthermore, the construction of the localized expansion regions is closely tied up to the point distribution process and, hence, incorporated into the automated sequence. This also allows for the optimization of the RBF free parameter on a local basis to achieve a desired level of accuracy in the expansion. In addition, an optimized auto-segmentation process is adopted to distribute and balance the problem loads throughout a parallel computational environment while minimizing communication requirements.

REFERENCES

1. Fletcher, C.A.J., Computational Techniques for fluid Dynamics, Vol. 1 and 2, Springer Verlag, 1991.
2. Tannehill, J.C., Anderson, D.A., and Pletcher, R.H., Computational Fluid Mechanics and Heat Transfer, McGraw Hill Book Co., New York, 1997.
3. Zienkiewicz, O.C. and Taylor, R.L., The Finite Element Method, Vol. 1 and 2, McGraw Hill Book Co., New York, 1989.
4. Brebbia, C.A., Telles, J.C.F. and Wrobel, L.C., Boundary Element Techniques, Springer-Verlag, Berlin, 1984.
5. Wrobel, L.C. and Aliabadi, M.A., The Boundary Element Method, Vol. 1 & 2, Wiley, New York, 2002.
6. Divo, E.A. and Kassab, A.J., Boundary Element Method for Heat Conduction with Applications in Nonhomogeneous Media, Wessex Institute of Technology Press, Southampton, UK, and Boston, USA, 2003.
7. Divo, E., Kassab, A.J., Mitteff, E., and Quintana, L. "A Parallel Domain Decomposition Technique for Meshless Methods Applications to Large-Scale Heat Transfer Problems," ASME Paper: HT-FED2004-56004.
8. Soni, B., Thompson, J., & Weathermill, N., Handbook of GRID GENERATION. CRC Press, 1999.
9. Cheng, A.H.-D., Golberg, M.A., Kansa, E.J., Zammito, G., "Exponential Convergence and H-c Multiquadric Collocation Method for Partial Differential Equations," Numerical Methods in Partial Differential Equations, Vol. 19, No. 5, pp. 571-594, 2003.
10. Hardy, R.L., "Theory and Applications of the multiquadric biharmonic method: 20 years of discovery," Computers and Mathematics with Applications, Vol. 19, No. 8/9, 1990, pp. 163-208.
11. Dirichlet, G.L., Uber die Reduction der positiven Quadratischen formen mit drei Underestimmten Ganzen Zahlan, Z. ReineAngew. Mathematics, Vol. 40, No 3, pp. 209-227, 1850.

12. Kansa, E.J., "Multiquadrics- a scattered data approximation scheme with applications to computational fluid dynamics I- surface approximations and partial derivative estimates," *Comp. Math. Appl.*, Vol. 19, 1990, pp. 127-145.
13. Divo, E. and Kassab, A.J., "Effective Domain Decomposition Meshless Formulation of Fully-Viscous Incompressible Fluid Flows," *Boundary Elements XVII*, Kassab, A., Brebbia, C.A. and Divo, E. (eds.), WIT Press, 2005, pp. 67-77.
14. Divo, E. and Kassab, A.J., "A Meshless Method for Conjugate Heat Transfer," *Proceedings of ECCOMAS Coupled Problems 2005*, M. Papadrakakis, E. Oñate and B. Schrefler (Eds.), Santorini, Greece, 2005.
15. Divo, E. and Kassab, A.J., "A Meshless Method for Conjugate Heat Transfer Problems," *Engineering Analysis*, Vol. 29, No. 2, 2005, pp. 136-149.