Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 188 (2007) 865–876
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

Symmetric Gauss Legendre quadrature formulas for composite numerical integration over a triangular surface

H.T. Rathod [a,*], K.V. Nagaraja [b], B. Venkatesudu [b]

[a] *Department of Mathematics, Central College Campus, Bangalore University, Bangalore 560001, India*
[b] *Department of Mathematics, Amrita School of Engineering, # 26 & 27, Kasavanahalli, Carmelram Post, Bangalore 560035, India*

Abstract

This paper first presents a Gauss Legendre quadrature method for numerical integration of $I = \int\int_T f(x,y)\,\mathrm{d}x\,\mathrm{d}y$, where $f(x,y)$ is an analytic function in x, y and T is the standard triangular surface: $\{(x,y)|0 \leqslant x, y \leqslant 1, x+y \leqslant 1\}$ in the Cartesian two dimensional (x,y) space. We then use a transformation $x = x(\xi,\eta)$, $y = y(\xi,\eta)$ to change the integral I to an equivalent integral $\int\int_S f(x(\xi,\eta),y(\xi,\eta))\frac{\partial(x,y)}{\partial(\xi,\eta)}\mathrm{d}\xi\,\mathrm{d}\eta$, where S is now the 2-square in (ξ,η) space: $\{(\xi,\eta)| -1 \leqslant \xi,\eta \leqslant 1\}$. We then apply the one dimensional Gauss Legendre quadrature rules in ξ and η variables to arrive at an efficient quadrature rule with new weight coefficients and new sampling points. We then propose the discretisation of the standard triangular surface T into n^2 right isosceles triangular surfaces T_i $(i = 1(1)n^2)$ each of which has an area equal to $1/(2n^2)$ units. We have again shown that the use of affine transformation over each T_i and the use of linearity property of integrals lead to the result:

$$I = \sum_{i=1}^{n\times n} \int\int_{T_i} f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \frac{1}{n^2}\int\int_T H(X,Y)\,\mathrm{d}X\,\mathrm{d}Y,$$

where $H(X,Y) = \sum_{i=1}^{n\times n} f(x_i(X,Y),y_i(X,Y))$ and $x = x_i(X,Y)$ and $y = y_i(X,Y)$ refer to affine transformations which map each T_i in (x,y) space into a standard triangular surface T in (X,Y) space. We can now apply Gauss Legendre quadrature formulas which are derived earlier for I to evaluate the integral $I = \frac{1}{n^2}\int\int_T H(X,Y)\,\mathrm{d}X\,\mathrm{d}Y$. We observe that the above procedure which clearly amounts to Composite Numerical Integration over T and it converges to the exact value of the integral $\int\int_T f(x,y)\,\mathrm{d}x\,\mathrm{d}y$, for sufficiently large value of n, even for the lower order Gauss Legendre quadrature rules. We have demonstrated this aspect by applying the above explained Composite Numerical Integration method to some typical integrals.

Keywords: Quadrature rules; Weight function; Numerical integration; FEM; Triangle

* Corresponding author.
E-mail address: nagarajaitec123@yahoo.com (K.V. Nagaraja).

0096-3003/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2006.10.041

1. Introduction

In recent years, the Finite Element Method (FEM) has become a very powerful tool for the approximate solution of boundary value problems governing the diverse physical phenomena. Its use in industry and research is extensive and without it many practical problems in science and engineering would be incapable of solution. The triangular elements with either straight sides or curved sides are very widely used in finite element analysis [1–3]. The basic problem of integrating a function of two variables over the surface of the triangle were first given by Hammer et al. [4] and Hammer and Stroud [5,6]. With the advent of finite element method, the triangular elements are proved to be versatile and there has been considerable interest in the area of numerical integration schemes over triangles. Cowper [7] provided a table of Gaussian quadrature formulae for symmetrically placed integration points. Lyness and Jespersen [8] made an elaborate study of symmetric quadrature rules and provided integration formulas with a precision of upto degree 11 by formulating the problem in terms of polar coordinates. Lannoy [9] discussed the symmetric 4-point integration rule, which is presented in Ref. [7]. Laursen and Gellert [10] also gave some new higher order formulas of precision upto degree ten. Dunavant [11] presented some extensions to the integration formulas given by Lyness and Jespersen [8] and also gave tables of integration formulas with precisions of degree from 11 to 20. Laurie [12] derived a 7-point formula and discussed the numerical error in integrating some functions. Sylvester [13] derived some numerical integration formulas for triangles as product of one dimensional Newton Cotes rules of closed type as well as open type. The precision of these integration formulas is again limited to degree ten at most for various reasons. Lethor [14] and Hillion [15] derived formulas for triangles as product of one dimensional Gauss Legendre and Gauss Jacobi quadrature rules. The precision of these formulas is again limited to a degree seven. We also note that higher order quadrature rules of this type cannot be derived beyond degree $15 = 2 \times 8 - 1$ as the abscissas and weights of 1-D Gauss Jacobi quadrature rules are not tabulated even in the standard reference work of Abramowicz and Stegun [16] for a order higher than eighth. Reddy [17] and Reddy and Shippy [18] derived some 3-point, 4-point, 6-point and 7-point formulas of precision 3, 4, 6 and 7, respectively, which gave improved accuracy as compared to some earlier works. Since all the above information on integration formulas which is documented in the works [4–18] is limited to a precision of degree at most 20 and it is not likely that the techniques proposed by these authors can be extended much further to give greater accuracy which may be demanded in future we have taken a significant note from the recent work of Lague and Baldur [19] on the above aspect who gave substantial reasons in favour of the product formulas based only on roots and weights of Gauss Legendre quadrature rules. The use of proposed method on product formulas [19] will remove the restrictions on the derivation of high precision numerical integration formulas and it is clear that now one can obtain formulas of very high degree of precision as the methods rely on standard Gauss Legendre quadrature rules. However Lague and Baldur [19] have not worked out explicit weights and abscissas required for this purpose. Rathod et al. [20–22] provided this information in a systematic manner in their recent works, for the first time.

Integration formulas resulting from interval subdivision and repeated application of a low order formula are called composite numerical integration formulas [23–26]. One way to reduce the error associated with low order integration formula in one dimension is to subdivide the interval of integration, say, $[a, b]$ into smaller intervals and then to use the formula repeatedly on each subinterval. We adopt a strategy similar to the above which is normally used for the treatment of line integrals over arbitrary shaped curves to evaluation of double integrals also. We segment the given region into subregions and effect a transformation over each subregion into a standard region. The success of this strategy follows from the linearity property of double integrals. Repeated application of low order formula is usually preferred to the single application of a high order formula partly because of the lower order formulas and partly because of the computational difficulties one such difficulty is due to the errors introduced because of only a fixed usually small number of digits can be retained after each computer operation. In addition there exist many functions for which the magnitude of the derivative increases without bound as the order of differentiation increases. Therefore a higher order formula may produce a larger error than a lower order one. It is in view of this that the numerical integration formulas employing more than eight points (for Newton Cotes rules) are almost never used. We feel that these important details cannot be simply ignored, and they need to be addressed in great rigor. Hence the derivation of algorithms for composite numerical integration formulas over dimensions higher than one is important for practical applications and it should be used wherever necessary. It is the main purpose of this paper to evolve a practical and workable algorithm for composite numerical integration over triangular surfaces by using the well known Gauss Legendre quadrature rules. We have demonstrated the effectiveness of the above algorithm by applying it to some typical integrals.

2. Formulation of integrals over a triangular area

The finite element method for two dimensional problems with triangular elements requires the numerical integration of shape functions, product of shape function derivatives and rational functions whose denominators are bivariate polynomials, etc. Since an affine transformation makes it possible to transform any triangle into the two dimensional standard triangle T with coordinates $(0,0)$, $(0,1)$, $(1,0)$ in Cartesian frame of (x,y) space (say), we have just to consider numerical integration on T. The integral of an arbitrary function, f, over the surface of a triangle T is given by

$$I = \int\int_T f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \int_0^1 \mathrm{d}x \int_0^{1-x} f(x,y)\,\mathrm{d}y = \int_0^1 \mathrm{d}y \int_0^{1-y} f(x,y)\,\mathrm{d}x. \tag{1}$$

It is now required to find the value of the integral by a quadrature formula

$$I = \sum_{m=1}^{N} c_m f(x_m, y_m), \tag{2}$$

where c_m are the weights associated with sampling points (x_m, y_m) and N is the number total sampling points related to the required precision. One of these methods which have an optimum precision upto a degree 20 is reported in recent work [11]. The other method is approximation of I by product formulas [14,15] which is of type 1(2) based on the roots and weights of Gauss Legendre and Gauss Jacobi quadrature rules. The reported precision of these formulas is limited to a degree seven. This is because the weights and roots of Gauss Jacobi quadrature rules are not tabulated even in the standard reference books of Abramowicz and Stegun [16] beyond a order of precision eight. Use of these will enable us to derive formulas of precision $2 \times 8 - 1 = 15^\circ$ only. The product formulas proposed in this paper and in the recent work [20] are based on the sampling points and weight coefficients of Gauss Legendre quadrature formulas, as this enables us to obtain formulas of very high degree of precision, as Gauss Legendre quadrature rules of order as large as 96 are well documented in Abramowicz and Stegun [16].

The integral I of Eq. (1) can be transformed into an integral over the surface of the square: $\{(u,v)|0 \leqslant u, v \leqslant 1\}$ by the substitution (see Fig 1)

$$x = uv, \quad y = u(1-v). \tag{3}$$

Then the determinant of the Jacobian and the differential area are

$$|J| = \frac{\partial(x,y)}{\partial(u,v)} = \frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u} = (v)(-u) - u(1-v) = u \quad \text{and} \quad \mathrm{d}x\,\mathrm{d}y = \frac{\partial(x,y)}{\partial(u,v)}\,\mathrm{d}u\,\mathrm{d}v = -u\,\mathrm{d}u\,\mathrm{d}v. \tag{4}$$

Then on using Eqs. (3) and (4) in Eq. (1), we have

$$I = \int_0^1 \int_0^{1-x} f(x,y)\,\mathrm{d}y\,\mathrm{d}x = \int_0^1 \int_0^1 f(uv, u(1-v)u\,\mathrm{d}u\,\mathrm{d}v. \tag{5}$$

The integral I of Eq. (5) can be further transformed into an integral over the standard 2-square: $\{(\xi,\eta)| -1 \leqslant \xi, \eta \leqslant 1\}$ by the substitution (see Fig 1)

$$u = (1+\xi)/2, \quad v = (1+\eta)/2. \tag{6}$$

Then clearly the determinant of the Jacobian and the differential area are

$$\begin{aligned} \frac{\partial(u,v)}{\partial(\xi,\eta)} &= \frac{\partial u}{\partial \xi}\frac{\partial v}{\partial \eta} - \frac{\partial u}{\partial \eta}\frac{\partial v}{\partial \xi} = (1/2)(1/2) - (0)(0) = 1/4, \\ \mathrm{d}u\,\mathrm{d}v &= \frac{\partial(u,v)}{\partial(\xi,\eta)}\mathrm{d}\xi\mathrm{d}\eta = \frac{1}{4}\mathrm{d}\xi\mathrm{d}\eta. \end{aligned} \tag{7}$$

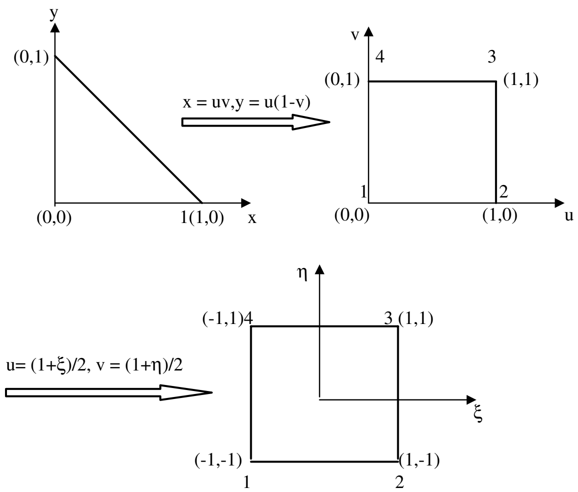

Fig. 1. Transformation of standard triangle T into equivalent 1-square in (u,v) space and 2-square in (ξ,η) space.

Now on using Eqs. (6) and (7) in Eq. (5), we have

$$I = \int_0^1 \int_0^{1-x} f(x,y)\,\mathrm{d}y\,\mathrm{d}x = \int_0^1 \int_0^1 f(uv, u(1-v))|u|\,\mathrm{d}u\,\mathrm{d}v$$
$$= \int_{-1}^1 \int_{-1}^1 f\left(\frac{(1+\xi)(1+\eta)}{4}, \frac{(1+\xi)(1-\eta)}{4}\right)\left(\frac{1+\xi}{8}\right)\mathrm{d}\xi\,\mathrm{d}\eta. \tag{8}$$

Eq. (8) represents an integral over the surface of a standard 2-square: $\{(\xi,\eta)|\ -1 \leqslant \xi,\eta \leqslant 1\}$. Now efficient Gauss Legendre quadrature rules are readily available over the 2-square in the literature so that any desired accuracy can be readily obtained for the integral I of Eq. (1) [16].

From Eq. (8), we can write

$$\begin{aligned} I &= \int_{-1}^1 \int_{-1}^1 f(x(\xi,\eta), y(\xi,\eta))\left(\frac{1+\xi}{8}\right)\mathrm{d}\xi\,\mathrm{d}\eta, \\ I &= \sum_{i=1}^{s}\sum_{j=1}^{s}\left(\frac{1+\xi_i}{8}\right) w_i w_j f(x(\xi_i,\eta_j), y(\xi_i,\eta_j)), \end{aligned} \tag{9}$$

where ξ_i, η_j are Gaussian points in the ξ,η directions and w_i and w_j are the corresponding weight coefficients.

We can rewrite Eq. (9) as

$$I = \sum_{k=1}^{N=s\times s} c_k f(x_k, y_k), \tag{10}$$

where, c_k, x_k and y_k can be obtained from the relations

$$c_k = \frac{(1+\xi_i)}{8} w_i w_j, \quad x_k = \frac{(1+\xi_i)(1+\eta_j)}{4}, \quad y_k = \frac{(1+\xi_i)(1-\eta_j)}{4},$$
$$(k = 1,2,\ldots,N)\ \ (i,j = 1,2,3,\ldots..,n). \tag{11}$$

The weighting coefficients c_k and sampling points (x_k, y_k) of various order can be now easily computed by formulas of Eqs. (10) and (11).We have listed here a C-Program which generates c_k, x_k and y_k and then computes the integral $\int\int_T f(x,y)\,\mathrm{d}x\,\mathrm{d}y$.We have also given the sample output of the program for $n = 2,3,4,5$.

C-Program

```
# include<stdio.h>
# include<conio.h>
# include<math.h>
void main ( )
{
  double c[10][10], x[10][10], y[10][10], p[10], q[10], w1[10], w2[10];
int k, i, j, n;
clrscr( );
printf (input n\n);
scanf (%d, & n);
printf (enter % d p values\n,n);
for (i = 0; i< n; ++i){
scanf (% lf, & x[i]);
}
printf (enter % d q values\n,n);
for (i = 0; i< n; ++i){
scanf (% lf, & y[i]);
}
printf (enter % d w1 values\n,n);
for (i = 0; i< n; ++i){
scanf (% lf, & w[i]);
}
printf (enter % d w1 values\n,n);
for (i = 0; i< n; ++i){
scanf (% lf, & w2[i]);
}
for (i = 0; i< n; ++i){
for (j = 0; j< n; ++j)
{
c[i][j] = ((1 + p[i])/8.0)*(w1[i]*w2[j]);
x[i][j] = ((1+ p[i])*(1+ q[j]))/4.0;
y[i][j] = ((1 + p[i])*(1- q[j]))/4.0;
}
for (i = 0; i< n; ++i){
for (j = 0; j< n; ++j)
{
printf ({% 0.15lf\T % 0.15 lf\t % 0.15 lf\n, c[i][j], x[i][j], y[i][j]);
}}
getch ( );
}
```

k	Sample output		
	c_k	x_k	y_k
$s = 2$			
1	0.052 831 216 351 297	0.044 658 198 738 520	0.166 666 666 666 667
2	0.052 831 216 351 297	0.166 666 666 666 667	0.044 658 198 738 520
3	0.197 168 783 648 703	0.166 666 666 666 667	0.622 008 467 928 146
4	0.197 168 783 648 703	0.622 008 467 928 146	0.166 666 666 666 667

(*continued on next page*)

(*continued*)

k	Sample output		
	c_k	x_k	y_k
$s = 3$			
1	0.008 696 116 155 807	0.012 701 665 379 258	0.100 000 000 000 000
2	0.013 913 785 849 291	0.056 350 832 689 629	0.056 350 832 689 629
3	0.008 696 116 155 807	0.100 000 000 000 000	0.012 701 665 379 258
4	0.061 728 395 061 728	0.056 350 832 689 629	0.443 649 167 310 371
5	0.098 765 432 098 765	0.250 000 000 000 000	0.250 000 000 000 000
6	0.061 728 395 061 728	0.443 649 167 310 371	0.056 350 832 689 629
7	0.068 464 377 671 354	0.100 000 000 000 000	0.787 298 334 620 741
8	0.109 543 004 274 166	0.443 649 167 310 371	0.443 649 167 310 371
9	0.068 464 377 671 354	0.787 298 334 620 741	0.100 000 000 000 000
$s = 4$			
1	0.002 100 365 244 475	0.004 820 780 989 426	0.064 611 063 213 548
2	0.003 937 685 608 733	0.022 913 166 676 413	0.046 518 677 526 561
3	0.003 937 685 608 733	0.046 518 677 526 561	0.022 913 166 676 413
4	0.002 100 365 244 475	0.064 611 063 213 548	0.004 820 780 989 426
5	0.018 715 815 315 013	0.022 913 166 676 413	0.307 096 311 531 159
6	0.035 087 705 252 933	0.108 906 255 706 834	0.221 103 222 500 738
7	0.035 087 705 252 933	0.221 103 222 500 738	0.108 906 255 706 834
8	0.018 715 815 315 013	0.307 096 311 531 159	0.022 913 166 676 413
9	0.037 997 147 647 950	0.046 518 677 526 561	0.623 471 844 265 867
10	0.071 235 620 499 740	0.221 103 222 500 738	0.448 887 299 291 690
11	0.071 235 620 499 740	0.448 887 299 291 690	0.221 103 222 500 738
12	0.037 997 147 647 950	0.623 471 844 265 867	0.046 518 677 526 561
13	0.028 150 383 076 926	0.064 611 063 213 548	0.865 957 092 583 479
14	0.052 775 277 354 230	0.307 096 311 531 159	0.623 471 844 265 867
15	0.052 775 277 354 230	0.623 471 844 265 867	0.307 096 311 531 159
16	0.028 150 383 076 926	0.865 957 092 583 479	0.064 611 063 213 548
$s = 5$			
1	0.000 658 316 657 301	0.002 200 555 327 023	0.044 709 521 703 645
2	0.001 329 900 683 819	0.010 825 220 107 480	0.036 084 856 923 188
3	0.001 580 694 532 071	0.023 455 038 515 334	0.023 455 038 515 334
4	0.001 329 900 683 819	0.036 084 856 923 188	0.010 825 220 107 480
5	0.000 658 316 657 301	0.044 709 521 703 645	0.002 200 555 327 023
6	0.006 542 197 529 252	0.010 825 220 107 480	0.219 940 124 839 679
7	0.013 216 243 082 027	0.053 252 644 428 581	0.177 512 700 518 577
8	0.015 708 573 902 135	0.115 382 672 473 579	0.115 382 672 473 579
9	0.013 216 243 082 027	0.177 512 700 518 577	0.053 252 644 428 581
10	0.006 542 197 529 252	0.219 940 124 839 679	0.010 825 220 107 480
11	0.016 848 134 048 440	0.023 455 038 515 334	0.476 544 961 484 666
12	0.034 035 816 568 844	0.115 382 672 473 579	0.384 617 327 526 421
13	0.040 454 320 987 654	0.250 000 000 000 000	0.250 000 000 000 000
14	0.034 035 816 568 844	0.384 617 327 526 421	0.115 382 672 473 579
15	0.016 848 134 048 440	0.476 544 961 484 666	0.023 455 038 515 334
16	0.021 807 802 470 748	0.036 084 856 923 188	0.733 149 798 129 653

(*continued*)

k	Sample output		
	c_k	x_k	y_k
17	0.044 055 107 973 971	0.177 512 700 518 577	0.591 721 954 534 264
18	0.052 363 059 235 553	0.384 617 327 526 421	0.384 617 327 526 421
19	0.044 055 107 973 971	0.591 721 954 534 264	0.177 512 700 518 577
20	0.021 807 802 470 748	0.733 149 798 129 653	0.036 084 856 923 188
21	0.013 375 270 558 306	0.044 709 521 703 645	0.908 380 401 265 687
22	0.027 020 099 316 181	0.219 940 124 839 679	0.733 149 798 129 653
23	0.032 115 573 564 810	0.476 544 961 484 666	0.476 544 961 484 666
24	0.027 020 099 316 181	0.733 149 798 129 653	0.219 940 124 839 679
25	0.013 375 270 558 306	0.908 380 401 265 687	0.044 709 521 703 645

3. Composite integration over standard triangle T

We can discretise T in (x,y) space into $n \times n = n^2$ right isosceles triangle T_i each of area $1/(2n^2)$. This is depicted in Fig. 2.

By use of the linearity property of integrals, we can write from Eq. (1) and from the above discretisation of Fig. 1, we have

$$I = \int\int_T f(x,y)\,dx\,dy = \int_0^1\int_0^{1-x} f(x,y)\,dy\,dx = \int_0^1\int_0^{1-y} f(x,y)\,dx\,dy = \sum_{i=1}^{s\times s}\int\int_{T_i} f(x,y)\,dx\,dy$$
$$= \frac{1}{n^2}\int_0^1\int_0^{1-x} H(X,Y)\,dY\,dX = \frac{1}{n^2}\int_0^1\int_0^{1-y} H(X,Y)\,dX\,dY, \tag{12}$$

where

$$H(X,Y) = \sum_{j=0}^{n-1}\sum_{i=1}^{n-j} f\left(\frac{i-1}{n}+\frac{X}{n}, \frac{j}{n}-\frac{Y}{n}\right) + \sum_{j=1}^{n-1}\sum_{i=1}^{n-j} f\left(\frac{i}{n}-\frac{X}{n}, \frac{j}{n}-\frac{Y}{n}\right). \tag{13}$$

We can now apply Gauss Legendre quadrature rules on the integral, in a manner similar to the procedure we already developed for integral $\int\int_T f(x,y)\,dx\,dy$. Following the method already developed in previous section, we have now on using the transformation

$$X = (1+\xi)/2, \quad Y = (1-\xi)(1+\eta)/4. \tag{14}$$

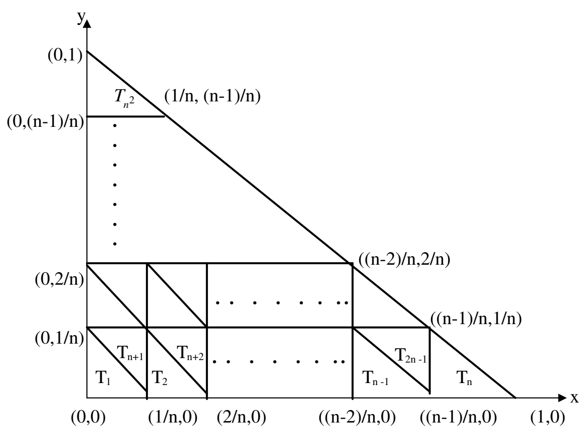

Fig. 2. Discretisation of T into n^2 subtriangles T_i.

The integral I in Eq. (12) can be written as

$$I = \int\int_T f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \frac{1}{n^2}\int\int_T H(X,Y)\,\mathrm{d}X\,\mathrm{d}Y = \frac{1}{n^2}\int_{-1}^{1}\int_{-1}^{1}\left(\frac{1-\xi}{8}\right)H(X(\xi,\eta),Y(\xi,\eta))\,\mathrm{d}\xi\,\mathrm{d}\eta$$
$$= \frac{1}{n^2}\sum_{p=1}^{s}\sum_{q=1}^{s}\left(\frac{1-\xi_p}{8}\right)W_pW_qH(X(\xi_p,\eta_q),Y(\xi_p,\eta_q)), \tag{15}$$

where

$$H(X,Y) = \sum_{j=0}^{n-1}\sum_{i=1}^{n-j} f\left(\frac{i-1}{n}+\frac{X}{n},\frac{j}{n}+\frac{Y}{n}\right) + \sum_{j=1}^{n-1}\sum_{i=1}^{n-j} f\left(\frac{i}{n}-\frac{X}{n},\frac{j}{n}-\frac{Y}{n}\right),$$
$$X(\xi_p,\eta_q) = \frac{(1+\xi_p)(1+\eta_q)}{4}, \tag{16}$$
$$Y(\xi_p,\eta_q) = \frac{(1+\xi_p)(1-\eta_q)}{4} \quad (p,q = 1,2,3\ldots s).$$

From Eqs. (13)–(15), it is clear that, we have obtained the following composite integration rule:

$$I = \frac{1}{n^2}\sum_{k=1}^{N=s\times s} c_k H(x_k,y_k), \tag{17}$$

where

$$H(x_k,y_k) = \sum_{j=0}^{n-1}\sum_{i=1}^{n-j} f\left(\frac{i-1}{n}+\frac{x_k}{n},\frac{j}{n}+\frac{y_k}{n}\right) + \sum_{j=1}^{n-1}\sum_{i=1}^{n-j} f\left(\frac{i}{n}-\frac{x_k}{n},\frac{j}{n}-\frac{y_k}{n}\right),$$
$$c_k = \frac{(1+\xi_p)}{8}w_pw_q, \quad x_k = \frac{(1+\xi_p)(1+\eta_q)}{4}, \quad y_k = \frac{(1+\xi_p)(1-\eta_q)}{4}, \tag{18}$$
$$(k = 1,2,\ldots,N), \quad (p,q = 1,2,3,\ldots,s).$$

We have listed here a *C-Program* which computes integral $\int\int_T f(x,y)\,\mathrm{d}x\,\mathrm{d}y$ by the above explained Composite Numerical Integration method:

C-Program

```
#include<stdio.h>
#include<alloc.h>
#include<math.h>
#include<complex.h>
#include<conio.h>
double fun(double X, double Y)
{
return (double) sqrt(X+Y);
}
void main( )
{
double u, v, x[25], y[25], w, f, s, C[200];
double F1[15][20][25];
double far *FP1,*FP2;
double F2[15][20][25];
double t1, t2, t3, t4, t6, t7, t8, t9, t11;
double sum1=0.0;
```

```
static double SUM11[200],SUM22[200],SUM33[200];
double far *S11, *S22, *S33;
int i, j, k, l, m, n;
clrscr( );
printf(input m and n\n);
scanf(%d%d, & m, & n);
FP1 = (double*)malloc(m*n*sizeof(double));
FP1 = & F1[0][0][0];
FP2 = (double*)malloc(m*n*sizeof(double));
FP2 = & F2[0][0][0];
S11 = (double*)malloc(m*m*sizeof(double));
S11 = & SUM11[1];
S22 = (double*)malloc(m*m*sizeof(double));
S22 = & SUM22[1];
printf(Input x,y,C\n);
for(k = 1;k <= m*m;++k){
fflush(stdin);
scanf(%lf%lf%lf,& x[k],& y[k],& C[k]);
}
for(k = 1;k <= m*m;++k){
for(j = 0;j <= n - 1;++j){
  for(i = 1;i <= n - j;++i){
    t1 = (float)(i-1)/n;
    t2 = (float)j/n;
    t3 = (float)(x[k]/n);
    t4 = (float)(y[k]/n);
    *(*(*(F1 + k) + j) + i) = fun(t1 + t3, t2 + t4);
    *(S11 + k) += *(*(*(F1 + k) + j) + i);
    }}}
    for(k = 1;k <= m*m;++k)
    for(j = 1;j <= n-1;++j)
    for(i = 1;i <= n - j;++i){
    t6 = (float)i/n;
    t7 = (float)j/n;
    t8 = (float)x[k]/n;
    t9 = (float)y[k]/n;
    *(*(*(F2+k) + j) + i) = fun(t6-t8,t7-t9);
    *(S22 + k) += *(*(*(F2 + k) + j) + i);
    }
    for(k = 1;k <= m*m;++k){
    SUM33[k] = *(S11 + k) + *(S22 + k);
    SUM33[k]* = C[k];
    sum1+ = SUM33[k];
    }
    t11 = (float)1/(n*n);
    printf(The solution is: %12.12lf \n,t11*sum1);
    getch();
    }
```

4. Some numerical results

We consider some typical integrals with known exact values [13]:

$$I_1 = \int_0^1 \int_0^{1-y} (x+y)^{\frac{1}{2}}\,\mathrm{d}x\,\mathrm{d}y = 0.400\,000\,000,$$

$$I_2 = \int_0^1 \int_0^{1-y} (x+y)^{\frac{-1}{2}}\,\mathrm{d}x\,\mathrm{d}y = 0.666\,666\,667,$$

$$I_3 = \int_0^1 \int_0^{y} (x^2+y^2)^{\frac{-1}{2}}\,\mathrm{d}x\,\mathrm{d}y = 0.881\,373\,587,$$

$$I_4 = \int_0^{\frac{\pi}{2}} \int_0^{y} \sin(x+y)\,\mathrm{d}x\,\mathrm{d}y = 1.000\,000\,000,$$

$$I_5 = \int_0^1 \int_0^{y} \mathrm{e}^{|x+y-1|}\,\mathrm{d}x\,\mathrm{d}y = 0.71828183.$$

These integrals were evaluated using the two integration schemes of previous Sections 2 and 3 derived in the present paper and it is found that excellent convergence occurs to the exact value. The results are summarized in Tables 1–4

Table 1
Numerical results of double integration ($s = 2 =$ order of Gauss Legendre quadrature rule)

n^2	I_1	I_2	I_3	I_4	I_5
1^2	0.398773985	0.673887339	0.784678327	0.990476629	0.741130436
2^2	0.399774578	0.669239502	0.832879825	0.999463357	0.724537717
3^2	0.399917340	0.668068874	0.849030932	0.999896006	0.721108882
4^2	0.399959554	0.667577770	0.857113936	0.999967309	0.719881314
5^2	0.399976793	0.667318700	0.861965085	0.999986650	0.719308254
6^2	0.399985267	0.667162723	0.865199543	0.999993573	0.718995664
7^2	0.399989969	0.667060334	0.867509993	0.999996534	0.71880741
8^2	0.399992811	0.666988887	0.869242878	0.999997970	0.718683944
9^2	0.399994642	0.666936708	0.870590701	0.999998733	0.718599673
10^2	0.399995882	0.666897235	0.871668970	0.999999169	0.718539355
20^2	0.399999271	0.666748187	0.876521255	0.999999948	0.71834662
40^2	0.399999871	0.666695488	0.878947420	0.999999996	0.718297942
60^2	0.399999953	0.666682355	0.879756142	0.999999999	0.718288990
80^2	0.399999977	0.666676856	0.880160503	0.999999999	0.718285857
100^2	0.399999986	0.666673958	0.880403120	0.999999999	0.718284406
150^2	0.399999995	0.666670635	0.880726609	0.999999999	0.718282974
180^2	0.399999996	0.666669686	0.880834438	0.999999999	0.718282624

Table 2
Numerical results of double integration ($s = 3 =$ order of Gauss Legendre quadrature rule)

n^2	I_1	I_2	I_3	I_4	I_5
1^2	0.399812412	0.669179634	0.830150053	1.000145446	0.695312789
2^2	0.399966763	0.667555451	0.855760342	0.999999400	0.712492884
3^2	0.399987936	0.667150469	0.864298061	0.999999948	0.715704859
4^2	0.399994123	0.666980906	0.868566939	0.999999990	0.716831464
5^2	0.399996635	0.666891518	0.871128268	0.999999997	0.717353351
6^2	0.399997867	0.666837717	0.872835821	0.999999999	0.717636690
7^2	0.399998549	0.666802405	0.874055502	0.999999999	0.717808007
8^2	0.399998961	0.666777767	0.874970262	0.999999999	0.717919038
9^2	0.399999226	0.666759775	0.875681743	0.999999999	0.717995168
10^2	0.399999405	0.666746164	0.876250927	0.999999999	0.718049627
20^2	0.399999894	0.666694773	0.878812257	0.999999999	0.718223773
40^2	0.399999981	0.666676604	0.880092922	0.999999999	0.718267314
60^2	0.399999993	0.666672076	0.880519810	0.999999999	0.718275378
80^2	0.399999996	0.666670180	0.880733255	0.999999999	0.718278200
100^2	0.399999998	0.666691806	0.880861321	0.999999999	0.718279506

Table 3
Numerical results of double integration ($s = 4 =$ order of Gauss Legendre quadrature rule)

n^2	I_1	I_2	I_3	I_4	I_5
1^2	0.399950385	0.667827645	0.849816063	0.999998699	0.705297478
2^2	0.399982352	0.667170624	0.860150053	0.999999446	0.713312710
3^2	0.399996763	0.666555451	0.865760542	0.999999900	0.716501184
4^2	0.399999436	0.666850469	0.874298062	0.999999999	0.716704859
5^2	0.399999823	0.666780912	0.876566940	0.999999999	0.717921421
6^2	0.399999935	0.666741512	0.878128268	0.999999999	0.717353351
7^2	0.399999967	0.666717717	0.879835201	0.999999999	0.717636690
8^2	0.399999989	0.666692405	0.880055502	0.999999999	0.718008007
9^2	0.399999991	0.666687767	0.880270262	0.999999999	0.718069038
10^2	0.399999993	0.666679775	0.880481421	0.999999999	0.718195168
20^2	0.399999995	0.666676164	0.880650920	0.999999999	0.718249627
40^2	0.399999997	0.666674721	0.880812257	0.999999999	0.718263721
60^2	0.399999999	0.666672604	0.880992922	0.999999999	0.718277314
80^2	0.399999999	0.666671076	0.881089810	0.999999999	0.718280878
100^2	0.399999999	0.666660180	0.881133220	0.999999999	0.718281200

Table 4
Numerical results of double integration ($s = 5 =$ order of Gauss Legendre quadrature rule)

n^2	I_1	I_2	I_3	I_4	I_5
1^2	0.399982448	0.667296789	0.860047531	0.999999981	0.709124295
2^2	0.399995552	0.666170653	0.865150053	0.999999996	0.715356716
3^2	0.399999763	0.666785451	0.875761545	0.999999999	0.717051154
4^2	0.399999936	0.666750420	0.876458062	0.999999999	0.717184859
5^2	0.399999963	0.666720912	0.878066940	0.999999999	0.717521421
6^2	0.399999995	0.666701520	0.880128268	0.999999999	0.717853351
7^2	0.399999998	0.666697717	0.880335221	0.999999999	0.718096620
8^2	0.399999999	0.666678406	0.880455505	0.999999999	0.718100054
9^2	0.399999999	0.666675767	0.880570267	0.999999999	0.718169037
10^2	0.399999999	0.666673785	0.880481421	0.999999999	0.718205162
20^2	0.399999999	0.666672164	0.880750923	0.999999999	0.718279625
40^2	0.399999999	0.666670751	0.880952267	0.999999999	0.718280721
60^2	0.399999999	0.666667603	0.881092925	0.999999999	0.718281414
80^2	0.399999999	0.666667071	0.881189810	0.999999999	0.718281678
100^2	0.399999999	0.666666180	0.881213220	0.999999999	0.718281800

5. Conclusions

We have derived various orders ($s = 2, 3, 4, 5, \ldots$) extended numerical integration rules based on classical Gauss Legendre quadrature. This is made possible by transforming the triangular surface: $0 \leqslant x,\ y \leqslant 1$, $x + y \leqslant 1$ to a standard 2-square; $-1 \leqslant \xi, \eta \leqslant 1$. Over the 2-square, the Gauss Legendre quadrature rule of all orders is applicable. It is the main purpose of this paper to evolve a practical and workable algorithm for composite numerical integration over triangular surfaces and it converges to the exact value of the integral, for sufficiently large value of n, even for the lower order Gauss Legendre quadrature rules.

References

[1] O.C. Zienkiewicz, The Finite Element Method, third ed., McGraw-Hill, London, 1977.
[2] J.N. Reddy, An Introduction to the Finite Element Method, McGraw Hill Book Company, New York, 1984.
[3] T.J.R. Hughes, The Finite Element Method, Static and Dynamic Analysis, Prentice-Hall, Englewood Cliffs. N.J, 1987.
[4] P.C. Hammer, O.J. Marlowe, A.H. Stroud, Numerical integration over simplexes and cones, Math. Tables Other Aids to Computation 10 (1956) 130–136.
[5] P.C. Hammer, A.H. Stroud, Numerical integration over simplexes, Math. Tables Other Aids to Computation 10 (1956) 137–139.

[6] P.C. Hammer, A.H. Stroud, Numerical Evaluation of multiple integrals, Math. Tables Other Aids to Computation 12 (1958) 272–280.
[7] G.R. Cowper, Gaussian quadrature formulas for triangles, Int. J. Numer. Methods Eng. 7 (1973) 405–408.
[8] J.N. Lyness, D. Jespersen, Moderate degree symmetric quadrature rules for the triangle, J. Inst. Math. Appl. 15 (1975) 19–32.
[9] F.G. Lannoy, Triangular finite elements and numerical integration, Comput. Struct. 7 (1977) 613.
[10] M.E. Lauresn, M. Gellert, Some criteria for numerically integrated matrices and quadrature formulas for triangles, Int. J. Numer. Methods Eng. 12 (1978) 67–76.
[11] D.A. Dunavant, High degree efficient symmetrical Gaussian Quadrature Rules for the triangle, Int. J. Numer. Methods Eng. 21 (1985) 1129–1148.
[12] D.P. Laurie, Automatic numerical integration over a triangle, CSIR Special Report WISK 273, National Institute for Mathematical Sciences, Pretoria, 1977.
[13] P. Sylvester, Symmetric quadrature formulae for simplexes, Math. Comput. 24 (1970) 95–100.
[14] F.G. Lethor, Computation of double integrals over a triangle, J. Comput. Appl. Math. 2 (1976) 219–224.
[15] P. Hillion, Numerical integration on a triangle, Int. J. Numer. Methods Eng. 11 (1977) 797–815.
[16] M. Abramowicz, I.A. Stegun (Eds.), Handbook of Mathematical Functions, Dover Publications, 1964.
[17] C.T. Reddy, Improved three point integration schemes for triangular finite elements, Int. J. Numer. Methods Eng. 12 (1978) 1890–1896.
[18] C.T. Reddy, D.J. Shippy, Alternative integration formulae for triangular finite elements, Int. J. Numer. Methods Eng. 17 (1981) 133–139.
[19] G. Lague, R. Baldur, Extended numerical integration method for triangular surfaces, Int. J. Numer. Methods Eng. 11 (1977) 388–392.
[20] H.T. Rathod, K.V. Nagaraja, Symmetric Gauss Legendre quadrature over a triangle, J. Sci. Eng. Technol. 17 (1) (2005) 1–8.
[21] H.T. Rathod, K.V. Nagaraja, B. Venkatesudu, N.L. Ramesh, Gauss Legendre quadrature over a triangle, J. Indian Inst. Sci. 84 (2004) 183–188.
[22] H.T. Rathod, K.V. Nagaraja, B. Venkatesudu, Gauss Legendre quadrature over a triangle, Acharya Nagarjuna Int. J. Math. Inform. Technol. 1 (1) (2004) 33–52.
[23] Brice Carnahan, H.A. Luther, James.O. Wilkes, Applied Numerical Methods, John Wiley and Sons, 1969.
[24] C.F. Gerald, P.O. Wheatley, Applied Numerical Analysis, sixth ed., Lowprice ed., Pearson Education, Asia, 1999.
[25] S.C. Chapra, R.P. Canale, Numerical Methods for Science and Engineering, Tata McGraw-Hill, New Delhi, 2000.
[26] R.L. Burden, J.D. Fairs, Numerical Analysis, fourth ed., PWS-KENT Publishing Company, 1989.