Available online at www.banglajol.info

Bangladesh J. Sci. Ind. Res. **46(3)**, 399-405, 2011

BCSIR

BANGLADESH JOURNAL OF SCIENTIFIC AND INDUSTRIAL RESEARCH

E-mail: bjsir07@gmail.com

Gauss-Legendre Numerical Integrations over a Quadrilateral Element in Closed Form

M.. S. Islam[a],* G. Saha[a] and N. Akter[b]

[a]Department of Mathematics, University of Dhaka, Dhaka-1000, Bangladesh.
[b]Department of Mathematics, Dhaka City College, Dhaka, Bangladesh

Abstract

In this paper we investigate the stiffness matrix of a general quadrilateral element in closed form using $n \times n$ Gauss-Legendre quadrature rule. For this, we propose four types of nodal coordinate transformation. The terms of the matrix are divided into two groups, namely - diagonal and non-diagonal. Only one term (called leading) from each group is computed, and then the remaining fourteen terms are computed from these two leading terms exploited one of the proposed types of coordinate transformation. This leads us a great savings in computational time and memory space. In order to compute the matrix we use these transformations in two ways, and thus two algorithms are given to generate the matrix. Finally, numerical example is given to verify the effectiveness of the present formulation.

Keywords: Gauss-Legendre quadrature, Numerical integration, Quadrilateral finite element, Stiffness matrix, Closed form.

Introduction

In the evaluation of the element/stiffness matrix in Finite Element Method (FEM), various integrals are determined numerically. Among various numerical integration schemes, the use of Gauss quadrature is attractive and it can evaluate exactly the $(2n-1)$th order simple polynomials with n points. The integrals in practical situations are not always simple but rational expressions in which the lower order quadrature scheme cannot evaluate exactly (Hacker *et al,* 1989; Yagawa *et al,* 1990; Zienkiewicz, 1977). For this some researchers have attempted to develop analytical integration formula (Hacker *et al,* 1989 and Rathod and Islam 2002).) for limited finite elements. These integrals usually involve a huge amount of computing time and memory space. Thus symbolic computing techniques (Brlzer, 1990; Videla and Cerrolaza, 1996; Yang, 1994 and Yew *et al* 1995) are applied to save the computational costs. In this aspect, Griffiths (Griffiths, 1994) introduced coordinate transformation in closed form, and recently Islam and Akter (2008) used this technique again but these are limited only for 2×2 Gauss quadrature rule. Since we do not know the exact order of the quadrature rule in which the integrands can be evaluated exactly, higher order quadrature is essential, and an important task that one how can handle easily and efficiently to get the desired accuracy.

Videla and Cerrolaza (Videla and Cerrolaza, 1996) explicit numerical integration with the Derive symbolic manipulation code. Yagawa *et al* (1990) presented a combined approach based on both conventional numerical and symbolic integration. They reported 15 percent savings in CPU times to their test problem. Yang (1994) developed a transformation method that replaces the integral form of the stiffness matrix by its algebraic form only for triangular elements. The analytical integration stiffness matrix has been also investigated by Yew *at el* (1995) in closed form using *Mathematica* but it is confined for the mixed finite elements. Thus the explicit numerical integration is an essential task to provide a balance between efficiency and accuracy in the generation of stiffness matrices.

However, this paper describes how the stiffness matrix of a general quadrilateral element can be expressed in closed form using the Gauss quadrature numerical integration summation. For this, some basic idea of the formulation of stiffness matrix is described. Nodal coordinate transformation method is then employed to replace the associated algebraic form. Two algorithms are proposed to generate the matrix and to save the computational times. These algorithms are executed by *Mathematica.* Numerical accuracy and efficien-

* Corresponding author: E-mail: mdshafiqul@yahoo.com

cy are demonstrated by comparing it with conventional Gaussian quadrature through numerical example.

Formulation of Stiffness Matrix

Let us consider an arbitrary four node linear quadrilateral element in the global system (x, y) which is mapped into a 2-Square in the local parametric system (ξ, η) as shown in the Fig. 1. Then the isoparametric coordinate transformation from (x, y) plane to (ξ, η) plane is given by,

$$x = \sum_{i=1}^{4} x_i N_i(\xi,\eta) \qquad \text{and} \qquad y = \sum_{i=1}^{4} y_i N_i(\xi,\eta) \tag{1}$$

where (x_i, y_i), i = 1 - 4, are the vertices of the element in (x, y) -plane and N_i (ξ, η) denotes the 2D bilinear basis functions (Bickford, 1990 and Zienkiewicz, 1977) with (ξ, η) as the natural coordinates in (ξ, η) -plane such that

$$N_i(\xi,\eta) = \tfrac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i), \quad i = 1-4$$

Now from equation (1) we have

$$x = \tfrac{1}{4}[a_x + b_x\xi + c_x\eta + d_x\xi\eta] \quad \text{and}$$

$$y = \tfrac{1}{4}[a_y + b_y\xi + c_y\eta + d_y\xi\eta] \tag{2a}$$

where,

$$\begin{aligned} a_x &= x_1+x_2+x_3+x_4 & a_y &= y_1+y_2+y_3+y_4 \\ b_x &= -x_1+x_2+x_3-x_4 & b_y &= -y_1+y_2+y_3-y_4 \\ c_x &= -x_1-x_2+x_3+x_4 & c_y &= -y_1-y_2+y_3+y_4 \\ d_x &= x_1-x_2+x_3-x_4 & d_y &= y_1-y_2+y_3-y_4 \end{aligned} \tag{2b}$$

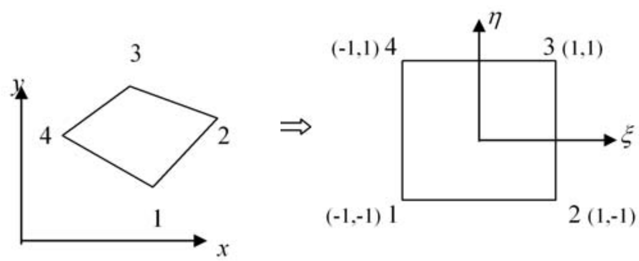

Fig. 1: Original 4-node quadrilateral element and its configuration in ξ - η plane.

Also from equation (2a),we have

$$\frac{\partial y}{\partial \xi} = \tfrac{1}{4}[b_y + d_y\eta] \qquad \text{and} \qquad \frac{\partial y}{\partial \eta} = \tfrac{1}{4}[c_y + a$$

$$\frac{\partial y}{\partial \xi} = \tfrac{1}{4}[b_y + d_y\eta] \qquad \text{and} \qquad \frac{\partial y}{\partial \eta} = \tfrac{1}{4}[c_y + d_y\xi] \tag{3}$$

Hence the Jacobian J can be expressed as:

$$J = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}$$

$$= \alpha_0 + \alpha_1\xi + \alpha_2\eta \tag{4}$$

where,

$$\begin{aligned} \alpha_0 &= \tfrac{1}{8}[(x_4-x_2)(y_1-y_3) - (y_4-y_2)(x_1-x_3)] \\ \alpha_1 &= \tfrac{1}{8}[(x_4-x_3)(y_2-y_1) + (x_1-x_2)(y_4-y_3)] \\ \alpha_2 &= \tfrac{1}{8}[(x_4-x_1)(y_2-y_3) + (x_3-x_2)(y_4-y_1)] \end{aligned} \tag{5}$$

Using the chain rule of calculus and Eqns.(3), we obtain the global derivatives

$$\frac{\partial N_i}{\partial x} = \frac{1}{J}\left[\frac{\partial y}{\partial \eta}\frac{\partial N_i}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial N_i}{\partial \eta}\right]$$

$$= \frac{1}{J}[B^0_{iy} + B^1_{iy}\xi + B^2_{iy}\eta], \quad i = 1-4 \tag{6a}$$

and similary,

$$\frac{\partial N_i}{\partial y} = \frac{1}{J}[B^0_{ix} + B^1_{ix}\xi + B^2_{ix}\eta] \quad i = 1-4 \tag{6b}$$

where

$$\begin{aligned} B^0_{3x} &= \tfrac{1}{8}(x_2-x_4) & B^0_{4x} &= \tfrac{1}{8}(x_3-x_1) \\ B^1_{3x} &= \tfrac{1}{8}(x_2-x_1) & B^1_{4x} &= \tfrac{1}{8}(x_1-x_2) \\ B^2_{3x} &= \tfrac{1}{8}(x_1-x_4) & B^2_{4x} &= \tfrac{1}{8}(x_3-x_2) \end{aligned}$$

$$\begin{aligned} B^0_{3x} &= \tfrac{1}{8}(x_2-x_4) & B^0_{4x} &= \tfrac{1}{8}(x_3-x_1) \\ B^1_{3x} &= \tfrac{1}{8}(x_2-x_1) & B^1_{4x} &= \tfrac{1}{8}(x_1-x_2) \\ B^2_{3x} &= \tfrac{1}{8}(x_1-x_4) & B^2_{4x} &= \tfrac{1}{8}(x_3-x_2) \\ B^0_{1y} &= \tfrac{1}{8}(y_2-y_4) & B^0_{2y} &= \tfrac{1}{8}(y_3-y_1) \\ B^1_{1y} &= \tfrac{1}{8}(y_4-y_3) & B^1_{2y} &= \tfrac{1}{8}(y_3-y_4) \\ B^2_{1y} &= \tfrac{1}{8}(y_3-y_2) & B^2_{2y} &= \tfrac{1}{8}(y_1-y_4) \end{aligned} \tag{7a}$$

$$B_{1y}^{0}=\tfrac{1}{8}(y_2-y_4) \qquad B_{2y}^{0}=\tfrac{1}{8}(y_3-y_1)$$
$$B_{1y}^{1}=\tfrac{1}{8}(y_4-y_3) \qquad B_{2y}^{1}=\tfrac{1}{8}(y_3-y_4) \qquad (7b)$$
$$B_{1y}^{2}=\tfrac{1}{8}(y_3-y_2) \qquad B_{2y}^{2}=\tfrac{1}{8}(y_1-y_4)$$

In order to obtain the finite element stiffness matrix using quadrilateral elements due to second order linear Partial Differential Equation via Galerkin weighted residual formulation (Bickford, 1990 and Zienkiewicz, 1977) the integrals of the product of global derivatives are of the form (Islam *et al* 2008):

$$P_{i,j}^{x,x}=\int\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}\mathrm{d}x\mathrm{d}y, \quad P_{i,j}^{y,y}=\int\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\mathrm{d}x\mathrm{d}y,$$
$$P_{i,j}^{x,y}=\int\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial y}\mathrm{d}x\mathrm{d}y \qquad (8)$$

Since we are restricted ourselves only to consider the 4-node quadrilateral element, the element matrix will be symmetric, so only those terms, on and above on the main diagonal, will need to be evaluated. This symmetric matrix is of the form

$$K=[k_{i,j}]=\begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} & k_{1,4}\\ & k_{2,2} & k_{2,3} & k_{2,4}\\ symm & & k_{3,3} & k_{3,4}\\ & & & k_{4,4}\end{bmatrix} \qquad (9)$$

where each component, $k_{i\text{-}j}$ of the matrix K is the linear combination of the integrals defined in eqns. (8). Generally, this can be written symbolically as

$$k_{i,j}=\frac{1}{64}\int_{-1}^{1}\int_{-1}^{1}\frac{1}{J}[B_{i,j}^{00}+B_{i,j}^{10}\xi+B_{i,j}^{01}\eta+B_{i,j}^{20}\xi^2+$$
$$B_{i,j}^{11}\xi\eta+B_{i,j}^{02}\eta^2]\,d\xi d\eta \qquad i,j=1-4 \qquad (10)$$

Since each coefficient, $B_{i,j}^{mn}$ is constant depending on the four vertices of the quadrilateral obtained using eqns.(7 - 8).

Evaluation of Stiffness Matrix

Since in Eq.(10), the denominator (the *Jacobian* defined in Eq. (4) is a function of two variables ξ and η, the numerator is also a function of two variables ξ and η, so applying *nxn* Gaussian quadrature(Bickford, 1990 and Zienkiewicz, 1977) the numerical integration of Eq. (10) is then

$$k_{i,j}=\frac{1}{64}\sum_{q=1}^{n}\sum_{p=1}^{n}\frac{B_{i,j}^{00}+B_{i,j}^{10}\xi_p+B_{i,j}^{01}\eta_q+B_{i,j}^{20}\xi_p^2+B_{i,j}^{11}\xi_p\eta_q+B_{i,j}^{02}\eta_q^{\,2}}{\alpha_0+\alpha_1\xi_p+\alpha_2\eta_q}w_p w_q$$

where (ξ_p, η_q) are the Gaussian integration points, and w_p, w_q, are the corresponding weights.

Before evaluating the terms of the matrix defined in Eq. (9), we split the terms into two different groups, namely A (diagonal), and B (non-diagonal). Then the Eq.(9) may be redefined as

$$K=[k_{i,j}]=\begin{bmatrix} A & B & B & B\\ & A & B & B\\ sym & & A & B\\ & & & A\end{bmatrix} \qquad (12)$$

We compute the leading terms $k_{1,1}$ ($i=j=1$) and $k_{1,2}$ ($i=1$, $j=2$) of the groups A, and B, respectively. The other terms of each group can be obtained using one of the four nodal coordinate transformations (Zienkiewicz, 1977) listed in Table I. The notation used in Table 1 is that the symbol ==> means "is replaced by".

Table I: Types of Nodal Coordinate Transformation

Type 1	Type 2	Type 3	Type 4
$(x_1,y_1)\Rightarrow(x_4,y_4)$	$(x_1,y_1)\Rightarrow(y_3,x_3)$	$(x_1,y_1)\Rightarrow(y_3,x_3)$	$(x_1,y_1)\Rightarrow(x_2,y_2)$
$(x_2,y_2)\Rightarrow(x_1,y_1)$	$(x_2,y_2)\Rightarrow(y_2,x_2)$	$(x_2,y_2)\Rightarrow(y_4,x_4)$	$(x_2,y_2)\Rightarrow(x_4,y_4)$
$(x_3,y_3)\Rightarrow(x_2,y_2)$	$(x_3,y_3)\Rightarrow(y_1,x_1)$	$(x_3,y_3)\Rightarrow(y_1,x_1)$	$(x_3,y_3)\Rightarrow(x_1,y_1)$
$(x_4,y_4)\Rightarrow(x_3,y_3)$	$(x_4,y_4)\Rightarrow(y_4,x_4)$	$(x_4,y_4)\Rightarrow(y_2,x_2)$	$(x_4,y_4)\Rightarrow(x_3,y_3)$

Using four types of transformation the computational relation between the two terms of each groups matrix K in Eq. (9), is shown in Table II.

Table II: Relation between two terms of matrix K

Group	To compute	To compute	To compute
A	$k_{4.4}$	k_{11}	Type 1
	$k_{3.3}$	$k_{4.4}$	Type 1
	$k_{2.2}$	$k_{3.3}$	Type 1
B	$k_{2.3}$	$k_{1.2}$	Type 2
	$k_{1.4}$	$k_{2.3}$	Type 3
	$k_{4.4}$	$k_{1.4}$	Type 2
	$k_{1.3}$	$k_{3.4}$	Type 4
	$k_{2.4}$	$k_{1.3}$	Type 1

Now we give two brief methods below to compute the complete matrix for a general four node quadrilateral described in Figure 1 on the basis of the above information. We also write a Mathematical program for each, which are available upon request to the corresponding authour.

Method 1:

For this method first we write down the explicit form of the coefficients of k_{11} and $k_{1,2}$ as follows:

For $k_{1,1}$:

$$B_{1,1}^{00} = (y_2 - y_4)^2 + (x_4 - x_2)^2$$
$$B_{1,1}^{10} = 2\{(y_2 - y_4)(y_4 - y_3) + (x_4 - x_2)(x_3 - x_4)\}$$
$$B_{1,1}^{01} = 2\{(y_2 - y_4)(y_3 - y_2) + (x_4 - x_2)(x_2 - x_3)\}$$
$$B_{1,1}^{20} = (y_4 - y_3)^2 + (x_3 - x_4)^2 \tag{12a}$$
$$B_{1,1}^{11} = 2\{(y_4 - y_3)(y_3 - y_2) + (x_3 - x_4)(x_2 - x_3)\}$$
$$B_{1,1}^{02} = (y_3 - y_2)^2 + (x_2 - x_3)^2$$

For $k_{1,2}$:

$$B_{1,2}^{00} = (y_2 - y_4)(y_3 - y_1) + (x_4 - x_2)(x_1 - x_3)$$
$$B_{1,2}^{10} = (y_2 - y_4)(y_3 - y_4) + (y_4 - y_3)(y_3 - y_1) + (x_4 - x_2)(x_4 - x_3) + (x_3 - x_4)(x_1 - x_3)$$
$$B_{1,2}^{01} = (y_2 - y_4)(y_1 - y_4) + (y_3 - y_2)(y_3 - y_1) + (x_4 - x_2)(x_4 - x_1) + (x_2 - x_3)(x_1 - x_3)$$
$$B_{1,2}^{20} = -(y_4 - y_3)^2 - (x_4 - x_3)^2 \tag{12b}$$
$$B_{1,2}^{11} = (y_4 - y_3)(y_1 - y_4) + (y_3 - y_2)(y_3 - y_4) + (x_3 - x_4)(x_4 - x_1) + (x_2 - x_3)(x_4 - x_3)$$
$$B_{1,2}^{02} = (y_3 - y_2)(y_1 - y_4) + (x_2 - x_3)(x_4 - x_1)$$

Algorithm 1:

Step 1. Input: The coordinates of the quadrilateral element; $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$,

Step 2. Define $B_{1,2}^{m,n}$ and $B_{1,2}^{m,n}$ for $k_{1,1}$ and $k_{1,2}$ respectively, defined in Eq.(13)

Step 3. Define the Gauss points.

Step 4. Define the leading terms $k_{1,1}$ and $k_{1,2}$ defined in Eq. (10).

Step 5. Write down the nodal coordinate transformation types.

Step 6. Store data in the variables.

Step 7. Calculate $k_{i,j}$ $1 \leq$ i, j ≤ 4. .

Method 2:

Now express the Eq. (11) using the quadrature rule for $n =$ 2,3 and 4, successively as given below for this paper. One can proceed for n ≥ 5, , in a similar fashion.

For : $n = 2$:

$$k_{i,j} = \frac{1}{32}\left(\frac{s_1 f_0 - t_1 f_1}{3f_0^{\,2} - f_1^2} + \frac{s_2 f_0 - t_2 f_2}{3f_0^{\,2} - f_2^2}\right) \tag{14a}$$

where

$$f_0 = \alpha_0 \qquad s_1 = 3B_{i,j}^{00} + B_{i,j}^{02} + B_{i,j}^{20} - B_{i,j}^{11}$$
$$f_1 = \alpha_1 - \alpha_2 \qquad s_2 = 3B_{i,j}^{00} + B_{i,j}^{02} + B_{i,j}^{20} + B_{i,j}^{11}$$
$$f_2 = \alpha_1 + \alpha_2$$
$$t_1 = B_{i,j}^{10} - B_{i,j}^{01}$$
$$t_2 = B_{i,j}^{10} + B_{i,j}^{01} \tag{14b}$$

For $n = 3$:

$$k_{i,j} = \frac{s_0}{81\alpha_0} + \frac{5}{324}\left(\frac{\alpha_0 s_1 - \alpha_1 s_2}{f_1} + \frac{\alpha_0 s_3 - \alpha_2 s_4}{f_2}\right) + \frac{25}{2592}\left(\frac{\alpha_0 t_1 - f_5 t_2}{f_3} + \frac{\alpha_0 t_3 - f_6 t_4}{f_4}\right) \quad (15a)$$

where

$$\begin{aligned}
&f_1 = 5\alpha_0{}^2 - 3\alpha_1^2 && s_0 = B_{i,j}^{00} && t_1 = 5B_{i,j}^{00} + 3B_{i,j}^{2,0} + 3B_{i,j}^{02} - 3B_{i,j}^{11}\\
&f_2 = 5\alpha_0{}^2 - 3\alpha_2^2 && s_1 = 5B_{i,j}^{00} + 3B_{i,j}^{20} && t_2 = 3(B_{i,j}^{10} - B_{i,j}^{01})\\
&f_3 = 5\alpha_0{}^2 - 3(\alpha_1 - \alpha_2)^2 && s_2 = 3B_{i,j}^{10} && t_3 = 5B_{i,j}^{00} + 3B_{i,j}^{2,0} + 3B_{i,j}^{02} + 3B_{i,j}^{11}\\
&f_4 = 5\alpha_0{}^2 - 3(\alpha_1 + \alpha_2)^2 && s_3 = 5B_{i,j}^{00} + 3B_{i,j}^{02} && t_4 = 3(B_{i,j}^{10} + B_{i,j}^{01})\\
&f_5 = \alpha_1 - \alpha_2 && s_4 = 3B_{i,j}^{01} &&\\
&f_6 = \alpha_1 + \alpha_2 && &&
\end{aligned} \quad (15b)$$

For $n = 4$:

$$\begin{aligned}
k_{i,j} = &\frac{c^2}{32}\left(\frac{s_0 + a^2(\alpha_0 s_1 - f_1 s_2)}{f_2} + \frac{s_0 + a^2(\alpha_0 s_3 - f_4 s_4)}{f_5}\right)\\
&+ \frac{d^2}{32}\left(\frac{s_0 + b^2(\alpha_0 s_1 - f_1 s_2)}{f_3} + \frac{s_0 + b^2(\alpha_0 s_3 - f_4 s_4)}{f_6}\right)\\
&+ \frac{cd}{32}\left[\frac{s_0 + \alpha_0 t_1 - f_7 t_2}{f_8} + \frac{s_0 + \alpha_0 t_3 - f_9 t_4}{f_{10}} + \frac{s_0 + \alpha_0 t_5 - f_{11} t_6}{f_{12}} + \frac{s_0 + \alpha_0 t_7 - f_{13} t_8}{f_{14}}\right]
\end{aligned} \quad (16a)$$

where

$$\begin{aligned}
&f_1 = \alpha_1 + \alpha_2 && f_5 = \alpha_0{}^2 - a^2 f_4^2 && f_9 = a\alpha_1 - b\alpha_2 && f_{13} = b\alpha_1 - a\alpha_2\\
&f_2 = \alpha_0{}^2 - a^2 f_1^2 && f_6 = \alpha_0{}^2 - b^2 f_4^2 && f_{10} = \alpha_0{}^2 - f_9^2 && f_{14} = \alpha_0{}^2 - f_{13}^2\\
&f_3 = \alpha_0{}^2 - b^2 f_1^2 && f_7 = a\alpha_1 + b\alpha_2 && f_{11} = b\alpha_1 + a\alpha_2 &&\\
&f_4 = \alpha_1 - \alpha_2 && f_8 = \alpha_0{}^2 - f_7^2 && f_{12} = \alpha_0{}^2 - f_{11}^2 &&
\end{aligned} \quad (16b)$$

$$\begin{aligned}
&s_0 = \alpha_0 B_{i,j}^{00} && t_1 = a^2 B_{i,j}^{20} + b^2 B_{i,j}^{02} + abB_{i,j}^{11} && t_5 = b^2 B_{i,j}^{20} + a^2 B_{i,j}^{02} + abB_{i,j}^{11}\\
&s_1 = B_{i,j}^{20} + B_{i,j}^{02} + B_{i,j}^{11} && t_2 = aB_{i,j}^{10} + bB_{i,j}^{01} && t_6 = bB_{i,j}^{10} + aB_{i,j}^{01}\\
&s_2 = B_{i,j}^{10} + B_{i,j}^{01} && t_3 = a^2 B_{i,j}^{2,0} + b^2 B_{i,j}^{02} - abB_{i,j}^{11} && t_7 = b^2 B_{i,j}^{2,0} + a^2 B_{i,j}^{02} - abB_{i,j}^{11}\\
&s_3 = B_{i,j}^{20} + B_{i,j}^{02} - B_{i,j}^{11} && t_4 = aB_{i,j}^{10} - bB_{i,j}^{01} && t_8 = bB_{i,j}^{10} - aB_{i,j}^{01}\\
&s_4 = B_{i,j}^{10} - B_{i,j}^{01} && &&
\end{aligned} \quad (16c)$$

$$a = \sqrt{\frac{15 + 2\sqrt{30}}{35}}, \qquad b = \sqrt{\frac{15 - 2\sqrt{30}}{35}}, \qquad c = \frac{3\sqrt{30} + 5}{6\sqrt{30}}, \qquad d = \frac{3\sqrt{30} - 5}{6\sqrt{30}} \quad (16d)$$

Here α and $B_{1,2}^{m,n}$ are defined in Eqs. (5) and (13), respectively. The functions *f, s* and *t* depend on the nodal co-ordinates. Also note that the transformations are always applied to the '*s* ', '*t* 'and '*f* ' functions.

Algorithm 2:

Step 1 Input: The coordinates of the quadrilateral element; $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$.

Step 2. Define the leading terms $k_{1,1}$ and $k_{1,2}$ defined in Eqs. (14 - 16) for 2x2, 3x3 and 4x4 Gauss points, respectively.

Step 3. Write down the nodal transformation types.

Step 5. Store data in the variables.

Step 6. Set the expressions for *f, s, t* .

Step 7. Calculate $k_{i,j}$ $1\leq$ i, j ≤ 4.

Test Example

We compute the element matrices using the integration formula presented in this paper to compare with the existing solutions. For this, we consider a simple one-element example to evaluate the element matrices for two-dimensional Laplace's equation. The finite element formulation (Beltzer, 1990 and Hacker *et al.*, 1989)of the Laplace's equation is

$$K_{i,j} = \iint_R \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right) \mathrm{d}x\mathrm{d}y$$

$$= P_{i,j}^{x,x} + P_{i,j}^{y,y} \quad i,j = 1,2,3,4 \tag{17}$$

where *R* is the typical four-node isoparametric element shown in Fig. 2

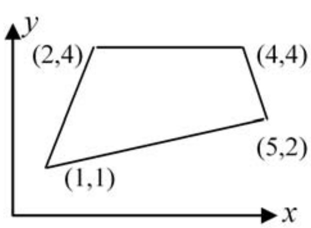

Fig. 2: Element geometry for Laplacian matrices

The complete matrices (symmetric) for this geometry are shown in Table III, obtained by the proposed algorithms for *n* =2, 3, 4 Gauss points. The accuracy of the results is similar to the existing solutions (Beltzer, 1990).

Table III: Element matrix for the geometry shown in Fig. 2

Gauss points	Matrix K = [$k_{i,j}$], $1\leq i, j \leq 4$.
2x2	$\begin{bmatrix} 0.48413 & 0.02353 & -0.29314 & -0.21471 \\ 0.02353 & 0.71471 & -0.31029 & -0.42794 \\ -0.29314 & -0.31029 & 0.81887 & -0.21544 \\ -0.21471 & -0.42794 & -0.21544 & 0.86809 \end{bmatrix}$
3x3	$\begin{bmatrix} 0.48519 & 0.02212 & -0.29072 & -0.21668 \\ 0.02212 & 0.71668 & -0.31392 & -0.42498 \\ -0.29072 & -0.31392 & 0.82552 & -0.22088 \\ -0.21668 & -0.42498 & -0.22088 & 0.86254 \end{bmatrix}$
4x4	$\begin{bmatrix} 0.48522 & 0.02218 & -0.29066 & -0.21674 \\ 0.02218 & 0.71674 & -0.31402 & -0.42490 \\ -0.29066 & -0.31402 & 0.83570 & -0.22102 \\ -0.21674 & -0.42490 & -0.22102 & 0.86266 \end{bmatrix}$

Table IV: Comparison of running time

Gauss Points	CPU Time (sec)		
	Conventional	Present Methods	
	(Eq.11)	Method 1	Method 2
2x2	0.016	0.014	0.001
3x3	0.031	0.015	0.001
4x4	0.281	0.094	0.047

The running time of the test example is listed in Table IV. Observe that the savings in computational time is obvious. In this study, Method 1 can save the computational time 12.5%, 51.6% and 66.5% compare to the conventional (direct) numerical integration method for 2x2, 3x3 and 4x4 Gauss points, respectively. On the other hand, the savings in computational time by Method 2 are 90.7%, 96.7% and 83.3% for the same, respectively. Note that the computational time requires 118.343 seconds for analytical results (Rathod et al., 2008) while 4x4 Gaussian quadrature rule requires 0.047 seconds using algorithm 2 in this paper to get the same accuracy. Thus the performance of the present methods is excellent.

Conclusion

In this paper, two simple and efficient methods are focused on the reduction of computing time and space in the stiffness matrix for a general four node quadrilateral element through pre- and post-integration process, and by symbolic computation. Among the 16 terms of a 4×4 stiffness matrix, only two terms are simplified and are calculated by $n \times n$ Gaussian quadrature rule. Nodal coordinate transformation method is thus developed in this study so that the explicit expressions of these two terms can be replaced by the associated algebraic form obtained pre- and post-numerical integration. Thus we can save greatly the computational time and memory space in two ways: we no need to, (i) compute the coefficients of the element equation (11), and (ii) simplifying the Eq.(11) using $n \times n$ Gaussian quadrature rule, for the remaining fourteen terms of the matrix. Here, the explicit numerical integration is used for the improvement of the efficiency of numerical integration, not for the integration itself. The procedure is rather simple and it may be carried out to optimize the explicit integration formulas also for the other finite elements. The comparison of the computing time also confirmed one of the main advantages of the symbolic integration approach.

References

Beltzer A. I. (1990). Variational and Finite element Methods, A Symbolic Computation Approach, Springer-Verlag, New York Berlin Heidelberg (1990).

Bickford W. B. A. (1990). First Course in the Finite Element Method, Irwin, Illinois (1990).

Hacker W. L., Schreyer H. L. (1989). Eigenvalue analysis of compatible and incompatible rectangular four-node quadrilateral elements, *Int. J. Numer. Methods Engrg*. **28**(3): 687-703 (1989).

Griffiths D. V. (1994). Stiffness matrix of the four-node quadrilateral element in closed form, *Int. J. Numer. Methods Engrg.* **37**(6), 1027 - 1038 (1994).

Islam M. S. and Akter N. (2008) - Closed form numerical integration formulae for a four-node convex quadrilateral finite element, *The Dhaka University Journal of Science,* **56** (2) 165 - 169.

Rathod H. T., Islam M. Shafiqul (2002). Reduction of rational integrals related to linear and convex quadrilateral finite elements, Numer. *Methods Partial Differ. Eqns*. **18**: 759 - 770 (2002).

Videla L. and Cerrolaza M. (1996). Explicit integration of the stiffness matrix of a four nodded plane elasticity finite element, Commn. Numer. *Methods Engrg.* **12**: 731 - 743 (1996).

Yagawa G., Ye G. W. and Yoshimura S. (1990). A numerical integration scheme for finite element method based on symbolic manipulation, *Int. J. Numer. Methods Engrg.* **29**, 1539 - 1549.

Yang C. Y. (1994). An algebraic-expressed finite element model for symbolic computation, *Comput. Struct*. **52**(5): 1069 - 1077.

Yew C. K., Boyle J. T. and Mackenzie D. (1995). Closed form integration of element stiffness matrices using a computer algebra system, *Comput. Struct.* **56**(4): 529 - 539

Zienkiewicz O. C. (1977). The Finite Element Method, 3rd Edn., Mc Graw Hill Inc.

Received : September 16, 2008;
Accepted : October 19, 2010