J. Nig. Ass. Math. Phys.
Vol. 4 (2000)

GENERATING SEMI-GAUSSIAN TWO DIMENSIONAL QUADRATURE RULES

V. U. AIHIE
DEPARTMENT OF MATHEMATICS
UNIVERSITY OF BENIN, BENIN-CITY, NIGERIA

G. A. EVANS
DEPARTMENT OF MATHEMATICS,
LOUGHBOROUGH UNIVERSITY OF TECHNOLOGY,
LOUGHBOROUGH, LEICESTERSHIRE, LE11 3TU

ABSTRACT

This paper generates semi-Gaussian two-dimensional quadrature rules. It is based on monomial equations and experimentally executed. The rules generated were tested and compared favourably with Gaussian product rules.

1. PHILOSOPHY OF THE APPROACH

Two dimensional quadrature rules lack the philosophical elegance, which encompasses the one-dimensional case. In one dimension the application of desirable properties to a general rule such as

$$\int_a^b w(x)f(x)dx = \sum_{i=1}^{N} W_i f(x_i) + E \qquad w(x) \succ 0 \quad on[a,b] \qquad (1.1)$$

leads to solutions which exist and generally have good stability properties. Hence freeing W_i and x_i to force E to be zero for a suitable set of monomials gives a set of non-linear equations which, may be surprisingly, not only has a unique solution but also have positive W_i (and hence stability) and moreover all the x_i's belong to the interval [a,b]. The Gaussian formulae so generated form a very powerful set.[1]

There also exist weaker forms of Gaussian rules such as Radan and Lobatto in which end points are fixed in the set $\{x_i\}$ so reducing the unknowns and hence the number of exactly satisfied monomials.[2]

The alternative approach in common use is based on the results from approximation theory concerning Chebyshev polynomials, $T_n(x)$. If a function is approximated by a series such that the remainder is a Chebyshev polynomial such as

$$f(x) = \sum_{i=0}^{N} a_i g_i(x) + T_{N+1}(x),$$

for some suitable $g_i(x)$ (possibly $T_i(x)$), then TN+1 has N + 2 equal and opposite maxima and minima, and therefore satisfies the oscillation theorem. Hence the above approximation to $f(x)$ is best in the uniform norm sense and it would seem appropriate to integrate this result to obtain a quadrature rule. The result is the Clenshaw-Curtis quadrature rule [3]

$$\int_{-1}^{1} f(x)dx = \sum_{i=0}^{N}{}'' a_i \int_{-1}^{1} T_i(x)dx,$$

where

$$a_i = \frac{2}{N}\sum{}''\cos\frac{\pi is}{N} f\left(\cos\frac{\pi s}{N}\right).$$

Once again the weights are all positive and a most accurate general purpose progressive rules ensues. For the same point number N the highest exactly integrable monomial is only about half the power of that achieved by Gauss, but as test on the Clenshaw-Curtis formula have shown this is not a necessary end to achieve a powerful rule.

In two dimensions the equivalent approaches seem to lead to either non existent solutions or abscissae outside the interval of integration. [4] No equivalent of a true Gaussian formula exists, nor is there an equivalent two-dimensional best approximation. We do have Gaussian product rules and many of the equations which need to be satisfied to make the relevant monomials exactly integrable are dependent. Hence extra equations are required to obtain a set from which the Gauss product rules can be re-generated. These are various symmetry relations between the x_i's and y_i's and in some cases also between the W_i's.

The philosophy is therefore to use a formula which is known to exist to see what equations need to be set up to regenerate the rule. These conditions can then be altered in stages to introduce more desirable properties to the quadrature rule and to generate extended rules somewhat along the lines of Patterson.[5]

Two problems arise in setting up monomial equations. Some of these are dependent, so it becomes necessary to know which ones of a sequence to omit. For every omission an extra equation is needed. Should this then be a symmetry condition on the abscissae in each variable, a similar weight condition – possibly with the aim of forcing positive weights, or a higher order monomial, remembering that the full Gaussian set will have no solution?

The second related proble is that some monomial equations outside the intended range will prove to be satisfied exactly. In general the more of these 'free' conditions that are obtained the better the quadrature rule.

2. A PROGRAMME – AN EXPERIMENTAL BASE

The requirement here is a straightforward solution of a set of partially linear equations. That is, of the total number m of equations, $n(n < m)$ of the unknowns appear linearly with coefficients which may be non-linear combinations of the other variables. Hence the set of $(m - n)$ non-linear equations can be solved by a Newton algorithm, which will be efficient, as the nature of the problem will yield good first estimates. The current iterates of the non-linear variables are used to compute the coefficients of the linear variables. These can then be solved by conventional elimination, to be substituted into the $(m - n)$ non-linear equations in the Newton algorithm to generate the next set of iterates. Estimated partial derivatives were used based on a first order finite difference form.

By this means the number of expensive to solve non-linear equations is kept to a minimum, especially in view that long arithmetic was used to counter any possible (and expected) ill-conditioning in the defining equations.

For flexibility the programme was designed so that monomial pairs of the form $x^i y^j$ which are required to be integrated exactly by the resulting formula could be entered as a collateral of pairs (i,j). It then becomes trivial to add or delete such equations. The symmetry conditions are also laid out in a straightforward fashion to allow easy alterations. It is important that if a zero pivot occurs in the eliminator a suitable interrupt is raised as this implies linear dependence and an adjustment to the equations is needed.

In order to monitor any ill conditioning the condition numbers of the various sets of linear equations involved were found after off-loading the relevant coefficients. In all cases quoted the level of ill-conditioning was found to be very small, though clearly increasing for higher point number formulae.

3. RESULTS

A number of successful formulae were found from the above procedure. Inevitably the choice of which monomials to make exact depends on a variety of factors. In the first instance all the lower order monomials are included. Hence the formula

$$\int_{-1}^{1}\int_{-1}^{1} x^i y^i dxdy = \sum_{k=1}^{N} w_k x_k^i y_k^j$$

is forced to hold for the pairs (i,j) taking values such as (0,0), (0,1), (1,0), (1,1) and so on. Depending on the symmetries satisfied by the abscissae (x_k,y_k), some of the monomial equations will be dependent on each other and not be able to be used. In this case higher order ones are included.

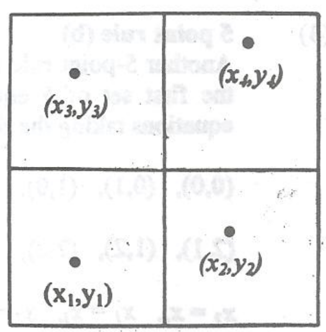
The results are presented in the tables below in which the first line of pairs gives the set of monomial equations treated as linear in the wights w_i – for an N point rule there will be N of these. The second line gives the other monomial equations and then the various imposed symmetries are quoted below. Finally a small diagram and table of the actual abscissae arae given.

(1) 4-point rule

(0,0), (0,10, (1,0), (1,1), (2,2)

$x_2 = x_4,\ \ x_1 = x_3,\ \ x_1 = -x_2,\ \ x_2 = y_3$

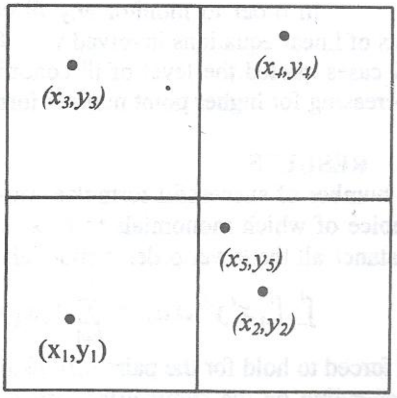
$x_3 = y_2,\ \ x_1 = y_1,\ \ x_4 = y_4,$



i	w_i	x_i	y_i
1	1.0	-0.57735026918962576467	-5.5773502618962576467
2	1.0	0.57735026918962576467	-0.5773502618962576467
3	1.0	-0.57735026918962576467	0.5773502618962576467
4	1.0	0.57735026918962576467	0.5773502618962576467

This result is in total agreement with the two-point Gaussian product and so the above model formed a validated basis for this work.

(2) 5-point rule (a)



(0,0), (0,1), (1,0), (1,1), (2,0),

(2,1), (1,2), (2,2), (3,0), (0,3),

$x_2 = x_4,\quad x_1 = x_3,\quad x_1 = -x_2,\quad x_1 = y_1,$

$x_4 = y_4,$

i	w_i	x_i	y_i
1	0.64265722402873640779	-0.66673264565535991723	-0.66673264565535991723
2	0.85704591459275138404	0.6667326456553599734	-0.49995052065537574118
3	0.85704591459275138426	-0.66673264565535991745	0.49995052065537574107
4	0.64265722402873640704	0.66673264565535991745	0.66673264565535991752
5	1.0005937227570244172	0.0	0.0

(3) 5 point rule (b)

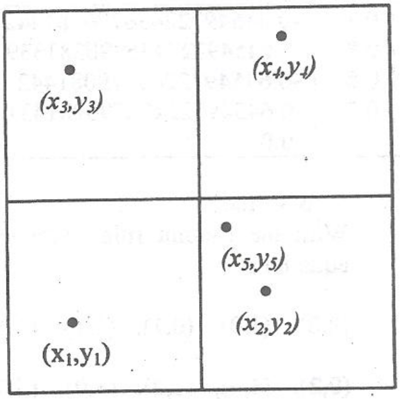
Another 5-point rule can be produced with weight conditions introduced. Here the first set of 5 equations is the same as in (2) but the second set of ten equations taking the form below

(0,0), (0,1), (1,0), (1,1), (2,0),

(2,1), (1,2), (2,2), (0,3),

$x_2 = x_4,\quad x_1 = x_3,\quad x_1 = -x_2,\quad x_1 = y_1,$
$x_4 = y_4$

$w_1 = w_2$



i	w_i	x_i	y_i
1	0.68465843842649082437	-0.65339533882747565289	-0.65339533882747565213
2	0.87689437438233944937	0.65339533882747565289	-0.51015566461111166064
3	0.87689437438233945078	-0.6533953388274755258	0.51015566461111165977
4	0.68465843842649082556	0.6533953388274565267	0.65339533882747565288
5	0.87689437438233944981	0.0	0.0

(4) 5-point rule (c) – with equal weights

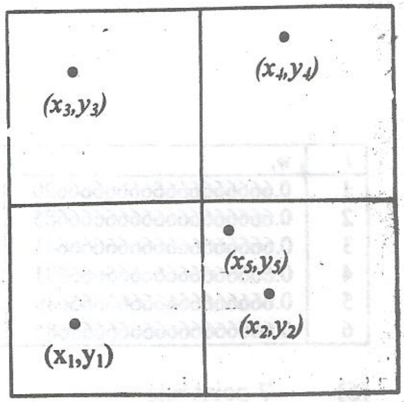
By imposing the condition that all the weights be equal the set of equations we have for a five point rule are:

(0,0), (1,0), (0,1) (1,1), (2,0),

(0,2), (2,1(, (1,2), (3,0) (0,3),

(3,1),

$w_1 = w_2,\quad w_3 = w_4,\quad w_5 = w_1,\ w_2 = w_4$



i	w_i	x_i	y_i
1	0.8	-0.64549722436790281442	-0.64549722436790281431
2	0.8	0.64549722436790281339	-0.64549722436790281442
3	0.8	-0.6454972243679081442	0.64549722436790281491
4	0.8	0.64549722436790281431	0.64549722436790281420
5		0.0	0.0

5 **6-point rule**

With the 6-point rule (with equal weights) we have the following sets of equations:

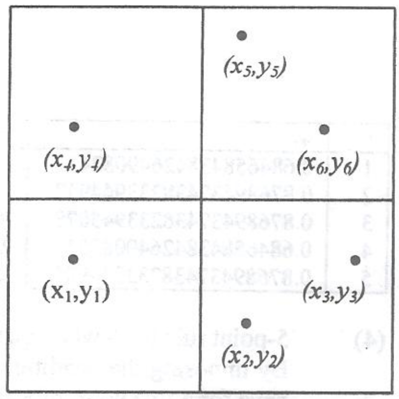
(0,0), (1,0), (0,1), (1,1), (2,1),

(0,2), (4,0), (1,2), (3,0), (5,0),

(0,5), (1,3), (0,4),

$w_1 = w_2$, $w_3 = w_4$, $w_5 = w_6$, $w_4 = w_2$,

$w_1 = w_5$,



i	w_i	x_i	y_i
1	0.66666666666666666620	-0.74008280449228525035	-0.35002117458154067810
2	0.66666666666666666685	0.0	-0.868890300722201204915
3	0.66666666666666666641	0.74008280449228525024	-0.35002117458154067815
4	0.66666666666666666631	-0.74008280449228525089	0.3500211745815406818
5	0.66666666666666666639	0.0	0.86889030072220120495
6	0.66666666666666666685	0.74008280449228525057	0.35002117458154067761

(6) **7-point rule**

With the 7-point rule (with equal weights) we have the following sets of equations:

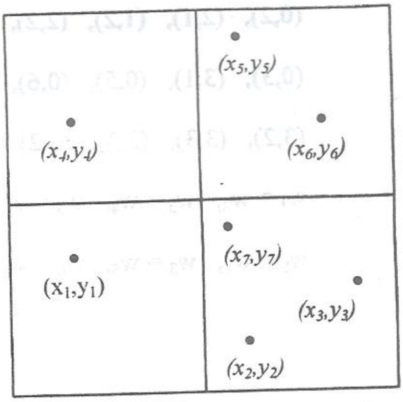
(0,0), (1,0), (0,1), (1,1), (2,0),

(0,2), (2,1), (5,0), (0,5), (5,1),

(1,5), (1,4), (5,4), (5,5), (4,5),

$w_1 = w_2, \quad w_3 = w_4, \quad w_5 = w_6, \quad w_4 = w_2,$

$w_3 = w_2, \quad w_5 = w_7,$



i	w_i	x_i	y_i
1	0.57142549470901745452	-0.76377634150326558871	-0.298334285119026504075
2	0.57142906531711892620	0.0	-0.994303573074995978
3	0.57142549470901745474	0.76377634150326558827	-0.29834285119026503994
4	0.57143164836930384893	-0.7637748899020107145	0.29836108534558329668
5	0.57142806588149074965	0.0	0.99430285531030323598
6	0.57143164336930384796	0.763374889902010	0.29836108534558329510
7	0.5714285264474771877	0.0	-0.0004043936282357150

Some ill-conditioning is expected here as the errors in the supposedly equal weights show.

(7) 7-point rule
By replacing (0,5), (5,1), (1,5) by (0,25), (25,11), (11,25) a similar set of formulae is generated as shown below:

i	w_i	x_i	y_i
1	0.57142857066576083983	-0.76376261922451045344	-0.299999502852020127785
2	0.57142857155633176749	0.0	-0.9933126218235589107
3	0.57142857066576084016	0.76376261922451045810	-0.29999950285020127687
4	0.57142857219138202998	-0.76376261242743619996	0.29999950738891820168
5	0.57142857130419388910	0.0	0.99331125910324016284
6	0.57142857219138203074	0.76376261242743619562	0.29999950738891820374
7	0.57142857142518860313	0.0	-0.0000000716192790792

(8) 9-point rule with equal weights
For a nine-point rule the set of equations involved are as follows:

(0,0), (1,0) (0,1), (1,1), (2,0),

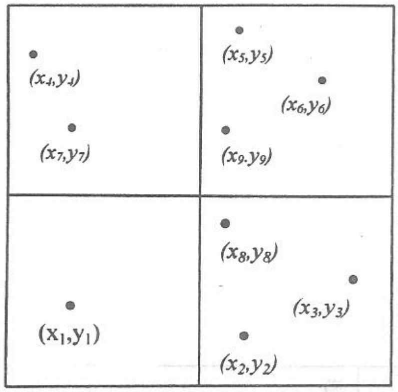
(0,2), (2,1), (1,2), (2,2), (3,0),

(0,3), (3,1), (0,5), (0,6), (0,7),

(3,2), (3,3), (2,3), (5,2),

$w_1 = w_2$, $w_3 = w_4$, $w_5 = w_6$, $w_7 = w_8$,

$w_9 = w_1$, $w_3 = w_5$, $w_6 = w_7$, $w_8 = w_2$,



i	w_i	x_i	y_i
1	0.4444241582707352855	-0.8137089875143209672	-0.6146905493843060025
2	0.4444518803265492236	0.00031081203690234615	-0.86279453582773435807
3	0.4444130287316860095	0.8140162587744951551	-0.61435756439783453006
4	0.4448050817388163561	-0.812338166693188049714	0.61534747325323288633
5	0.4444248869405269037	0.00031764783683278901	0.86212930789494299406
6	0.4444308630563611389	-0.81269280438417541379	0.61500983479212011016
7	0.44415956445468844269	-0.42237173212933225069	0.00485904890168844906
8	0.44463381551067402060	-0.00083786962247918029	-0.01042840689288041142
9	0.44540809587801552911	-0.42007286129450621307	0.0048861036966911564

(9) 9-point rule

By changing (3,0) to (0.9) the following set is obtained:

i	w_i	x_i	y_i
1	0.4444444444444446292	-0.71730585014582323527	-0.69729904195897098290
2	0.4444444444444433895	-0.00000000000000014943	-0.72608726096049206855
3	0.44444444444444550278	0.71730585014582331365	-0.69729904195897097130
4	0.44444444444444336165	-0.71577695410879641109	0.698291634125419729
5	0.44444444444444530779	0.00000000000000012946	0.7246311372027099358
6	0.4444444444444443316	0.71577695410879648200	-0.69829163403125419501
7	0.44444444444444564536	-0.68784858022192045686	-0.00215427648444910291
8	0.44444444444444464019	-0.00000000000000030024	-0.00483761335568257135
9	0.44444444444444325848	0.687848580221920294131	0.00215427648444912185

Here the earlier ill-conditioning is largely removed.

	I_1	I_2	I_3	I_4	I_5	I_6
Integral	$\int_{-1}^{1}\int_{-1}^{1}\frac{dxdy}{\sqrt{4-x-y}}$	$\int_{-1}^{1}\int_{-1}^{1}\frac{dxdy}{\sqrt{4+x+y}}$	$\int_{-1}^{1}\int_{-1}^{1}\frac{dxdy}{\sqrt{3-x^2-y^2}}$	$\int_{-1}^{1}\int_{-1}^{1}e^x \cos ydxd$	$\int_{-1}^{1}\int_{-1}^{1}e^x \sin ydxdy$	$\int_{-1}^{1}\int_{-1}^{1}e^{xy}dxdy$
Exact	2.0958446	1.0469963	2.6555866	3.955591	0.0	4.2290035
4-point Gauss Product	2.0896379	1.0454546	2.6186146	3.9259455	0.0	4.22428715
9-point Gauss Product	2.0953358	1.0464731	2.6514268	3.95562539	0.0	4.22896940
16-point Gauss Product	2.99579670	1.04649577	2.65509516	3.95558965	0.0	4.2290034
5-point (c)	2.093217447	1.04651128	2.635852121	3.907445679	0.0	4.28181988
6-point rule	2.09075384	1.04645067	2.6369527	4.08377866	0.0	4.08997363
7-point rule	2.09243792	1.0444148	2.6338496	4.047420824	-0.000000461	4.23357717
9-point rule	2.0930289	1.0460852	2.6381227	3.944323412	-0.00007166	4.2268906
13-point rule	2.0974149	1.0462532	2.6558956	3.96033815	-0.011936	4.2287850

REFERENCES

[1] Stroud, A. H. and Secrest, D., "Caussian quadrature formulas", (1966), Prentice-Hall.

[2] Hildebrand, F. B., "Introduction to numerical analysis", (1974), McGraw-Hill, New York.

[3] Clenshaw, C. W. and Curtis, A. R., "A method for numerical integration on an automatic computer", Numer. Math 2 (1960), 197.

[4] Stroud, A. H., "Numerical integration formulae of degree two" Math. Comp. 14 (1960), 21 – 26.

[5] Patterson, T. N. L., "The optimum addition of points to quadrature formulae", Maths. Comp. 22 (1968),213.