A Metric for Mesh Quality in Finite Element Analysis with Isoparametric Elements

Marcal, P.V.¹, Rainsberger R.², *Fong, J.T.³

Abstract Much effort has been expended in generating meshes a priori as well as post analysis to obtain good analysis results. In some recent work, the authors have provided methods for ensuring good results by using convergence at the limit [1,2,3]. Yet the method of determining a good mesh with elements with good ‘aspect ratios’ remain mired in hearsay and individual practices. The problem is that it has been difficult to measure the effect of a mesh without a rational metric for the mesh quality. There have been guidelines proposed to measure the quality of a mesh a posteori. Olivera [4] pointed out that the optimum mesh resulted in orthogonal contours of specific energy. McNeice and Marcal[5], optimized the mesh by minimizing the Potential Energy by varying the mesh coordinates. It was found to be a nonlinear problem. There are two factors that influence the quality of the results in a FEA. The first is the truncation error determined by the polynomial function selected for the element. The second is the quality of a mesh, we shall call this the mesh metric. We will review briefly, the errors due to the first factor. Then we will propose a mesh metric for isoparametric elements based on the determinant of the Jacobian. We show that the use of isoparametric elements results also in determining the truncation errors in the element. This means that the value of the metric must be kept constant within the element and also kept uniform throughout the mesh. Finally we show some results for the metric and point out other uses for the mesh metric in large deformation nonlinear analysis.

Keywords: Finite Elements, Mesh quality, Error Analysis, Mesh Metric

¹MPACT Corp., Oak Park, CA, USA pedrovmarcal@gmail.com
²XYZ Corp., Pleasant Hills, CA USA
³NIST, Gaithersburg, MD, USA

*Contribution of the U.S. National Institute of Standards & Technology. Not subject to copyright
Theoretical Considerations  Most of the discussions on accuracy of the finite element method start with the completeness of the element representation using the adaptation of the Pascal Triangle [6], which in turn links back to errors of approximation by truncation of the terms in a Taylor expansion. We have found that the characterization in terms of the Taylor expansion is too coarse because for any order of approximation this metric does not take into account the value of the contribution of some of the polynomials towards reducing the error. The discussion follows the truncation errors as given by Crandall[7]. Here we explain the expansion in two dimensions using the Pascal Triangle and then generalize it to three dimensions.

The expansion for a square (may be isoparametric) is given by all the terms included by the diamond pattern as shown in Fig.1.

So for a quadratic square we have the following formula:

\[ u = [ 1, x, xy, y, x^2, x^2y, xy^2, y^2 ] \{a\} + h^3(0) \]  \hspace{1cm} (1)

where \( u \) is a single degree of freedom in an element, \( \{a\} \) are nine undetermined coefficients, and \( h^3(0) \) is the order of the truncation error. The corresponding element in 3D is the 27-node hexahedron.

![The Pascal Triangle of the Order of Truncation Errors](image)

Fig 1 THE PASCAL TRIANGLE OF THE ORDER OF TRUNCATION ERRORS (AFTER [6], FIG. 5.5)

The order of the truncation error can be read off from the Pascal Triangle as the lowest order that is not included in the Pascal Diamond for the square elements and by the triangle for simplex elements. By this we can write a similar expression for the quadratic triangle as

\[ u = [ 1, x, xy, y, x^2, y^2 ] \{a\} + h(0) \]  \hspace{1cm} (2)

where \( \{a\} \) are six undetermined coefficients, and \( h(0) \) is the order of the truncation error, e.g. the truncated terms that are linear in \( x \) and \( y \) have been left out in the following terms \( x^3y, x^2y^3, xy^2 \) in (1) have been left out. We note that this is different than the metric based on a Taylor expansion since the truncated term is the highest order term that has been left out, i.e. \( h(0)^2 \).

As shown by Zienkiewicz and Taylor [6], the so-called ‘serendipity’ elements are among the most popular of elements. In the two dimensional case these elements are obtained by dropping the \( xy \) term so that the eight undetermined coefficients \( \{a\} \) result in a truncation error of \( h(0) \) compared to that of \( h^3(0) \) for the full nine terms. The corresponding element does not have the center node. We
shall refer to the elements with the full series expansions as Taylor elements in contrast to the Serendipity elements.

The Pascal Triangle can be generalized to the third dimension by considering a third axis for z. Then the terms to be included or excluded are obtained by multiplying the z terms in the third axis by the respective triangle or diamond for the first two axes as shown by the Fig. 1. All terms are obtained by cyclic permutation. As an example, we consider the case for the quadratic serendipity element in three directions. Here the excluded seven terms are

\[ xy, yz, zx, x^2yz, x^2y, xyz, x yz \]  

This is a new result from the Pascal Triangle, for the 20-node hexa element and because of the missing linear terms in x,y,z the a truncation error is \( h(0) \) and not \( h^2(0) \) as previously thought.

This may be compared to the full expansion quadratic element resulting in a 27-node hexa with a truncation error of \( h^3(0) \). This element has three layers of nine nodes each with the middle layer at the mid-plane. Though the value of the truncation errors are problem dependent, we can conclude qualitatively that the serendipity elements are a poor trade off because they result in a lower order of truncation error for the sake of approximately dropping of a quarter of the terms in the case of the 20-node hexa. This approximation adopted for convenience and reduction of computing in the early FEA days is now so widely accepted that many analysts are surprised by its poor performance.

We introduce the method of undetermined coefficients. There are as many coefficients as there are nodes.

For the 9 node quadrilateral, quad9,

\[ u = l, 1, x, xx, y, xy, xyy, yy, xyy \{a\} \]  

where \( u \) is the displacement in the x direction, and \( \{a\} \) is the vector of undetermined coefficients.

Eq. (1) describes the displacement in the element.

Using the three displacement at the nodes,

\[ \{u\}=[F(x,y)]\{a\} \]  

where \([F(x, y)]\) represents the matrix formed by substitution of the x, y values at the nodes.

By Inversion of this matrix we have

\[ \{a\}=[F(x,y)]^{-1}\{u\} \]

The matrix \([F]\) remains unchanged for displacements in the y directions so we can write the matrix equation that includes the displacements \( u \) and \( v \).

\[ \{a_0, a_1\}=[F(x,y)]^{-1}\{u, v\} \]

We generalize the vector values by just writing

\[ [a]=[F(x,y)]^{-1}[u] \]
Substituting in eq. (9) we obtain the interpolating function for the value at any point in the element.

\[ x = \left[ f(x,y,z) \right] \left[ f(x,y,z) \right]^{-1} (x) = [N][x] \] 

(16)

If we examine the Pascal Triangle for this element, we note that all four edges are defined by quadratic values in x or y. For the interpolation at each edge due to different shapes in the adjacent elements, we require that the edge be a straight line. Otherwise holes will open in our displacement field. This is too restrictive for our analysis, so we choose to map our displacements to a rectangular parametric space as proposed by Taig[6] and Irons[7]. In particular we choose the same function as in eq. 9 to represent the mapping of our displacements and coordinates. We call this the isoparametric plane and we have

\[ x = \left[ N(\xi \eta) \right][x] \] 

(17)

and

\[ u = \left[ N(\xi \eta) \right][u] \] 

(18)

We further simplify the definition, by giving coordinates +1 and -1 values at the corners.

Therefore \( \xi \) is given a normalized value of -1,0,1 in each axis direction by implicit linear scaling. The unit values represent values of the nodes at the corners of the element. They are the nodal values for a quadratic element.

The resulting scaling alters the interpolation function \( N \) in (17,18) and is compensated for in all operations with the function and in particular when performing integration of the element.

By standard bookwork we can show that the partial differential vectors w.r.t. the isoparametric values and the coordinates are related to the Jacobian matrix \( J \).

\[ \{ \partial N / \partial \xi \} = J \{ \partial N / \partial x \} \] 

(19)

And

\[ \{ \partial N / \partial x \} = J^{-1} \{ \partial N / \partial \xi \} \] 

(20)

Because (20) maps the values required for calculating strains from displacements, the Jacobian matrix and its inverse is an important matrix and it may be measured by its determinant, \( \text{det} J \). We therefore choose \( \text{det} J \) at any point to be the metric for our element representation in both the mesh and its displacement. It carries the same truncation errors as described above.

In its shape we require that \( \text{det} J \) be positive definite, this means that the element should not fold on itself. Now we see that by our isoparametric mapping we have made the shape or aspect ratio of our element in Cartesian coordinates have the same errors as that of our displacements. In this case, we give meaning to our intuition that elements with bad aspect ratios result in bad strain values in our calculation. We could also insist that the isoparametric mapping be applied to any element type with consistent reductions in our undetermined coefficients, so that \( \text{det} J \) becomes a universal metric for measuring the aspect ratio of its element as well as its strains.
The det J may then also be applied to measure the rate of convergence of a solution w.r.t. its mesh size as demonstrated through the logistic function fit in [1,2,3].

Work is progressing on the use of the det J and will be reported below. The results above impinge greatly on the theoretical and practical application of FEA to be considered.

**Program for Jacobian Metric** We have developed a python program to calculate the det J for all the nodes in each element. The values are divided by the volume of an element to compensate for the choice of nodal coordinates in the isoparametric space. This volume is normalized by the smallest length in any axis direction in order to be unaffected by any perfectly cube element. Finally we have averaged the results for each node in the usual way and written the results in a format that can be plotted by MpaveMesh.

In order to look at some preliminary results, we have modeled a square beam (100 X 10 X 10) with four equal elements. On one of its midside edge nodes, we have increased its x value to 55. We present the effect of this change in the hexa8, hexa27 contour plot of det J, respectively.

![Fig. 2 plot for det J for hexa8 beam](image)
Because of all the smoothing, the plot for hexa8 shows a smooth transition. The value of det_J increases with relative shortening of the element because of the reduction in the scaling volume. The plot for the hexa27 shows its ability to model more severe distortions without too high a penalty in the value of the metric. The plot shows that the effect of a distortion is spread through the element and not confined to the node, as to be expected by the use of interpolating functions.

Finally, we return to our problem of the barrel vault[1]. We use the data corresponding to a 2 X 2 element mesh. The Jacobian Plot is shown in Fig. 8.
The value of the $\text{det}_J$ is fairly constant indicating a good mesh. Even though the value of the $\text{det}_J$ is reduced by 1,000. This indicates that it is a thin shell. The steady value indicates that the results have the same truncation errors and so may be relied upon. Finally, we compare the mesh metric for a sphere modeled with ten node tetra10 compared with that using hexa27 elements. A constant size was suggested for both meshes.

![Fig. 5 Det_J for tetra10](image1)

We see that the metric varies with an order difference of 100 for tetra10. Whereas the hexa27 mesh is essentially uniform (factor of 2). This is in fact partly due to the projection method used in TrueGrid, which allows uniform isoparametric elements to be generated.

![Fig. 6 Det_J for hexa27](image2)
Conclusions A mesh metric, viz. the Determinant of the Jacobian has been demonstrated theoretically and numerically. The metric measures the uniformity of the mesh within the element as well as throughout the mesh. Because the metric measures the uniformity between elements it may be used as a gauge of when a mesh requires to be re-meshed during problems of large deformation.

References


