

Hp-Meshless Cloud Method

T.J. Liszka

C.A.M. Duarte

W.W. Tworzydło

Mar 20, 1996

1 Introduction

The idea of the interpolation of randomly scattered data has been investigated for a long time. The first use of such interpolation was in topography, and in the interpolation of various experimental data. It has been recognized that such kinds of interpolation could be used in numerical methods; for some long time, however, the only successful method was based on the finite element (FE) approach: subdivide the domain into a large number of simple subdomains with data points the vertices of these subdomains, and then define interpolation independently within each subdomain. Two main drawbacks of this approach are the limited continuity of the approximation on the internal boundaries, and the difficult process of generation of the mesh of consistent subdomains.

It is becoming increasingly apparent that, as analysts attempt to solve larger and more difficult problems, several barriers exist to advances in computational simulation methods. The main problem hampering the desired analysis is the need to discretize the domain to be simulated using a computational mesh. Topologically complex domains can be discretized automatically with tetrahedral elements in many cases; however, these elements typically provide low accuracy and are difficult to refine automatically. Moreover, changes in the domain geometry and/or topology, e.g. in shape optimization, free surface flows, fracture analysis, and body fragmentation, are difficult, if not impossible, to accommodate with existing meshing techniques.

Over the past two decades, various researchers have come to realize that so-called *meshless* methods can be developed that remove the problems of traditional meshing. These methods still require a set of computational nodes to be 'sprinkled' through the domain, but *do not* require any pre-specified connectivity of the nodes, or locally regular topological structure as is required of traditional meshing. These nodes can be generated automatically using a variety of schemes including oct-trees, tetra-trees, regular topological kernels,

*The Computational Mechanics Company (COMCO), Austin, Texas

etc. The meshless analysis method removes the disadvantages of standard approaches, and leads to the possibility of truly automatic analysis. Fully automated analysis frees the analyst/engineer to focus on engineering issues, and not on time-consuming computational considerations. Finally, meshless analysis is uniquely suited to coupling with other codes and/or disciplines and thus enables truly multi-disciplinary analysis.

First attempts to formulate approximation methods for the solution of partial differential equations on “randomly” scattered nodal points (not yet recognized as ‘meshless’ methods) were published in early seventies [10,17,20,21]. The basic idea of the method was finally interpreted as an approximation method, and published in two similar papers [14,16]. Although several possible applications of the method in the numerical solution of partial differential equations have been presented at that time [11,17,18,20,21,23,25], the method did not gain any significance, mainly due to overwhelming success of the FE method.

Recently, the meshless technology has been rediscovered due to the slow progress in domain meshing, and several ‘meshless’ methods of numerical PDE solution have been presented [1,3,4,8,9,19]. Most of these methods are based on the Moving Least Square (MLS) interpolation method, presented in details in [14]. The main emphasis of that paper was on interpolation, and application for the solution of PDE’s requires further analysis which will be outlined here.

In this work, we present some new results obtained on the structure of the MLS, and of the method presented in [18], which is similar in concept, but differs mainly in the choice of the weighting functions, and in the selection of the neighboring data points. These two differences may have a significant effect on the final results of the method, especially when applied to the numerical analysis of PDE’s. The method is implemented for the solution of a wide class of engineering problems, with special emphasis on structural mechanics problems.

Section 2 presents the basic equations of the approximation method, emphasizing the differences between MLS and IMLS (interpolating MLS), and its application to the solution of PDE’s. Two possible discretization methods are outlined, equivalent to the classical collocation method, and a Galerkin type formulation. The Galerkin formulation is very similar to the FE method, and is relatively simple to implement. Most of the knowledge from FEM, including theoretical proofs, can be extended or modified to apply to this version. However, the cost of a Galerkin formulation is, at the present time, significantly higher than FEM. This paper focuses on the collocation method, as it provides a much more efficient algorithm. Several cases have shown the method to be on the order of 10 times faster than FEM. Details of the formulation relevant to the discretization of the boundary conditions, especially of the Neumann type are presented, together with numerical tests for convergence of analysis for simple structural problems. The presentation is limited to two-dimensional cases, but extension to 3 and more dimensions are straightforward.

The next section follows with an original generalization of the MLS. By introduction of spectral degrees of freedom, equivalent to Hermite interpolation, the quality of the approximation at and near the boundary is significantly improved. An outline of the formulation,

optimization of various parameters, and convergence tests are presented.

Several practical applications of the method are presented in section 5. They include nonlinear elasticity, geometrically and physically nonlinear membranes, and a transient heat conduction problem. These examples clearly show the versatility and robustness of the method in engineering analysis. Even at this early stage of development, the method competes well with finite element technique and may surpass them in performance in various applications.

2 Basic formulation of the meshless method

2.1 Moving least squares approximation

The basic idea of the MLS approximation method is very simple. Assume that we have function $g = g(x)$ defined at some finite set of points, and we are looking for the approximate value of g at point X_0 ($f(X_0) \simeq g(X_0)$). One may use all available data points, but for the purpose of efficiency, it is reasonable to select some small subset of data points (X_i), (called the *stencil* or *star*), preferably in the neighborhood of X . We assume that the function f is smooth enough to be approximated in some well defined functional basis (local expansion around point X_0):

$$f(X) = \sum_{i=0}^{n_p} (a_i * p_i(X - X_0)) \tag{2.1}$$

Usually $p_i(x)$, $x = X - X_0$; is a set of simple polygons, and the eq. (2.1) can be interpreted as a Taylor expansion of the unknown function $f(X)$ around the point $X = X_0$:

$$p_i(x) = [1, x, x^2/2, x^3/6, \dots]$$

in the one dimensional case, or

$$p_i(x, y) = [1, x, y, x^2/2, xy, y^2/2, x^3/6, \dots]$$

in two dimensions.

With this choice, the coefficients a_i (which are in general functions of the point X_0 , $a_i = a_i(X_0)$), can be understood as the approximation of the unknown function and its derivatives at the point X_0 .

It will be useful to define two distinctly different cases, depending on the existence of data point at X_0 :

2.1.1 Method A: nodal approximation

This method will be suitable only if there is a data point at X_0 and we know the value of the function $f(X_0)$. In this case

$$a_0 = f(X_0), \quad (2.2)$$

and we need to find the values of $a_i, i = 1, \dots, n_p$.

We want, if possible, for the approximation to pass through the remaining data points, i.e. to satisfy

$$f(X_j) = \sum_{i=0}^n (a_i * p_j(X_j - X_0)) = y_j, \quad j = 1, \dots, n \quad (2.3)$$

where y_j denotes value of data point.

In order to solve the set of equations (2.3), we need the number of data points (n) to be no less than the number of terms (n_p) in the expansion (2.1).

$$n \geq n_p.$$

We also need the positions of the data points to satisfy certain geometrical criteria [11, 21, 25], so the linear set (2.3) is not singular. Instead of trying to define and satisfy these criteria, one selects more data points than equations, thus changing (2.3) into an overdetermined set of linear equations, and consequently solves the system using a traditional minimum least squares method, weighting each equation with an appropriately chosen weight w_i . The weights are defined using the weight function, which may, in general, depend on the location of the node $W = W(X, X_j)$, and it is sometimes advantageous to favor data points in one direction (e.g. along the boundary layer). We will limit the discussion to simple, isotropic weight function of the distance from the central point:

$$w_i = W(\rho_i), \quad \rho = |X - X_j|$$

At minimum, the weight function should be non-increasing with its argument:

$$W(x) \geq W(y) \iff x < y,$$

so the data points farther from the center have smaller influence on the approximation. Note again, that the eq. (2.2) is satisfied exactly in this approach, so that the approximate value of the function is exactly equal to the known data value at point X_0 . The only new information obtained from the method is related to derivatives of f at point X_0 .

For a two-dimensional domain, and the choice of the basis functions outlined above, the final set of linear equations to solve is as follows:

$$[\mathbf{A}]\{\mathbf{Df}\} - \{\mathbf{F}\} = \{0\}, \quad (2.4)$$

$$\begin{aligned}
[\mathbf{A}] &= \begin{bmatrix} h_1 & k_1 & h_1^2/2 & k_1^2/2 & h_1k_1 \\ h_2 & \cdots & \cdots & \cdots & \cdots \\ & \vdots & & & \\ h_m & & & & \end{bmatrix}, \\
\{\mathbf{F}\}^T &= \{f_1 - f_0, f_2 - f_0, \cdots, f_m - f_0\},
\end{aligned} \tag{2.5}$$

where

$$\begin{aligned}
f &= f(x, y), & f_0 &= f(x_0, y_0), \\
h_i &= x_i - x_0, & k_i &= y_i - y_0, & \rho &= \sqrt{h_i^2 + k_i^2}.
\end{aligned}$$

and the five unknown derivatives at the point (x_0, y_0) are

$$\{\mathbf{Df}\}^T = \left\{ \frac{\partial f_0}{\partial x}, \frac{\partial f_0}{\partial y}, \frac{\partial^2 f_0}{\partial x^2}, \frac{\partial^2 f_0}{\partial y^2}, \frac{\partial^2 f_0}{\partial x \partial y} \right\}. \tag{2.6}$$

This version of the MLS is sufficient to implement the analog of the Finite Difference Method (FDM) on an irregular grid of nodes [16, 18], more precisely it allows you to solve an arbitrary PDE using the nodal collocation method. The main drawback of this approach is its discreteness: the approximation is defined only at the data points, and nothing is known about the solution between these points (this matches the traditional understanding of the FDM).

2.1.2 Method B: pointwise approximation

One can easily modify the above procedure to define a dense approximation, i.e. for any point X , whether coinciding with a data point or located arbitrarily in the domain. All that is necessary is removing special equation (2.2) from the final set of equations, and treat uniformly equations for all data points, even if one of them coincides with the point X . The set of unknown consists of all coefficients $a_i, i = 0, 1, \dots, n_p$.

Considering again the two-dimensional example, we have to solve the following set of equations (similar to (2.5) above):

$$\begin{aligned}
[\mathbf{A}]\{\mathbf{Df}\} - \{\mathbf{F}\} &= \{0\}, \\
[\mathbf{A}] &= \begin{bmatrix} 1 & h_1 & k_1 & h_1^2/2 & k_1^2/2 & h_1k_1 \\ 1 & h_2 & \cdots & \cdots & \cdots & \cdots \\ & \vdots & & & & \\ 1 & h_m & & & & \end{bmatrix}, \\
\{\mathbf{F}\}^T &= \{f_1, f_2, \cdots, f_m\},
\end{aligned} \tag{2.7}$$

and the six unknowns are

$$\{\mathbf{Df}\}^T = \left\{ f_0, \frac{\partial f_0}{\partial x}, \frac{\partial f_0}{\partial y}, \frac{\partial^2 f_0}{\partial x^2}, \frac{\partial^2 f_0}{\partial y^2}, \frac{\partial^2 f_0}{\partial x \partial y} \right\}. \quad (2.8)$$

This approach defines an approximate value of the function $f_0 = f(X)$, and its derivatives, at any point in the domain, but, in general, *the value of the approximation does not match the data value of any data point*. Then, we have an approximation method but without the interpolation property. This method has been, with reasonable success, applied to the solution of various PDE's using a Galerkin type of approach [4, 19], but the lack of the interpolation property caused significant difficulties in satisfying the essential boundary conditions [4, 12].

2.1.3 Continuity of the approximation.

It has been shown [14] that the resulting approximation resulting from method A or B is continuous under the following conditions:

- i) The basis functions $p_i(\cdot) \in C^m$.
- ii) The weighting function $W(\rho) \in C^l$.
- iii) All data points from the whole domain are included in the stencil.

Then the resulting approximation $f(X) \in C^{\min(m,l)}$.

In the MLS formulation it is often assumed that the weight function has a support limited to a finite ball of radius r :

$$W(\rho) = 0 \iff \rho > r$$

In that case the above condition iii) may be modified as follows:

iii') All data point from the local support of the weight function are included in eq. (2.8). This condition implies, that, if the weight function is isotropic (depends only on the distance ρ), then all data points included in the local ball surrounding point X_0 (see [9]) have to be included in the approximation.

As shown in [14], a discontinuity of the weight function at the center of the approximation (at point X_0) of type $1/\rho^{2\nu}$, does not cause the discontinuity. This type of the weight function is very useful, because it defines an approximation with the interpolation property (we can see that the method B converges to method A in the limit as X_0 converges to the data point).

Another possible discontinuity exists in the formulation presented in [17, 18] due to not satisfying condition iii') above. The weight function has infinite support but the stencils include only a small, fixed number of data points. It has to be noted also, that due to a

change in the local configuration of the data points, in particular, due to a change of their local density, the weight function depends in general on the position of the central point

$$W(\rho) = W(\rho, X_0).$$

For example: in [7] the weight is defined as the continuous function:

$$W_R(\rho) = \begin{cases} \sqrt{4/\pi} (1.0 - (\rho/R)^2)^4, & \forall \rho < R \\ 0, & \forall \rho \geq R \end{cases}$$

but the radius of the ‘cloud’ R is adjusted pointwise to contain enough data points to avoid a singularity in solving 2.8. The weight function $W(\rho, X_0)$ should be defined as continuous with respect to both its arguments, which in this case is not.

2.2 Practical aspects of the method for the solution of PDE’s

2.2.1 Modifications for computational efficiency

It has been observed that the method as outlined above is relatively slow when applied to the solution of PDE’s by the generalized Galerkin formulation [4,9]. This is mainly due to the inability of developers to formulate efficient numerical quadrature rules. The collocation method (as presented in [16]) is generally much faster, but requires a more elaborate selection of stencils.

In both versions, significant optimization of CPU time can be obtained by use of very small, carefully selected stencils (the requirements and possible selection algorithms are discussed below). For an irregular grid with a varying density of data points, the ‘optimal’ stencils usually violate the continuity requirements. In order to use all of the theoretical results presented in [1,9,15] we have to reinterpret the continuity requirements, and fit them to this practical algorithm.

The ‘reinterpretation’ will be based on the observation, that in both methods, one uses only a finite set of points, where the approximation is computed (eg., Gauss points in the Galerkin method). It is therefore always possible to define what happens between these points in such a way, that the continuity requirements are satisfied. In particular, one should define the weight function so it will smoothly go to zero in the shaded area in fig. 2.1. These additional definitions will not have any impact on the computer code, but they will serve to prove accuracy and convergence of the algorithm.

2.2.2 ‘Optimal’ selection of stars

A proper approximation of the solution requires that support for each data point (circular or spherical clouds as in [7]) should cover all the domain, so that the solution at any point in the domain is properly defined. Actually, they have to sufficiently overlap (Fig. 2.2), and

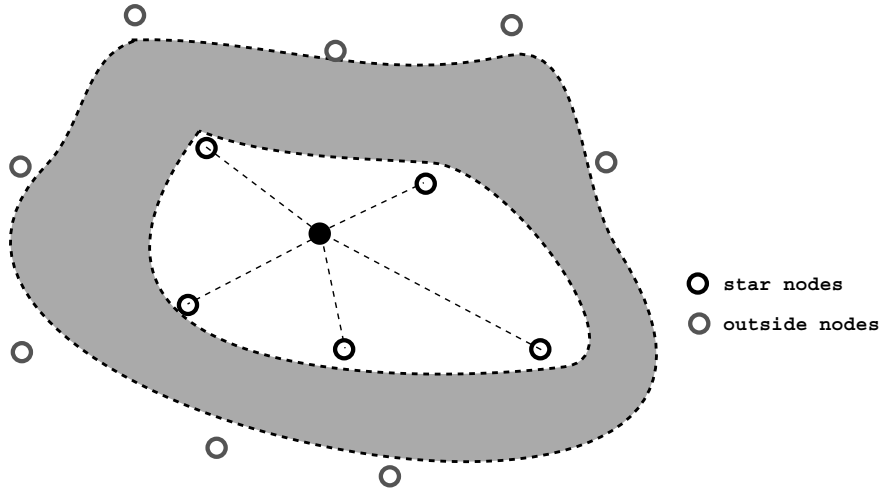


Figure 2.1: Transition zone for the weight function

the overlap areas have to be accounted for in the choice of collocation points or integration quadrature. Whenever there is not enough integration/collocation points in the overlap zone (shaded area in Fig. 2.2), the solution will not converge. In the case of the collocation method it is intuitively obvious that left and right halves of the domain will not be “tied together” and the solution will exhibit a spurious discontinuity along the shaded area. In extreme cases it may even result in a singularity in the global system of linear equations.

This difficulty has been recognized from the very beginning, and various algorithms to ensure correct stars have been introduced ([10,21]). The idea of “sufficient overlap” is not easily quantifiable, section 3.4 presents some analysis of the problem. In the present work we use a relatively simple and effective star selection algorithm presented by Liszka [16], which selects the star according to the “four quadrants” criterium, (Fig. 2.3). This method can easily be modified to “eight sectors” in three dimensional analysis. Near the boundary, one can also modify the method to account for unavoidable unsymmetry of the stars, by selecting an appropriate number of nodes at the boundary, and then selecting the remaining nodes from sectors extending into the computational domain (Fig. 2.4).

2.2.3 Anisotropic approximation

In most applications, the accuracy of the approximation can be improved by increasing the size of stencils (i.e., the number of data points and the number of base functions) whenever the solution is sufficiently smooth. This amounts to p -adaptivity, which is much simpler in the meshless method than in the Finite Element Method. For long, thin domains, it is possible to increase the order of approximation along the longer dimension of the domain, while along the shorter dimension there may not be enough data points to obtain desired approximation order. It is important, therefore, to define an algorithm, which will allow one to obtain different orders of approximation in different directions (and for these directions

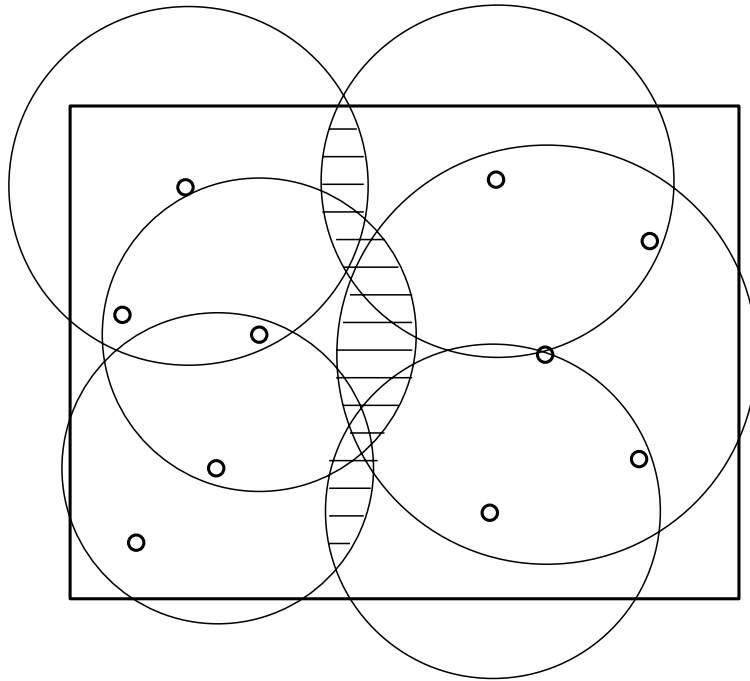


Figure 2.2: Overlap zone between two parts of the domain

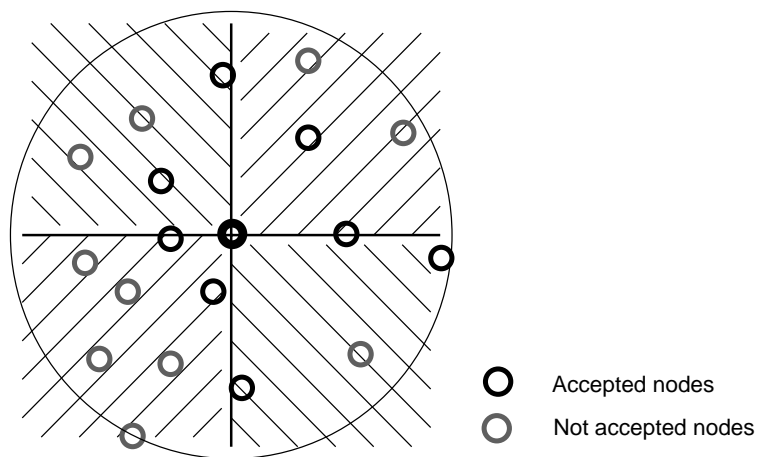


Figure 2.3: Selection of a star by four quadrants criterion.

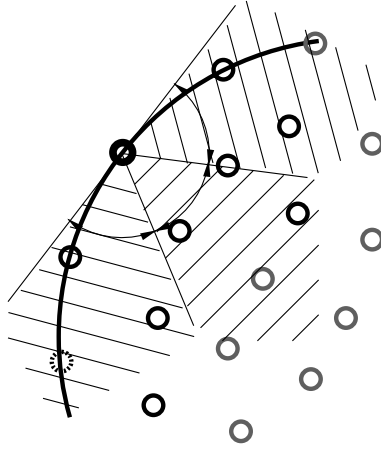


Figure 2.4: Selection of a star on the boundary

to be arbitrary - non-aligned with the global coordinate system).

One possible solution for an anisotropic approximation is presented in [9], where local approximations with varying orders in different directions are used. This formulation requires that the orientation of the local coordinate systems be defined a priori, while in practical application it would be advantageous to be able to select nodes based on local configurations and then compute the best possible approximation, even if the orientation of the domain is not explicitly known. For this purpose the computer code used to (pseudo-)invert the matrix $[\mathbf{A}]$ in eq. 2.4, should use the Singular Value Decomposition (SVD) technique to automatically detect ill-conditioned components of the approximation [22]. Of course, for efficiency, SVD should be used only when the standard algorithm indicates that the star is not sufficient for the requested order of approximation.

2.2.4 Implementation of Neumann boundary conditions

The implementation of boundary conditions in the collocation version of the MLS method is considerably different than in the Galerkin approach and requires a different methodology for efficient and accurate handling.

The application of *Dirichlet* boundary conditions, such as prescribed displacements, is quite straightforward. It amounts to specifying the value of the function on the boundary and adding the relevant equation to the system. No special treatment, such as a penalty method usually associated with Galerkin methods, is needed.

However, for *Neumann* boundary conditions, such as prescribed tractions, some special attention is required. Formally, on each boundary a relevant equation needs to be explicitly discretized. For example, to prescribe a typical traction boundary condition one simply discretizes the equation:

$$\sigma_{ij}n_j = \bar{t}_i \quad (2.9)$$

where the components of the stress tensor σ_{ij} should be expressed as relevant derivatives of displacements, n_j are the components of a normal vector, and \bar{t}_i is a prescribed traction vector. Clearly, implementation of this boundary condition requires:

- components of the traction vector $\bar{\mathbf{t}}$,
- components of the normal vector \mathbf{n} , and
- selection of an appropriate boundary “stencil” for discretization of the derivatives.

The above condition needs to be explicitly implemented even on free boundaries where the prescribed traction vector is zero. This is quite different than in the Galerkin method, where free boundaries require no calculation of boundary integrals.

Direct implementation of the above boundary condition, although it is formally straightforward, creates some effects that may reduce the accuracy of the collocation method in its’ most straightforward form. This is because:

1. Discretization of the boundary condition at the boundary node precludes discretization of the principal governing equation at this node (one equation is generated per one degree of freedom). Thus, only the internal nodes support the governing equations (Navier equations in the case of linear elasticity).
2. The boundary stencils tend to be of strongly unsymmetric shapes, with the centers of “mass” shifted towards the inside of the domain.

The above phenomena may cause loss of accuracy and sensitivity of the results to the selection of stars on the boundary. As an introductory illustration of this phenomenon, we present a numerical solution to the classical problem of a plate with a hole under a uniform tensile stress.

In the first example, the problem is solved on a rather coarse grid with quite irregular distribution of points near the hole. Figure 2.5 shows the mesh used for the upper half of the domain. The MLS approximation was of second order, with 9 points selected for each star (both interior and boundary stars). On this mesh, the stars tend to be rather unsymmetric shapes, especially near the hole. As a result, the solution on this mesh is of low quality – in particular, it does not preserve the expected symmetry w.r.t. the vertical y-axis. This solution is shown in figure 2.6. A more careful examination of the results shows that this effect is caused primarily by irregular and nonsymmetric shapes of stars used to discretize the Neumann boundary condition.

The accuracy of the solution can be improved by increasing the number of points in a star, which improves their conditioning and reduces sensitivity to star unsymmetry. Indeed, after increasing this number to 13, the accuracy of the solution improves dramatically (see figure 2.7) and retains the expected symmetry w.r.t. the vertical axis, even though the

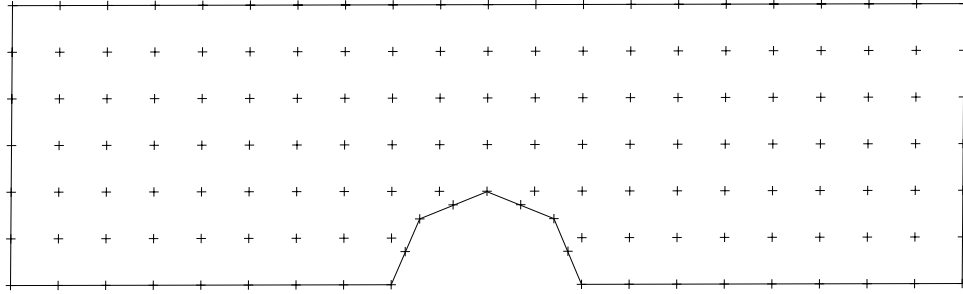


Figure 2.5: An irregular grid of points for the plate with a hole problem.

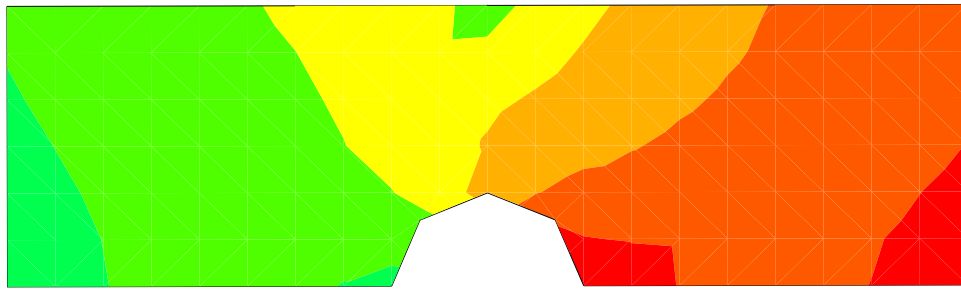


Figure 2.6: Plate with a hole: displacement u_x calculated with 9-noded stars.

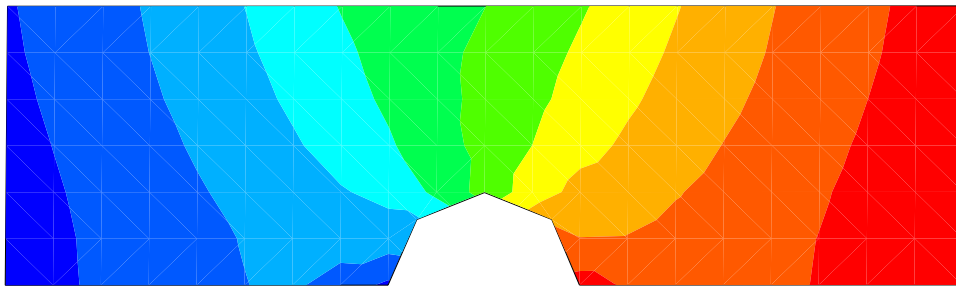


Figure 2.7: Plate with a hole: displacement u_x calculated with 13 per star.

geometries of the stars do not have this symmetry. Importantly, higher numbers of nodes per star maintain this improved quality result.

A different way to increase the order of the approximation, without increasing the size of the stencil, may be used, when there is information about the derivatives of the unknown function at (some) data points. This method will be presented in more details in the following section.

3 Version of the MLS for the spectral DOF's

3.1 MLS Approximations with Derivative Degrees of Freedom

As discussed in previous section one main difficulty in the successful application of the moving least squares approach is the proper selection of stars to avoid singular or ill-conditioned stencils. The addition of more nodes to the stars generally improves the conditioning of the matrices \mathbf{A} but also significantly increases the computation cost to compute the derivatives $\{\mathbf{D}\mathbf{f}\}$. Therefore the number of nodes in the stars should be kept to a minimum.

The control of the conditioning of the stencils become even more delicate for points near or on the boundary of the domain. Also the slenderness of the domain may adversely affect the conditioning of the stencils. Very slender domains generally lead to a loss of accuracy in the computation of the derivatives $\{\mathbf{D}\mathbf{f}\}$ specially for points near or on the boundary.

The ill-conditioning of the stencils for boundary nodes is a consequence of not having ‘the center of gravity’ of the star near the node but biased toward the inside of the domain. The following approach have been developed to improve the conditioning of these stencils.

As previously, each node in the domain represents a degree of freedom corresponding to the value of the solution. In addition, the nodes at the boundary, or at selected part of it, contain additional degree of freedom, corresponding to the derivative of the solution in the direction normal to the boundary. More general approximation can be used (both directional derivatives, second derivatives, curvature,...), but it will not be discussed here.

In addition to Equation (2.5) the following condition is imposed on the MLS approximation built at a boundary node (x, y) with unit normal $\mathbf{n} = (n_x, n_y)$

$$\begin{aligned}
 \frac{\partial f}{\partial \mathbf{n}}(x, y) &= \frac{\partial f}{\partial x}(x, y)n_x + \frac{\partial f}{\partial y}(x, y)n_y \\
 &= \frac{\partial f_0}{\partial x}n_x + \frac{\partial f_0}{\partial y}n_y + h\frac{\partial^2 f_0}{\partial x^2}n_x + \frac{\partial^2 f_0}{\partial x\partial y}(kn_x + hn_y) \\
 &+ k\frac{\partial^2 f_0}{\partial y^2}n_y
 \end{aligned} \tag{3.1}$$

where, as before,

$$\begin{aligned} f &= f(x, y), & f_0 &= f(x_0, y_0), \\ h &= x - x_0, & k &= y - y_0. \end{aligned} \quad (3.2)$$

Evaluating (2.5) for each of the nodes in the star and (3.1) for each of the boundary nodes in the star results in the following system of equations

$$[\mathbf{A}]\{\mathbf{Df}\} = \{\mathbf{F}\} \quad (3.3)$$

where

$$[\mathbf{A}] = \begin{bmatrix} 1 & h_1 & k_1 & h_1^2/2 & h_1 k_1 & k_1^2/2 \\ 1 & h_2 & k_2 & h_2^2/2 & h_2 k_2 & k_2^2/2 \\ \vdots & & & & & \\ 1 & h_m & k_m & h_m^2/2 & h_m k_m & k_m^2/2 \\ 0 & n_{1x} & n_{1y} & h_1 n_{1x} & (k_1 n_{1x} + h_1 n_{1y}) & k_1 n_{1y} \\ 0 & n_{2x} & n_{2y} & h_2 n_{2x} & (k_2 n_{2x} + h_2 n_{2y}) & k_2 n_{2y} \\ \vdots & & & & & \\ 0 & n_{lx} & n_{ly} & h_l n_{lx} & (k_l n_{lx} + h_l n_{ly}) & k_l n_{ly} \end{bmatrix} \quad (3.4)$$

$$\{\mathbf{F}\} = \left\{ f_1, f_2, \dots, f_m, \frac{\partial f_1}{\partial n}, \frac{\partial f_2}{\partial n}, \dots, \frac{\partial f_l}{\partial n} \right\}^T \quad (3.5)$$

$$\{\mathbf{Df}\} = \left\{ f_0, \frac{\partial f_0}{\partial x}, \frac{\partial f_0}{\partial y}, \frac{\partial^2 f_0}{\partial x^2}, \frac{\partial^2 f_0}{\partial x \partial y}, \frac{\partial^2 f_0}{\partial y^2} \right\}^T \quad (3.6)$$

To ensure proper rank of $[\mathbf{A}]$ we require

$$m + l \geq n_p, \quad (\text{here: } n_p = 6)$$

In the above it was assumed that there are m nodes in the star, and that the first l nodes were on the boundary $\partial\Omega$ (or equivalently that there are m nodes in the star with function value degrees of freedom and l nodes with rotational degrees of freedom).

The effect of the l additional equations (one for each node with rotational degree of freedom) is to move “the center of gravity” of the star towards the boundary and therefore to improve the conditioning of the stencils for the boundary nodes. Some numerical experiments with this new approach are reported below.

One important benefit of using the approach described above is on the imposition of Neumann boundary conditions. This approach implies that the boundary nodes will have not only displacement degrees of freedom but also normal derivatives degrees of freedom (cf.

Equation (3.5)). Therefore the enforcement of Neumann boundary conditions can, in many cases, be done in the same straightforward way as standard Dirichlet boundary conditions. In collocation method we follow the algorithm, which is customary in classical FDM, we impose two equations at all boundary nodes: one equation resulting from the boundary condition, and one from the PDE (i.e. the same as in the interior of the domain).

3.2 Numerical Experiments

In this section the conditioning of the stencils for boundary nodes when using Equation (2.4) or Equation (3.3) is analyzed. Similar tests were performed for higher orders of approximation with identical conclusions, but we will present here only the case for second order approximation.

In order to be considered acceptable the computed values of $\{Df\}$ have to satisfy the following criteria:

- (i) The sum of the MLS functions must be one, since they represent a partition of unity [9].
- (ii) The sum of the derivatives of the MLS must be zero for the same reason as in (i).

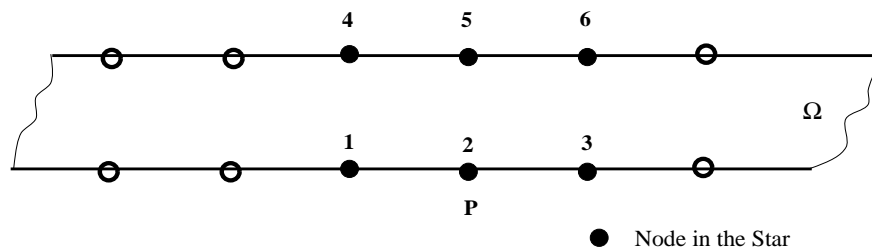


Figure 3.1: Star used to compute a MLS approximation of degree $p = 2$.

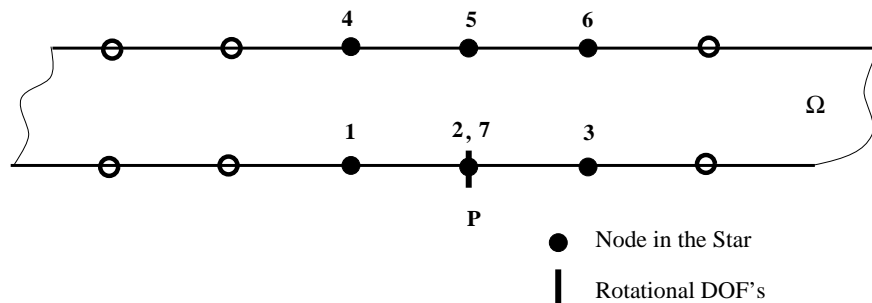


Figure 3.2: Well conditioned star with a rotational degree of freedom at point P .

Suppose that we want to build a moving least squares approximation at point P , of the slender domain represented in Fig. 3.1, using polynomials of degree less than or equal to $p = 2$. Therefore at least $(p+1)(p+2)/2 = 6$ nodes have to be used. Suppose that the 6 nodes

star represented in Fig. 3.1 is chosen. If Equation (2.4) is used to build the moving least squares approximation the corresponding system of equations will be very poorly conditioned and the computed values for $\{\mathbf{D}\mathbf{f}\}$ do not satisfy the above criteria. Table 3.1 contains the generalized finite difference formulas, i.e. MLS functions (only for first m nodes with function value degrees of freedom), computed at point \mathbf{P} using Equation (3.3). The columns Φ_I , $\Phi_{I,x}$, $\Phi_{I,y}$, etc, in the table contain the values and indicated derivatives at point \mathbf{P} of the MLS functions associated with nodes $I = 1, 6$. The last row of the table contains the sum of the MLS functions and their derivatives at point \mathbf{P} . The poor conditioning of the star is reflected by the very large values of some of the computed derivatives (those in the y direction). It can be observed that condition (i) is satisfied but condition (ii) is not. Nonetheless, if just one node with a rotational degree of freedom is added at point \mathbf{P} (node 7 in Fig. 3.2) the stencil becomes well conditioned and the above criteria are satisfied with errors $\mathcal{O}(10^{-13})$ or less. The corresponding star with this rotational degree of freedom is represented in Fig. 3.2. Table 3.2 contains the corresponding generalized finite difference formulas computed at point \mathbf{P} using this star and Equation (3.3). It can be observed from Table 3.2 that the additional MLS function shape associated with the rotational degree of freedom has some of the properties of Hermite shape functions used in C^1 finite elements – the function is zero at the node and its derivative in the normal direction to the boundary is 1.0.

Node I	Φ_I	$\Phi_{I,x}$	$\Phi_{I,y}$	$\Phi_{I,xx}$	$\Phi_{I,xy}$	$\Phi_{I,yy}$
1	1.416E-06	-1.04697	5.754E+15	3.363E-04	2.858	-2.301E+16
2	0.9999	9.537E-02	-1.150E+16	-6.727E-04	-1.716	4.603E+16
3	1.552E-06	0.9523	5.754E+15	3.363E-04	-1.141	-2.301E+16
4	-1.416E-06	4.768E-02	-5.754E+15	3.999	-2.858	2.301E+16
5	2.988E-06	-9.537E-02	1.150E+16	-7.999	1.716	-4.603E+16
6	-1.279E-06	4.768E-02	-5.754E+15	3.999	1.141	2.301E+16
$\sum_{I=1}^6$	0.999999	6.8250E-14	-376.0	1.3322E-15	-1.880E-13	1504.0

Table 3.1: MLS functions (GFD formulas) computed using the star of Fig. 3.1.

Node I	Φ_I	$\Phi_{I,x}$	$\Phi_{I,y}$	$\Phi_{I,xx}$	$\Phi_{I,xy}$	$\Phi_{I,yy}$
1	0.000000	-1.000000	0.000000	3.636363	2.000000	-0.727273
2	1.000000	0.000000	0.000000	-7.272726	0.000000	-6.545453
3	0.000000	1.000000	0.000000	3.636363	-2.000000	-0.727273
4	0.000000	0.000000	0.000000	0.363637	-2.000000	0.727273
5	0.000000	0.000000	0.000000	-0.727274	0.000000	6.545453
6	0.000000	0.000000	0.000000	0.363637	2.000000	0.727273
$\sum_{I=1}^6$	1.000	6.506E-14	-1.337E-19	-2.378E-13	-1.301E-13	4.785E-14
7	0.000000	0.000000	-1.000000	0.000000	0.000000	4.000000

Table 3.2: MLS functions (GFD formulas) computed using the star of Fig. 3.2.

The results of this tests show that it is possible to build accurate (second or third order) approximation using only nodes on the boundary. This could be very useful when modeling three-dimensional slender domains, like plates or shells.

3.3 Selecting weight functions

For the homogeneous approximation (i.e., without the spectral dof's), the choice of the weight function is relatively straightforward. Based on the paper by Lancaster [14], in order to obtain an interpolating version of the approximation, one has to use weight functions with a singularity at the origin (in the center of the star), and an inverse of even power of the distance gives the proper order of the singularity and guarantees full continuity of the approximation. Introduction of spectral dof's requires that the equations related to the standard dof's have different weight functions than those related to spectral dof's, if only to satisfy non-dimensionality of the final results. In addition, as we have found during recent numerical tests, spectral dof's should have their weights lowered to improve the quality of the approximation.

In our tests (see the next section), we have assumed weight functions of the following form:

$$W_i(\rho) = \frac{1}{\rho_i^p}$$

for non-spectral dof's, and

$$W_i(\rho) = \frac{1}{\rho_i^{p-1}c}$$

for the spectral dof's.

The exponent p varied in the range 1 – 8, and correction weight c was tested as one of: 1.0, 10.0, 24.0, 100.0, 10.0×10^4 . The original choice of $p = 6$ ([16]), produced stars with poor stability, and the best tested results were obtained with $p = 3$, $c = 24.0$ although $p = 2$ and $p = 4$ produced usually similiar results..

3.4 Estimate of the star quality

In the collocation method, the stars have to satisfy certain (unknown) geometrical criteria. Intuitively, the requirements are that stars:

- are closely connected to all direct neighbors,
- have a central node nicely surrounded by peripheral nodes,
- have ‘center of the gravity’ close to the central node of the star.

None of these ideas is easily quantifiable, although some of them can be useful in the selection methods (e.g. four quadrants criterion presented in figure 2.3). The paper by Demkowicz [6], contains a proof of the convergence of the method based on the assumption that the coefficients of the ‘meshless shape function’ for the Laplace operator satisfy a ‘discrete maximum principle’:

- coefficient for the center node is negative

$$a_{center} < 0$$

- coefficients for all other nodes in the star are positive

$$a_i > 0, i \neq center$$

- the sum of all coefficients is equal zero.

$$\sum a_i = 0$$

These are by no means necessary conditions; in fact, except for very regular meshes, the stars do not satisfy the maximum principle, and the solution nevertheless converges and does not exhibit local discontinuities. The error in satisfying the maximum principle, defined as:

$$\epsilon = \left| \frac{a_{center} - \sum_{i \neq center} |a_i|}{a_{center}} \right|$$

can, in most cases be up to 0.5 for several stars, and on the order of 10 for some localized stars, with the solution still converging. On the other hand, whenever the solution contains (local) instabilities, then the ϵ for the stars in the vicinity were non-zero (at least 0.1).

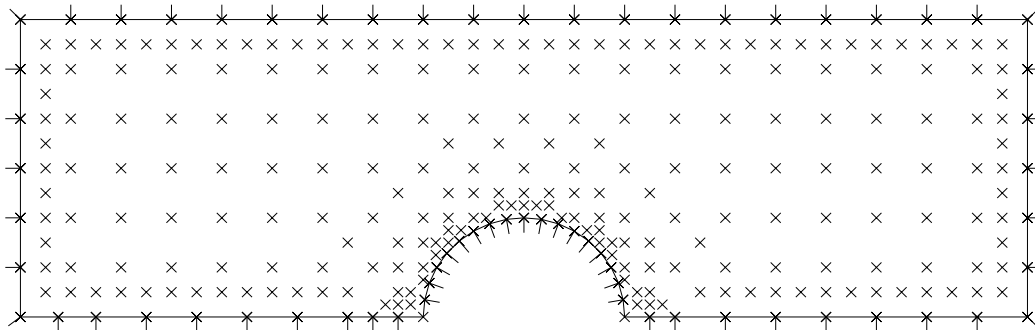


Figure 3.1: Mesh with irregularly spaced nodes

It is easily observed that the stars, which satisfy our intuitive criteria, satisfy also the maximum principle. For some ‘not-so-nice’ stars, the maximum principle may or may

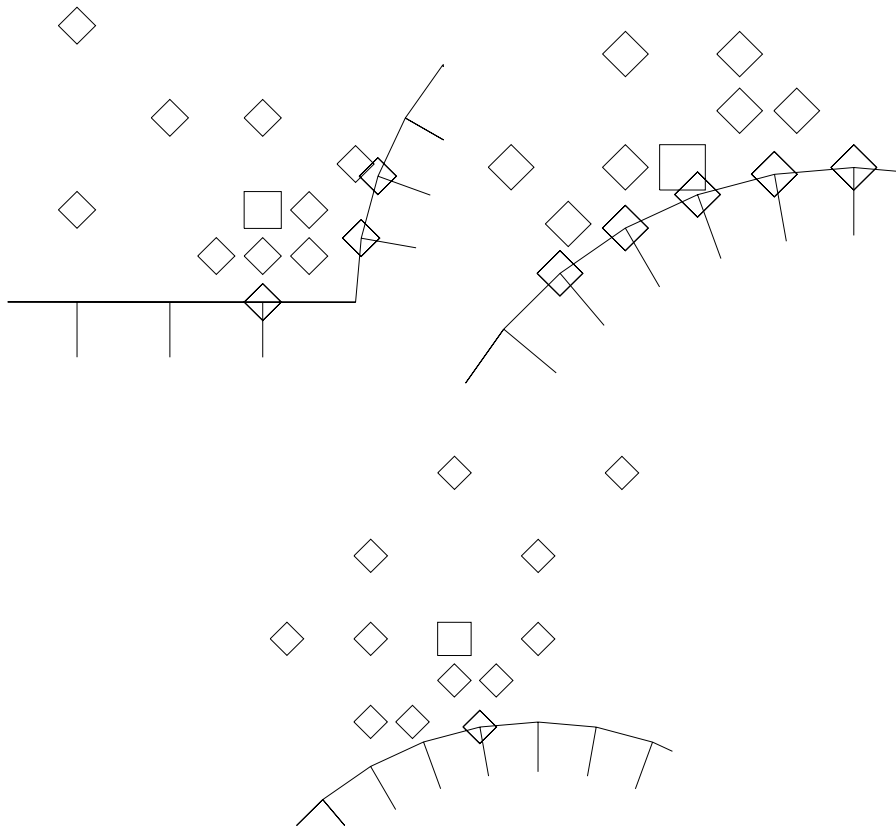


Figure 3.2: Selected 'poor quality' stars (a, b, c in Table 3.1)

not be satisfied, depending on the particular choice of the weight functions. In our tests we selected the mesh shown in the figure 3.1, which contain some irregularities around the central cutoff. We also selected some stars, which tended to produce 'worst case' results (fig. 3.2), and tested how ϵ varied for the different weight functions described above.

Table 3.1 below shows some selected results. We can see from this table that improvements in the selected measure for the star quality were obtained for the value of $p = 2$, while the best correction factor was $c = 24.0$. General observations from these tests may be summarized as follows:

- A quality measure of 1.0 or above is an indication that the center of the star is misplaced and a different node takes role of the "central", i.e. it has the biggest coefficient. In some cases a true central node (origin of the Taylor expansion) had negligible coefficients when compared to the others in the star. This resulted in poor convergence when using iterative methods, and usually in local instability of the solution.

p	c	star a	star b	star c	worst case
6	10.0	10.8814	0.7349	3.5924	10.8814
4	24.0	0.1425	18.7392	0.2775	18.7392
2	10.0	0.2186	0.2378	0.0015	0.3118
2	24.0	0.0571	0.1207	0.0	0.1555

Table 3.1: Star quality for different approximation order and spectral weight.

- Without spectral nodes, all selected cases produced bad approximations for the boundary stars (the stars were obviously unsymmetric).
- The introduction of spectral dof's immediately improved stars on the boundary but the stars next to the boundary (but those containing spectral dof's) still exhibit large values of ϵ .
- The correction factor improved the quality measure of stars close to the boundary without destroying stars on the boundary.

4 Solution of Boundary-Value Problems

4.1 Plate with a Hole

As a first numerical example we consider the problem of an infinite plate with a hole of radius a loaded at infinity by a traction σ_0 in the x direction. The analytical solution of this problem is given by [5]

$$\begin{aligned}
\sigma_{xx} &= \sigma_0 \left\{ 1 - \frac{a^2}{r^2} \left[\frac{3}{2} \cos(2\theta) + \cos(4\theta) \right] + \frac{3a^4}{2r^4} \cos(4\theta) \right\}, \\
\sigma_{yy} &= -\sigma_0 \left\{ \frac{a^2}{r^2} \left[\frac{1}{2} \cos(2\theta) - \cos(4\theta) \right] + \frac{3a^4}{2r^4} \cos(4\theta) \right\}, \\
\sigma_{xy} &= -\sigma_0 \left\{ \frac{a^2}{r^2} \left[\frac{1}{2} \sin(2\theta) + \sin(4\theta) \right] - \frac{3a^4}{2r^4} \sin(4\theta) \right\},
\end{aligned} \tag{4.1}$$

$$\begin{aligned}
u_r &= \frac{\sigma_0}{4G} \left\{ r \left[\frac{(\kappa - 1)}{2} + \cos(2\theta) \right] + \frac{a^2}{r} [1 + (1 + \kappa) \cos(2\theta)] - \frac{a^4}{r^3} \cos(2\theta) \right\}, \\
u_\theta &= \frac{\sigma_0}{4G} \left[(1 - \kappa) \frac{a^2}{r} - r - \frac{a^4}{r^3} \right] \sin(2\theta),
\end{aligned} \tag{4.2}$$

where

$$G = \frac{E}{2(1 + \nu)}, \quad \kappa = \begin{cases} 3 - 4\nu & \text{plane strain} \\ \frac{3 - \nu}{1 + \nu} & \text{plane stress} \end{cases}$$

and (r, θ) is a polar coordinate system with the origin at the center of the hole.

The tractions associated with the stress field (4.1) are applied at the boundary of a finite plate with dimensions 10×10 and with a hole of radius $a = 1$. Due to symmetries only the upper right quadrant is modeled. A plane stress state is assumed. The material properties were taken to be $E = 1000$ and $\nu = 0.3$.

4.1.1 h Convergence and the Influence of the Star Size on the Quality of the Approximation

In this section we analyze the convergence in the energy norm of the Generalized Finite Difference (GFD) solution when the number of nodes in the discretization is increased and the local order of the approximation is kept fixed.

Three discretizations, with 64, 117 and 365 nodes, were used to analyze the h convergence of the method for this problem. The corresponding number of degrees of freedom are 86, 156 and 447, respectively (the additional degrees of freedom correspond to the rotational degrees of freedom used on $\partial\Omega$). In all cases, a second order ($p = 2$) local approximation and stars with 9, 16 and 36 nodes are used to build the MLSF.

Figure 4.3 shows the relative error in the energy norm,

$$\frac{\|\mathbf{u} - \mathbf{u}_h\|_E}{\|\mathbf{u}\|_E} := \frac{\sqrt{B(\mathbf{u} - \mathbf{u}_h, \mathbf{u} - \mathbf{u}_h)}}{\sqrt{B(\mathbf{u}, \mathbf{u})}},$$

where $B(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \sigma_{ij}(\mathbf{u}) \epsilon_{ij}(\mathbf{v}) d\Omega$, associated with the various discretizations. Figure 4.3 also shows the rate of convergence obtained when using stars with 9 and 16 nodes. It can be observed that, for a fixed number of degrees of freedom, that the error in the energy norm and the rate of convergence may considerably vary depending on the number of nodes used in the stars. This same kind of phenomena has been observed in the element free Galerkin method (EFGM) [13]. In this example, the rate of convergence in the energy norm is 1.69 and 0.83 for stars with 9 and 16 nodes, respectively. Belytschko and colleagues have reported a convergence rate of 1.66 when using the element free Galerkin method with a quadratic approximation to solve this problem [4]. However, if nodal integration is used in the EFGM the rate of convergence in the L^2 norm is only 0.65 [2]. The rate of convergence in the energy norm was not reported when using nodal integration but it is reasonable to expect that it would be smaller than in the L^2 norm case.

Figure 4.4 shows the 365 nodes discretization. The contour plot of σ_{xx} corresponding to this discretization is shown in Fig. 4.5. Figure 4.6 shows the computed and exact values of σ_{xx} along the direction $\theta = \pi/2$. The computed values correspond to the 365 nodes discretization with a second order local approximation ($p = 2$) and stars with 9 nodes. It can be observed that the computed values are in good agreement with the analytic results computed from (4.1). The maximum stress computed is equal to 3.12 which is close to the

exact value 3.0 given by (4.1). Figure also 4.7 shows the computed and exact values of the radial component of the displacement vector in the direction $\theta = 0$.

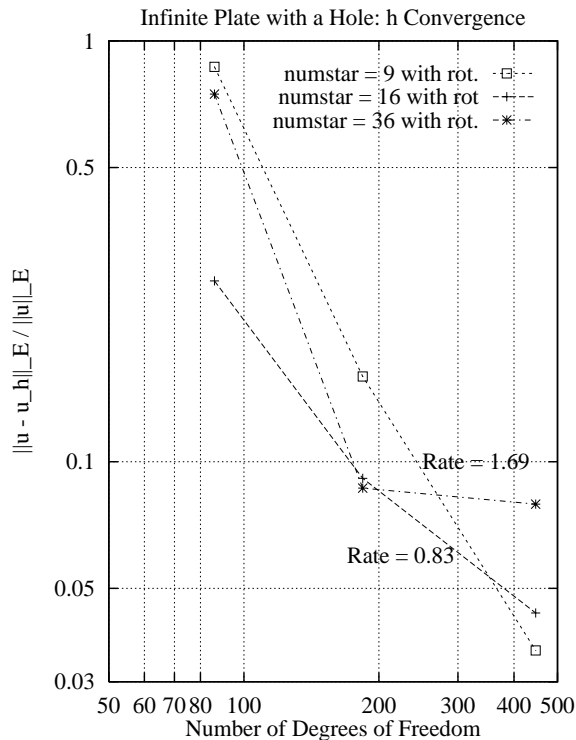


Figure 4.3: h convergence for infinite plate problem.

4.1.2 GFDM with High Order Local Approximation

We investigate in this section the effect of using high order local approximations in the moving least squares method. The plate with a hole problem, as described before, is solved using local approximations ranging from $p = 2$ to $p = 6$. Figure 4.8 shows the relative error in the energy norm associated with discretizations of 117 and 365 nodes for various values of p . The number of nodes in the stars is equal to $(p + 2)^2$. The rate of convergence corresponding to various values of p is also shown. It can be observed that in general the rate of convergence increases with p . Figure 4.9 shows the corresponding results when instead of Neumann boundary conditions, the displacement field given by (4.2) is applied on $\partial\Omega$. The conclusions are the same as before.

4.1.3 Experiments with Hermite Type MLSF

In this section, we investigate the quality of moving least squares (MLS) approximations when using or not derivative degrees of freedom. The plate with a hole problem is used as

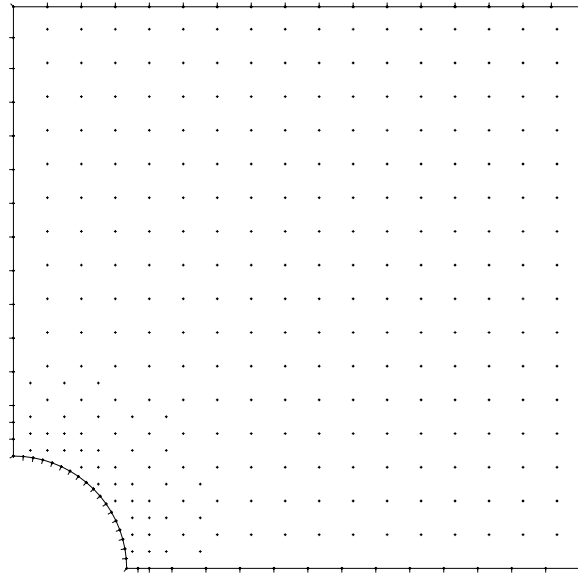


Figure 4.4: 365 nodes discretization.

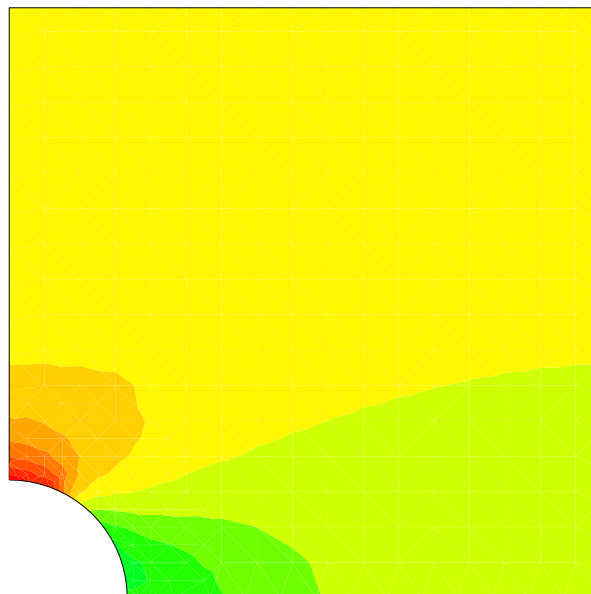


Figure 4.5: Contour plot of σ_{xx} .

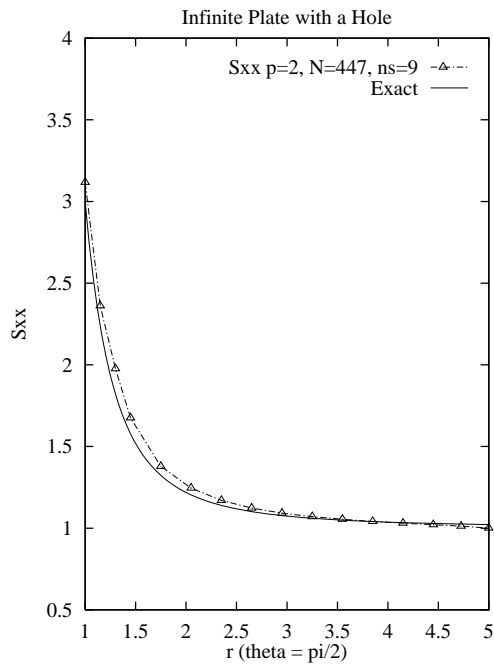


Figure 4.6: Stress along the direction $\theta = \pi/2$.

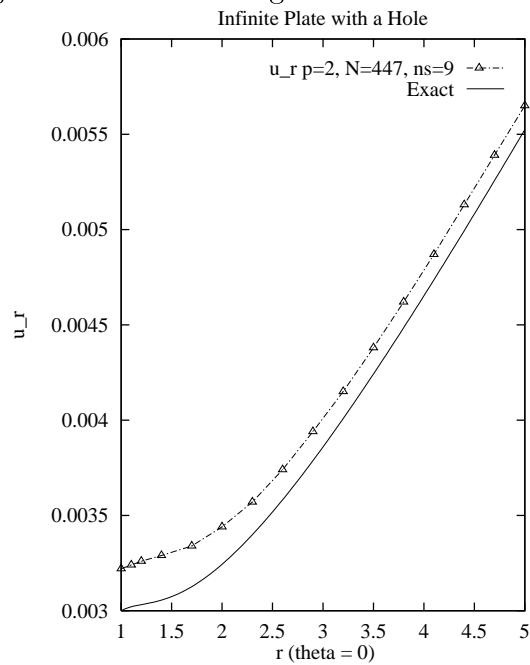


Figure 4.7: Radial displacement along the direction $\theta = 0$.

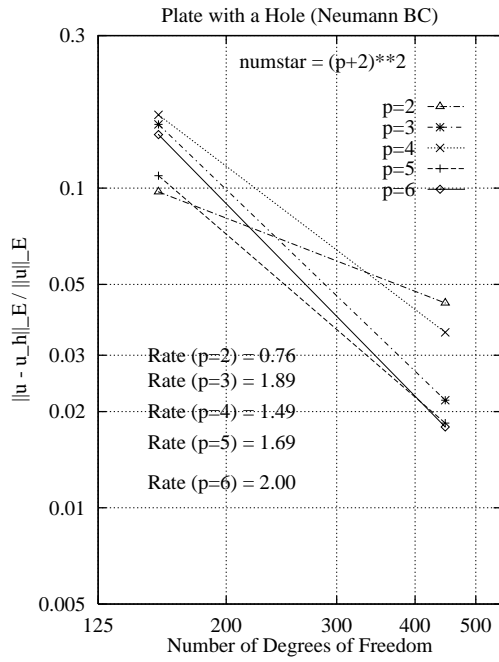


Figure 4.8: GFDm with high p and Neumann Boundary Condition.

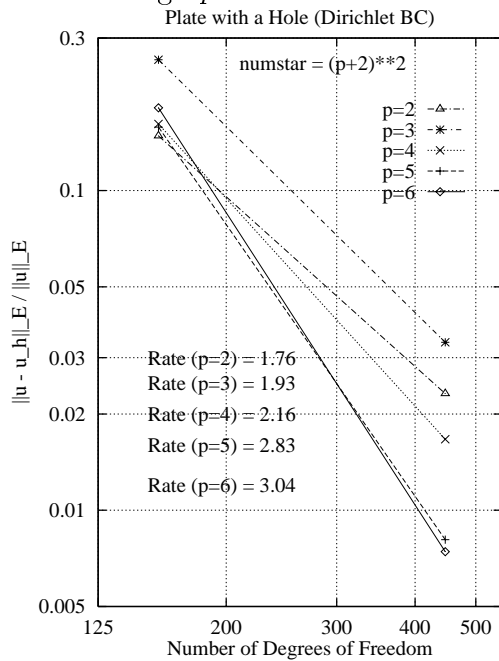


Figure 4.9: GFDm with high p and Dirichlet Boundary Condition.

	numstar = 9	numstar = 16	numstar = 36
w/out der. dof	0.64735377	0.71665284	0.55436978
w/ der. dof	0.10947508	0.096612338	0.093279968

Table 4.2: Relative error in the energy norm for the 117 nodes discretization.

a model problem. Figure 4.10 shows the 117 nodes discretization with derivative degrees of freedom at the boundary $\partial\Omega$ used in the experiments.

As a first experiment, we investigate the effect of the approach developed in Section 3.1 on the quality of the stars associated with the 117 nodes discretization. The star associated with a boundary node at $(5, 2)$ is shown in Fig. 4.11.

As in Section 3.1, the stencils are considered acceptable or not if

- (i) The sum of the MLS functions must be one, since they represent a partition of unity [9].
- (ii) The sum of the derivatives of the MLS must be zero for the same reason as in (1).

In addition, we also impose that at a node \mathbf{x}_I

$$\begin{aligned}
 a_J &:= \Delta\Phi_J = \Phi_{J,xx} + \Phi_{J,yy} > 0 \quad J = 1, \dots, \text{numstar}, \quad J \neq I \\
 \sum_{J=1, J \neq I}^{\text{numstar}} |a_J| &= -a_I
 \end{aligned} \tag{4.3}$$

where numstar is equal to the number of nodes in the star and Φ_J is the MLSF associated with node \mathbf{x}_J in the star.

Demkowicz et al. [6] have shown that if (but not only if) the above criteria is satisfied, the approximate solution obtained by the generalized finite difference method converges to the exact solution.

The values of a_J for the star associated with the node at $(x = 5.0, y = 2.0)$ are shown Fig. 4.12 (the values are scaled by $h^2 = 0.25$, h being the local distance between the nodes). The values correspond to the case where only standard MLS functions are used, i.e., no Hermite type MLS function is used. It can be observed that some of the a_J 's are negative and therefore the condition (4.3) is violated. Figure 4.13 shows the corresponding values of a_J when Hermite type MLS functions are used. In this case the condition (4.3) is satisfied.

The relative error in the energy norm when Hermite type MLS functions are used is shown in Table 4.2. For comparison, the results when only standard MLS functions are used is also included in the table. The errors correspond to the 117 nodes discretization shown in Fig. 4.10 with stars of various sizes. It can be observed that, for this problem, the addition of the Hermite type MLS functions significantly improves the results.

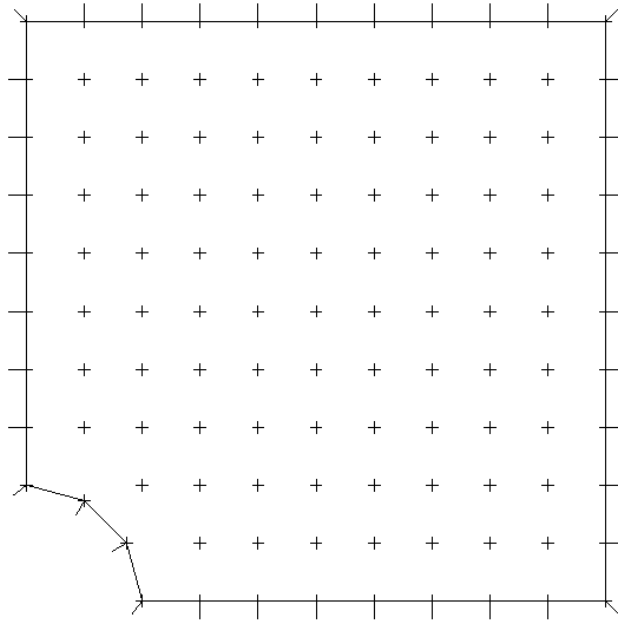


Figure 4.10: 117 nodes discretization with rotational degrees of freedom at the boundary.

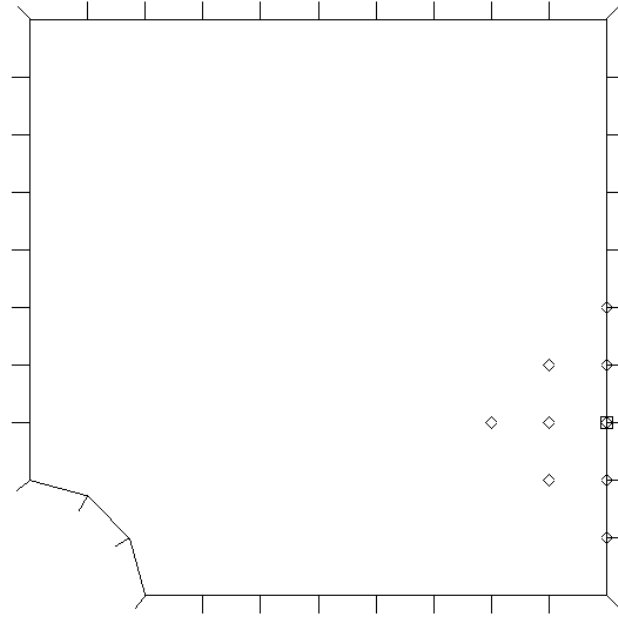


Figure 4.11: Star analyzed.

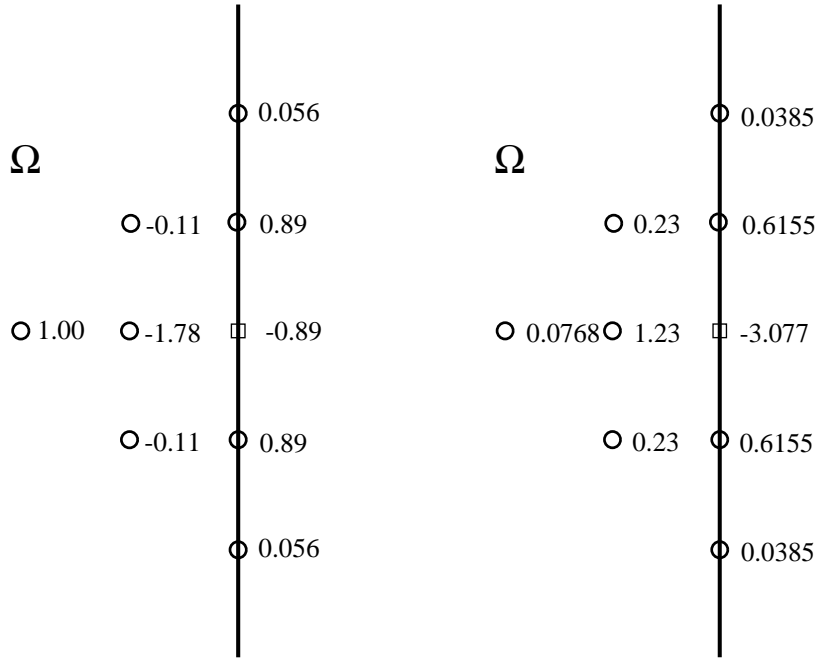


Figure 4.12: Values of $a'_j s$ when only standard MLS functions are used.

Figure 4.13: Values of $a'_j s$ when Hermite type MLS functions are used.

Figure 4.14 compares the convergence in the energy norm when using MLS functions with and without derivative degrees of freedom. The results correspond to the 64 and 117 nodes discretizations and stars with 9, 16 and 36 nodes. It is clear that the use of Hermite type MLS functions is advantageous.

4.2 Cracked Panel

Let us consider the problem of a cracked panel.

We assume a state of plane stress and apply as boundary conditions on $\partial\Omega$ the first symmetric mode of the asymptotic expansion of the solution around the crack tip,

$$\begin{aligned}
 u_x &= \frac{K_I}{2\mu} \sqrt{\frac{r}{2\pi}} \cos\left(\frac{\theta}{2}\right) \left[\kappa - 1 + 2 \sin^2\left(\frac{\theta}{2}\right) \right] \\
 u_y &= \frac{K_I}{2\mu} \sqrt{\frac{r}{2\pi}} \sin\left(\frac{\theta}{2}\right) \left[\kappa + 1 - 2 \cos^2\left(\frac{\theta}{2}\right) \right]
 \end{aligned}$$

where K_I is the first stress intensity factor, $\mu = E/(2(1+\nu))$ and $\kappa = (3-\nu)/(1+\nu)$. (r, θ) is a polar coordinate system centered at the crack tip. The corresponding stresses are given

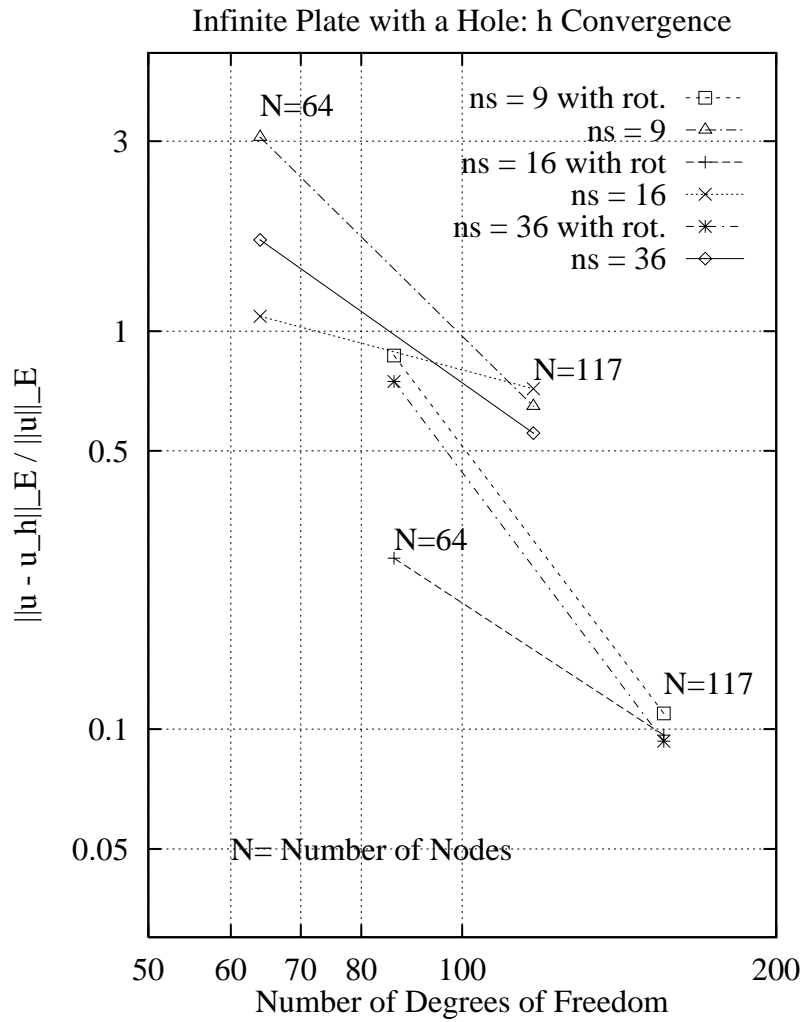


Figure 4.14: h convergence for discretizations with and without Hermite type MLSF.

by [24]:

$$\sigma_{ij} = \frac{K_I}{\sqrt{2\pi r}} f_{ij} \theta \quad (4.4)$$

$$f_{xx}(\theta) = \cos\left(\frac{\theta}{2}\right) \left[1 - \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{3\theta}{2}\right) \right] \quad (4.5)$$

$$f_{yy}(\theta) = \cos\left(\frac{\theta}{2}\right) \left[1 + \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{3\theta}{2}\right) \right] \quad (4.6)$$

$$f_{xy}(\theta) = \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{3\theta}{2}\right) \quad (4.7)$$

We adopted $E = 1000$ and $\nu = 0.3$.

Due to symmetry, only the upper half of the panel is modeled. Figures 4.15 and 4.16 show the two nodal arrangement used.

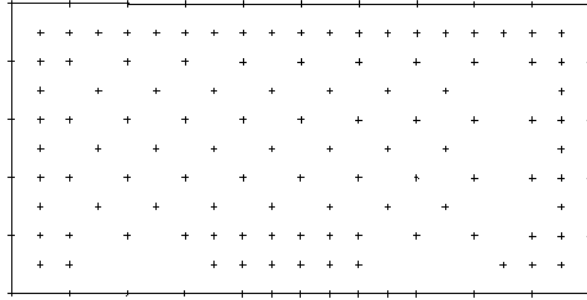


Figure 4.15: 137 nodes discretization.

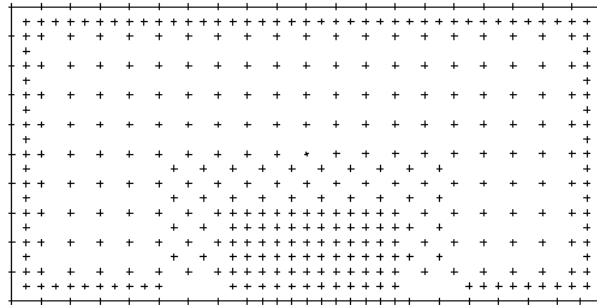


Figure 4.16: 411 nodes discretization.

Figure 4.17 shows the component σ_{yy} of the stress tensor along the direction $\theta = 0$. One GFD solution corresponds to the coarse discretization with $p = 2$ and 16 nodes stars. The other two solutions correspond to the finer mesh with $p = 2$ and stars with 16 and 36

nodes. It can be observed that the quality of the solution, in this case, does not change with the size of the stars. Both discretizations agree well with the closed form solution for $r > 0.01$ and are able to capture the strong gradient of the solution near the crack tip. Figure 4.18 shows equivalent results for the component σ_{xx} of the stress tensor.

Figures 4.19 and 4.20 show contour plots of σ_{xx} and σ_{yy} obtained with the finer discretization and stars with 36 nodes.

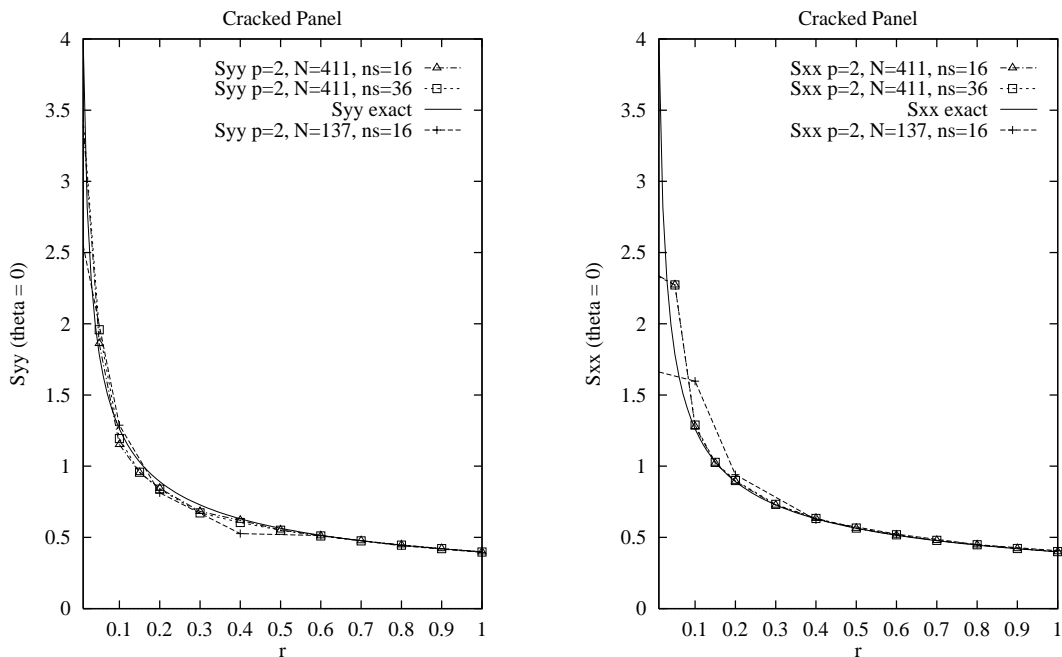


Figure 4.17: Computed values of S_{yy} .

Figure 4.18: Computed values of S_{xx} .

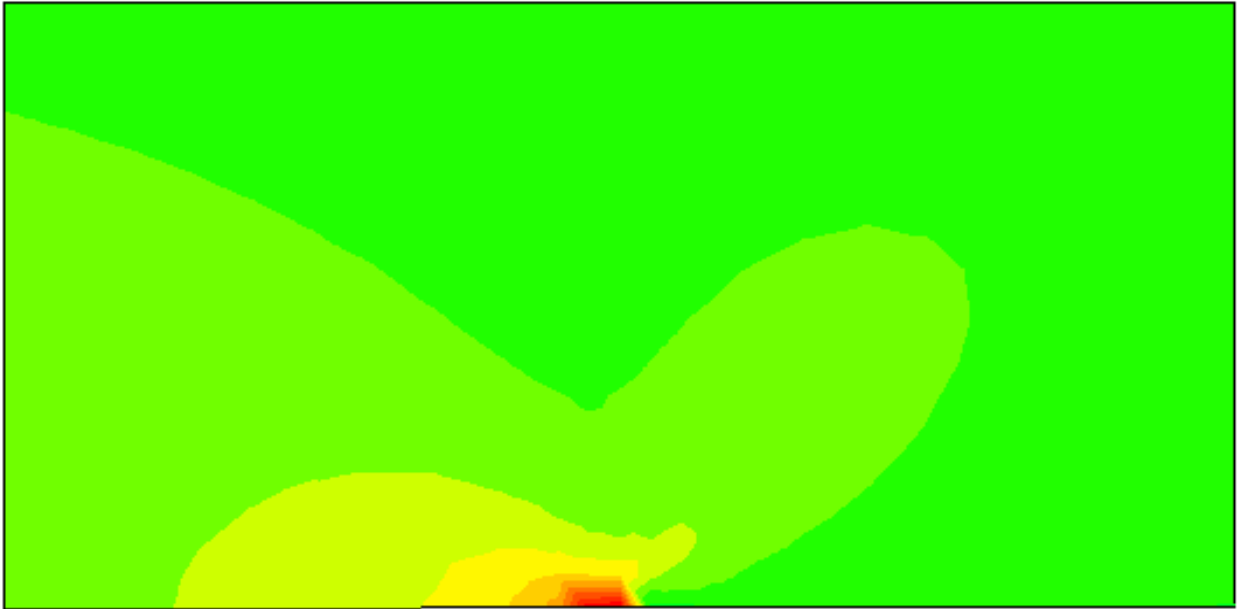


Figure 4.19: Contour plot of σ_{xx} .

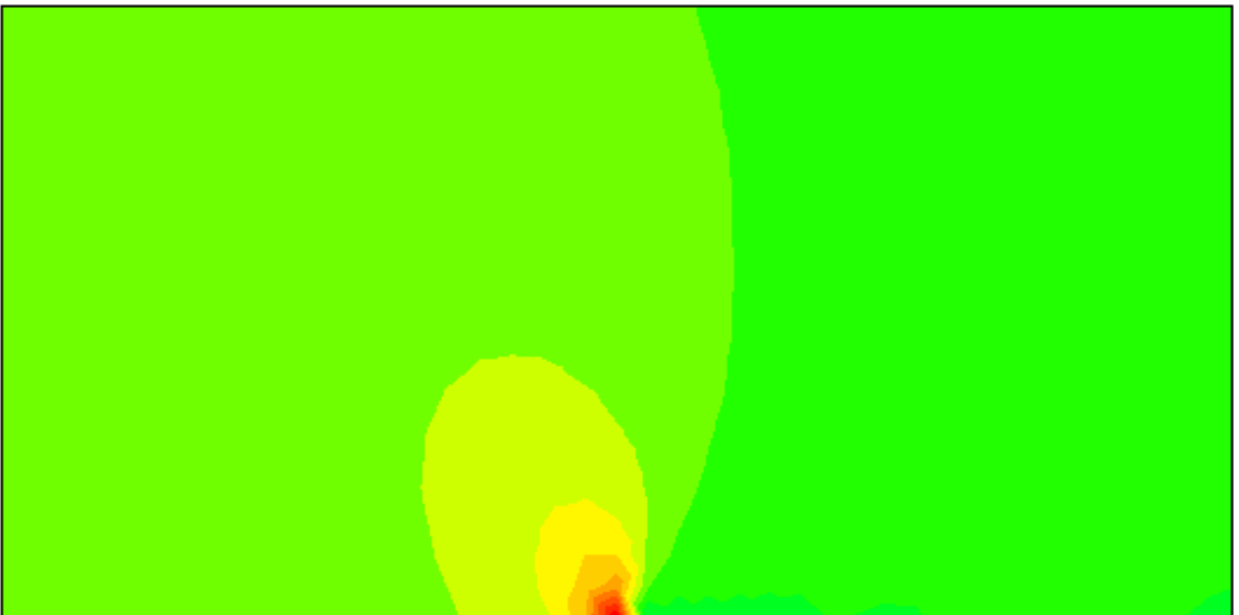


Figure 4.20: Contour plot of σ_{yy} .

5 Other Examples

The majority of examples presented in this paper are focused on linear elasticity and fracture mechanics. However, the meshless methods presented here are applicable to a much broader class of problems. In fact, virtually any boundary-value problem governed by a system of PDE's of second or higher order can be effectively solved using this method. As an illustration of such applications, we present several solutions of various problems in nonlinear structural mechanics and heat transfer.

The meshless methods are fully applicable to the solution of nonlinear problems, such as large deformations of membrane shells, governed by strongly nonlinear geometrical and constitutive equations. An example of a pneumatic covering formed in the shape of a semicylinder with spherical end sections is shown in figure 5.1. The radius of the cylinder is 5m, its length is 10m and the radii of the spherical end segments are 5m. The material behavior is governed by a Mooney-Rivlin constitutive law. The covering is subjected to an internal pressure $p = 500 Pa$ and loaded with the pressure of a wind blowing in the z^1 direction at a velocity of $42m/sec$. The grid of points used to solve this problem on a flat reference configuration is presented in figure 5.2 (only one quarter of the domain is shown). The wind loading is represented by a quasi-static steady-state model, with the coefficients of the wind pressure shown in figure 5.3. The calculated shapes of the cross-sections of this covering are shown in full scale in figures 5.4.

A nontrivial nonlinear application of the meshless method is the solution of inverse problems at large deformations, where the initial configuration is calculated for a prescribed final configuration of the membrane shell. In the present formulation of the meshless method, the formulation of such a problem is quite straightforward – for details see reference [26]. However, the solutions to this type of problems are, in general, non-unique. This is illustrated in the example shown in figure 5.5, where an initial configuration was calculated for a membrane with a prescribed final configuration (ideal hemisphere of radius $r = 10$) under an internal pressure loading. This problem was solved using two grids with 16 and 36 nodes, on one quarter of a circle. The calculated initial configurations are quite different for these two grids – this is a reflection of the non-uniqueness of the solutions to this problem.

As an example of an application of meshless method to problems with nonlinear constitutive laws, an elasto-plastic torsion of a prismatic bar is shown in figure 5.6. In particular, the figure shows the shape of elastic-plastic boundaries and distributions of shear stress along various cross-sections.

It is important to note that the meshless methods described here can also be applied to the solution of transient problems, such as heat transfer or nonlinear dynamics. An example of a time-dependent temperature distribution for a rail-shaped bar is illustrated in figure 5.7. The figure shows the shape of the domain and calculated temperature profiles at several time steps.

Figure 5.1: A cylindrical pneumatic covering.

Figure 5.2: The grid used in the analysis of pneumatic covering.

Figure 5.3: Presumed coefficients of wind pressure.

Figure 5.4: Deformation of the pneumatic covering under internal pressure and wind load ($42m/sec$).

Figure 5.5: Solution of an inverse problem for a hemispherical membrane.

Figure 5.6: Torsion of a prismatic bar: elasto-plastic boundaries and shear stress.

Figure 5.7: Time-dependent heat flow in a rail-shaped bar.

Acknowledgements

The partial support of this work by the Office of Naval Research under the SBIR contract N00014-95-C-0373 is gratefully acknowledged.

6 References

- [1] I. Babuska and J. M. Melenk. The partition of unity finite element method. Technical Report BN-1185, Inst. for Phys. Sc. and Tech., University of Maryland, July 1995.
- [2] T. Belytschko and S. Beissel. Nodal integration of the element-free Galerkin method. pre print, 1995.
- [3] T. Belytschko, Y. Y. Lu, and L. Gu. Crack propagation by element free Galerkin methods. In *Advanced Computational Methods for Material Modeling*, pages 191–205, 1993. AMD-Vol. 180/PVP-Vol. 268, ASME 1993.
- [4] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.
- [5] A. P. Boresi and P. P. Lynn. *Elasticity in Engineering Mechanics*. Prentice-Hall, New Jersey, 1974.
- [6] L. Demkowicz, A. Karafiat, and T. Liszka. On some convergence results for FDM with irregular mesh. *Computer Methods in Applied Mechanics and Engineering*, 42:343–355, 1984.
- [7] C. A. M. Duarte. A review of some meshless methods to solve partial differential equations. Technical Report 95-06, TICAM, The University of Texas at Austin, 1995.
- [8] C. A. M. Duarte and J. T. Oden. Hp clouds—a meshless method to solve boundary-value problems. *Numerical Methods for Partial Differential Equations* (to appear).
- [9] C. A. M. Duarte and J. T. Oden. Hp clouds—a meshless method to solve boundary-value problems. Technical Report 95-05, TICAM, The University of Texas at Austin, 1995.
- [10] P. S. Jensen. Finite difference techniques for variable grids. *Computers and Structures*, 2:17–29, 1972.
- [11] Z. Kaczkowski and R. Tribillo. A generalization of the finite difference formulas. *International Journal for Numerical Methods in Engineering*, 3(9):581–599, 1975.
- [12] Y. Krongauz and T. Belytschko. Enforcement of essential boundary conditions in meshless approximations using finite elements. to appear, 1995.
- [13] P. Krysl and T. Belytschko. Analysis of thin plates by the element-free Galerkin method. Submitted to *Computational Mechanics*, 1995.
- [14] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981.
- [15] P. Lancaster and K. Salkauskas. *Curve and Surface Fitting, an Introduction*. Academic Press, San Diego, 1986.

- [16] T. Liszka. An interpolation method for an irregular net of nodes. *International Journal for Numerical Methods in Engineering*, 20:1599–1612, 1984.
- [17] T. Liszka and J. Orkisz. Finite difference method for arbitrary irregular meshes in nonlinear problems of applied mechanics. In *IV SMiRt*, San Francisco, 1977.
- [18] T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures*, 11:83–95, 1980.
- [19] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.
- [20] V. Pavlin and N. Perrone. Finite difference energy techniques for arbitrary meshes applied to linear plate problems. *International Journal for Numerical Methods in Engineering*, 14:647–664, 1979.
- [21] N. Perrone and R. Kao. A general finite difference method for arbitrary meshes. *Computers and Structures*, 5:45–58, 1975.
- [22] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
- [23] J. Szmelter and Z. Kurowski. A complete program for solving systems of linear partial differential equations in plain domains. In *Proceedings of the 3rd Conference on Computational Methods in Structural Mechanics*, pages 237–247, Opole, Poland, 1977. in Polish.
- [24] H. Tada, P. Paris, and G. Irwin. *The Stress Analysis of Cracks Handbook*. Del Research Corporation, St. Louis, Missouri, 1973.
- [25] R. Tribillo. *Application of Algebraic Structures to a Generalized Finite Difference Method*. Politechnika Bialostocka, Poland, 1976. in Polish.
- [26] W. Tworzydło. Analysis of large deformations of membrane shells by the generalized finite difference method. *Computers and Structures*, 27(1):39–59, 1987.