Meshfree Methods of Airframe Stress Analysis

N S Kalutskiy
Researcher, Moscow Aviation Institute (National Research University), Moscow, Russia

E-mail: n.kalutskii@yandex.ru

Abstract. Present-day meshfree methods of numerical computer analysis of structures have been analyzed in this article. Principal mathematical ratios used in meshfree methods have been provided. Building of a shape function, one of the principal problems of the meshfree method, has been discussed, and an algorithm for the building of a shape function by using polynomial basis functions has been proposed. The algorithm developed has been tested on solving an elementary problem. The Conclusion section of this article contains solution results of a practically important problem of mechanical stress concentrations in a typical aircraft stringer. Comparative analysis of the accuracy of the results obtained by the conventional FEM and the meshfree method proposed has been conducted.

1. Introduction
Whenever it is possible to describe a natural phenomenon mathematically, it is most likely that the result will be one or more differential equations. This is true for a wide range of phenomena. Unfortunately, the number of differential equation types, that might be solved analytically, remains very limited. There exist many problems that have been formulated in a form of differential equations of partial derivatives, but their exact solutions still can’t be found. This leads to a wide usage of numerical methods, along with analytical and approximation methods. Numerical methods have gained the leading position due to the vast advances of computers in the recent decades. Common accessibility of high-performance computers contributes to both successful practical application of existing numerical methods and the development of advanced analysis methods.

To provide solutions of the partial derivatives equations, the finite difference method (FDM) was introduced in the 1930s. The finite element method (FEM) and the finite volume method (FVM) have been used since the 1960s until present. All of the above methods (hereinafter referred to as the “mesh methods”) have one thing in common – the substitution of the continuous variable function with a finite (discrete) set of points (nodes) connected to a pattern or mesh of elements. The function determined at the mesh nodes only, and not a continuous argument function, is considered. Thus, a differential equation system is substituted with a set of algebraic equations here.

The FEM is currently an industrial standard for solving a wide range of scientific and engineering problems. The FEM has been implemented with many commercially available software packages, including ANSYS, NASTRAN, LS-DYNA, ABAQUS and others used globally. In this paper, we are going to refer FEM as the baseline for newly proposed method assessment.

The necessity of mesh generation is the main disadvantage of the FEM and the mesh methods. It takes engineers who use FEM packages most of their working time to build a finite element mesh.
Special mesh-generating software may provide some rate of automation to the process; yet, they are no substitute for humans. As the cost of machine time is much lower than that of “manual” labor of high-skilled engineers, the labor intensity of a “manual” mesh generation done by an engineer accounts for a key cost contribution to the CAD project.

Since 1990s various scientific papers reported an emerge of advanced meshfree methods (see Babenko 1979; Atluri et al., 1998; Liu et al., 1999), promising to reduce considerably the costs and the durations of development of new industrial products. A method that does not imply finite element mesh generation could lead to almost complete automation of engineering analysis.

2. Meshfree methods algorithms

Fig.1 below shows a comparison between two approaches. View (a) shows a model prepared for a FEM solver and is essentially a finite element mesh. View (b) shows the same model free of all the connections between the nodes (that is, the elements), while retaining the nodes from which a solution of the problem is to be found by the meshfree method.

![Figure 1. A comparison between FEM and the meshfree methods.](image)

The FEM uses discrete function values at the nodes of a finite element for the approximation of the function values at any point of the element in question. To do that, special approximation functions are introduced, called shape functions. There exist shape functions for various finite element types – from basic planar shapes, e.g., lines, triangles, rectangles, to complex ones, e.g., tetrahedrons, 20-node prismatic elements, etc. Using the FEM implies finding problem solution, based on predefined shape functions. Things are different when there are no known connections between nodes, that is, when no element mesh exists. In this case, a shape function must be calculated by solver. The is the main challenge on the way to implementation of meshfree methods.

As the meshfree approach is free of connections between nodes, the value of the field variable of, e.g., the displacement \( \mathbf{u} \) at an arbitrary point \( (x, y, z) \) that lies within the problem domain, must be found by interpolation of the known values of the field variable by means of neighboring nodes that are located closely enough

\[
\mathbf{u}(x, y) = \sum_{i=1}^{n} \phi_i \mathbf{u}_i ,
\]  

(1)
where \( n \) is the number of nodes included in the support domain, \( U_j \) is the value of the field variable at the \( i^{th} \) node, and \( \phi_i \) the value of the shape function \( \phi_i \).

The concept of a support domain must be introduced into the meshfree method to provide a clear definition of the neighboring nodes, see Fig.2 below.

**Figure 2.** The concept of the reference domain.

The size of the support domain is found from the expression

\[
d_s = \alpha d_i,
\]

where \( \alpha \) is the dimensionless coefficient of the support domain value; and \( d_i \) is the average nodal spacing about a point of interest.

Despite the self-evident advantages of the meshfree methods over the conventional ones, the former is not yet being commonly used due to a number of challenging problems that remain unsolved and therefore hinder the development of commercial meshfree software packages. E.g., the work (Onate 1996; Liu et al. 2002; Liu et al 2003) treat the shape function construction as the principal problem under investigation of the method.

This paper discusses some of the shape function construction methods applicable to subsequent usage for solving problems by the meshfree method.

### 3. Shape function definition

Let us write down a function to be approximated as a series. Let the function \( u(x, y) \), be determined within the problem domain \( \Omega \). If arbitrarily located points with known values of the function (let us call those points nodes) are determined within the problem domain, then the interpolated value of the function can be expressed as

\[
u(x, y) = \sum_{i=1}^{n} B_i(x, y) \cdot \alpha_i,
\]

where \( B_i(x, y) \) is the basis function, and \( \alpha_i \) are the basis function coefficients.

Now let us consider how the two basic functions, namely, the polynomial (Liu 1999) and the radial (Liu 2002) basis functions are used.

#### 3.1. Polynomial basis function

The use of polynomials as basic functions has been known for a long time, e.g., in the FEM.
Let us consider the continuous function \( u(x, y) \), determined within the problem domain \( \Omega \). Suppose that the values of the field function at certain points (nodes) are known. Then, the function \( u(x, y) \), value at any arbitrary point can be found from the expression below

\[
\begin{bmatrix}
  a_1 \\
  a_2 \\
  \vdots \\
  a_m
\end{bmatrix} = \mathbf{P}^T \mathbf{a},
\]

(4)

where \( p_i(x, y) \), are monomials, \( m \) is the number of monomials, and \( \alpha_i \) are the coefficients found by interpolations. The polynomial in a general form is:

\[
p^i(x, y) = \{1 \ x \ y \ x^2 \ xy \ y^2 \ \ldots \ \ x^p \ y^p\},
\]

where \( p \) is the order of the polynomial.

Further, we are to find the coefficients \( \alpha_i \). To do that, a support domain is to be determined in the neighborhood of a point of interest, and the \( n \) nodes that fall within the support domain, see Fig. 2 above.

It is worth noting that the method in question always uses as many monomials of the polynomial \( m \) as the nodes \( n \) contained by the support domain, that is, \( n = m \). The coefficients \( \alpha_i \) of Equation (3) are to be selected so that the function \( u(x, y) \), fits exactly to the known values at each of the \( n \) nodes – “strict” interpolation. Hence, a system of \( n \) equations is deduced

\[
\begin{align*}
  u_1 &= a_1 p_1(x_1, y_1) = a_1 + a_2 x_1 + a_3 y_1 + \cdots + a_m p_m(x_1, y_1) \\
  u_2 &= a_1 p_1(x_2, y_2) = a_1 + a_2 x_2 + a_3 y_2 + \cdots + a_m p_m(x_2, y_2) \\
  \vdots \\
  u_n &= a_1 p_1(x_n, y_n) = a_1 + a_2 x_n + a_3 y_n + \cdots + a_m p_m(x_n, y_n)
\end{align*}
\]

(5)

System (5) can be rewritten in a matrix form

\[
\mathbf{U}^T = \mathbf{P} \mathbf{a},
\]

(6)

Solving Equation (6) against \( \mathbf{a} \), we find that

\[
\mathbf{a} = \mathbf{P}^T \mathbf{U}^T.
\]

(7)

While at this stage, we suppose that the inversed matrix \( \mathbf{P}^T \) of Equation (7) does exist and is not a degenerate matrix.

The coefficients \( \mathbf{a} \) of Equation (7) are constants that do not depend on interpolation points location within the support domain. Substituting Equation (7) into Equation (4), keeping in mind that \( n = m \), we find that

\[
u(x, y) = \mathbf{P}^T(x, y)\mathbf{P}^T \mathbf{U}^T = \sum_{i=1}^{m} \phi_i u_i = \Phi^T(x, y)\mathbf{U}^T,
\]

(8)

Where \( \Phi^T(x, y) \), is the vector of values of the shape function, and is found from

\[
\Phi^T(x, y) = \mathbf{P}^T(x, y)\mathbf{P}^T = \{\phi_1(x, y) \ \phi_2(x, y) \ \ldots \ \phi_m(x, y)\},
\]

(9)

Derivatives of a shape function of a required order can be easily found, because, as it was stated earlier, shape functions in this case have a polynomial expression.
3.1.1. An example of building a shape function by the radial point interpolation method (RPIM). Let us provide an example of finding a shape function and its derivatives by using the radial point interpolation method (RPIM). Fig. 4 below shows a square-shaped problem domain limited within the coordinates $x \in [-1,1]$ and $y \in [-1,1]$. 25 nodes ($5 \times 5$) are determined within the domain, distributed evenly at a span 0.5 along the axes $x$ and $y$. Using the RPIM, we are able to find the shape function value at an arbitrary point of the set domain. In this case, we will limit ourselves to $61 \times 61 = 3,721$ points evenly distributed along the axes $x$ and $y$.

\[
\Phi^{(m)}(x, y) = \begin{cases} 
\phi_{i}^{(m)}(x, y) \\
\phi_{i}^{(m)}(x, y) \\
\vdots \\
\phi_{i}^{(m)}(x, y) 
\end{cases} = \frac{\partial^m P(x, y)}{\partial x^i \partial y^j} P^{-1},
\]  

(10)

Figure 4. A total of 25 nodes evenly distributed across the problem domain to build a shape function.

To solve the problem, we used the RPIM supplemented with the multiquadric function (13), for which the following shape parameters were set: $\alpha = 0.03; \ d_j = 0.5; \ q = 2.03$. The problem was solved by computer program written in FORTRAN 2003. Please follow the links (GitHub 2019; Shape Function Plot 2019; Shape Function Derivative Plot 2019) for the program’s source code and the research findings. Fig. 5 below shows a shape function diagram built.
Figure 5. The shape function $\phi$ found by the RPIM for node No 13 (point $x=0$, $y=0$).

4. Problem of stress concentrations in an aircraft stringer
Stress concentrations were evaluated for a radius area of a typical aircraft stringer as part of this research.
At the beginning of the research, a problem of elastic stresses around the central hole in an infinite plate – the Kirsch problem – for which an analytical solution was known, see, e.g., (Demidov, 1979), was solved to evaluate the accuracy of the method proposed.

Figure 6. A solution of the Kirsch problem by the meshfree method.
The resulting stress curve (Fig. 6 above) shows that the numerical values are in good accordance with the theoretical ones.
To solve a more complex problem of stress concentrations in an aircraft stringer, two models were built and solved, each having its node density, namely, Model One containing 180 nodes, and Model Two containing 460 nodes – see Fig. 7 below.

Figure 7. Model Two (460 nodes)
The graphical results of the Model Two solution (the first principal stresses, MPa) are shown on Fig. 8 below.

Figure 8. The results of stress calculation.
Comparing the results for the two Models, let us evaluate how the value found for maximum stresses changes as numbers of nodes in Models increase.

\[ e = \frac{\sigma_{\text{max,1}} - \sigma_{\text{max,2}}}{\sigma_{\text{max,1}}} = \frac{1.4228 - 1.3587}{1.4228} = 0.045 = 4.5\%. \]

Thus, the error that occurs when using Model One (180 nodes) is approx. 5%, compared to that yielded by a more computational resource-consuming Model Two (460 nodes).

4.1. \textit{Comparing with the FEM results}

Finite element method aircraft stringer models were built to verify the results found by the meshfree method. The dimensions of standard finite elements of the models and the numbers of their respective nodes were as follows: FEM Model One (3 mm, and 390 nodes); FEM Model Two (1 mm, and 3000 nodes), and FEM Model Three (0.5 mm, and 7600 nodes). The number of the nodes of the roughest model (390 nodes) was close to that of Model Two – the adjusted meshfree model containing 2, 460 nodes.

Then, we were to evaluate the error that occurred in transition from FEM Model Three to a rougher model – Model One.

\[ e = \frac{\sigma_{\text{max,FEM,3}} - \sigma_{\text{max,FEM,1}}}{\sigma_{\text{max,FEM,1}}} = \frac{1.384 - 1.201}{1.201} = 0.152 \approx 15\%. \]

The error was 15%, which was more than three times that of the meshfree method proposed.

5. \textit{Discussion}

The meshfree method used in this research allows evaluate quickly the stress concentrations in a geometric concentrator (here, a radius area) without building a complex finite element model. The whole process can be automated. All it takes to solve a problem by the meshfree method is to define the nodes (the points) at the boundary of the problem domain, and nodes arbitrarily located within the calculation domain. This preserves the geometrical parameters (the radius bend), which is critical to stress concentration. Arbitrary condensation of nodes near the domain of interest is available, which adds considerably to the complicity of the FEM. This article demonstrates that the meshfree method is less dependent on the nodes density, the error being 5% as against 15% of the FEM.

In the FEM, stress values change discretely (in leaps and bounds) in transition from element to element. To eliminate the disadvantage in question, neighboring element stress averaging is used, which contributes to inaccuracy of the total value of the effective stresses. In the FEM, the result accuracy is highest at integration points that coincide with the centers of gravity of a triangular and a quadrilateral element. Stress values at the nodes contain an accumulated error of stress interpolation inside an element. On the contrary, the meshfree method yields conservative results. Stress values are lower as nodes become packed more densely, while in the FEM, stresses increase as the mesh density increases.

6. \textit{Conclusion}

Based on the results shown in this paper the meshfree methods might be proposed as a promising alternative to the labor-intensive FEM-approach in the future.

7. \textit{References}

[1] Demidov S P 1979 \textit{The Elasticity Theory} (Moscow: Higher School) p 302
[2] Babenko K I 1979 Theoretical foundations and design of numerical algorithms for problems of mathematical physics (Moscow: Science) p 295
[3] Atluri S N and Zhu T A 1998 New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics (\textit{Computational Mechanics}) vol 22 issue 2 pp117–127
[4] Liu G R and Gu Y T 1999 A Point Interpolation Method \textit{Proc of the APCOM'99: Fourth Asia-
Pacific Conference on Computational Mechanics, 13-15 December 1999, edited by C.M. Wang, K.H. Lee and K.K. Ang. (Singapore: Elsevier) pp1009–1014

[5] Liu G R and Gu Y T 2002 Comparisons of Two Meshfree Local Point Interpolation Methods for Structural Analyses (Computational Mechanics) vol 29 release 2 pp 107–121

[6] Onate E and Idelsohn S and Zienkiewicz O C and Taylor R L 1996 A Finite Point Method in Computational Mechanics Applications to Convective Transport and Fluid Flow International Journal for Numerical Methods in Engineering vol. 39 release 2 pp 3839–3866

[7] Liu G R 2002 Meshfree Methods: Moving Beyond the Finite Element Method (USA: CRC Press) p 712

[8] Liu G R and Dai K Y and Lim K M and Gu Y T 2003 A Radial Point Interpolation Method for Simulation of Two-Dimensional Piezoelectric Structures Smart Materials and Structures vol 12 release 2 pp 171–180

[9] GitHub 2019 MFree Source Code accessed 11 March 2019 <https://github.com/Buran1988/MFree>

[10] Shape Function Plot accessed 11 March 2019 <https://plot.ly/~Nikita1988/257.embed>

[11] Shape Function Derivative Plot accessed 11 March 2019 <https://plot.ly/~Nikita1988/248.embed>