INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 21, 1129–1148 (1985)

HIGH DEGREE EFFICIENT SYMMETRICAL GAUSSIAN QUADRATURE RULES FOR THE TRIANGLE

D. A. DUNAVANT[†]

School of Civil Engineering, Purdue University, West Lafayette, Indiana, U.S.A.

SUMMARY

Gaussian quadrature is required for the computation of matrices based on the isoparametric formulation of the finite element method. A brief review of existing quadrature rules for the triangle is given, and the method for the determination of high degree efficient symmetrical rules for the triangle is discussed. New quadrature rules of degree 12–20 are presented, and a short FORTRAN program is included.

INTRODUCTION

Gaussian quadrature is employed when the integration of a function in terms of elementary functions cannot be performed without difficulty. Quadrature is a widely used method of numerical analysis, and is required for the computation of various matrices based on the isoparametric formulation of the finite element method.

A shape of elements frequently used is triangular, and we consider the natural triangle of area A as shown in Figure 1, where natural co-ordinates (α, β, γ) are

$$\alpha = \frac{A_1}{A}, \quad \beta = \frac{A_2}{A}, \quad \gamma = \frac{A_3}{A} \tag{1}$$

and

$$0 = \alpha + \beta + \gamma - 1 \tag{2}$$

The following integral is frequently required:

$$\int_A f(\alpha, \beta, \gamma)\,\mathrm{d}A \tag{3}$$

Integration is performed by a Gaussian quadrature rule such that

$$\int_A f(\alpha, \beta, \gamma)\,\mathrm{d}A = A \sum_{i=1}^{ng} w_i f(\alpha_i, \beta_i, \gamma_i) \tag{4}$$

where for the ith Gaussian point of location $(\alpha_i, \beta_i, \gamma_i)$, there corresponds a Gaussian weight w_i and functional evaluation $f(\alpha_i, \beta_i, \gamma_i)$. All but two[1,2] of the previously developed rules are based on the assumption that f is a simple and complete polynomial of highest order p. The error in quadrature is zero if the number of points ng is of sufficient magnitude.

[†] Assistant Professor of Structural Engineering.

0029–5981/85/061129–20$02.00

Received 3 April 1984
Revised 11 July 1984

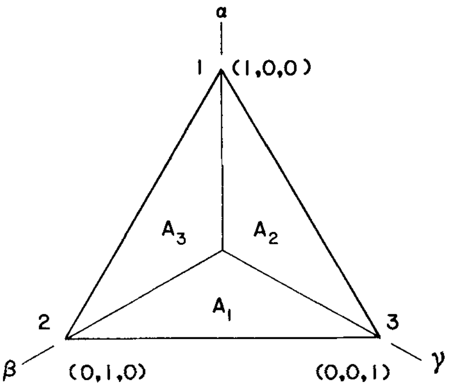

Figure 1. Natural triangle and co-ordinates

Isoparametric formulation involves a mapping transformation between a different co-ordinate system, say x–y, in which integration is required, and the natural co-ordinate system. Since the assumed displacement functions are usually expressed by polynomial representation, f will include zero or more products of zero or more derivatives of those polynomials. For example, a consistent mass matrix of order p results from assumed displacement functions of order $p/2$. The function f will also include J, the determinant of the Jacobian. If the sides of the x–y triangle are straight and the nodal spacing uniform, the mapping transformation will be linear, J scalar, f non-rational and non-singular polynomials, and the error in approximation by quadrature zero for sufficient ng. If the sides are curved, the map will be nonlinear, J and also f rational and non-singular polynomials, and the error will converge to zero as ng is progressively increased. However, if the sides are straight and the nodes are placed in certain locations,[3] the map will be nonlinear, J and also f rational and singular polynomials, and the error will not necessarily converge as ng is progressively increased. Specialized rules suitable for the strength of singularity[1] should be used if accuracy is necessary.

GAUSSIAN PRODUCT AND SYLVESTER QUADRATURE RULES

Product and Sylvester rules are sometimes used for quadrature over a triangular area. There are advantages and disadvantages to both.

1. Gaussian product rules may be formed by successive application of one-dimensional Gaussian rules. First, since it is known that[4]

$$\mathrm{d}A = 2A\,\mathrm{d}\alpha\,\mathrm{d}\beta \tag{5}$$

Equation (3) may be written

$$2A\int_{\alpha=0}^{1}\int_{\beta=0}^{1-\alpha} f(\alpha,\beta,\gamma)\,\mathrm{d}\beta\,\mathrm{d}\alpha \tag{6}$$

The two changes of variables are

$$\alpha=\frac{1+u}{2} \tag{7a}$$

$$\begin{aligned}\beta&=\frac{(1-\alpha)(1+v)}{2}\\&=\frac{(1-u)(1+v)}{4}\end{aligned} \tag{7b}$$

Substitution yields

$$\frac{A}{4}\int_{u=-1}^{1}(1-u)\int_{v=-1}^{1}f(\alpha,\beta,\gamma)\,\mathrm{d}v\,\mathrm{d}u \tag{8}$$

By successive application of one-dimensional rules

$$\int_A f(\alpha,\beta,\gamma)\,\mathrm{d}A=\frac{A}{4}\sum_{i=1}^{m}w_i(1-u_i)\sum_{j=1}^{n}w_jf(\alpha_i,\beta_i,\gamma_i) \tag{9}$$

The Gaussian points and weights in the u or α direction are u_i and w_i, and in the v or β direction are v_j and w_j. The number of points m and n in each direction may be different, but the lower value will control. The type of rule in each direction may also be different and, for example, Radau and Legendre rules have been used.[4] Usually, m and n will be identical, as will the two types of rules.

Product rules have several advantages. Their derivation and application is straightforward. They are versatile in that many one-dimensional rules are available for several different integrands.[5] Extremely high-order polynomials may be evaluated, although precision may be limited since most references provide points and weights to 20 significant digits at most.

The primary disadvantage is inefficiency since for high p, a relatively large number of points is required, and other quadrature rules are available with many fewer points. For one-dimensional Legendre rules, quadrature will be exact with n points if the $2n$th derivative of the integrand is zero.[5] For Legendre product rules, quadrature will be exact with n^2 points if the $2n$th derivative of the function f is zero. The secondary disadvantage is that the location of the points is unsymmetrical. Except for rules of low degree, a large number of points will be concentrated in a relatively small region near one vertex. Such an arrangement, although correct, may be considered aesthetically undesirable.

2. Sylvester rules[6] are extensions to the triangle of one-dimensional Newton–Cotes rules of either closed or open type. Although locations of points are symmetrical, such rules require more points than product rules. A triangular layout of points for which there are $n+1$ points along a side, and thus a total number of $n(n+1)/2$ points, would integrate a polynomial of degree n.

MOMENT EQUATIONS FOR EFFICIENT SYMMETRICAL GAUSSIAN QUADRATURE RULES

The beginning of the development of such rules is apparently Reference 7, where it is stated that 'the general theory is missing, but a general type of method is proposed for which illustrations are provided'. The illustrations are rules for $p=1,2,3$ and 5. It is also stated that 'there are known hyperefficient formulas', which is now thought to be unlikely. The general method is briefly discussed in Reference 8, and rules for $p=4$, 6 and 7 are given. Additional rules are also given for $p=3$, 4 and 5, but have more points than are necessary.

The method will be described as follows. An arbitrary complete polynomial of order p having np terms is assumed for the function f, where

$$np=\frac{(p+1)(p+2)}{2} \tag{10}$$

For example, if $p=2$,

$$f(\alpha,\beta,\gamma)=[\alpha^2\,\beta^2\,\gamma^2\,\alpha\beta\,\beta\gamma\,\gamma\alpha]\{a\} \tag{11}$$

It is more convenient, and equivalent because of (2), to alternatively assume

$$f(\alpha,\beta)=[1\ \alpha\ \beta\ \alpha^2\ \alpha\beta\ \beta^2]\{a\} \tag{12}$$

The arbitrary polynomial coefficients are $\{a\}$. With the following relationship[9]:

$$\int_A \alpha^i\beta^j\,\mathrm{d}A=2A\frac{i!j!}{(i+j+2)!} \tag{13}$$

The left-hand side of (4) is then

$$\int_A f(\alpha,\beta)\,\mathrm{d}A=\frac{A}{12}[12\,4\,4\,2\,1\,2]\{a\} \tag{14}$$

The right-hand side of (4) is

$$\begin{aligned}A\sum_{i=1}^{ng}w_i f(\alpha_i,\beta_i)&=Aw_1[1\ \alpha_1\ \beta_1\ \alpha_1^2\ \alpha_1\beta_1\ \beta_1^2]\{a\}+\cdots\\&+Aw_i[1\ \alpha_i\ \beta_i\ \alpha_i^2\ \alpha_i\beta_i\ \beta_i^2]\{a\}+\cdots\\&+Aw_{ng}[1\ \alpha_{ng}\ \beta_{ng}\ \alpha_{ng}^2\ \alpha_{ng}\beta_{ng}\ \beta_{ng}^2]\{a\}\end{aligned} \tag{15}$$

For (4) to hold true, we equate the right-hand sides of (14) and (15), and for arbitrary $\{a\}$, the resulting moment equations are

$$\begin{aligned}0&=\sum_{i=1}^{ng}w_i-1\\0&=\sum_{i=1}^{ng}w_i\alpha_i-\tfrac{1}{3}\\0&=\sum_{i=1}^{ng}w_i\beta_i-\tfrac{1}{3}\\0&=\sum_{i=1}^{ng}w_i\alpha_i^2-\tfrac{1}{6}\\0&=\sum_{i=1}^{ng}w_i\alpha_i\beta_i-\tfrac{1}{12}\\0&=\sum_{i=1}^{ng}w_i\beta_i^2-\tfrac{1}{6}\end{aligned} \tag{16a–f}$$

The system of nonlinear equations is not independent, and reduces to

$$0=\sum_{i=1}^{ng}w_i-1 \tag{17a}$$

$$0=\sum_{i=1}^{ng}w_i\gamma_i^2-\tfrac{1}{6} \tag{17b}$$

For higher values of p, a similar reduction in the number of equations becomes difficult by algebraic manipulation. However, the reduction can be accomplished directly if, instead of the natural triangle, the triangle shown in Figure 2 is considered, and the polynomials are expressed with deMoivre's theorem in polar co-ordinates.[11] Two different equations have been developed[10,11] which provide identical results for m, the number of independent equations, the

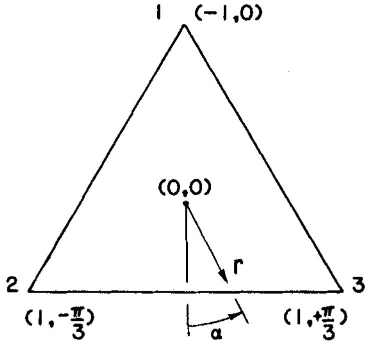

Figure 2. Alternative triangle and polar co-ordinates

simpler being[11]

$$m = \frac{(p+3)^2 + \alpha_p}{12} \tag{18}$$

where

$$\alpha_p = +3, -4, -1, 0, -1, -4 \tag{19}$$

for

$$p = 0, 1, 2, 3, 4, 5 \tag{20}$$

and

$$\alpha_{p+6} = \alpha_p \tag{21}$$

The moment equations are,[11] with a minor correction

$$w_0 + \sum_{i=1}^{n_1} w_i + \sum_{i=n_1+1}^{n_1+n_2} w_i = v_{0,0} = 1 \tag{22a}$$

$$\sum_{i=1}^{n_1} w_i r_i^j + \sum_{i=n_1+1}^{n_1+n_2} w_i r_i^j \cos(3k\alpha_i) = v_{j,3k} \tag{22b}$$

for

$$2 \leqslant j \leqslant p \tag{23}$$

and

$$0 \leqslant 3k \leqslant j \tag{24}$$

subject to

$$j + 3k = \text{even} \tag{25}$$

The polar moments $v_{j,3k}$ are[11]

$$v_{j,3k} = \frac{1}{A}\int_A r^j \mathrm{e}^{i3k\alpha} r \,\mathrm{d}r\,\mathrm{d}\alpha \tag{26}$$

$$= \frac{1}{A}\int_A r^{j+1} \mathrm{e}^{i3k\alpha} \mathrm{d}r\,\mathrm{d}\alpha \tag{27}$$

By symmetry, we need only consider the region of the triangle $-\pi/3 \leqslant \alpha \leqslant +\pi/3$[12], and with the relationship which applies only the edge

$$r = \frac{1}{2\cos\alpha} \tag{28}$$

It follows that

$$v_{j,3k} = \frac{3}{A}\int_{-\pi/3}^{\pi/3}\int_{0}^{1/2\cos\alpha} r^{j+1}\,\mathrm{d}r e^{i3k\alpha}\,\mathrm{d}\alpha \tag{29}$$

With the substitutions of

$$A = \frac{3\sqrt{3}}{4} \tag{30}$$

and

$$e^{i3k\alpha} = \cos 3k\alpha + i\sin 3k\alpha \tag{31}$$

For real $v_{j,3k}$, we obtain

$$v_{j,3k} = \frac{1}{\sqrt{3}(j+2)2^{j}}\int_{-\pi/3}^{\pi/3}\frac{\cos 3k\alpha}{(\cos\alpha)^{j+2}}\,\mathrm{d}\alpha \tag{32}$$

Calculation of $v_{j,3k}$ for relatively small values of j and $3k$ is straightforward, but becomes increasingly difficult for larger values. It becomes necessary to perform the integration with a symbolic algebraic manipulation program.[13] However, it is noted that in doing so, it is necessary to use program options which expand a multiple angle function to functions of a single angle. Values of m, $(j, 3k)$ and $v_{j,3k}$ for up to $m = 44$ corresponding to $p = 20$ are given in Table I. Gaussian points and weights can be classified into one of three groups, as given in Table II.

The sum of the weights per groups is w_i. The radial distance from the centre to a point in the group is r_i. For the n_1 groups, all three points lie on the trisecting medians of the triangle. For the median $\alpha = 0$, if $r_i > 0$ then the triangle formed by the three points as vertices is termed 'inverted'

Table I. Polar moments $v_{j,3k}$

p	m	$j, 3k$	$v_{j,3k}$
1	1	0, 0	+ 1
2	2	2, 0	+ 1/4
3	3	3, 3	− 1/10
4	4	4, 0	+ 1/10
5	5	5, 3	− 2/35
	6	6, 0	+ 29/560
6	7	6, 6	+ 1/28
7	8	7, 3	− 1/28
	9	8, 0	+ 11/350
8	10	8, 6	+ 1/40
	11	9, 3	− 37/1540
9	12	9, 9	− 1/55
	13	10, 0	+ 13/616
10	14	10, 6	+ 1/55
	15	11, 3	− 49/2860
11	16	11, 9	− 2/143
	17	12, 0	+ 425/28028
	18	12, 6	+ 137/10010
12	19	12, 12	+ 1/91
	20	13, 3	− 64/5005
13	21	13, 9	− 1/91
	22	14, 0	+ 523/45760
	23	14, 6	+ 85/8008
14	24	14, 12	+ 1/112
	25	15, 3	− 6733/680680
	26	15, 9	− 109/12376
15	27	15, 15	− 1/136
	28	16, 0	+ 217/24310
	29	16, 6	+ 209/24752
16	30	16, 12	+ 1/136
	31	17, 3	− 2909/369512
	32	17, 9	− 65/9044
17	33	17, 15	− 2/323
	34	18, 0	+ 66197/9237800
	35	18, 6	+ 8069/1175720
	36	18, 12	+ 317/51680
18	37	18, 18	+ 1/190
	38	19, 3	− 3769/587860
	39	19, 9	− 77/12920
19	40	19, 15	− 1/190
	41	20, 0	+ 83651/14226212
	42	20, 6	+ 11303/1989680
	43	20, 12	+ 92/17765
20	44	20, 18	+ 1/220

Table II. Groups of Gaussian points and weights

No. of groups	No. of points per group	Unknowns per group	No. of unknowns per group	Remarks
n_0	1	w_0	1	$n_0 = 0, 1$
n_1	3	w_i, r_i	2	
n_2	6	w_i, r_i, α_i	3	$0 < \alpha_i < \pi/3$

relative to the triangle of Figure 2. If $r_i < 0$, the triangle is 'upright'. For the n_2 groups, $r_i > 0$, and $0 < \alpha_i < \pi/3$. The total number of unknowns is n[11]:

$$n = n_0 + 2n_1 + 3n_2 \tag{33}$$

and the number of Gaussian points is ng[11]:

$$ng = n_0 + 3n_1 + 6n_2 \tag{34}$$

SOLUTION OF MOMENT EQUATIONS FOR HIGH-DEGREE GAUSSIAN QUADRATURE RULES

Solution of (22) requires that at least as many unknowns be provided as there are equations:

$$m \leqslant n \tag{35}$$

Appropriate values must first be selected for n_0, n_1 and n_2. For $m = n$, Reference 10 gives those values and corresponding quadrature rules for $p \leqslant 10$, and prediction of values of n_0, n_1 and n_2 for $p \geqslant 11$. Also for $m = n$, Reference 11 gives values and rules for $p \leqslant 9$ and 11, and equations for

Table III. Predicted values of (n_0, n_1, n_2) and confirmed values of n_2 min

	Product	Reference 10				Reference 11				Confirmed
p	ng	n_0	n_1	n_2	ng	n_0	n_1	n_2	ng	n_2 min
1	1	1	0	0	1	1	0	0	1	0
2	4	0	1	0	3	0	1	0	3	0
3	4	1	1	0	4	1	1	0	4	0
4	9	0	2	0	6	0	2	0	6	0
5	9	1	2	0	7	1	2	0	7	0
6	16	0	2	1	12	0	2	1	12	1
7	16	1	2	1	13	1	2	1	13	1
8	25	1	3	1	16	1	3	1	16	1
9	25	1	4	1	19	1	4	1	19	1
10	36	0	4	2	24	0	4	2	24	2
11	36	0	5	2	27	0	5	2	27	2
12	49	0	5	3	33	0	5	3	33	3
13	49	0	6	3	36	0	6	3	36	3
14	64	1	7	3	40	0	6	4	42	4
15	64	0	9	3	45	1	7	4	46	5
16	81	0	9	4	51	1	7	5	52	5
17	81	1	10	4	55	1	7	6	58	6
18	100	0	11	5	63	0	8	7	66	7
19	100	1	12	5	67	1	9	7	70	8
20	121	1	14	5	73	0	10	8	78	8

prediction of values for $p \geqslant 12$. Table III gives both sets of values for $p \leqslant 20$, and for each the corresponding number of points ng. We also give the number of points required for a Legendre product rule.

Both references give identical values of n_0, n_1 and n_2 for $p \leqslant 10$, and the quadrature rules for $p \leqslant 9$ given in each are identical. For $p = 10$, Reference 10 was unable to obtain a rule for 24 points, but a 25-point rule was given. Reference 11 did not provide a rule for $p = 10$, but a rule for $p = 11$ was given.

The correctness of rules for $p \leqslant 11$ was confirmed by writing moment equations based on (12) for which the derivation is much simpler than (22). Various library packages are available for the solution of systems of nonlinear equations, which is one of the most difficult of all areas of numerical analysis, and a comparative study has been made.[14] The IMSL library,[15] containing subroutines ZSCNT, ZSPOW and ZXSSQ, was used and all three require an initial estimate of the solution.

Knowing for each value of p the values of n_0, n_1 and n_2, it was assumed that all weights were zero. For the n_1 groups, the number of triangles assumed inverted is inv:

$$0 \leqslant \text{inv} \leqslant n_1 \tag{36}$$

For the inv inverted triangles, one point was assumed at $r = 1/2$, with equal spaces between the remaining points. Similarly, for the remaining upright triangles, one point was assumed at $r = -1$, with equal spaces between the remaining points. For the n_2 triangles, it was assumed that the angular spacing was equal between radial lines, and the points were located closely to the outer edge.

Computations were performed in double precision on a DEC VAX 11/780, and the results are given in Table IV, where X indicates that a rule was not obtained, and a number indicates that a rule was obtained, for that order. With the exception of $p = 2$, a rule was never obtained, when the number of inverted triangles exceeded the number of upright triangles:

$$\text{inv} \leqslant \text{int}\left(\frac{n_1}{2}\right) \tag{37}$$

where int denotes the integer part.

ZXSSQ proved to be the superior subroutine. It converged much faster than ZSCNT. It appeared to converge when the other two at times failed, a conclusion reached elsewhere[14] for similar subroutines based on a Levenberg–Marquardt algorithm. Further, ZXSSQ did not require that the number of equations and unknowns be equal, and relaxed the restriction of $m = n$ imposed by ZSCNT and ZSPOW.

All three subroutines employ iterative techniques, and the computational time for solution increased dramatically as p approached 11. It was evident that if rules for higher p were to be found, a substantial reduction of time was required. Execution was performed on the much faster CDC 6500 and 6600 machines, but the time was unacceptable. Automatic vectorization of ZXSSQ and execution on a Cyber 205 supercomputer still required excessive computational time. At this point, the moment equations (22) were used, which reduced the number of np equations to a minimum by a factor of about six.

We now discuss how new rules for $p \geqslant 12$ were achieved. Table III shows that the two predictions of n_0, n_1 and n_2 are different for $p \geqslant 14$. The allowance of $m \leqslant n$ of ZXSSQ made it possible to experimentally confirm that Reference 11 was correct. For example, it is not possible to obtain a rule for $p = 6$ regardless of the value of n_1 if $n_2 = 0$. An identical procedure was used for $p \leqslant 20$, and the only unconfirmed values were $p = 15$ and 19, although Reference 11 is probably correct. In addition to (35), the following requirement must be met:

$$n_2 \geqslant n_2 \min \tag{38}$$

Table IV. Values of (n_0, n_1, n_2) and inv for quadrature rules

					inv					
p	n_0	n_1	n_2	ng	0	1	2	3	4	5
1	1	0	0	1	1					
2	0	1	0	3	2	2				
3	1	1	0	4	3	X				
4	0	2	0	6	X	4	X			
5	1	2	0	7	X	5	X			
6	0	2	1	12	6	X	X			
7	1	2	1	13	7	X	X			
8	1	3	1	16	X	8	X	X		
9	1	4	1	19	X	X	9	X	X	
10	0	4	2	24	X	X	X	X	X	
10	1	4	2	25	X	X	X	X	X	
10	1	2	3	25	10	10	X			
11	0	5	2	27			11			
12	0	5	3	33			12			
13	0	6	3	36	X	X	X	X	X	
13	1	6	3	37	X	X	X	13		
14	0	6	4	42	X	X	14	X		
15	1	7	4	46	X	X	X	X	X	X
15	0	8	4	48	X	X	X	X	X	
15	0	6	5	48	X	X	15			
16	1	7	5	52	X	X	X	16		
17	1	7	6	58	X	X	X	X	X	
17	0	8	6	60					X	
17	0	6	7	60			X			
17	1	8	6	61					17	
18	0	8	7	66				X		
18	1	8	7	67			X	X	X	
18	1	6	8	67			X	X		
18	0	7	8	69			X	X		
18	1	9	7	70					18	
19	1	9	7	70				X	X	
19	1	10	7	73						X
19	1	8	8	73				19	X	
20	0	10	8	78	X	X			X	X
20	1	10	8	79					20	X

The closeness of the initial estimate to the rule became more important than for $p \leqslant 11$. For Legendre rules, the weights are dependent on the location of the points, and a graphical relationship has to be established.[16] However, the only relationship found here was that w_i decreased with increasing r_i. As p approached 20, the weights were adjusted accordingly, otherwise it was sufficient to assume equal weights for all ng points:

$$w_0 = \frac{1}{ng} \qquad n_0 = 1$$

$$w_i = \frac{3}{ng} \qquad 1 \leqslant i \leqslant n_1 \tag{39a–c}$$

$$w_i = \frac{6}{ng} \qquad n_1 + 1 \leqslant i \leqslant n_1 + n_2$$

The initial estimate of the point locations appeared to be more critical than the weights, which suggests a similarity to Legendre rules, for which the weights depend on the point locations. Orthogonal polynomials for the triangle have been developed,[17,18] and a correlation has been established between the zeros of the polynomials and quadrature rules.[18] It has also been stated that initial estimates may be obtained by a graphical means.[18] However, attempts to extend either of these two techniques to rules for higher values of p did not prove successful.

The point locations of Legendre rules are more closely spaced away from the centroid. Similarly, as p approached 20, the initial estimate of the n_1 points were adjusted accordingly, otherwise it was sufficient to assume equal spacing as discussed previously. Similarly, the estimate of the n_2 points followed two guidelines developed from observation of previously existing rules. First, points preferred locations closer to the edge of the triangle than the centroid. Second, locations were preferred closer to the median $\alpha = \pi/3$ than $\alpha = 0$. The result was that points were spaced along the edge, and the spacing was closer near the vertex.

Quadrature rules are optimal when the number of Gaussian points ng equals that predicted by Reference 11. Attempts to locate such rules first assumed inv to be maximum from (37). If a rule was not found, inv was decreased until it was felt that an optimal rule would not be found. The next larger number of Gaussian points was assumed, for which combinations of values of n_0, n_1 and n_2 satisfied (35) and (38). The procedure was repeated until a rule was found. The results are given in Table IV, and show that optimal rules are increasingly difficult to locate as p is increased. However, in all instances rules were located which are quasi-optimal, since ng was only slightly increased.

Quadrature rules for $1 \leqslant p \leqslant 20$ are given in Appendix II. The FORTRAN program source listing is given in Appendix III, and may be used either to calculate the rules to greater precision if necessary, or to extend the rules to higher degree. All computations were performed on CDC 6500 and 6600 machines. The CDC Extended Fortran Version 4 compiler was used, since other compilers sometimes aborted execution with a negative square root argument. Execution in single precision eventually resulted in roundoff problems, and so double precision with approximately 28 digits of precision was used. However, this considerably increased the execution time, which was nearly 3 hours for $p = 20$.

CONCLUSIONS

The Gaussian quadrature rules discussed in this paper are symmetrical and are at present the most economically available. The number of moment equations for which the solution is the Gaussian points and weights have been reduced to a minimum by Lyness and Jespersen.[11] However, the solution of such a system of nonlinear equations requires an initial estimate and employs iterative techniques which are computationally intensive. Guidelines are presented for the estimation of the solution, and available quadrature rules for the triangle have been increased from degree 11 to degree 20.

ACKNOWLEDGEMENTS

The author wishes to thank Professor R. H. Lee of Purdue University for generously given advice and assistance during the course of this investigation. He also is grateful to Professor D. R. Brown for the provision of resources of the Purdue University Computing Center.

APPENDIX I: NOTATION FOR APPENDIX II

p	= Order of complete polynomial.
ng	= Number of Gaussian points and weights.
nsig	= Number of significant digits.
ssq	= Sum of the squares of the residuals of the moment equations.
error	= Relative difference between exact integration and numerical quadrature.
ifn	= Number of calls to subroutine func.
infer	= 1 indicates that nsig criterion of 15 was achieved.
time	= CPU time, in seconds, required for solution of the moment equations.

APPENDIX II

```
 p= 1 ng= 1 nsig=29.0 ssq=0.      error= 0.      ifn=    2 infer=4 time=     0
      weight             alpha                beta                  gamma
 1.000000000000000  0.333333333333333  0.333333333333333  0.333333333333333

 p= 2 ng= 3 nsig=29.0 ssq=0.      error= 8.d-29 ifn=    3 infer=4 time=     0
      weight             alpha                beta                  gamma
 0.333333333333333  0.666666666666667  0.166666666666667  0.166666666666667

 p= 3 ng= 4 nsig=22.5 ssq=2.d-58 error= 2.d-28 ifn=   95 infer=1 time=     0
      weight             alpha                beta                  gamma
-0.562500000000000  0.333333333333333  0.333333333333333  0.333333333333333
 0.520833333333333  0.600000000000000  0.200000000000000  0.200000000000000

 p= 4 ng= 6 nsig=23.0 ssq=2.d-58 error= 2.d-28 ifn=   86 infer=1 time=     0
      weight             alpha                beta                  gamma
 0.223381589678011  0.108103018168070  0.445948490915965  0.445948490915965
 0.109951743655322  0.816847572980459  0.091576213509771  0.091576213509771

 p= 5 ng= 7 nsig=24.1 ssq=4.d-58 error= 2.d-28 ifn=  111 infer=1 time=     0
      weight             alpha                beta                  gamma
 0.225000000000000  0.333333333333333  0.333333333333333  0.333333333333333
 0.132394152788506  0.059715871789770  0.470142064105115  0.470142064105115
 0.125939180544827  0.797426985353087  0.101286507323456  0.101286507323456

 p= 6 ng=12 nsig=22.7 ssq=2.d-58 error= 2.d-28 ifn=  233 infer=1 time=     2
      weight             alpha                beta                  gamma
 0.116786275726379  0.501426509658179  0.249286745170910  0.249286745170910
 0.050844906370207  0.873821971016996  0.063089014491502  0.063089014491502
 0.082851075618374  0.053145049844817  0.310352451033784  0.636502499121399

 p= 7 ng=13 nsig=26.1 ssq=2.d-57 error= 3.d-28 ifn=  452 infer=1 time=     2
      weight             alpha                beta                  gamma
-0.149570044467682  0.333333333333333  0.333333333333333  0.333333333333333
 0.175615257433208  0.479308067841920  0.260345966079040  0.260345966079040
 0.053347235608838  0.869739794195568  0.065130102902216  0.065130102902216
 0.077113760890257  0.048690315425316  0.312865496004874  0.638444188569810

 p= 8 ng=16 nsig=15.2 ssq=7.d-59 error= 4.d-28 ifn=  410 infer=1 time=     6
      weight             alpha                beta                  gamma
 0.144315607677787  0.333333333333333  0.333333333333333  0.333333333333333
 0.095091634267285  0.081414823414554  0.459292588292723  0.459292588292723
 0.103217370534718  0.658861384496480  0.170569307751760  0.170569307751760
 0.032458497623198  0.898905543365938  0.050547228317031  0.050547228317031
 0.027230314174435  0.008394777409958  0.263112829634638  0.728492392955404

 p= 9 ng=19 nsig=16.3 ssq=1.d-57 error= 6.d-28 ifn=  713 infer=1 time=     7
      weight             alpha                beta                  gamma
 0.097135796282799  0.333333333333333  0.333333333333333  0.333333333333333
 0.031334700227139  0.020634961602525  0.489682519198738  0.489682519198738
 0.077827541004774  0.125820817014127  0.437089591492937  0.437089591492937
 0.079647738927210  0.623592928761935  0.188203535619033  0.188203535619033
 0.025577675658698  0.910540973211095  0.044729513394453  0.044729513394453
 0.043283539377289  0.036838412054736  0.221962989160766  0.741198598784498
```

```
p=10 ng=25 nsig=19.0 ssq=5.d-58 error= 6.d-28 ifn= 1059 infer=1 time=   39
     weight              alpha              beta                gamma
0.090817990382754  0.333333333333333  0.333333333333333  0.333333333333333
0.036725957756467  0.028844733232685  0.485577633383657  0.485577633383657
0.045321059435528  0.781036849029926  0.109481575485037  0.109481575485037
0.072757916845420  0.141707219414880  0.307939838764121  0.550352941820999
0.028327242531057  0.025003534762686  0.246672560639903  0.728323904597411
0.009421666963733  0.009540815400299  0.066803251012200  0.923655933587500

p=11 ng=27 nsig=23.5 ssq=4.d-58 error= 6.d-28 ifn= 1668 infer=1 time=   34
     weight              alpha              beta                gamma
0.000927006328961 -0.069222096541517  0.534611048270758  0.534611048270758
0.077149534914813  0.202061394068290  0.398969302965855  0.398969302965855
0.059322977380774  0.593380199137435  0.203309900431282  0.203309900431282
0.036184540503418  0.761298175434837  0.119350912282581  0.119350912282581
0.013659731002678  0.935270103777448  0.032364948111276  0.032364948111276
0.052337111962204  0.050178138310495  0.356620648261293  0.593201213428213
0.020707659639141  0.021022016536166  0.171488980304042  0.807489003159792

p=12 ng=33 nsig=17.8 ssq=9.d-58 error= 1.d-27 ifn= 2439 infer=1 time=   74
     weight              alpha              beta                gamma
0.025731066440455  0.023565220452390  0.488217389773805  0.488217389773805
0.043692544538038  0.120551215411079  0.439724392294460  0.439724392294460
0.062858224217885  0.457579229975768  0.271210385012116  0.271210385012116
0.034796112930709  0.744847708916828  0.127576145541586  0.127576145541586
0.006166261051559  0.957365299093579  0.021317350453210  0.021317350453210
0.040371557766381  0.115343494534698  0.275713269685514  0.608943235779788
0.022356773202303  0.022838332222257  0.281325580989940  0.695836086787803
0.017316231108659  0.025734050548330  0.116251915907597  0.858014033544073

p=13 ng=37 nsig=16.9 ssq=2.d-57 error= 8.d-28 ifn= 3804 infer=1 time=  188
     weight              alpha              beta                gamma
0.052520923400802  0.333333333333333  0.333333333333333  0.333333333333333
0.011280145209330  0.009903630120591  0.495048184939705  0.495048184939705
0.031423518362454  0.062566729780852  0.468716635109574  0.468716635109574
0.047072502504194  0.170957326397447  0.414521336801277  0.414521336801277
0.047363586536355  0.541200855914337  0.229399572042831  0.229399572042831
0.031167529045794  0.771151009607340  0.114424495196330  0.114424495196330
0.007975771465074  0.950377217273082  0.024811391363459  0.024811391363459
0.036848402728732  0.094853828379579  0.268794997058761  0.636351174561660
0.017401463303822  0.018100773278807  0.291730066734288  0.690169159986905
0.015521786839045  0.022233076674090  0.126357385491669  0.851409537834241

p=14 ng=42 nsig=18.7 ssq=8.d-58 error= 7.d-28 ifn= 5612 infer=1 time=  278
     weight              alpha              beta                gamma
0.021883581369429  0.022072179275643  0.488963910362179  0.488963910362179
0.032788353544125  0.164710561319092  0.417644719340454  0.417644719340454
0.051774104507292  0.453044943382323  0.273477528308839  0.273477528308839
0.042162588736993  0.645588935174913  0.177205532412543  0.177205532412543
0.014433699669777  0.876400233818255  0.061799883090873  0.061799883090873
0.004923403602400  0.961218077502598  0.019390961248701  0.019390961248701
0.024665753212564  0.057124757403648  0.172266687821356  0.770608554774996
0.038571510787061  0.092916249356972  0.336861459796345  0.570222290846683
0.014436308113534  0.014646950055654  0.298372882136258  0.686980167808088
0.005010228838501  0.001268330932872  0.118974497696957  0.879757171370171
```

```
p=15 ng=48 nsig=23.8 ssq=3.d-57 error= 8.d-28 ifn= 9874 infer=1 time=  634
     weight               alpha               beta                gamma
0.001916875642849 -0.013945833716486  0.506972916858243  0.506972916858243
0.044249027271145  0.137187291433955  0.431406354283023  0.431406354283023
0.051186548718852  0.444612710305711  0.277693644847144  0.277693644847144
0.023687735870688  0.747070217917492  0.126464891041254  0.126464891041254
0.013289775690021  0.858383228050628  0.070808385974686  0.070808385974686
0.004748916608192  0.962069659517853  0.018965170241073  0.018965170241073
0.038550072599593  0.133734161966621  0.261311371140087  0.604954466893291
0.027215814320624  0.036366677396917  0.388046767090269  0.575586555512814
0.002182077366797 -0.010174883126571  0.285712220049916  0.724462663076655
0.021505319847731  0.036843869875878  0.215599664072284  0.747556466051838
0.007673942631049  0.012459809331199  0.103575616576386  0.883964574092416

p=16 ng=52 nsig=15.1 ssq=4.d-57 error= 1.d-27 ifn=21626 infer=1 time= 1669
     weight               alpha               beta                gamma
0.046875697427642  0.333333333333333  0.333333333333333  0.333333333333333
0.006405878578585  0.005238916103123  0.497380541948438  0.497380541948438
0.041710296739387  0.173061122901295  0.413469438549352  0.413469438549352
0.026891484250064  0.059082801866017  0.470458599066991  0.470458599066991
0.042132522761650  0.518892500060958  0.240553749969521  0.240553749969521
0.030000266842773  0.704068411554854  0.147965794222573  0.147965794222573
0.014200098925024  0.849069624685052  0.075465187657474  0.075465187657474
0.003582462351273  0.966807194753950  0.016596402623025  0.016596402623025
0.032773147460627  0.103575692245252  0.296555596579887  0.599868711174861
0.015298306248441  0.020083411655416  0.337723063403079  0.642193524941505
0.002386244192839 -0.004341002614139  0.204748281642812  0.799592720971327
0.019084792755899  0.041941786468010  0.189358492130623  0.768699721401368
0.006850054546542  0.014317320230681  0.085283615682657  0.900399064086661

p=17 ng=61 nsig=27.8 ssq=8.d-54 error= 3.d-27 ifn=33194 infer=1 time= 4693
     weight               alpha               beta                gamma
0.033437199290803  0.333333333333333  0.333333333333333  0.333333333333333
0.005093415440507  0.005658918886452  0.497170540556774  0.497170540556774
0.014670864527638  0.035647354750751  0.482176322624625  0.482176322624625
0.024350878353672  0.099520061958437  0.450239969020782  0.450239969020782
0.031107550868969  0.199467521245206  0.400266239377397  0.400266239377397
0.031257111218620  0.495717464058095  0.252141267970953  0.252141267970953
0.024815654339665  0.675905990683077  0.162047004658461  0.162047004658461
0.014056073070557  0.848248235478508  0.075875882260746  0.075875882260746
0.003194676173779  0.968690546064356  0.015654726967822  0.015654726967822
0.008119655318993  0.010186928826919  0.334319867363658  0.655493203809423
0.026805742283163  0.135440871671036  0.292221537796944  0.572337590532020
0.018459993210822  0.054423924290583  0.319574885423190  0.626001190286228
0.008476868534328  0.012868560833637  0.190704224192292  0.796427214974071
0.018292796770025  0.067165782413524  0.180483211648746  0.752351005937729
0.006665632004165  0.014663182224828  0.080711313679564  0.904625504095608
```

```
p=18 ng=70 nsig=26.3 ssq=7.d-46 error=-2.d-23 ifn=25597 infer=1 time= 3460
     weight              alpha              beta              gamma
 0.030809939937647  0.333333333333333  0.333333333333333  0.333333333333333
 0.009072436679404  0.013310382738157  0.493344808630921  0.493344808630921
 0.018761316939594  0.061578811516086  0.469210594241957  0.469210594241957
 0.019441097985477  0.127437208225989  0.436281395887006  0.436281395887006
 0.027753948610810  0.210307658653168  0.394846170673416  0.394846170673416
 0.032256225351457  0.500410862393686  0.249794568803157  0.249794568803157
 0.025074032616922  0.677135612512315  0.161432193743843  0.161432193743843
 0.015271927971832  0.846803545029257  0.076598227485371  0.076598227485371
 0.006793922022963  0.951495121293100  0.024252439353450  0.024252439353450
-0.002223098729920  0.913707265566071  0.043146367216965  0.043146367216965
 0.006331914076406  0.008430536202420  0.358911494940944  0.632657968856636
 0.027257538049138  0.131186551737188  0.294402476751957  0.574410971510855
 0.017676785649465  0.050203151565675  0.325017801641814  0.624779046792512
 0.018379484638070  0.066329263810916  0.184737559666046  0.748933176523037
 0.008104732808192  0.011996194566236  0.218796800013321  0.769207005420443
 0.007634129070725  0.014858100590125  0.101179597136408  0.883962302273467
 0.000046187660794 -0.035222015287949  0.020874755282586  1.014347260005363

p=19 ng=73 nsig=21.1 ssq=5.d-58 error= 1.d-27 ifn=31956 infer=1 time=10502
     weight              alpha              beta              gamma
 0.032906331388919  0.333333333333333  0.333333333333333  0.333333333333333
 0.010330731891272  0.020780025853987  0.489609987073006  0.489609987073006
 0.022387247263016  0.090926214604215  0.454536892697893  0.454536892697893
 0.030266125869468  0.197166638701138  0.401416680649431  0.401416680649431
 0.030490967802198  0.488896691193805  0.255551654403098  0.255551654403098
 0.024159212741641  0.645844115695741  0.177077942152130  0.177077942152130
 0.016050803586801  0.779877893544096  0.110061053227952  0.110061053227952
 0.008084580261784  0.888942751496321  0.055528624251840  0.055528624251840
 0.002079362027485  0.974756272445543  0.012621863777229  0.012621863777229
 0.003884876904981  0.003611417848412  0.395754787356943  0.600633794794645
 0.025574160612022  0.134466754530780  0.307929983880436  0.557603261588784
 0.008880903573338  0.014446025776115  0.264566948406520  0.720987025817365
 0.016124546761731  0.046933578838178  0.358539352205951  0.594527068955871
 0.002491941817491  0.002861120350567  0.157807405968595  0.839331473680839
 0.018242840118951  0.223861424097916  0.075050596975911  0.701087978926173
 0.010258563736199  0.034647074816760  0.142421601113383  0.822931324069857
 0.003799928855302  0.010161119296278  0.065494628082938  0.924344252620784

p=20 ng=79 nsig=25.6 ssq=4.d-47 error=-5.d-24 ifn=49869 infer=1 time=10258
     weight              alpha              beta              gamma
 0.033057055541624  0.333333333333333  0.333333333333333  0.333333333333333
 0.000867019185663 -0.001900928704400  0.500950464352200  0.500950464352200
 0.011660052716448  0.023574084130543  0.488212957934729  0.488212957934729
 0.022876936356421  0.089726636099435  0.455136681950283  0.455136681950283
 0.030448982673938  0.196007481363421  0.401996259318289  0.401996259318289
 0.030624891725355  0.488214180481157  0.255892909759421  0.255892909759421
 0.024368057676800  0.647023488009788  0.176488255995106  0.176488255995106
 0.015997432032024  0.791658289326483  0.104170855336758  0.104170855336758
 0.007698301815602  0.893862072318140  0.053068963840930  0.053068963840930
-0.000632060497488  0.916762569607942  0.041618715196029  0.041618715196029
 0.001751134301193  0.976836157186356  0.011581921406822  0.011581921406822
 0.016465839189576  0.048741583664839  0.344855770229001  0.606402646106160
 0.004839033540485  0.006314115948605  0.377843269594854  0.615842614456541
 0.025804906534650  0.134316520547348  0.306635479062357  0.559048000390295
 0.008471091054441  0.013973893962392  0.249419362774742  0.736606743262866
 0.018354914106280  0.075549132909764  0.212775724802802  0.711675142287434
 0.000704404677908 -0.008368153208227  0.146965436053239  0.861402717154987
 0.010112684927462  0.026686063258714  0.137726978828923  0.835586957912363
 0.003573909385950  0.010547719294141  0.059696109149007  0.929756171556853
```

APPENDIX III

```
      program main
      implicit double precision (a-h,o-z)
      real t1,t2
      external func
      dimension parm(4),x(45),f(44),xjac(44,45),xjtj(1035),
     +          work(1348),w(19),alpha(19),beta(19),gamma(19)
      common n0,n1,n2,i1b,i1e,i2b,i2e,inu(44,2),dnu(44)
      data parm/4*0.d0/,x/45*0.d0/,f/44*0.d0/,xjac/1980*0.d0/,
     +     xjtj/1035*0.d0/,work/1348*0.d0/,
     +     w/19*0.d0/,alpha/19*0.d0/,beta/19*0.d0/,gamma/19*0.d0/
c     calculate the order of the polynomial
      read (5,10) ip
10    format(i5)
c     calculate the number of groups of multiplicity 1,3 and 6
      read (5,20) n0,n1,n2
20    format(3i5)
c     calculate the sum of the number of groups
      nsum=n0+n1+n2
c     calculate the pointers for n1 and n2
      i1b=n0+1
      i1e=i1b+2*n1-1
      i2b=i1e+1
      i2e=i2b+3*n2-1
c     calculate the number of gaussian points and weights
      ng=n0+3*n1+6*n2
c     calculate the number of unknowns
      n=n0+2*n1+3*n2
c     calculate the number of equations
      ipcopy=ip
      do 30 i=1,10
30    if((ipcopy-6).ge.0) ipcopy=ipcopy-6
      if(ipcopy.eq.0) ialpha=3
      if((ipcopy.eq.1).or.(ipcopy.eq.5)) ialpha=-4
      if((ipcopy.eq.2).or.(ipcopy.eq.4)) ialpha=-1
      if(ipcopy.eq.3) ialpha=0
      m=((ip+3)**2+ialpha)/12
c     calculate the initial approximation
      if(n0.eq.1) read (5,40) x(1)
40    format(d10.0)
      if(n1.ne.0) read (5,50) (x(i),i=i1b,i1e)
50    format(2d10.0)
      if(n2.ne.0) read (5,60) (x(i),i=i2b,i2e)
60    format(3d10.0)
      if(n2.eq.0) go to 80
c     calculate the n2 angle in radians
      pi=4.d0*datan(1.d0)
      do 70 i=i2b,i2e,3
70    x(i+2)=x(i+2)*pi/180.d0
c     calculate inu and dnu
80    call nu(ip,inu,dnu)
c     calculate the polar system weights and points
      call second(t1)
      call zxssq(func,m,n,15,0.d0,0.d0,100000,0,
     +            parm,x,ssq,f,xjac,44,xjtj,work,infer,ier)
      call second(t2)
      itime=int(t2-t1)
      ifn=idint(work(2))
c     calculate the area system weights and points
      call area (x,w,alpha,beta,gamma)
c     calculate the relative error of the integral
      call poly (ip,n0,n1,n2,nsum,w,alpha,beta,gamma,error)
      write(6,90) ip,ng,work(3),ssq,error,ifn,infer,itime
90    format('  p=',i2,' ng=',i2,' nsig=',f4.1,' ssq=',1pd7.0,
     +    ' error=',1pd7.0,' ifn=',i5,' infer=',i1,' time=',i5)
      write(6,100)
100   format(/7x,'weight',14x,'alpha',14x,'beta',15x,'gamma')
      write(6,110) (w(i),alpha(i),beta(i),gamma(i),i=1,nsum)
110   format(4f19.15)
      stop
      end

      subroutine nu(ip,inu,dnu)
      implicit double precision (a-h,o-z)
      dimension inu(44,2),dnu(44)
c     calculate the ith pair (j,3k)
      icount=1
      ipp1=ip+1
      do 10 j=1,ipp1
      jp1=j+1
      do 10 k3=1,jp1,3
      if(mod(j-1+k3-1,2).eq.1) go to 10
      inu(icount,1)=j-1
      inu(icount,2)=k3-1
      icount=icount+1
10    continue
c     calculate dnu(i),i=1,44
      dnu( 1)=+    1.d0
      etc
      dnu(44)=+    1.d0/     220.d0
      return
      end

      subroutine func(x,m,n,f)
      implicit double precision (a-h,o-z)
      dimension x(n),f(m)
      common n0,n1,n2,i1b,i1e,i2b,i2e,inu(44,2),dnu(44)
      do 40 im=1,m
      f(im)=-dnu(im)
      if((im.eq.1).and.(n0.eq.1)) f(1)=f(1)+x(1)
      if(n1.eq.0) go to 20
      do 10 i=i1b,i1e,2
10    f(im)=f(im)+x(i)*x(i+1)**inu(im,1)
20    if(n2.eq.0) go to 40
      do 30 i=i2b,i2e,3
      dk3=dble(float(inu(im,2)))
30    f(im)=f(im)+x(i)*x(i+1)**inu(im,1)*dcos(dk3*x(i+2))
```

```
40    continue
      return
      end

      subroutine area (x,w,alpha,beta,gamma)
      implicit double precision (a-h,o-z)
      dimension x(45),w(19),alpha(19),beta(19),gamma(19)
      common n0,n1,n2,i1b,i1e,i2b,i2e,inu(44,2),dnu(44)
      icount=1
      if(n0.eq.0) go to 10
      w(icount)=x(1)
      alpha(icount)=1.d0/3.d0
      beta(icount)=alpha(icount)
      gamma(icount)=alpha(icount)
      icount=icount+1
10    if(n1.eq.0) go to 30
      do 20 i=i1b,i1e,2
      w(icount)=x(i)/3.d0
      alpha(icount)=(1.d0-2.d0*x(i+1))/3.d0
      beta(icount)=(1.d0-alpha(icount))/2.d0
      gamma(icount)=beta(icount)
20    icount=icount+1
30    if(n2.eq.0) go to 50
      do 40 i=i2b,i2e,3
      w(icount)=x(i)/6.d0
      r=x(i+1)
      a=x(i+2)
      alpha(icount)=(1.d0-2.d0*r*dcos(a))/3.d0
      beta(icount)=(1.d0+r*dcos(a)-dsqrt(3.d0)*r*dsin(a))/3.d0
      gamma(icount)=1.d0-alpha(icount)-beta(icount)
40    icount=icount+1
50    return
      end

      subroutine poly (ip,n0,n1,n2,nsum,w,alpha,beta,gamma,error)
      implicit double precision (a-h,o-z)
      dimension w(19),alpha(19),beta(19),gamma(19)
c     calculate the exact integral
      exact=2.d0*fac(ip)/fac(ip+2)
c     calculate the approximate integral
      sum=0.d0
      if(n0.ne.0) sum=sum+w(1)*alpha(1)**ip
      if(n1.eq.0) go to 20
      n01=n0+1
      n0n1=n0+n1
      do 10 i=n01,n0n1
10    sum=sum+w(i)*(alpha(i)**ip+2.d0*beta(i)**ip)
20    if(n2.eq.0) go to 40
      n0n11=n0+n1+1
      n0n1n2=n0+n1+n2
      do 30 i=n0n11,n0n1n2
30    sum=sum+w(i)*2.d0*(alpha(i)**ip+beta(i)**ip+gamma(i)**ip)
40    approx=sum
```

```
      error=(exact-approx)/exact
      return
      end

      double precision function fac(n)
      implicit double precision (a-h,o-z)
      fac=1.d0
      if(n.eq.0) return
      do 10 i=1,n
10    fac=fac*dble(float(i))
      return
      end

Sample input for p=20

  20
   1   10    8
0.019
0.021     +0.5
0.048     +0.425
0.051     +0.325
0.072     +0.2
0.080     -0.275
0.067     -0.45
0.061     -0.625
0.036     -0.8
0.019     -0.925
0.005     -1.
0.110      0.45     30.
0.068      0.55     30.
0.134      0.375    40.
0.041      0.7      45.
0.098      0.6      50.
0.005      0.8      52.5
0.050      0.725    55.
0.015      0.925    57.5
```

REFERENCES

1. P. Hillion, 'Numerical integration on a triangle', *Int. j. numer. methods eng.*, **11**, 797–815 (1977).
2. C. T. Reddy and D. J. Shippy, 'Alternative integration formulae for triangular finite elements', *Int. j. numer. methods eng.*, **17**, 133–139 (1981).
3. Y. Yamada, Y. Ezawa, I. Nishiguchi and M. Okabe, 'Reconsiderations on singularity or crack tip elements', *Int. j. numer. methods eng.*, **14**, 1525–1544 (1979).
4. R. G. Anderson, B. M. Irons and O. C. Zienkiewicz, 'Vibration and stability of plates using finite elements', *Int. J. Solids Struct.*, 1031–1055 (1968).
5. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, 9th printing, Dover Publications, New York.
6. P. Sylvester, 'Symmetric quadrature formulae for simplexes', *Math. Comput.*, **24**, 95–100 (1970).
7. P. C. Hammer, O. J. Marlowe and A. H. Stroud, 'Numerical integration over simplexes and cones', *Math. Tabl. Natn. Res. Coun. Wash.*, **10**, 130–137 (1956).
8. G. R. Cowper, 'Gaussian quadrature formulas for triangles', *Int. j. numer. methods eng.*, **7**, 405–408 (1973).
9. M. A. Eisenberg and L. E. Malvern, 'On finite element integration in natural coordinates', *Int. j. numer. methods eng.*, **7**, 574–575 (1973).
10. M. E. Laursen and M. Gellert, 'Some criteria for numerically integrated matrices and quadrature formulas for triangles', *Int. j. numer. methods eng.*, **12**, 67–76 (1978).
11. J. N. Lyness and D. Jespersen, 'Moderate degree symmetric quadrature rules for the triangle', *J. Inst. Maths, Applics.*, **15**, 19–32 (1975).

12. J. N. Lyness, Private communication.
13. *MACSYMA Primer*, Laboratory For Computer Science, Massachusetts Institute of Technology, 1978.
14. K. L. Hiebert, 'An evaluation of mathematical software that solves systems of nonlinear equations', *ACM Trans. Math. Software*, **8**, 5–20 (1982).
15. *IMSL Library Reference Manual*, 9th edn, International Mathematical and Statistical Libraries, Houston, Texas, 1981.
16. B. Irons and S. Ahmad, *Techniques of Finite Elements*, Ellis Horwood, Chichester, U.K., 1980.
17. G. M. L. Gladwell, 'Orthogonal polynomials related to triangles', *Proc. Symp. on Computer Aided Engineering*, Univ. of Waterloo, 207–214 (1971).
18. T. Moan, 'Experiences with orthogonal polynomials and best numerical integration formulas on a triangle, with particular reference to finite element approximations', *ZAMM*, **54**, 501–508 (1974).