The algebraic construction of a minimal cubature formula of degree 11 for the square

Pierre Verlinden
Ronald Cools

Report TW 155, September 1991

Department of Computer Science, K.U.Leuven

ABSTRACT

Earlier, an iterative zero-finder has been applied succesfully, i.e. with some luck, for the construction of a minimal cubature formula of degree 11 with R_4-symmetry for the square. Here we show how the formula can be found with little computational effort by algebraic methods: the theory of linear representations of finite groups together with Morrow and Patterson's construction method of minimal cubature formulae of degree $8n+1$ with matrix eigenvalues. Meanwhile, it is proven that there exists no other minimal cubature formula of degree 11 with R_4-symmetry.

1 Introduction

We consider cubature formulae

$$Q[f] = \sum_{k=1}^{N} w_k f(x_k, y_k)$$

approximating the integral

$$I[f] = \int_{-1}^{1} \int_{-1}^{1} f(x, y) dx dy,$$

which are of degree ≥ 11, i.e. such that

$$Q[p] = I[p]$$

for all polynomials p of degree ≤ 11. The points (x_k, y_k) are called the nodes of the cubature formula and the numbers w_k are called the weights. According to Möller [4], the minimal number of nodes is 24 in this case.

We consider the group R_4 acting on the space of the polynomials in 2 variables and generated by the substitution r, where

$$rp(x, y) = p(-y, x).$$

We are only interested in cubature formulae Q that are invariant under R_4, i.e. such that

$$Q[rp] = Q[p]$$

for all polynomials p. Let Q be a minimal cubature formula of degree 11 for the square, that is invariant w.r.t R_4. Q is thus of the form

$$Q[f] = \sum_{k=1}^{6} w_k Q_k[f],$$

where

$$Q_k[f] = f(x_k, y_k) + f(-y_k, x_k) + f(-x_k, -y_k) + f(y_k, -x_k).$$

The cubature formula depends on 18 parameters: the 6 weights and the coordinates of 6 nodes. In [1] it is shown that these parameters have to satisfy 18 polynomial equations in order for the cubature formula to be of degree ≥ 11. An iterative zero-finder has been applied succesfully, i.e. with some luck, for searching a solution to this system.

Here we show how the formula can be found with little computational effort by algebraic methods: the theory of linear representations of groups [2] together with Morrow and Patterson's construction method of minimal cubature formulae of degree $8n + 1$ with matrix eigenvalues [5]. Meanwhile, it is proven that there exists no other minimal cubature formula of degree 11 with R_4-symmetry

2 The algebraic construction of the formula

We use the following notations

$$\begin{aligned}
\mathcal{P}_s &= \text{the space of the polynomials of degree } \leq s \\
\mathcal{O}_s &= \text{the space of the orthogonal polynomials of degree } s \\
\mathcal{H}_s &= \text{the space of the homogeneous polynomials of degree } s \\
P^{k,l} &= \text{the unique orthogonal polynomial of degree } k+l \text{ such that} \\
&\quad P^{k,l}(x,y) - x^k y^l \text{ is a polynomial of degree } < k+l.
\end{aligned}$$

Let $\mathcal{A}_s$ be the space of the polynomials of degree $\leq s$, vanishing at the nodes of a minimal R_4-invariant cubature formula Q of degree 11. If $p(x,y) \in \mathcal{A}_6$ and $q(x,y) \in \mathcal{P}_5$, then

$$I[pq] = Q[pq] = 0$$

because pq is of degree ≤ 11 and vanishes at the nodes of the cubature formula. Consequently, $\mathcal{A}_6$ can only contain orthogonal polynomials of degree 6. According to Möller [4], there are 4 linearly independent orthogonal polynomials of degree 6, vanishing at the nodes of a minimal cubature formula of degree 11 and thus $\mathcal{A}_6$ has dimension 4. As Q is R_4-invariant, so is $\mathcal{A}_6$. The group R_4 has 4 irreducible characters $\psi_0, \psi_1, \psi_2, \psi_3$, where

$$\psi_k(r^j) = i^{kj} \quad \text{with} \quad i = \sqrt{-1}.$$

The representation of R_4 in $\mathcal{P}_6$ has the character:

$$\chi^{\mathcal{P}_6} = 8\psi_0 + 6\psi_1 + 8\psi_2 + 6\psi_3.$$

As $\dim \mathcal{A}_6 = 4 = 28 - 24$, there exists for every node of the cubature formula, a unique polynomial in $\mathcal{P}_6$, taking the value 1 at that node and vanishing at the other nodes. The conjugacy classes of these polynomials modulo $\mathcal{A}_6$ form a basis for the quotient space $\mathcal{P}_6/\mathcal{A}_6$. In this basis, we find that the character of the representation of R_4 in $\mathcal{P}_6/\mathcal{A}_6$ is

$$\chi^{\mathcal{P}_6/\mathcal{A}_6}(r^j) = \begin{cases} 24 & \text{if } j = 0 \\ 0 & \text{if } j = 1,2,3 \end{cases}$$

or, equivalenty,

$$\chi^{\mathcal{P}_6/\mathcal{A}_6} = 6\psi_0 + 6\psi_1 + 6\psi_2 + 6\psi_3.$$

Hence, we have

$$\chi^{\mathcal{A}_6} = \chi^{\mathcal{P}_6} - \chi^{\mathcal{P}_6/\mathcal{A}_6} = 2\psi_0 + 2\psi_2.$$

Let

$$\begin{aligned}
\phi_A(x,y) &= x^2 + y^2, \\
\phi_B(x,y) &= 1, \\
\phi_C(x,y) &= x^2 - y^2, \\
\phi_D(x,y) &= xy
\end{aligned}$$

and

$$\begin{aligned}\mathcal{Z}_6 &= \{\phi_A \otimes R_A + \phi_B \otimes R_B + \phi_C \otimes R_C + \phi_D \otimes R_D \in (\mathcal{H}_0 \oplus \mathcal{H}_2) \otimes \mathcal{A}_6 : \\ &\quad \phi_A R_A + \phi_B R_B + \phi_C R_C + \phi_D R_D = 0\},\end{aligned}$$

where $\otimes$ denotes the Kronecker-product [2]. We have that

$$\begin{aligned}\mathcal{Z}_6 &= \{\phi_A \otimes R_A + \phi_B \otimes R_B + \phi_C \otimes R_C + \phi_D \otimes R_D : R_A, R_C, R_D \in \mathcal{A}_6 \\ &\quad \text{and } R_B = -\phi_A R_A - \phi_C R_C - \phi_D R_D \in \mathcal{P}_7\}.\end{aligned}$$

Hence, $\mathcal{Z}_6$ has the same character as the kernel of the map

$$L : \mathcal{H}_2 \otimes \mathcal{A}_6 \to \mathcal{P}_8/\mathcal{P}_7 : \phi_A \otimes R_A + \phi_C \otimes R_C + \phi_D \otimes R_D \mapsto \phi_A R_A + \phi_C R_C + \phi_D R_D + \mathcal{P}_7,$$

which is surjective according to Möller [4]. We have

$$\begin{aligned}\chi^{\text{Ker } L} + \chi^{\text{Im } L} &= \chi^{\mathcal{H}_2 \otimes \mathcal{A}_6} \\ \chi^{\mathcal{Z}_6} + \chi^{\mathcal{H}_8} &= \chi^{\mathcal{H}_2} \chi^{\mathcal{A}_6} \\ \chi^{\mathcal{Z}_6} + 5\psi_0 + 4\psi_2 &= (\psi_0 + 2\psi_2)(2\psi_0 + 2\psi_2) \\ \chi^{\mathcal{Z}_6} &= \psi_0 + 2\psi_2.\end{aligned}$$

$\mathcal{A}_6$ is a subspace of the space $\mathcal{O}_6$ and $\mathcal{Z}_6$ is a subspace of the space

$$\begin{aligned}\mathcal{S}_6 &= \{\phi_A \otimes R_A + \phi_B \otimes R_B + \phi_C \otimes R_C + \phi_D \otimes R_D \in (\mathcal{H}_0 \oplus \mathcal{H}_2) \otimes \mathcal{O}_6 : \\ &\quad \phi_A R_A + \phi_B R_B + \phi_C R_C + \phi_D R_D = 0\}.\end{aligned}$$

We are going to construct a basis for $\mathcal{O}_6$ and $\mathcal{S}_6$. These spaces are invariant with respect to the group D_4, generated by r and s, where

$$s\, p(x, y) = p(-x, y).$$

R_4 is a subgroup of D_4. D_4 splits $\mathcal{O}_6$ in 4 subspaces, generated by the components of the vectors

$$\begin{aligned}p_1 &= [P^{6,0} + P^{0,6}, P^{4,2} + P^{2,4}], \\ p_2 &= [P^{5,1} - P^{1,5}], \\ p_3 &= [P^{6,0} - P^{0,6}, P^{4,2} - P^{2,4}], \\ p_4 &= [P^{5,1} + P^{1,5}, P^{3,3}].\end{aligned}$$

The character of $\mathcal{O}_6$ is thus $2\chi_1 + \chi_2 + 2\chi_3 + 2\chi_4$,where the irreducible characters $\chi_1, \chi_2, \chi_3, \chi_4$ are given by the table

	r^k	$r^k s$
χ_1	1	1
χ_2	1	-1
χ_3	$(-1)^k$	$(-1)^k$
χ_4	$(-1)^k$	$(-1)^{k+1}$

$\mathcal{S}_6$ is the kernel of the surjective map

$$L' : (\mathcal{H}_0 \oplus \mathcal{H}_2) \otimes \mathcal{O}_6 \to \mathcal{O}_4 \oplus \mathcal{O}_6 \oplus \mathcal{O}_8 :$$
$$\phi_A \otimes R_A + \phi_B \otimes R_B + \phi_C \otimes R_C + \phi_D \otimes R_D \mapsto \phi_A R_A + \phi_B R_B + \phi_C R_C + \phi_D R_D$$

and hence has the following character

$$\begin{aligned}
\chi^{\mathcal{S}_6} &= \chi^{(\mathcal{H}_0 \oplus \mathcal{H}_2) \otimes \mathcal{O}_6} - \chi^{\mathcal{O}_4 \oplus \mathcal{O}_6 \oplus \mathcal{O}_8} \\
&= (\chi^{\mathcal{H}_0} + \chi^{\mathcal{H}_2})\chi^{\mathcal{O}_6} - (\chi^{\mathcal{O}_4} + \chi^{\mathcal{O}_6} + \chi^{\mathcal{O}_8}) \\
&= (2\chi_1 + \chi_3 + \chi_4)(2\chi_1 + \chi_2 + 2\chi_3 + 2\chi_4) \\
&\quad -(2\chi_1 + \chi_3 + \chi_4 + 2\chi_1 + \chi_2 + 2\chi_3 + 2\chi_4 + 3\chi_1 + 2\chi_2 + 2\chi_3 + 2\chi_4) \\
&= \chi_1 + 2\chi_2 + 2\chi_3 + 2\chi_4.
\end{aligned}$$

Consequently, $\mathcal{S}_6$ admits a basis $\{\sigma_{1,1}, \sigma_{2,1}, \sigma_{2,2}, \sigma_{3,1}, \sigma_{3,2}, \sigma_{4,1}, \sigma_{4,2}\}$, where

$$\begin{aligned}
s_1 &= [\sigma_{1,1}] &&= \phi_A \otimes p_1 A_{11} + \phi_B \otimes p_1 B_{11} + \phi_C \otimes p_3 C_{31} + \phi_D \otimes p_4 D_{41} \\
s_2 &= [\sigma_{2,1}, \sigma_{2,2}] &&= \phi_A \otimes p_2 A_{22} + \phi_B \otimes p_2 B_{22} + \phi_C \otimes p_4 C_{42} + \phi_D \otimes p_3 D_{32} \\
s_3 &= [\sigma_{3,1}, \sigma_{3,2}] &&= \phi_A \otimes p_3 A_{33} + \phi_B \otimes p_3 B_{33} + \phi_C \otimes p_1 C_{13} + \phi_D \otimes p_2 D_{23} \\
s_4 &= [\sigma_{4,1}, \sigma_{4,2}] &&= \phi_A \otimes p_4 A_{44} + \phi_B \otimes p_4 B_{44} + \phi_C \otimes p_2 C_{24} + \phi_D \otimes p_1 D_{14}.
\end{aligned}$$

The computation of the matrices $A_{i,j}, B_{i,j}, C_{i,j}, D_{i,j}$ requires only the solution of linear systems.

Let $s_1 k_1 + s_2 k_2$ ($k_1 \in \mathbb{R}^{1\times 1}, k_2 \in \mathbb{R}^{2\times 1}$) generate the one-dimensional subspace $\mathcal{Z}_{6,0}$ of $\mathcal{Z}_6$ with the character ψ_0. Then $p_3 C_{31} k_1 + p_4 C_{42} k_2$ and $p_3 D_{32} k_2 + p_4 D_{41} k_1$ belong to the subspace $\mathcal{A}_{6,2}$ with the character $2\psi_2$. Now, we proceed like Morrow and Patterson [5]. The matrices A_{33} and A_{44} being invertible, we have that $s_3 A_{33}^{-1} C_{31} k_1 + s_4 A_{44}^{-1} C_{42} k_2$ and $s_3 A_{33}^{-1} D_{32} k_2 + s_4 A_{44}^{-1} D_{41} k_1$ belong to the subspace $\mathcal{Z}_{6,2}$ of $\mathcal{Z}_6$ with the character $2\psi_2$. Consequently, $p_3 B_{33} A_{33}^{-1} C_{31} k_1 + p_4 B_{44} A_{44}^{-1} C_{42} k_2$ and $p_3 B_{33} A_{33}^{-1} D_{32} k_2 + p_4 B_{44} A_{44}^{-1} D_{41} k_1$ belong also to $\mathcal{A}_{6,2}$. The matrices $B_{33} A_{33}^{-1}$ and $B_{44} A_{44}^{-1}$ appear to have the same characteristic polynomial

$$\lambda^2 + \frac{5908}{5397}\lambda + \frac{900}{5397},$$

which admits 2 distinct real roots λ_1 and λ_2. We have the matrix decompositions

$$\begin{aligned}
B_{33} A_{33}^{-1} &= X_3^{-1} \Lambda X_3 \\
B_{44} A_{44}^{-1} &= X_4^{-1} \Lambda X_4,
\end{aligned}$$

where

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}.$$

Let

$$\begin{aligned}
f_3 &= X_3 C_{31} k_1 \qquad g_3 = X_3 D_{32} k_2 \\
f_4 &= X_4 C_{42} k_2 \qquad g_4 = X_4 D_{41} k_1.
\end{aligned}$$

Then the space spanned by the polynomials

$$p_3X_3^{-1}f_3+p_4X_4^{-1}f_4, p_3X_3^{-1}g_3+p_4X_4^{-1}g_4, p_3X_3^{-1}\Lambda f_3+p_4X_4^{-1}\Lambda f_4, p_3X_3^{-1}\Lambda g_3+p_4X_4^{-1}\Lambda g_4$$

is at most of dimension 2, since it is a subspace of $\mathcal{A}_{6,2}$. Hence, the rank of the matrix

$$\begin{bmatrix} f_3 & g_3 & \Lambda f_3 & \Lambda g_3 \\ f_4 & g_4 & \Lambda f_4 & \Lambda g_4 \end{bmatrix} = \begin{bmatrix} \phi_{3,1} & \gamma_{3,1} & \lambda_1\phi_{3,1} & \lambda_1\gamma_{3,1} \\ \phi_{3,2} & \gamma_{3,2} & \lambda_2\phi_{3,2} & \lambda_2\gamma_{3,2} \\ \phi_{4,1} & \gamma_{4,1} & \lambda_1\phi_{4,1} & \lambda_1\gamma_{4,1} \\ \phi_{4,2} & \gamma_{4,2} & \lambda_2\phi_{4,2} & \lambda_2\gamma_{4,2} \end{bmatrix}$$

is at most 2. This is only possible if one of the following three conditions is satisfied

1. $\begin{vmatrix} \phi_{3,1} & \gamma_{3,1} \\ \phi_{4,1} & \gamma_{4,1} \end{vmatrix} = \begin{vmatrix} \phi_{3,2} & \gamma_{3,2} \\ \phi_{4,2} & \gamma_{4,2} \end{vmatrix} = 0$
2. $\phi_{3,1} = \phi_{4,1} = \gamma_{3,1} = \gamma_{4,1} = 0$
3. $\phi_{3,2} = \phi_{4,2} = \gamma_{3,2} = \gamma_{4,2} = 0.$

The second and the third condition cannot be satisfied. The first condition is a system of 2 homogeneous quadratic equations in the coefficients of the vector

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \kappa_{1,1} \\ \kappa_{2,1} \\ \kappa_{2,2} \end{bmatrix}.$$

As the system is invariant with respect to the transformation

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \mapsto \begin{bmatrix} k_1 \\ -k_2 \end{bmatrix},$$

the solutions of the system occur in pairs. The system admits 2 pairs of projective solutions, one of the pairs is complex, the other is real. Choosing one of the real solutions, we have immediately 4 polynomials in $\mathcal{A}_6$

$$\begin{aligned} &p_1A_{11}k_1 + p_2A_{22}k_2 \\ &p_1B_{11}k_1 + p_2B_{22}k_2 \\ &p_3C_{31}k_1 + p_4C_{42}k_2 \\ &p_4D_{41}k_1 + p_3D_{32}k_2 \end{aligned}$$

These orthogonal polynomials are linearly independent and satisfy Möller's conditions [4] for generating a minimal cubature formula. The cubature formula they generate, is either the cubature formula $Q[f]$, described in [1] or $Q'[f] = Q[sf]$. The computations have been done with the computer algebra system REDUCE [3] and are listed at the end of this paper.

References

[1] R. Cools and A. Haegemans. Another step forward in searching for cubature formulae with a minimal number of knots for the square. *Computing*, 40:139–146, 1988.

[2] K. Gatermann. Gruppentheoretische Konstruction von symmetrischen Kubaturformeln. Technical Report TR 90-1, Konrad-Zuse-zentrum für Informationstechnik Berlin, 1990.

[3] A.C. Hearn. *REDUCE User's Guide, Version 3.3.* The RAND Corporation, 1988.

[4] H.M. Möller. Kubaturformeln mit minimaler Knotenzahl. *Numer. Math.*, 25:185–200, 1976.

[5] C.R. Morrow and T.N.L. Patterson. Construction of algebraic cubature rules using polynomial ideal theory. *SIAM J. Numer. Anal.*, 15:953–976, 1978.

```
on gcd;

on ezgcd;

let L(0) = 0;

*** L declared operator

for all k,m let u*p(k,m) = p(k+1,m) + L(k)*p(k-1,m);

*** P declared operator

for all k,m let v*p(k,m) = p(k,m+1) + L(m)*p(k,m-1);

for all a,k,l let co(a,k,l) = coeffn(a,p(k,l),1);

*** CO declared operator

for all a let comp(a) = sub({x=u,y=v,fD = u*v,fA = u^2 + v^2, fC = u^2 - v^2},a);

*** COMP declared operator

procedure a(k,l);
begin
  scalar res;
  res := (fA+fC)*p(k,l+1) - 2*fD*p(k+1,l);
  res := res - comp(res);
  return res;
end;

A

for all a such that numberp(a) let rev(a) = a;

*** REV declared operator

for all a let rev(-a) = -rev(a);

for all k let rev(L(k)) = L(k);

let rev(x) = -y;

let rev(y) = x;

let rev(fA) = fA;

let rev(fC) = -fC;

let rev(fD) = -fD;

for all k,l let rev(p(k,l)) = if (evenp(k)) then p(l,k) else -p(l,k);

for all a, b let rev(a*b) = rev(a)*rev(b);

for all a, b let rev(a+b) = rev(a)+rev(b);

for all a, b let rev(a^b) = rev(a)^rev(b);

for all a, b let rev(a/b) = rev(a)/rev(b);

syz1 := a0*(a(4,1)+rev(a(4,1)))+a1*(a(2,3)+rev(a(2,3)))+a2*(a(0,5)+rev(a(0,5)))$

syz2 := b0*(a(5,0)+rev(a(5,0)))+b1*(a(3,2)+rev(a(3,2)))+b2*(a(1,4)+rev(a(1,4)))$

syz3 := c0*(a(4,1)-rev(a(4,1)))+c1*(a(2,3)-rev(a(2,3)))+c2*(a(0,5)-rev(a(0,5)))$

syz4 := d0*(a(5,0)-rev(a(5,0)))+d1*(a(3,2)-rev(a(3,2)))+d2*(a(1,4)-rev(a(1,4)))$

comp(syz1);

0

comp(syz2);

0

comp(syz3);

0

comp(syz4);

0

clear a;

sol := solve( { co(syz1,4,0) = 0, co(syz1,2,2) = 0},{a1,a2} );

                 L(4)*A0
SOL := {{A1=----------,
                 L(3)

                    A0*(L(5)*L(3) - L(4)*L(2))
          A2= - ------------------------------}}
                          L(5)*L(3)

syz1 := sub( sol, syz1)$
```

```
sol := solve(  co(syz2,3,1) = 0 , b2 );

                 L(5)*L(4)*B0 - L(4)*L(2)*B1 - L(3)*L(2)*B1
SOL := {B2= - ------------------------------------------------}
                                 L(4)*L(2)

syz2 := sub( sol, syz2)$

sol := solve(  co(syz3,4,0) = 0 , c2 );

              L(5)*C0 + L(2)*C1
SOL := {C2=-------------------}
                   L(5)

syz3 := sub( sol, syz3)$

sol := solve(  co(syz4,3,1) = 0 , d2 );

              L(5)*L(4)*D0 - L(4)*L(2)*D1 + L(3)*L(2)*D1
SOL := {D2=-------------------------------------------}
                              L(4)*L(2)

syz4 := sub( sol, syz4)$

matrix s1(1,1), s2(1,2), s3(1,2), s4(1,2);

s1(1,1) := sub(a0=1,syz1)$

s2(1,1) := sub({b0=1,b1=0},syz2)$

s2(1,2) := sub({b0=0,b1=1},syz2)$

s3(1,1) := sub({c0=1,c1=0},syz3)$

s3(1,2) := sub({c0=0,c1=1},syz3)$

s4(1,1) := sub({d0=1,d1=0},syz4)$

s4(1,2) := sub({d0=0,d1=1},syz4)$

clear syz1,syz2,syz3,syz4,a0,a1,b0,b1,b2,c0,c1,c2,d0,d1,d2;

matrix p1(1,2),p2(1,1),p3(1,2),p4(1,2);

p1(1,1) := p(6,0) + p(0,6)$

p1(1,2) := p(4,2) + p(2,4)$

p2(1,1) := p(5,1) - p(1,5)$

p3(1,1) := p(6,0) - p(0,6)$

p3(1,2) := p(4,2) - p(2,4)$

p4(1,1) := p(5,1) + p(1,5)$

p4(1,2) := p(3,3)$

let projs(1) = s1(1,1);

*** PROJS declared operator

let projs(2) = s2(1,1);

let projs(3) = s2(1,2);

let projs(4) = s3(1,1);

let projs(5) = s3(1,2);

let projs(6) = s4(1,1);

let projs(7) = s4(1,2);

for all a let projm(a,1) = coeffn(a,fA,1);

*** PROJM declared operator

for all a let projm(a,2) = sub({fA = 0, fC = 0, fD = 0},a);

for all a let projm(a,3) = coeffn(a,fC,1);

for all a let projm(a,4) = coeffn(a,fD,1);

for all a let projp(a,1) = (co(a,6,0)+co(a,0,6))/2;

*** PROJP declared operator

for all a let projp(a,2) = (co(a,4,2)+co(a,2,4))/2;
```

```
for all a let projp(a,3) = (co(a,5,1)-co(a,1,5))/2;

for all a let projp(a,4) = (co(a,6,0)-co(a,0,6))/2;

for all a let projp(a,5) = (co(a,4,2)-co(a,2,4))/2;

for all a let projp(a,6) = (co(a,5,1)+co(a,1,5))/2;

for all a let projp(a,7) = co(a,3,3);

for all s,m,p let proj(s,m,p) = projp(projm(projs(s),m),p);

*** PROJ declared operator
matrix a11(2,1),a22(1,2),a33(2,2),a44(2,2);

matrix b11(2,1),b22(1,2),b33(2,2),b44(2,2);

matrix c31(2,1),c42(2,2),c13(2,2),c24(1,2);

matrix d41(2,1),d32(2,2),d23(1,2),d14(2,2);

a11(1,1) := proj(1,1,1)$

a11(2,1) := proj(1,1,2)$

a22(1,1) := proj(2,1,3)$

a22(1,2) := proj(3,1,3)$

a33(1,1) := proj(4,1,4)$

a33(2,1) := proj(4,1,5)$

a33(1,2) := proj(5,1,4)$

a33(2,2) := proj(5,1,5)$

a44(1,1) := proj(6,1,6)$

a44(2,1) := proj(6,1,7)$

a44(1,2) := proj(7,1,6)$

a44(2,2) := proj(7,1,7)$

b11(1,1) := proj(1,2,1)$

b11(2,1) := proj(1,2,2)$

b22(1,1) := proj(2,2,3)$

b22(1,2) := proj(3,2,3)$

b33(1,1) := proj(4,2,4)$

b33(2,1) := proj(4,2,5)$

b33(1,2) := proj(5,2,4)$

b33(2,2) := proj(5,2,5)$

b44(1,1) := proj(6,2,6)$

b44(2,1) := proj(6,2,7)$

b44(1,2) := proj(7,2,6)$

b44(2,2) := proj(7,2,7)$

c31(1,1) := proj(1,3,4)$

c31(2,1) := proj(1,3,5)$

c42(1,1) := proj(2,3,6)$

c42(2,1) := proj(2,3,7)$

c42(1,2) := proj(3,3,6)$

c42(2,2) := proj(3,3,7)$

c13(1,1) := proj(4,3,1)$

c13(2,1) := proj(4,3,2)$
```

```
c13(1,2) := proj(5,3,1)$

c13(2,2) := proj(5,3,2)$

c24(1,1) := proj(6,3,3)$

c24(1,2) := proj(7,3,3)$

d41(1,1) := proj(1,4,6)$

d41(2,1) := proj(1,4,7)$

d32(1,1) := proj(2,4,4)$

d32(2,1) := proj(2,4,5)$

d32(1,2) := proj(3,4,4)$

d32(2,2) := proj(3,4,5)$

d23(1,1) := proj(4,4,3)$

d23(1,2) := proj(5,4,3)$

d14(1,1) := proj(6,4,1)$

d14(2,1) := proj(6,4,2)$

d14(1,2) := proj(7,4,1)$

d14(2,2) := proj(7,4,2)$

matrix t1(1,1),t2(1,2),t3(1,2),t4(1,2);

t1 := s1 - fA*p1*a11 - p1*b11 -fC*p3*c31 -fD*p4*d41;

T1(1,1) := 0

t2 := s2 - fA*p2*a22 - p2*b22 -fC*p4*c42 -fD*p3*d32;

T2(1,1) := 0

T2(1,2) := 0

t3 := s3 - fA*p3*a33 - p3*b33 -fC*p1*c13 -fD*p2*d23;

T3(1,1) := 0

T3(1,2) := 0

t4 := s4 - fA*p4*a44 - p4*b44 -fC*p2*c24 -fD*p1*d14;

T4(1,1) := 0

T4(1,2) := 0

clear t1,t2,t3,t4,proj;

det(a33);

 L(5) + L(2)
-------------
    L(5)

det(b33);

4*L(3)*L(1)

lambeq3:= first(first(mateigen(b33/a33,lamb)));

LAMBEQ3 := 4*L(5)*L(3)*L(1) + 2*L(5)*L(3)*LAMB + 2*L(5)*L(2)*LAMB + 2
                                 2                                  2
*L(5)*L(1)*LAMB + L(5)*LAMB  + 2*L(4)*L(2)*LAMB + L(2)*LAMB

lambeq4:= first(first(mateigen(b44/a44,lamb)));

LAMBEQ4 := 4*L(5)*L(3)*L(1) + 2*L(5)*L(3)*LAMB + 2*L(5)*L(2)*LAMB + 2
                                 2                                  2
*L(5)*L(1)*LAMB + L(5)*LAMB  + 2*L(4)*L(2)*LAMB + L(2)*LAMB

lambeq3 - lambeq4;

0

%square with weight function 1
for all k let L(k) = k*k/((2*k+1)*(2*k-1));

eig := mateigen(b33/a33,lamb);


                  2
EIG := {{5397*LAMB  + 5908*LAMB + 900,

         1,
```

```
                    264*ARBCOMPLEX(3)
          MAT(1,1) := -----------------
                       771*LAMB + 250

          MAT(2,1) := ARBCOMPLEX(3)
          }}

lambeq := first(first(eig));

                        2
LAMBEQ := 5397*LAMB  + 5908*LAMB + 900

sol := solve(lambeq=0,lamb);

                  2*(2*SQRT(241801) + 1477)
SOL := {LAMB= - --------------------------,LAMB=
                           5397

          2*(2*SQRT(241801) - 1477)
         --------------------------}
                    5397

lamb1 := rhs(first(sol));

              2*(2*SQRT(241801) + 1477)
LAMB1 :=  - --------------------------
                         5397

lamb2 := rhs(second(sol));

            2*(2*SQRT(241801) - 1477)
LAMB2 := --------------------------
                      5397

sub(lamb=lamb1,lambeq);

0

sub(lamb=lamb2,lambeq);

0

clear lambeq;

matrix vec(2,1),x3(2,2);

let arbcomplex(1) = 1;

vec := third(first(eig))$

x3(1,1) := sub(lamb=lamb1,vec(1,1))$

x3(2,1) := sub(lamb=lamb1,vec(2,1))$

x3(1,2) := sub(lamb=lamb2,vec(1,1))$

x3(2,2) := sub(lamb=lamb2,vec(2,1))$

x3 := x3^(-1)$

b33/a33 - (x3^(-1))*(MAT((lamb1,0),(0,lamb2)))*x3;

MAT(1,1) := 0

MAT(1,2) := 0

MAT(2,1) := 0

MAT(2,2) := 0

eig := mateigen(b44/a44,lamb);

                      2
EIG := {{5397*LAMB  + 5908*LAMB + 900,

          1,

                      342*ARBCOMPLEX(4)
          MAT(1,1) := ---------------------
                      5*(257*LAMB + 150)

          MAT(2,1) := ARBCOMPLEX(4)
          }}

lambeq := first(first(eig));

                        2
LAMBEQ := 5397*LAMB  + 5908*LAMB + 900

sub(lamb=lamb1,lambeq);

0

sub(lamb=lamb2,lambeq);

0

matrix vec(2,1),x4(2,2);

let arbcomplex(2) = 1;

vec := third(first(eig))$

x4(1,1) := sub(lamb=lamb1,vec(1,1))$
```

```
x4(2,1) := sub(lamb=lamb1,vec(2,1))$

x4(1,2) := sub(lamb=lamb2,vec(1,1))$

x4(2,2) := sub(lamb=lamb2,vec(2,1))$

x4 := x4^(-1)$

b44/a44 - (x4^(-1))*(MAT((lamb1,0),(0,lamb2)))*x4;

MAT(1,1) := 0

MAT(1,2) := 0

MAT(2,1) := 0

MAT(2,2) := 0

matrix f3(2,1),f4(2,1),g3(2,1),g4(2,1),k1(1,1),k2(2,1);

k1(1,1) := k11$

k2(1,1) := k21$

k2(2,1) := k22$

f3 := x3*c31*k1$

f4 := x4*c42*k2$

g3 := x3*d32*k2$

g4 := x4*d41*k1$

%clear a11,a22,a33,a44,b11,b22,b33,b44,c31,c42,c13,c24,d41,d32,d23,d14;
clear x3,x4;

load groebner;

vgl := {f3(1,1)=0,f4(1,1)=0,g3(1,1)=0,g4(1,1)=0}$

sol := groebner(vgl,{k11,k21,k22});

SOL := {K11,K21,K22}
```

```
vgl := {f3(2,1)=0,f4(2,1)=0,g3(2,1)=0,g4(2,1)=0}$

sol := groebner(vgl,{k11,k21,k22});

SOL := {K11,K21,K22}

let k11 = 1;

vgl := {f3(1,1)*g4(1,1)=f4(1,1)*g3(1,1),f3(2,1)*g4(2,1)=f4(2,1)*g3(2,1)}$

sol := groebner(vgl,{k21,k22});

                                                  3
SOL := {173600000*K21 + 277306261533*K22  - 1362342400*K22,

                                  4                        2
        1819406381918013*K22  - 8906436777600*K22  - 17221120000}

eq1 := first(sol);

                                                  3
EQ1 := 173600000*K21 + 277306261533*K22  - 1362342400*K22

k22 := sqrt(sqk22);

K22 := SQRT(SQK22)

eq2 := second(sol);

                                  2
EQ2 := 1819406381918013*SQK22  - 8906436777600*SQK22 - 17221120000

sol := solve(eq2=0,sqk22);

                  1600*(750*SQRT(825385) - 424213)
SOL := {SQK22= - ---------------------------------,SQK22=
                              277306261533

         1600*(750*SQRT(825385) + 424213)
        ---------------------------------}
                  277306261533

sqk22 := rhs(first(reverse(sol)));

          1600*(750*SQRT(825385) + 424213)
SQK22 := ---------------------------------
                   277306261533

k22;

 40*SQRT(750*SQRT(825385) + 424213)
-----------------------------------
         81*SQRT(42265853)
```

```
k21 := rhs(first(solve(eq1=0,k21)));

K21 :=  - (2*SQRT(750*SQRT(825385) + 424213)*(250*SQRT(825385) -
              142417))/(146475*SQRT(42265853))
k1;

MAT(1,1) := 1
k2;

MAT(1,1) :=  - (2*SQRT(750*SQRT(825385) + 424213)*(250*SQRT(825385) -
                    142417))/(146475*SQRT(42265853))
MAT(2,1) :=
             40*SQRT(750*SQRT(825385) + 424213)
            -----------------------------------
                    81*SQRT(42265853)
u := s1(1,1) + k21*s2(1,1) + k22*s2(1,2)$

p := projm(u,1)$

projp(p,1);

 29
-----
 675
projp(p,2);

 161
-----
 81
projp(p,3);

 - (SQRT(750*SQRT(825385) + 424213)*(32125*SQRT(825385) - 13496747))/
(4833675*SQRT(42265853))
p := projm(u,2)$

projp(p,1);

 2
---
 3
projp(p,2);

   50
 - ----
   99
projp(p,3);

(2*SQRT(750*SQRT(825385) + 424213)*(125*SQRT(825385) - 41371))/(64449
   *SQRT(42265853))
p := projm(u,3)$

projp(p,4);

   29
 - -----
   675
projp(p,5);

 1
----
 81
projp(p,6);

 - (SQRT(750*SQRT(825385) + 424213)*(125*SQRT(825385) - 757471))/(
   690525*SQRT(42265853))
projp(p,7);

 80*SQRT(750*SQRT(825385) + 424213)
-----------------------------------
         81*SQRT(42265853)
p := projm(u,4)$

projp(p,4);

(4*SQRT(750*SQRT(825385) + 424213)*(250*SQRT(825385) - 142417))/(
   146475*SQRT(42265853))
projp(p,5);

(2*SQRT(750*SQRT(825385) + 424213)*(625*SQRT(825385) - 259369))/(
   193347*SQRT(42265853))
projp(p,6);
```

```
    1408
 - ------
    675

projp(p,7);

    320
 - -----
    81

on bigfloat;

precision 20;

20

on numval;

u := s1(1,1) + k21*s2(1,1) + k22*s2(1,2)$

let p(0,0) = 1;

let p(1,0) = x;

let p(0,1) = y;

let p(1,1) = x*y;

for all k,m such that m>1 let p(k,m) = y*p(k,m-1) - L(m-1)*p(k,m-2);

for all k,m such that k>1 let p(k,m) = x*p(k-1,m) - L(k-1)*p(k-2,m);

let x = 982639223540855472 95/(10^20);

let y = -69807610454956756478/(10^20);

u;

 - 1.355 E -21 *(FD - 0.670 40590 40590 40590 41*FA - 0.231 29151 291
                 51 29151 29*FC + 1.726 19926 19926 19926 2)
let x = 825775835902963937 30/(10^20);

let y = -939486382816736907 21/(10^20);

u;

1.948 E -21 *(FD - 0.529 46611 90965 09240 25*FA - 0.153 69609 85626
                 28336 76*FC + 1.572 38193 01848 04926 1)
let x = 188586138718641954 60/(10^20);

let y = -953539528201532015 85/(10^20);

u;

 - 1.348 E -21 *(FD + 3.495 54896 14243 32344 2*FA - 0.142 35905 0445
                  1 03857 57*FC - 3.264 09495 54896 14243 3)
let x = 812520548304813100 49/(10^20);

let y = -315623432915254195 99/(10^20);

u;

 - 1.536 E -21 *(FD + 1.084 63541 66666 66666 7*FA - 0.063 80208 3333
                  3 33333 333*FC - 0.530 59895 83333 33333 33)
let x = 525320250364547762 34/(10^20);

let y = -712001913075336306 55/(10^20);

u;

2.3 E -22*(FD + 1.776 95652 17391 30434 8*FA - 0.157 82608 69565 2173
           9 13*FC - 1.117 39130 43478 26087 )
let x = 4165807191202236827 4/(10^21);

let y = -424847248848669250 62/(10^20);

u;

1.24  E -22 *(FD + 16.24 19354 83870 96774 2*FA - 0.230 64516 12903 2
                 2580 65*FC - 2.822 58064 51612 90322 6)
end;
```

```
a11  =

   0.042962962962963
   1.987654320987654

b11  =

   0.666666666666667
  -0.505050505050505

c31  =

  -0.042962962962963
   0.012345679012346

d41  =

  -2.085925925925926
  -3.950617283950618

a22  =

   1.946969696969697  -2.012500000000000

b22  =

  -1.136363636363636   1.875000000000000

c42  =

   0.053030303030303   2.012500000000000
   0.000000000000000   2.000000000000000

d32  =

  -2.000000000000000   0.000000000000000
  -1.893939393939394   2.025000000000000

a33  =

  -1.000000000000000  -1.056000000000000
   1.000000000000000  -1.000000000000000

b33  =

   0.666666666666667   0.000000000000000
  -1.012987012987013   0.514285714285714

c13  =

   1.000000000000000   1.056000000000000
   1.000000000000000   1.000000000000000

d23  =

   0.000000000000000   2.112000000000000

a44  =

   1.946969696969697   0.012500000000000
   0.000000000000000   2.000000000000000

b44  =

  -1.136363636363636   0.525000000000000
   0.962000962000962  -1.015873015873016

c24  =

   0.053030303030303  -0.012500000000000

d14  =

  -2.000000000000000   0.000000000000000
  -1.893939393939394  -2.025000000000000

k1   =

     1.

k2   =

  -0.187069413008573
   0.079868926111675

a1 = [ a11*k1 b11*k1 ];

a2 = [ a22*k2 b22*k2 ];

[a1 ; a2]  =

   0.042962962962963   0.666666666666667
   1.987654320987654  -0.505050505050505
  -0.524954692157346   0.362333114878223

rank([a1 ; a2])  =

     2.

a3 = [ c31*k1 d32*k2 ];

a4 = [ c42*k2 d41*k1 ];

[a3 ; a4]  =

  -0.042962962962963   0.374138826017146
   0.012345679012346   0.516032706074196
   0.150815866140200  -2.085925925925926
   0.159737852223349  -3.950617283950618

rank([a3 ; a4])  =

     2.

z1 = k1; z2 = k2; z3 = inv(a33)*a3; z4 = inv(a44)*a4;

rank([z3;z4])  =

     2.

rank([a11*z1 b11*z1 c13*z3 d14*z4; a22*z2 b22*z2 c24*z4 d23*z3])  =

     2.

rank([a33*z3 b33*z3 c31*z1 d32*z2; a44*z4 b44*z4 c42*z2 d41*z1])  =

     2.
```

Related TW-Reports

TW 27: A. Haegemans. Tables of circularly symmetrical integration formulas of degree $2d-1$ for two-dimensional circularly symmetrical regions. TW 27, K.U. Leuven Applied Mathematics and Programming Division, 1975.

TW 28: A. Haegemans. Tables of symmetrical cubature formulas for the two-dimensional hexagon. Report TW 28, Dept. of Computer Science, K.U. Leuven, 1976.

TW 71: R. Cools and A. Haegemans. Construction of fully symmetric cubature formulae of degree $4k-3$ for fully symmetric planar regions. Report TW 71, Dept. of Computer Science, K.U. Leuven, 1985.

TW 76: R. Cools and A. Haegemans. Tables of degree $2k-1/2k+1$ pairs of cubature formulae for symmetric planar regions, obtained by optimal addition of knots. Report TW 76, Dept. of Computer Science, K.U. Leuven, 1986.

TW 77: R. Cools and A. Haegemans. Automatic computation of knots and weights of cubature formulae for circular symmetric planar regions. Report TW 77, Dept. of Computer Science, K.U. Leuven, 1986.

TW 83: R. Cools and A. Haegemans. Tables of sequences of cubature formulae for circular symmetric planar regions. Report TW 83, Dept. of Computer Science, K.U. Leuven, 1986.

TW 85: M. Beckers and A. Haegemans. Construction of three-dimensional invariant cubature formulae. Report TW 85, Dept. of Computer Science, K.U. Leuven, 1986.

TW 95: P. Verlinden and A. Haegemans. Computing the knots of a cubature formula as the solution of an eigenvalue problem. Report TW 95, Dept. of Computer Science, K.U. Leuven, 1987.

TW 96: R. Cools and A. Haegemans. Construction of minimal cubature formulae for the square and the triangle using invariant theory. Report TW 96, Dept. of Computer Science, K.U. Leuven, 1987.

TW 97: R. Cools and A. Haegemans. Construction of symmetric cubature formulae with the number of knots (almost) equal to Möller's lower bound (with proofs). Report TW 97, Dept. of Computer Science, K.U. Leuven, 1987.

TW 106: P. Verlinden, R. Cools, D. Roose, and A. Haegemans. The construction of cubature formulae for a family of integrals: a bifurcation problem. Report TW 106, Dept. of Computer Science, K.U. Leuven, 1988.

TW 111: R. Cools and A. Haegemans. A lower bound for the number of function evaluations in an error estimate for numerical integration. Report TW 111, Dept. of Computer Science, K.U. Leuven, 1988.

TW 112: R. Cools and A. Haegemans. On the construction of multi-dimensional embedded cubature formulae. Report TW 112, Dept. of Computer Science, K.U. Leuven, 1988.

TW 127: R. Cools and A. Haegemans. The construction of cubature formulae using continuation and bifurcation software. Report TW 127, Dept. of Computer Science, K.U. Leuven, 1989.

TW 128: M. Beckers and A. Haegemans. The construction of cubature formulae for the tetrahedron. Report TW 128, Dept. of Computer Science, K.U. Leuven, 1990.

TW 132: R. Cools and A. Haegemans. An embedded family of cubature formulae for n-dimensional product regions. Report TW 132, Dept. of Computer Science, K.U. Leuven, 1990.

TW 137: P. Verlinden and R. Cools. Minimal cubature formulae of degree $4k+1$ for integrals with circular symmetry. Report TW 137, Dept. of Computer Science, K.U. Leuven, 1990.

TW 145: M. Beckers and A. Haegemans. Lattice methods for numerical integration. Report TW 145, Dept. of Computer Science, K.U. Leuven, 1991.

TW 149: P. Verlinden and R. Cools. A cubature formula of degree 19 with 68 nodes for integration over the square. Report TW 149, Dept. of Computer Science, K.U. Leuven, 1991.

TW 150: R. Cools. A suite of codes for region collection management in adaptive numerical integration algorithms. Report TW 150, Dept. of Computer Science, K.U. Leuven, 1991.

TW 154: R. Cools and A. Haegemans. CUBPACK: Progress report. Report TW 154, Dept. of Computer Science, K.U. Leuven, 1991.

TW 155: P. Verlinden and R. Cools. The algebraic construction of a minimal cubature formula of degree 11 for the square. Report TW 155, Dept. of Computer Science, K.U. Leuven, 1991.

TW 157: M. Beckers and A. Haegemans. Transformation of integrands for lattice rules. Report TW 157, Dept. of Computer Science, K.U. Leuven, 1991.

TW 158: R. Cools. A survey of methods for constructing cubature formulae. Report TW 158, Dept. of Computer Science, K.U. Leuven, 1991.