GENERATION OF SHAPE FUNCTIONS BY OPTIMIZATION

O. C. AKEREMALE, O. A. OLAIJU, AND S. H. YEAK

ABSTRACT. In boundary value problems, the solution region is always discretized into finite elements. The polynomial chosen to interpolate the field variables over the element are called shape functions. The shape functions establish the relationship between the displacement at any point in the element with the nodal displacement of the element. However, the polynomial cannot guarantee the shape function of all the transition elements as the inverse of the matrix generated from some of the transition elements are not feasible. This paper offers an insight into the derivation of shape function using minimization theory. In the case of irregular elements, such as transition elements, improvements are made regarding the derivation so as to capture the peculiarities of the so-called transition elements. All the shape functions derived using minimization approach are validated according to interpolation properties.

1. INTRODUCTION

A finite element approximation of displacement is given by

\[ u(x) \approx u = \sum_i N_i(x)u_i = N(x)u_e \]

where \( N_i \) are element shape or interpolation functions, \( u_i \) are nodal displacements and the sum ranges over the number of nodes associated with an element.
Shape functions in finite element analysis depend on the dimensionality of the problem and type of elements used for discretization of the problem domain. These must also satisfy continuity requirements depending on the underlying PDE and form (strong or weak) of finite element formulation. Normally these are chosen as polynomial functions. In addition, the shape functions must satisfy the following properties.

1.1. **Partition of Unity.** Partition of Unity (by letting $u = 1$ in equation (1))

$$\sum_{k=1}^{n} N_k = 1.$$ 

1.2. **Kronecker Delta Property.**

$$N_i(x_k) = \delta_{ik},$$

where $n$ is the number of nodes in an element, $x_k$ represents the coordinates of $k^{th}$ node.

Convergence motivated some of the requirements of shape functions, since the improvement of the finite element solution to approach the analytical solution of the mathematical model in question requires mesh refinement. The compatibility requirement ensures the interpolation to be able to allow the field of displacements to be continual and derivable inside the element and across the element boundary ([5, 7, 10]). As pointed by ([3, 8, 9, 11, 12]) “The shape functions should provide displacement continuity between elements. Physically, these guarantees that no material gaps appear as the elements deforms. As the mesh is refined, such gaps would multiply and may absorb or release spurious energy.”

The completeness requirement is satisfied if the shape functions possess partition of unity property and the element is compatible [8].

There are different types of shape functions and several ways of deriving them which can be found in many finite element literatures ([1, 2, 4, 6, 13]). Shape functions forms the central part of any finite element formulation, which makes their derivation to be very important. In this paper, the derivation of serendipity shape functions is to be constructed via unconstrained optimization.
2. Shape Function

Shape function is an integral part of finite element method and hence, the required investigation. The efficiency of any finite element method depends to a large extent on the shape function.

2.1. Shape Function for 2-D Rectangular Elements.

2.1.1. Linear element. Consider the linear rectangular element in fig 1 below, the shape function corresponding to the points $P_1$, $P_2$, $P_3$ and $P_4$ can be obtained using the following equation or interpolation

$$u_i(\xi, \eta) = c_1 + c_2 \xi + c_3 \eta + c_4 \xi \eta$$

where $c_1$, $c_2$, $c_3$ and $c_4$ are constants, $u_1$, $u_2$, $u_3$ and $u_4$ are nodal solutions respectively and $\xi$, $\eta$ are the natural coordinates.

![Figure 1. 4 nodes linear element](image)

Calculating the function $u$ at each point gives

$$u_1 = c_1 - c_2 - c_3 + c_4$$
$$u_2 = c_1 + c_2 - c_3 - c_4$$
$$u_3 = c_1 + c_2 + c_3 + c_4$$
$$u_4 = c_1 - c_2 + c_3 - c_4$$

The coefficient of $c_1$, $c_2$, $c_3$ and $c_4$ can be written in matrix form as

$$A = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$
and we get

\[ \mathbf{c} = \mathbf{A}^{-1} \mathbf{u} \]

which gives

\[ u(\xi, \eta) = u_1(\frac{1}{4}(1-\xi)(1-\eta)) + u_2(\frac{1}{4}(\xi+1)(1-\eta)) + u_3(\frac{1}{4}(1-\xi)(\eta+1)) + u_4(\frac{1}{4}(\xi+1)(\eta+1)) \]

\[
\begin{align*}
N_1(\xi, \eta) &= \frac{1}{4}(\xi - 1)(1 - \eta) \\
N_2(\xi, \eta) &= \frac{1}{4}(\xi + 1)(1 - \eta) \\
N_3(\xi, \eta) &= \frac{1}{4}(1 - \xi)(\eta + 1) \\
N_4(\xi, \eta) &= \frac{1}{4}(\xi + 1)(\eta + 1)
\end{align*}
\]

2.2. Quadratic Element. By using the quadratic Ansatz

\[ u(\xi, \eta) = c_1 + c_2 \xi + c_3 \eta + c_4 \xi^2 + c_5 \xi \eta + c_6 \eta^2 + c_7 \xi^2 \eta + c_8 \xi \eta^2 \]

**Figure 2.** 8 nodes quadratic element

using the same approach by generating and finding the inverse of the matrix from the polynomials for various values of \( \xi \) and \( \eta \) and upon substituting for the constants in the polynomials and factorizing, we obtain the shape functions as below,
An $8 \times 8$ matrix is obtained by substituting the points $p_1, \ldots, p_8$ in equation (6), which after same procedure as in linear element gives the following shape functions

\[
\begin{align*}
N_1(\xi, \eta) &= -\frac{1}{4}(1 - \xi)(1 - \eta)(1 + \xi + \eta) \\
N_2(\xi, \eta) &= -\frac{1}{4}(1 + \xi)(1 - \eta)(1\xi + \eta) \\
N_3(\xi, \eta) &= -\frac{1}{4}(1 + \xi)(1 + \eta)(1 - \xi - \eta) \\
N_4(\xi, \eta) &= -\frac{1}{4}(1 - \xi)(1 + \eta)(1 + \xi - \eta) \\
N_5(\xi, \eta) &= -\frac{1}{2}(1 - \xi)(1 + \xi)(1 - \eta) \\
N_6(\xi, \eta) &= -\frac{1}{2}(1 + \xi)(1 + \eta)(1 - \eta) \\
N_7(\xi, \eta) &= -\frac{1}{2}(1 - \xi)(1 + \xi)(1 + \eta) \\
N_8(\xi, \eta) &= -\frac{1}{2}(1 - \xi)(1 + \eta)(1 - \eta)
\end{align*}
\]

3. Minimization Method

Minimization is the act of reducing the total user-defined error. While maximization is the reverse of minimization, both maximization and minimization are called optimization.

Shape functions can also be derived via unconstrained optimization. In this section, minimization approach was used to generate the shape functions. Four nodes rectangular element will be used in this section. Using equation (2), the objective function can be formulated as,

\[
\text{minimize} \quad f(\mathbf{a}) = (N_1(\mathbf{a}, p_1) - 1)^2 + (N_1(\mathbf{a}, p_2))^2 + (N_1(\mathbf{a}, p_3))^2 + (N_1(\mathbf{a}, p_4))^2 \\
+ (N_2(\mathbf{a}, p_1))^2 + (N_2(\mathbf{a}, p_2) - 1)^2 + (N_2(\mathbf{a}, p_3))^2 + (N_2(\mathbf{a}, p_4))^2 \\
+ (N_3(\mathbf{a}, p_1))^2 + (N_3(\mathbf{a}, p_2) - 1)^2(N_3(\mathbf{a}, p_3) - 1)^2 + (N_3(\mathbf{a}, p_4) - 1)^2 \\
+ (N_4(\mathbf{a}, p_1))^2 + (N_4(\mathbf{a}, p_2))^2 + (N_4(\mathbf{a}, p_3))^2 + (N_4(\mathbf{a}, p_4) - 1)^2,
\]
where,
\[ p_i = (x_i, y_i) \]
and
\[
N_1(a, x, y) = a_1 + a_2 x + a_3 y + a_4 xy \\
N_2(a, x, y) = a_5 + a_6 x + a_7 y + a_8 xy \\
N_3(a, x, y) = a_9 + a_{10} x + a_{11} y + a_{12} xy \\
N_4(a, x, y) = a_{13} + a_{14} x + a_{15} y + a_{16} xy
\]
and \( p_1 = (-1, -1), p_2(1, -1), p_3(1, 1), p_4(-1, 1) \) are the interpolation points, \((x, y)\) is represented by \( \xi, \eta \) and \( N_i(a, x, y) \) are the polynomial shape functions which coefficients \( a_i \) are to be determined. This minimization problem can be solved using the MATLAB code as below,

```matlab
% clear all
% clc;
[a,fval, exitflag, output] = fminunc (@objfun, a0)
function f = objfun(a)
  \% \textbf{a} = [a_1, a_2, \ldots, a_{16}];
  p_1 = [-1,-1];
  p_2 = [1,-1];
  p_3 = [1,1];
  p_4 = [-1,1];
  \%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  f = (N_1(a,p_1)-1)^2+(N_1(a,p_2))^2+(N_1(a,p_3))^2+(N_1(a,p_4))^2+...
  (N_2(a,p_1))^2+(N_2(a,p_2)-1)^2+(N_2(a,p_3))^2+(N_2(a,p_4))^2+...
  (N_3(a,p_1))^2+(N_3(a,p_2))^2+(N_3(a,p_3)-1)^2+(N_3(a,p_4))^2+...
  (N_4(a,p_1))^2+(N_4(a,p_2))^2+(N_4(a,p_3))^2+(N_4(a,p_4)-1)^2
end
\%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function f = N_1(a,x,y)
  f = a_1+a_2*x+a_3*y+a_4*x*y;
end
```
function f = N_2(a,x,y)
f = a_5+a_6*x+a_7*y+a_8*x*y;
end

function f = N_3(a,x, y)
f = a_9+a_{10}*x+a_{11}*y+a_{12}*x*y;
end

function f = N_4(a,x, y)
f = a_{13}+a_{14}*x+a_{15}*y+a_{16}*x*y
end

3.1. Output.

\[
a = \begin{bmatrix}
0.2500 & -0.2500 & -0.2500 & 0.2500 \\
0.2500 & 0.2500 & -0.2500 & -0.2500 \\
0.2500 & 0.2500 & 0.2500 & 0.2500 \\
0.2500 & -0.2500 & 0.2500 & -0.2500
\end{bmatrix},
\]

Then the solution of \(a\) in matrix form as shown above is transformed into row vector as below,

\[
a = \begin{bmatrix}
a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & a_8, & a_9, & a_{10}, & a_{11}, & a_{12}, & a_{13}, & a_{14}, & a_{15}, & a_{16}
\end{bmatrix}
\]

The output is the coefficients \(a_1\) to \(a_{16}\), substituting these values in equation (11) gives the required shape functions as shown in equation (6)

3.2. Irregular Shapes. In the case of irregular shapes, for example, quadrilateral element with six(6) nodes as shown in the figure 3 below, the minimization code is improved upon by taking care of the irregularity. The minimization method is very flexible

3.2.1. Improvement by integration. This improvement is done by adding additional functions to the objective function. The square of the difference between the integration of the any variable term of the polynomial i.e \(u\) and the integration of the sum of all the shape functions of the element i.e the total area covered by the element multiply with their respective coordinates is added to the objective function to become the new objective function. In the case of six
From equation (1), \( u = \sum_i N_i(\xi, \eta)u_i \), where \( u \) can be any function and \( N_i(\xi, \eta) \) is the set of shape functions. The area covered by such element is the sum of all the shape function as shown below;

\[
u = \sum_i N_i(\xi, \eta)u_i,
\]

where \( n \) is the number of nodes. Thus, the function \( u \) can be choosing from the following polynomials, that is, \( 1, \xi, \eta, \xi\eta, \xi^2, \eta^2, \xi\eta^2 \). Choosing \( u = 1 \), equation (13) becomes,

\[
1 = \sum_i N_i(\xi, \eta).
\]
Integrating both sides of equation (14) we have,
\[
\int_{-1}^{1} \int_{-1}^{1} 1 d\xi d\eta = \int_{-1}^{1} \int_{-1}^{1} \sum N_i(\xi, \eta) d\xi d\eta \\
\int_{-1}^{1} \int_{-1}^{1} 1 dA = \int_{-1}^{1} \int_{-1}^{1} \sum N_i \ dA.
\]
Let
\[
w^k = \sum_i N_i w^k_i,
\]
where,
\[
w^k \in W
\]
and
\[
W^k = [1 \ \xi \ \eta \ \xi \eta \ \xi^2 \ \eta^2 \ \xi^2 \eta \ \xi^2 \eta^2]
\]
The new objective function is as given below
\[
f = f + \sum_k \left( \int_{-1}^{1} \int_{-1}^{1} w^k dA - \int_{-1}^{1} \int_{-1}^{1} \sum N_i w^k_i dA \right)^2
\]
is added to the objective function for improvement. Likewise, for any \( u \) in the sequence \( 1, \xi, \eta, \xi \eta, \eta^2, \xi^2 \eta, \xi \eta^2 \) which in turn gives
\[
\left( \int_{-1}^{1} \int_{-1}^{1} \xi dA - 0 \right)^2, \left( \int_{-1}^{1} \int_{-1}^{1} \eta dA - 0 \right)^2, \left( \int_{-1}^{1} \int_{-1}^{1} \xi \eta dA - 0 \right)^2,
\]
\[
\left( \int_{-1}^{1} \int_{-1}^{1} \xi^2 dA - 4/3 \right)^2, \left( \int_{-1}^{1} \int_{-1}^{1} \eta^2 dA - 4/3 \right)^2, \left( \int_{-1}^{1} \int_{-1}^{1} \xi^2 \eta dA - 0 \right)^2,
\]
\[
\left( \int_{-1}^{1} \int_{-1}^{1} \xi \eta^2 dA - 0 \right)^2.
\]

3.2.2. Improvement through collocation points. Using equation (2) once again, the sum of all the polynomial shape functions must be equal to unity, that is
\[
\sum_{i=1}^{n} N_i(\xi, \eta) = 1.
\]
This equation shows that the sum of all the constant terms is equal to unity while the sum of the variable terms in each of the shape functions is equal zero. To further improve the shape functions, the square of the sum of the coefficient of each term in each of the polynomial shape functions is further added to the objective function.
In the case of the 6-nodes, 8 terms polynomial, the polynomial shape functions are given below

\[ N_1(\xi, \eta) = a_1 + a_2\xi + a_3\eta + a_4\xi\eta + a_5\xi^2 + a_6\eta^2 + a_7\xi\eta^2 + a_8\xi^2\eta \]

\[ N_2(\xi, \eta) = a_9 + a_{10}\xi + a_{11}\eta + a_{12}\xi\eta + a_{13}\xi^2 + a_{14}\eta^2 + a_{15}\xi^2\eta + a_{16}\xi\eta^2 \]

\[ N_3(\xi, \eta) = a_{17} + a_{18}\xi + a_{19}\eta + a_{20}\xi\eta + a_{21}\xi^2 + a_{22}\eta^2 + a_{23}\xi^2\eta + a_{24}\xi\eta^2 \]

\[ N_4(\xi, \eta) = a_{25} + a_{26}\xi + a_{27}\eta + a_{28}\xi\eta + a_{29}\xi^2 + a_{30}\eta^2 + a_{31}\xi^2\eta + a_{32}\xi\eta^2 \]

\[ N_5(\xi, \eta) = a_{33} + a_{34}\xi + a_{35}\eta + a_{36}\xi\eta + a_{37}\xi^2 + a_{38}\eta^2 + a_{39}\xi^2\eta + a_{40}\xi\eta^2 \]

\[ N_6(\xi, \eta) = a_{41} + a_{42}\xi + a_{43}\eta + a_{44}\xi\eta + a_{45}\xi^2 + a_{46}\eta^2 + a_{47}\xi^2\eta + a_{48}\xi\eta^2 \]

Thus, the coefficients of each term are generated as shown below where \( i \) represents the nodes ranging from 1-6 and \( C \) denotes the constant terms.

<table>
<thead>
<tr>
<th>C</th>
<th>( a(1 + (i - 1) \times 8) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \xi )</td>
<td>( a(2 + (i - 1) \times 8) )</td>
</tr>
<tr>
<td>( \eta )</td>
<td>( a(3 + (i - 1) \times 8) )</td>
</tr>
<tr>
<td>( \xi\eta )</td>
<td>( a(4 + (i - 1) \times 8) )</td>
</tr>
<tr>
<td>( \xi^2 )</td>
<td>( a(5 + (i - 1) \times 8) )</td>
</tr>
<tr>
<td>( \eta^2 )</td>
<td>( a(6 + (i - 1) \times 8) )</td>
</tr>
<tr>
<td>( \xi^2\eta )</td>
<td>( a(7 + (i - 1) \times 8) )</td>
</tr>
<tr>
<td>( \xi\eta^2 )</td>
<td>( a(8 + (i - 1) \times 8) )</td>
</tr>
</tbody>
</table>

Hence,

\[ s(C) = 1, s(\xi) = 0, s(\eta) = 0, s(\xi^2) = 0, s(\eta^2) = 0, s(\xi^2\eta) = 0, s(\xi\eta^2) = 0 \]

where \( s(*) \) is the sum of the coefficients \( a_i' s \). This can be added to the objective functions as in (9) in the form

\[ f = f + (s(C) - 1)^2 + (s(\xi))^2 + (s(\eta))^2 + (s(\xi^2))^2 + (s(\eta^2))^2 + (s(\xi^2\eta))^2 + (s(\xi\eta^2))^2 \]

Adding the two improvements to the equation gives better shape functions. The six (6) nodes element in fig 3 above gives the following shape functions after the modification of the code by addition of the two improvements that is section (3.2.1) and (3.2.2) respectively. With factorization, the shape function in Table
Table 2. Element shape function

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>ξ</th>
<th>η</th>
<th>ξη</th>
<th>ξ²</th>
<th>η²</th>
<th>ξ²η</th>
<th>η²ξ²</th>
</tr>
</thead>
<tbody>
<tr>
<td>N₁</td>
<td>0</td>
<td>−1/8</td>
<td>0</td>
<td>1/4</td>
<td>1/4</td>
<td>0</td>
<td>−1/4</td>
<td>−1/8</td>
</tr>
<tr>
<td>N₂</td>
<td>0</td>
<td>1/8</td>
<td>0</td>
<td>−1/4</td>
<td>1/4</td>
<td>0</td>
<td>−1/4</td>
<td>1/8</td>
</tr>
<tr>
<td>N₃</td>
<td>0</td>
<td>1/8</td>
<td>0</td>
<td>1/4</td>
<td>1/4</td>
<td>0</td>
<td>1/4</td>
<td>1/8</td>
</tr>
<tr>
<td>N₄</td>
<td>0</td>
<td>−1/8</td>
<td>0</td>
<td>−1/4</td>
<td>1/4</td>
<td>0</td>
<td>1/4</td>
<td>−1/8</td>
</tr>
<tr>
<td>N₅</td>
<td>1/2</td>
<td>0</td>
<td>−1/2</td>
<td>0</td>
<td>−1/2</td>
<td>0</td>
<td>1/2</td>
<td>0</td>
</tr>
<tr>
<td>N₆</td>
<td>1/2</td>
<td>0</td>
<td>1/2</td>
<td>0</td>
<td>1/2</td>
<td>0</td>
<td>−1/2</td>
<td>0</td>
</tr>
</tbody>
</table>

2 is re-written as

\[ N₁ = -\frac{1}{8} \xi (\eta - 1)(\eta - 1 + 2\xi), \]
\[ N₂ = \frac{1}{8} \xi (\eta - 1)(\eta - 1 - 2\xi), \]
\[ N₃ = \frac{1}{8} \xi (\eta - 1)(\eta - 1 - 2\xi), \]
\[ N₄ = -\frac{1}{8} \xi (\eta + 1)(\eta + 1 - 2\xi), \]
\[ N₅ = \frac{1}{2}(\xi - 1)(\xi + 1)(\eta - 1), \]
\[ N₆ = -\frac{1}{2}(\xi - 1)(\xi + 1)(\eta + 1). \]

4. Validation

It is obvious from the shape function above that both partition of unity and kronecker delta as the major two conditions that any set of shape function must obey are well satisfied as seen below.

\[ N = -\frac{1}{8} \xi (\eta - 1)(\eta - 1 + 2\xi) + \frac{1}{8} \xi (\eta - 1)(\eta - 1 - 2\xi) + \frac{1}{8} \xi (\eta - 1)(\eta - 1 - 2\xi) - \frac{1}{8} \xi (\eta + 1)(\eta + 1 - 2\xi) + \frac{1}{2}(\xi - 1)(\xi + 1)(\eta - 1) - \frac{1}{2}(\xi - 1)(\xi + 1)(\eta + 1) \]

\[ N = N₁ + N₂ + N₃ + N₄ + N₅ + N₆ = 1 \]

Also satisfy kronecker delta property, \( Nᵢ(ᵢⱼ) = δᵢⱼ \) as shown below

\[ N₁(−1, −1) = 1, \text{ but zero at any other coordinate} \]
\[ N₂(1, −1) = 1, \text{ but zero at any other coordinate} \]
\( N_3(1, 1) = 1 \), but zero at any other coordinate
\( N_4(-1, 1) = 1 \), but zero at any other coordinate
\( N_5(0, -1) = 1 \), but zero at any other coordinate
\( N_6(0, 1) = 1 \), but zero at any other coordinate.

5. Conclusion

There is a major shortcoming associated with the matrix inversion method used in the first part of this paper as pointed out by [13]. This shortcoming is the non-feasibility of the inverse of the matrix generated during the implementation of the method. The minimization method, however, has circumvented this shortcoming and can also produce many sets of shape functions which can be customized to user requirement. However, the method still needs further improvements especially those with irregular elements such as transition elements. It should be noted that the method proposed in this work is also applicable to Lagrange shape functions.

References


DEPARTMENT OF MATHEMATICAL SCIENCES
FEDERAL UNIVERSITY LAFIA, NIGERIA
PMB 146, LAFIA, NIGERIA

DEPARTMENT OF MATHEMATICAL SCIENCES
UNIVERSITI TEKNOLOGI MALAYSIA
81310, JOHOR BAHRU, SUDAI, MALAYSIA
Email address: collins.akeremale@science.fulafia.edu.ng

DEPARTMENT OF MATHEMATICAL SCIENCES
UNIVERSITI TEKNOLOGI MALAYSIA
81310, JOHOR BAHRU, SUDAI, MALAYSIA
Email address: olaiju@yahoo.com

DEPARTMENT OF MATHEMATICAL SCIENCES
FEDERAL POLYTECHNIC ILARO, NIGERIA
PMB 50, ILARO, NIGERIA
Email address: s.h.yeak@utm.my