2 Gradient and Integral of Interpolations

1.1 Introduction

The previous chapter reviewed the mathematical background for the theoretical process for evaluating the gradient of a quantity that is interpolated employing parametric coordinates. Likewise, it covered the integration of such polynomials. In finite element analysis it is necessary to do both of those things in every element in the mesh. Today, it is not rare for a mesh to include thousands or millions of elements. Therefore, to accomplish practical finite element solutions it is necessary to understand how to automate both of those types of calculations. Here, the theory of the previous chapter will be extended to mesh dependent implementation using totally numerical processes. Here most of the integrals will relate to scalar integrals like the geometric properties of shapes that are covered in introductory courses on calculus.

1.2 Numerical evaluation of the Jacobian matrix

In practical finite element applications it is always necessary to numerically form the geometry transformation Jacobian matrix, $J^e$, its determinant, $J^e$, and its inverse, $J^{-1}$. The determinant is required to define the differential physical volume to the non-dimensional differential parametric volume, while the inverse matrix is required to calculate the physical gradient components from the parametric gradient. The Jacobian matrix is calculated by a simple matrix product of two matrices, at any local parametric point inside an element. Part of the necessary data are the numerical values of the physical spatial coordinates of the nodes on the current element. The first matrix comes from numerically evaluating the local parametric derivatives of the interpolation functions that define the geometry mapping, at a point inside the element. That point is most commonly one of the tabulated quadrature points; because it can be proved that the most accurate location to evaluate a physical gradient happens to usually be a quadrature point.

The interpolation for the geometry mapping does not have to be the same as the interpolation for and unknown quantity. Each physical spatial coordinate interpolation function will be denoted by the row matrix, \( G(r,s,t) \), while a scalar unknown will be interpolated by the row matrix, \( H(r,s,t) \), and a vector unknown by the rectangular matrix \( N(r,s,t) \). Usually, the coordinates and scalar unknowns are interpolated by the same function, so then \( G(r,s,t) \) is the same as \( H(r,s,t) \). The vector interpolation \( N(r,s,t) \) always contains a large percentage of zeros and multiple copies of the functions in \( H(r,s,t) \), but can depend on other items, such as a radial position.

Having picked a type of the parametric element (the parametric shape and its number of nodes) the geometry interpolation functions can be looked up in a library of parametric functions. Since all the common parametric interpolation functions are know in terms of the parametric coordinates \( (r,s,t) \) that also means that all of the parametric local derivatives are known and can be stored in the same library of functions. Those parametric derivatives are denoted here as the vector (column vector) with as many rows as the dimension of the parametric space, \( n_p \):

\[
\partial \phi = \left\{ \frac{\partial \phi}{\partial r}, \frac{\partial \phi}{\partial s}, \frac{\partial \phi}{\partial t} \right\}
\]
Therefore, the parametric local derivatives of any interpolated item, say \( u(r, s, t) \), is
\[
\frac{\partial}{\partial \mathbf{u}} \mathbf{u}^e = \left\{ \frac{\partial (\ )/\partial r}{\partial H(r, s, t)/\partial r} \; \frac{\partial (\ )/\partial s}{\partial H(r, s, t)/\partial s} \; \frac{\partial (\ )/\partial t}{\partial H(r, s, t)/\partial t} \right\} \{\mathbf{u}^e\}.
\]

If \( u \) is replaced with the physical spatial coordinate \( x \), and \( H(r, s, t) \) is replaced with the geometry interpolation \( G(r, s, t) \), then this product becomes the first column of the geometric Jacobian matrix:
\[
\left\{ \frac{\partial x/\partial r}{\partial G(r, s, t)/\partial r} \; \frac{\partial x/\partial s}{\partial G(r, s, t)/\partial s} \; \frac{\partial x/\partial t}{\partial G(r, s, t)/\partial t} \right\} \{\mathbf{x}^e\}.
\]

Replacing the right column data with a rectangular array constructed by adding a column of the \( n_n \) coordinates \( y^e \) and a column for the \( z^e \) coordinates, the full geometric Jacobian matrix, for that specific element, is obtained:
\[
J^e = \begin{bmatrix} \frac{\partial x/\partial r}{\partial G(r, s, t)/\partial r} & \frac{\partial y/\partial r}{\partial G(r, s, t)/\partial s} & \frac{\partial z/\partial r}{\partial G(r, s, t)/\partial t} \\ \frac{\partial x/\partial s}{\partial G(r, s, t)/\partial r} & \frac{\partial y/\partial s}{\partial G(r, s, t)/\partial s} & \frac{\partial z/\partial s}{\partial G(r, s, t)/\partial t} \\ \frac{\partial x/\partial t}{\partial G(r, s, t)/\partial r} & \frac{\partial y/\partial t}{\partial G(r, s, t)/\partial s} & \frac{\partial z/\partial t}{\partial G(r, s, t)/\partial t} \end{bmatrix} = \begin{bmatrix} \partial G(r, s, t)/\partial r \\ \partial G(r, s, t)/\partial s \\ \partial G(r, s, t)/\partial t \end{bmatrix} \begin{bmatrix} x^e \\ y^e \\ z^e \end{bmatrix}.
\]

Note that the element Jacobian matrix, \( J^e \), is generally a rectangular matrix and is only a square (and invertible) matrix when the dimension of the parametric space, \( n_p \), equals the dimension of the physical space, \( n_s \). In other words, the element Jacobian matrix is square except when the element is a line element on a two- or three-dimensional curve, or if it is an area element on a three-dimensional surface. The vast majority of practical finite element calculations utilize a square Jacobian matrix, but there are times (like convection on a non-flat surface) when the rectangular format must be employed.

In the above equation, \( (r, s, t) \) represents any local point in the element where it is desired to evaluate the Jacobian matrix. Usually, the point is a tabulated quadrature point. To numerically evaluate the \( \partial G \) matrix, at a specific point, the local coordinates are provided as arguments to the function that contains the parametric interpolation equations, and the equations for their parametric derivatives. Those data, along with the parametric element type, that defines its spatial dimension, \( n_p \), and the number of nodes, \( n_n \), provide for its automatic evaluation. The rectangular array of element nodal coordinates is \textit{gathered} as a set of input numbers. Then, a simple numerical matrix multiplication provides all the numerical entries of the element Jacobian matrix, \( J^e \). Having a current numerical value for a square element Jacobian matrix, \( J^e \), it is straightforward to numerically evaluate its determinant, \( J^e \), and its inverse matrix, \( J^{-1} \). Recall that the determinant relates a physical differential volume to the corresponding parametric differential volume, at the point where it was computed. In simplex elements, (straight lines, straight edge triangles and tetrahedral in physical space) the determinant of the element Jacobian is constant everywhere in the element. Constant determinants can occur in other special cases, like straight edge rectangles and bricks with their edges parallel to the physical axes. Of course, a constant determinant implies that the Jacobian matrix, and its inverse, are also constant in such special element geometries in physical space.

Consider programming aspects of the above numerical matrix evaluation. To obtain the necessary geometric data there must be a function that reads and stores all the physical coordinates of all the nodes in the mesh. The list of coordinates defines the number of physical spatial dimensions, \( n_s \). There must also be another function that reads and stores the node connection list for each and every element in the mesh. That list of connections defines the number of nodes on each element, \( n_n \). For the element of interest, there must be another function that will gather the list of nodes on that particular element. That list (a vector subscript in programming terminology) is then used to extract the sub-set of coordinates on that particular element:
The concept of gathering those local coordinate data is illustrated in Figure 2.1, for a one dimensional mesh.

Those three vectors are substituted into the three columns of the rightmost matrix in the Jacobian matrix product. The numerical values of the left matrix are inserted by another function that evaluates the $\partial \mathbf{G}$ matrix at the specified local parametric point, $(r, s, t)$. Finally, the numerical matrix multiplication is executed to create the numerical values in the element Jacobian matrix.

Having a square element Jacobian matrix allows the physical gradient of any quantity to be numerically evaluated, at the specified parametric point, by inverting the Jacobian matrix and multiplying it times the parametric gradient of the quantity:

$$\partial_{\Omega}^e u = \mathbf{J}^{-1} \mathbf{\partial u}$$
2 Gradient and Integral of Interpolations

\[ \begin{align*}
(\frac{\partial u}{\partial x})^e & = J^{-1} (\frac{\partial u}{\partial \xi}) \\
(\frac{\partial u}{\partial y})^e & = J^{-1} (\frac{\partial u}{\partial \eta}) \\
(\frac{\partial u}{\partial z})^e & = J^{-1} (\frac{\partial u}{\partial \zeta})
\end{align*} \]

Here, the number of components in the gradient vector (and the size of the square matrix to invert) will be the same as the dimension of the physical space, \( n_s = n_p \).

As an example of the Jacobian matrix and physical gradient vector calculations at a point within a type element type consider a four noded \((n_n = 4)\) bi-linear Lagrangian quadrilateral \((n_p = 2)\) element. In unit coordinates, the parametric interpolation functions (also called shape functions) are an incomplete quadratic polynomial in the two-dimensional parametric space (they are missing the \( r^2 \) and \( s^2 \) terms):

\[
H(r, s) = \begin{bmatrix}
H_1(r, s) & H_2(r, s) & H_3(r, s) & H_4(r, s)
\end{bmatrix}
\]

\[
H_1(r, s) = 1 - r - s + rs \\
H_2(r, s) = r - rs \\
H_3(r, s) = rs \\
H_4(r, s) = s - rs
\]

and their parametric derivatives are

\[
\frac{\partial}{\partial \xi} H(r, s) = \begin{bmatrix}
(-1 + s) & (1 - s) & s & -s \\
(-1 + r) & -r & r & (1 - r)
\end{bmatrix}
\]

which is linear in the local coordinates. Thus, the element Jacobian matrix, for this element type, will not be constant, unless its sides happen to be input parallel to the physical axes.

Consider an element with the nodal coordinates of \( x^e^T = [4 \hspace{1em} 7 \hspace{1em} 8 \hspace{1em} 3] \) meters and \( y^e^T = [4 \hspace{1em} 5 \hspace{1em} 10 \hspace{1em} 8] \) meters and pressure values of \( p^e^T = [22 \hspace{1em} 24 \hspace{1em} 30 \hspace{1em} 28] \) N/meters². What is the physical pressure gradient at the center of that element, \((r = 1/2, s = 1/2)\)? To answer that question it is first necessary to numerically evaluate the Jacobian matrix for that element:

\[
J^e(r, s) = \frac{\partial}{\partial \xi} H(r, s)[x^e \hspace{1em} y^e]
\]

\[
J^e(r, s) = \begin{bmatrix}
(-1 + s) & (1 - s) & s & -s \\
(-1 + r) & -r & r & (1 - r)
\end{bmatrix}
\]

\[
\begin{bmatrix}
x_1 & y_1 \\
x_2 & y_2 \\
x_3 & y_3 \\
x_4 & y_4
\end{bmatrix}
\]
2 Gradient and Integral of Interpolations

J.E. Akin

\[ J^e(1/2, 1/2) = \begin{bmatrix} -1/2 & 1/2 & 1/2 & -1/2 \\ -1/2 & 1/2 & 1/2 & 1/2 \end{bmatrix} \text{meters} \]

\[ J^e(1/2, 1/2) = \begin{bmatrix} 4 & 3/2 \\ 0 & 9/2 \end{bmatrix} \text{meters} \]

So the determinant at that point is \( J^e = 18 \text{ meters}^2 \) and the inverse Jacobian matrix is

\[ J^{-1} = \frac{1}{18} \begin{bmatrix} 9/2 & -3/2 \\ 0 & 4 \end{bmatrix} \text{ 1/meter.} \]

The parametric pressure gradient components at this point is

\[ \frac{\partial}{\partial r} p(r, s) = \frac{\partial}{\partial s} H(r, s) p^e \]

\[ \begin{pmatrix} \frac{\partial p}{\partial r} \\ \frac{\partial p}{\partial s} \end{pmatrix}^e = \begin{pmatrix} -s & (1 - s) \\ (1 - r) & -r \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix}^e \]

\[ \begin{pmatrix} \frac{\partial p(1/2, 1/2)}{\partial r} \\ \frac{\partial p(1/2, 1/2)}{\partial s} \end{pmatrix}^e = \begin{pmatrix} -1/2 & 1/2 & 1/2 & -1/2 \\ -1/2 & -1/2 & 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} 9/2 \\ -3/2 \\ 0 \\ 4 \end{pmatrix} = \frac{22}{24} = \frac{2}{3} N/\text{meter}^2 \]

\[ \frac{\partial}{\partial r} p(1/2, 1/2) = \frac{2}{3} N/\text{meter}^2 /1 \]

and the physical pressure gradient components, at the point, are

\[ \vec{v}^e p = \frac{\partial}{\partial (x, y)} p = J^{-1} e (1/2, 1/2) \frac{\partial}{\partial (r, s)} p(1/2, 1/2) \]

\[ \begin{pmatrix} \frac{\partial}{\partial x} p \\ \frac{\partial}{\partial y} p \end{pmatrix}^e = \frac{1}{18} \begin{bmatrix} 9/2 \\ -3/2 \\ 0 \\ 4 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 0 \\ 4 \end{bmatrix} N/\text{meter}^2. \]

Here, the pressure gradient is different at every point in the element. If this quadrilateral was degenerated to a rectangle, parallel to the x-y axes, with the same nodal pressures a special case of a constant Jacobian and constant pressure gradient occurs. Let \( x_2 = (x_1 + \Delta x) = x_3, x_4 = x_1 \) and \( y_2 = y_1, y_3 = y_4 = (y_1 + \nabla y) \), then the Jacobian matrix becomes

\[ J^e(r, s) = \begin{pmatrix} -s & (1 - s) \\ (1 - r) & -r \end{pmatrix} \begin{pmatrix} x_1 & y_1 \\ x_1 + \Delta x & y_1 + \Delta y \end{pmatrix} \]

\[ J^e(r, s) = \begin{pmatrix} \Delta x & 0 \\ 0 & \Delta y \end{pmatrix} \text{meters} \]

which is a constant diagonal Jacobian. Its determinant is \( J^e = \Delta x \Delta y \text{ meters}^2 = A^e = A^e / 1 \) which is the physical area of the element. Constant Jacobian determinants are usually the ratio of the physical element volume over the parametric element volume.
1.3 Numerical evaluation of integrals

For the constant Jacobian element finding the integral of the pressure over the area (its resultant force) for the above element is easy:

\[ F_p^e = \iint_{A^e} p(x,y) \, dA = \iint_{A^e} p(r,s) \, f^e(r,s) \, d\Box \]

\[ F_p^e = \iint_{A^e} H(r,s) p^e \, f^e(r,s) \, d\Box = A^e \iint_{A^e} H(r,s) \, d\Box \, p^e = A^e I_\Box \cdot p^e \]

where the integral of the interpolation functions for the unit coordinate bilinear quadrilateral are

\[ I_\Box = \int_0^1 \int_0^1 H(r,s) \, dr \, ds = \int_0^1 \int_0^1 [(1-r-s+rs)(r-rs)(rs)(s-rs)] \, dr \, ds = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \]

so the resultant pressure force, for a constant Jacobian rectangle is

\[ F_p^e = \frac{A^e}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} p_1^e \\ p_2^e \\ p_3^e \\ p_4^e \end{bmatrix} \]

If the pressure is constant then the force is simply \( F_p^e = 4pA^e / 4 = pA^e \) Newtons.

In the general case, where the Jacobian determinant is not constant, integrals over an element are almost always evaluated by numerical integration. In this example, the resultant force becomes

\[ F_p^e = \iint_{A^e} p(x,y) \, dA = \iint_{A^e} p(r,s) \, f^e(r,s) \, d\Box = \sum_{q=1}^{n_q} p(q, s_q) \, f^e(q, s_q) \, w_q \]

\[ = \sum_{q=1}^{n_q} H(q, s_q) \, p^e \, f^e(q, s_q) \, w_q = \left[ \sum_{q=1}^{n_q} H(q, s_q) \, f^e(q, s_q) \, w_q \right] \, p^e \]

\[ 1 \times 1 = (1 \times n_n) \times (n_n \times 1), \]

where the constant nodal data (pressure) column matrix as been pulled out of the integral and the summation of the row matrices.

In this example, \( H \) is degree one in \( r \) and \( s \) while \( f^e \) is degree two in \( r \) and \( s \) so the integrand will be degree three in each parametric direction. For quadrilateral elements the integration rule is the tensor product of the two one-dimensional rules. For a degree three integrand in \( r \) the minimum number of quadrature points is \( 3 \geq (2n_R - 1) \), or \( n_R = 2 \). The same rule applies in the \( s \)-direction (\( n_s = 2 \)) and the total number of two-dimensional quadrature points is \( n_1 = n_R \times n_s = 4 \):

<table>
<thead>
<tr>
<th>( q )</th>
<th>( n_R )</th>
<th>( n_s )</th>
<th>( (r_q, s_q) )</th>
<th>( w_q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(r_s, s_1)</td>
<td>( w_1 ) * ( w_1 )</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>(r_s, s_1)</td>
<td>( w_2 ) * ( w_1 )</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>(r_s, s_1)</td>
<td>( w_1 ) * ( w_2 )</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>(r_s, s_1)</td>
<td>( w_2 ) * ( w_2 )</td>
</tr>
</tbody>
</table>
Quadrature rule tables are provided in the text Matlab scripts for parametric lines, squares and cubes (in both unit and natural coordinates) and for unit triangles and unit tetrahedral. For the unit coordinate two-point rule the tables provide $w_1 = w_2 = 1/2$ and the tabulated locations are $(\sqrt{3} \mp 1)/2\sqrt{3}$.

A rough estimate of the resultant force is to take the average nodal pressure ($25.1 \text{ N/m}^2$) and multiplying by the area, which can be found exactly using two triangles ($18.0 \text{ m}^2$) for a total force of about $451.8 \text{ N}$. Looping over the above four quadrature points using the data for the original quadrilateral gives the correct resultant force as $472.67 \text{ N}$, so the initial estimate was only in error by about 5%.

<table>
<thead>
<tr>
<th>q</th>
<th>$x_q$</th>
<th>$y_q$</th>
<th>$p_q$</th>
<th>$J^e_q * w_q$</th>
<th>$H_1$</th>
<th>$H_2$</th>
<th>$H_3$</th>
<th>$H_4$</th>
<th>$H_q p^e J^e_q w_q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.51</td>
<td>5.10</td>
<td>23.69</td>
<td>3.778</td>
<td>0.622</td>
<td>0.167</td>
<td>0.045</td>
<td>0.167</td>
<td>89.51</td>
</tr>
<tr>
<td>2</td>
<td>6.49</td>
<td>5.80</td>
<td>24.85</td>
<td>3.923</td>
<td>0.167</td>
<td>0.622</td>
<td>0.167</td>
<td>0.045</td>
<td>97.46</td>
</tr>
<tr>
<td>3</td>
<td>4.18</td>
<td>7.53</td>
<td>27.15</td>
<td>5.077</td>
<td>0.167</td>
<td>0.045</td>
<td>0.167</td>
<td>0.622</td>
<td>137.87</td>
</tr>
<tr>
<td>4</td>
<td>6.82</td>
<td>8.57</td>
<td>28.31</td>
<td>5.222</td>
<td>0.045</td>
<td>0.167</td>
<td>0.622</td>
<td>0.167</td>
<td>147.82</td>
</tr>
<tr>
<td>$\sum$</td>
<td></td>
<td></td>
<td></td>
<td>$18.000 \text{ m}^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$472.67 \text{ N}$</td>
</tr>
</tbody>
</table>

Table 2.2 Data at four quadrature points in pressure element