Applications of Gauss-Radau and Gauss-Lobatto Numerical Integrations Over a Four Node Quadrilateral Finite Element

Md. Shafiqul Islam* and Goutam Saha

Department of Mathematics, University of Dhaka, Dhaka-1000, Bangladesh

Abstract

In this paper Gauss-Radau and Gauss-Lobatto quadrature rules are presented to evaluate the rational integrals of the element matrix for a general quadrilateral. These integrals arise in finite element formulation for second order Partial Differential Equation via Galerkin weighted residual method in closed form. Convergence to the analytical solutions and efficiency are depicted by numerical example.

Key words: Gauss-Radau, Gauss-Lobatto, Rational Integrals, Quadrilateral Finite Element.

Introduction

Various integrals are determined numerically in the evaluation of the element/stiffness matrix, mass matrix, body force vector, initial stress vector, the surface load vector etc., arising in Finite Element Method (FEM). Since the integrals in practical situations are not always simple but rational expressions in which the lower order quadrature scheme cannot evaluate exactly. Also there is no order of Gauss quadrature that will evaluate these integrals exactly (Zienkiewicz 1977, Bickford 1990, Yagawa *et al.* 1990). The integration points have to be increased in order to improve the integration accuracy and it is desirable to make these evaluations by using as few Gauss points (Stroud 1974,

Burden and Faires 2002, Babolian *et al.* 2006) as possible, from the point of view of the computational efficiency.

In this aspect, a large number of articles are available in the literature using the quadrilateral element with straight sides, which is one of the most popular elements in FEM. The complications arise from two main sources (Barrett 1999): the large number of integrations that need to be performed and the presence of the determinant of the *Jacobian matrix* (which will be referred to simply as the *Jacobian* throughout this paper) in the denominator of the element matrices. For this some researchers have been attempted to

*Corresponding authour, Email: mdshafiqul@yahoo.com

develop analytical integration formula (Rathod and Islam 2002, Hacker and Schreyer 1989) for limited finite elements, but a huge amount of computing time and memory space are involved. Thus numerical integration is an essential task to strike a proper balance between accuracy and efficiency commonly applied for science and engineering problems.

However, this paper describes the finite element matrix by Galerkin weighted residual method using four node quadrilateral elements. Then a brief introduction of Gauss-Radau (Masjed-Jamei *et al.* 2005) and Gauss-Lobatto (Eslahchi et al. 2005) quadrature rules, which are previously applied to evaluate for single integrals, is presented in the next section. These numerical methods are exploited to evaluate the double integrals of rational functions belonging to the finite element matrix in this paper. A symbolic algebra package, Mathematica is used to generate this element matrix. Numerical accuracy and efficiency are demonstrated by comparing it with the conventional Gaussian quadrature as well as analytical method through numerical example.

Explicit formulation of element matrix

Let us consider an arbitrary four noded 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 following figure. Then the isoparametric coordinate transformation from (x,y) plane to (ξ, η) plane is given by,

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

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

$$N_i = \tfrac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i), \quad i = 1\text{-}4 \quad (2)$$

Now from equation (1) we have

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

$$y = \frac{1}{4}[a_y + b_y\xi + c_y\eta + d_y\xi\eta] \quad (3a)$$

where,

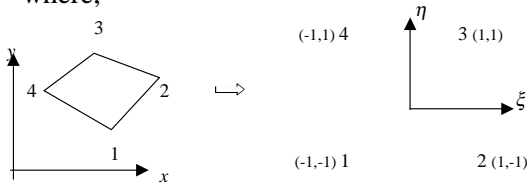**Fig. 1.   Original 4-node quadrilateral element and its configuration in ξ–η plane.**

$$a_x = x_1 + x_2 + x_3 + x_4 \qquad a_y = y_1 + y_2 + y_3 + y_4$$
$$b_x = -x_1 + x_2 + x_3 - x_4 \qquad b_y = -y_1 + y_2 + y_3 - y_4$$
$$c_x = -x_1 - x_2 + x_3 + x_4 \qquad c_y = -y_1 - y_2 + y_3 + y_4$$
$$d_x = x_1 - x_2 + x_3 - x_4 \qquad d_y = y_1 - y_2 + y_3 - y_4$$

$$(3b)$$

Hence the Jacobian can be expressed as:

$$J = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \frac{\partial x}{\partial\xi}\frac{\partial y}{\partial\eta} - \frac{\partial x}{\partial\eta}\frac{\partial y}{\partial\xi} = \alpha_0 + \alpha_1\xi + \alpha_2\eta$$

$$(4a)$$

where,

$$\alpha_0 = \tfrac{1}{8}[(x_4 - x_2)(y_1 - y_3) - (y_4 - y_2)(x_1 - x_3)]$$

$$\alpha_1 = \tfrac{1}{8}[(x_4 - x_3)(y_2 - y_1) + (x_1 - x_2)(y_4 - y_3)]$$

$$\alpha_2 = \tfrac{1}{8}[(x_4 - x_1)(y_2 - y_3) + (x_3 - x_2)(y_4 - y_1)]$$

(4b)

In order to obtain the finite element matrix using quadrilateral elements due to second order linear Partial Differential Equation via Galerkin weighted residual formulation the integrals of the symmetric 4x4 matrix (Zienkiewicz 1977, Bickford 1990) is

$$K = [k_{i,j}] = \begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} & k_{1,4} \\ & k_{2,2} & k_{2,3} & k_{2,4} \\ symm & & k_{3,3} & k_{3,4} \\ & & & k_{4,4} \end{bmatrix} \quad (5)$$

where each $k_{i,j}$, after a minor simplification, is of the form

$$k_{ij} = \frac{1}{64} \int_{-1}^{1}\int_{-1}^{1} \frac{1}{J}[B_{i,j}^{00} + B_{i,j}^{10}\xi + B_{i,j}^{01}\eta +$$

$$B_{i,j}^{01}\eta + B_{i,j}^{20}\xi^2 + B_{i,j}^{11}\xi\eta + B_{i,j}^{02}\eta^2]\,d\xi\,d\eta$$

$$i, j = 1-4 \qquad (6)$$

Here each coefficient, $B_{i,j}^{mn}$ $(0 \le m, n \le 2)$, of each $k_{i,j}$ $(1 \le i, j \le 4)$ depends on the four vertices of the physical quadrilateral, can be written explicitly. For example, the coefficients of $k_{1,1}$ are:

$$B_{1,1}^{00} = (y_2 - y_4)^2 + (x_4 - x_2)^2$$

$$B_{1,1}^{10} = 2\{(y_2 - y_4)(y_4 - y_3) + (x_4 - x_2)(x_3 - x_4)\}$$

$$B_{1,1}^{01} = 2\{(y_2 - y_4)(y_3 - y_2) + (x_4 - x_2)(x_2 - x_3)\}$$

$$B_{1,1}^{20} = (y_4 - y_3)^2 + (x_3 - x_4)^2$$

$$B_{1,1}^{11} = 2\{(y_4 - y_3)(y_3 - y_2) + (x_3 - x_4)(x_2 - x_3)\}$$

$$B_{1,1}^{02} = (y_3 - y_2)^2 + (x_2 - x_3)^2$$

others can be written similarly

Numerical integration

The most common numerical integration is given by (Stroud 1974, Burden and Faires 2002)

$$\int_{a}^{b} w(x)f(x)dx = \sum_{k=1}^{n} w_k f(x_k) + E[f] \quad (7)$$

known as the Gaussian quadrature, where x_k, w_k $(k = 1, 2, \text{L}, n)$ and $E[f]$ are called nodes (abscissas), weights, and error approximation, respectively. Since $w(x)$, called weight function, and $f(x)$ are integrable functions over the interval $[a,b]$, and x_k lies in $[a,b]$, the interval of integration. This gives us $2n$ parameters to choose. If the coefficients of a polynomial are considered parameters, the class of polynomials of degree at most $2n$ - 1 also contains parameters. This, then, is the largest class of polynomials for which it is reasonable to expect the formula to be exact (i.e. $E[f]=0$). With the proper choice of values and constants, exactness on this set can be obtained.

Assuming the exactness (i.e. $E[f]=0$) and for our convenient (by change of variable), and set $w(x)=1$, the eqn. (7) the can be written as,

$$\int_{-1}^{1} f(x)dx = \sum_{k=1}^{m} w_k f(x_k) + \sum_{k=m+1}^{n-m} w_k f(x_k)\ m \le$$

(8)

which is known as generalized Gauss quadrature rules. Particular quadratures are as follows:

1. If we put $m=0$, then the concluded numerical integration rule is called Gauss-Legendre (Burden and Faires 2002). Set $f(x)=x^i$, $i = 0,1,2,..., 2n-1$ in (8), which gives us $2n$ parameters (n weights and n nodes), and solving the nonlinear system to obtain the required nodes x_k, , and the corresponding weights w_k for each $k = 1,2,..., n$. since the nodes are symmetric.

2. If we put $m =1$, $x_1 = -1$ (*or* 1) then the concluded numerical integration rule is called Gauss-Radau (Masjed-Jamei *et al.* 2005). $f(x)= x^i$, $i = 0,1,2,...,2n-2$ Setting in (8) to obtain $2n-1$ parameters (n weights and $n-1$ nodes). Then solving the system, we can obtain required nodes

x_k, $k = 2,3,...n$ and the corresponding weights w_k, $k = 1,2,...n$. Observe that the nodes are unsymmetric.

3. Rearrange the eqn. (8) as

$$\int_{-1}^{1} f(x)dx = w_1 f(x_1) + \sum_{k=2}^{n-1} w_k f(x_k) + w_n f(x_n) \tag{9}$$

and putting $x_1 = -$ *and* $x_n = 1$ in (9), then the numerical integration rule is called Gauss-Lobatto (Eslahchi *et al.* 2005). If we set $f(x)=x^i$, $i =0,1,2,...2n-3$ in (9), we get $2n-2$ parameters (n weights and $2n-2$ nodes). Then solving the system, we can obtain required nodes x_k, $k =2,3,...n-1$, and the corresponding weights w_k, $k =1,2,...n$. In this case the nodes are symmetric.

We may summarize the nodes and corresponding weights for $n =3,4,5$ in Table - I, for this paper.

Table I. Nodes and weights for Gauss-Legendre, Gauss-Radau, and Gauss-Lobatto methods

n	Gauss-Legendre (G)		Gauss-Radau (GR)		Gauss-Lobatto (GL)	
	Nodes x_k	Weights w_k	Nodes x_k	Weights w_k	Nodes x_k	Weights w_k
3	0.000000	0.888889	- 1.000000	0.222222	±1.000000	0.333333
	±0.774597	0.555556	- 0.289898	1.0249717	0.000000	1.333333
			0.689898	0.7528061		
4	±0.339981	0.652145	- 1.000000	0.125000	±1.000000	0.166667
	±0.861136	0.347859	- 0.575319	0.657689	±0.447214	0.833333
			0.181066	0.776387		
			0.822824	0.440924		
5	0.000000	0.568889	- 1.000000	0.080000	±1.000000	0.100000
	±0.538469	0.478629	- 0.720480	0.446208	±0.447214	0.544444
	±0.906179	0.236927	0.167181	0.623653	0.000000	0.711111
			0.446314	0.562712		
			0.885792	0.287427		

In general, the integrals defined in eqn. (6) cannot be evaluated easily. Then each term, k_i, j of (5) gives us rational integral. In fact, these rational integrals are evaluated by Gauss quadratures. Since the denominator (the *Jacobian* defined in eqn. (4)) is a function of two variables ξ and η, the numerator is also a function of two variables ξ and η, so we assume that the eqn.(6) can be expressed as

$$I = \int_{-1}^{1} \int_{-1}^{1} f(\xi,\eta)\, d\xi\, d\eta \qquad (10)$$

In particular, for integration over the standard square, we can use the two dimensional Gaussian quadratures, which takes the form:

$$I = \sum_{q=1}^{n} \sum_{p=1}^{n} f(\xi_p,\eta_q)\, w_p\, w_q \qquad (11)$$

where (ξ_p, η_q) are the Gauss-Legendre, Gauss-Radau, and Gauss-Lobatto integration points and w_p, w_q, are the corresponding weighting factors independent of f.

Now compute the complete element matrix for a general four noded quadrilateral, described in Fig.1, on the basis of the above information using *Mathematica* program.

Test Example

In this section, we wish to compute the element matrix using closed form 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 matrix for two-dimensional Laplace's equation. The finite element formulation (Zienkiewicz 1977, Bickford 1990) of the Laplace's equation is

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

$$= P_{i,j}^{x,x} + P_{i,j}^{y,y} \quad i, j = 1,2,3,4 \qquad (12)$$
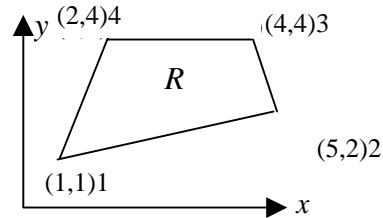
Fig. 2. Element geometry for Laplacian matrices

where R is the typical four-node isoparametric element shown in Fig. 2 [Bickford 1990, pp. 313-314]. In order to compare the accuracy, the complete element matrix (symmetric) for the geometry shown in Fig. 2 is given in Table II, and the error compare to the analytical method (Rathod and Islam 2002) obtained by *Mathematica* is also given in Table III obtained by the presented numerical integration methods.

Table II. A: Element matrix for the geometry in Fig. 2 using 3x3 order

Integration Methods	j i	1	2	3	4
Legendre (G)		0.48519217	0.02221173	-0.29072138	-0.21668250
Radau (GR)	1	0.48505441	0.02241839	-0.29110038	-0.21637241
Lobatto (GL)		0.48662137	0.02006795	-0.28679124	-0.21989807
Analytical		0.48521668	0.02217497	-0.29065411	-0.21673754
G			0.71668250	-0.31391790	-0.42497624
GR	2		0.71637241	-0.31334942	-0.42544137
GL			0.71989807	-0.31981314	-0.42015288
Analytical			0.71673754	-0.31401883	-0.42489369
G		**Symmetric**		0.82551620	-0.22087689
GR				0.82447395	-0.22002414
GL	3			0.83632409	-0.22971971
Analytical				0.82570118	-0.22102824
G					0.86253564
GR					0.86183793
GL	4				0.86977067
Analytical					0.86265947

Table: IIB: Element matrix using 4x4 order

Integration Methods	j i	1	2	3	4
G		0.48521592	0.02217615	-0.29065628	-0.216735816
GR	1	0.48521198	0.02218201	-0.29066701	-0.216726974
GL		0.48525153	0.02212446	-0.29056209	-0.216813894
Analytical		0.48521668	0.02217497	-0.29065411	-0.216737540
G			0.71673577	-0.31401559	-0.42489634
GR	2		0.71672699	-0.31399950	-0.42490949
GL			0.716815730	-0.31415962	-0.42478057
Analytical			0.71673754	-0.31401883	-0.42489369
G		**Symmetric**		0.82569520	-0.22102349
GR	3			0.82566576	-0.22099920
GL				0.82596022	-0.22123851
Analytical				0.82570118	-0.22102824
G					0.86265558
GR					0.86263572
GL	4				0.86283298
Analytical					0.86265947

Table. IIC: Element matrix using 5x5 order

Integration methods	j / i	1	2	3	4
G		0.48521671	0.02217505	-0.29065450	-0.21673770
GR	1	0.48521651	0.02217518	-0.29065448	-0.21673721
GL		0.48521675	0.02217337	-0.29065069	-0.21673944
Analytical		0.48521668	0.02217497	-0.29065411	-0.21673754
G			0.71673818	-0.31401899	-0.42489423
GR	2		0.71673715	-0.31401818	-0.42489414
GL			0.71673844	-0.31402247	-0.42488934
Analytical			0.71673754	-0.31401883	-0.42489369
G		**Symmetric**		0.82570170	-0.2210282
GR	3			0.82569997	-0.22102731
GL				0.82570736	-0.22103421
Analytical				0.82570118	-0.22102824
G					0.86265934
GR	4				0.86265866
GL					0.86266299
Analytical					0.86265947

Table III. Error estimation for the results in Table II

Order n x n	Integration methods	Errors			
	Legendre	0.00002445	0.00003669	0.00006728	0.0000550
		0.00003669	0.0000550	0.00010087	0.00004256
		0.00006728	0.00010087	0.00018492	0.0001513
		0.0000550	0.00008256	0.0001513	0.0001238
3 x 3	Radau	0.00016228	0.00024342	0.00044627	0.0003651
		0.00024342	0.0003651	0.00066940	0.0005477
		0.00044627	0.00066940	0.0012272	0.00100410
		0.0003651	0.0005477	0.00100410	0.0008215
	Lobatto	0.00140467	0.00210702	0.0038629	0.00316053
		0.00210702	0.00310651	0.0057943	0.00474082
		0.00038629	0.0057943	0.0106229	0.00869146
		0.00216053	0.00474082	0.00869146	0.0071112

Table III to be contd.

	Legendre	3.36×10^{-6}	$1. \times 10^{-6}$	3.8×10^{-6}	$6. \times 10^{-7}$
		$1. \times 10^{-6}$	4.56×10^{-6}	3.6×10^{-7}	5.16×10^{-6}
		3.8×10^{-6}	3.6×10^{-7}	2.66×10^{-6}	1.46×10^{-6}
		$6. \times 10^{-7}$	5.15×10^{-6}	1.46×10^{-6}	4.26×10^{-6}
4 x 4	Radau	4.7×10^{-6}	7.04×10^{-6}	0.00001290	0.0000106
		7.04×10^{-6}	0.00001055	0.0000193	0.0000158
		0.00001290	0.0000193	0.00003542	0.0000290
		0.0000106	0.0000158	0.0000290	0.00002375
	Lobatto	0.0000367	0.00005055	0.0000912	0.00007732
		0.00005055	0.0000803	0.00014125	0.00011148
		0.0000912	0.00014125	0.0002605	0.0002104
		0.00007732	0.00011148	0.0002104	0.00017622
	Legendre	$4. \times 10^{-7}$	$2. \times 10^{-7}$	5.9×10^{-7}	$3. \times 10^{-8}$
		$2. \times 10^{-7}$	4.7×10^{-7}	1.3×10^{-7}	3.9×10^{-7}
		5.9×10^{-7}	1.3×10^{-7}	$0. \times 10^{-9}$	4.5×10^{-7}
		$3. \times 10^{-8}$	7.9×10^{-7}	4.5×10^{-7}	3.1×10^{-7}
5 x 5	Radau	1.8×10^{-7}	2.1×10^{-7}	$4. \times 10^{-7}$	3.4×10^{-7}
		2.1×10^{-7}	4.07×10^{-7}	$6. \times 10^{-7}$	4.6×10^{-7}
		$4. \times 10^{-7}$	$6. \times 10^{-7}$	1.21×10^{-6}	9.3×10^{-7}
		3.4×10^{-7}	4.6×10^{-7}	9.3×10^{-7}	8.1×10^{-7}
	Lobatto	5.2×10^{-7}	1.5×10^{-6}	3.1×10^{-6}	2.1×10^{-6}
		1.5×10^{-6}	1.6×10^{-6}	3.85×10^{-6}	3.8×10^{-6}
		3.1×10^{-6}	3.85×10^{-6}	6.8×10^{-6}	6.02×10^{-6}
		2.1×10^{-6}	3.8×10^{-6}	6.02×10^{-6}	4.2×10^{-6}

In this paper, we use the notations *G, GR*, and *GL* for Gauss-Legendre, Gauss-Radau, and Gauss-Lobatto numerical integration methods, respectively.

The error is calculated as follows

$$Error = \left| (Analytical - Approximate) \right|$$

The computational CPU time for each matrix obtained in Table II, is also given in Table IV. We observe that the CPU time of the exact method is 300 times comparing to the numerical methods: G, GR and GL, even if we use higher order quadrature points. The errors given in Table III are very negligible and the accuracy of the presented methods in this paper is excellent.

Table IV. Comparison of CPU time for element matrix in Table II

Order	Methods and CPU time (in s)			
n x n	G	GR	GL	Exact
3 x 3	0.016	0.015	0.016	
4 x 4	0.031	0.031	0.031	9.437
5 x 5	0.031	0.031	0.031	

Conclusion

The integrals arising in finite element analysis usually involve a huge amount of computing time and memory space. In this paper, a simple and efficient method is 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. For this, three types of numerical Gaussian quadrature rules are used. Here, the numerical integration methods are used for the comparison purpose but not for the integration itself. The procedures are 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. Thus the authours' concluding remark is that not only Gauss-Legendre quadrature rule is efficient but also Gauss-Radau as well as Gauss-Lobatto quadrature rules may be used for the evaluation of element matrices for a general four-node linear convex quadrilateral element.

References

Babolian, E., Dehghan, M. M.R. Eslahchi (2006) Application of Gauss quadrature rule in finding bounds for solution of linear system of equations, *Appl. Math. Computation,* **179:** 707 - 713.

Barrett, K.E. (1999) Explicit eight-noded quadrilateral elements, Finite Elements in Analysis and Design, **31:** 209 - 222.

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

Burden, R.L. Faires, J.D. (2002) - Numerical Analysis, Brooks/Cole, 7th Ed.

Eslahchi, M.R., Masjed-Jamei, M. Babolian, E. (2005) On numerical improvement of Gauss-Lobatto quadrature rules, *Appl. Math. Computation,* **164:** 707 - 717.

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

Masjed-Jamei, M., Eslahchi, M.R. Dehghan, M. (2005) On numerical improvement of Gauss-Radau quadrature rules, *Appl. Math. Computation,* **168**: 51 - 64.

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.

Stroud, A.H. (1974) Numerical Quadrature and Solution of Ordinary Differential Equations, Springer-Verlag, New York Berlin Heidelberg.

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

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