Quadratic forms in PARI/GP

B. Allombert

IMB
CNRS/Université de Bordeaux

10/01/2025

Quadratic forms in GP

Quadratic forms are represented by their Gram matrix, which is symmetric.

```
? Q = [2,1;1,2];
? qfeval(Q,[x,y])
%2 = 2*x^2 + 2*y*x + 2*y^2
?  [x,y]*Q*[x,y]~
%3 = 2*x^2 + 2*y*x + 2*y^2
? qfeval(Q,[x1,y1],[x2,y2])
%4 = (2*x2 + y2)*x1 + (x2 + 2*y2)*y1
?  [x1,y1]*Q*[x2,y2]~
%5 = (2*x2 + y2)*x1 + (x2 + 2*y2)*y1
```

Invariants

```
? R = [1,2,3;2,4,2;3,2,5];
? matdet(R)
%7 = -16
? qfsign(R)
%8 = [2, 1]
? [U,V]=qfgaussred(R,1)
%9 = [[1, 2, 3; 0, 1, 0; 0, 1, 1], [1, 4, -4]]
?  U~*matdiagonal(V)*U == R
%10 = 1
```

$R(x, y, z)$ is equivalent to the form $x^2 + 4y^2 - 4z^2$.

Jacobi decomposition

Compute an orthogonal basis of eigenvectors.

```
? [L, V]=qfjacobi(Q)
%11 = [[1.0000,3.000]~,[0.707,0.707;-0.707,0.707]]
? norml2(V~*Q*V - matdiagonal(L))
%12 = 2.4181271954264444951E-76
? V~*V==1
%13 = 1
```

Lattices

If L is the basis of a lattice for the standard quadratic form, the Gram matrix is `L~*L`. Conversely, if M is positive definite, then `qfcholesky` gives an (approximate) basis of a suitable lattice L.

```
? L = qfcholesky(Q)
%14 = [1.414,0.707;0,1.224]
? L~*L==Q
%15 = 1
```

Gram-Schmidt orthogonalization

If L it is possible to compute the Gram-Schmidt orthogonalization using `matqr`.

```
? L = [1,0,0;0,1,0;4,5,6];
? [q,r]=matqr(L);
? q
%18 = [-0.24254  0.74848 -0.61721]
%     [       0 -0.63621 -0.77152]
%     [-0.97014 -0.18712  0.15430]
? r
%19 = [-4.1231 -4.8507 -5.8209]
%     [      0 -1.5718 -1.1227]
%     [      0       0 0.92582]
? norml2(q*r-L)
%20 = 0.E-75
```

LLL

`qflllgram` performs LLL reduction on a positive definite quadratic form, while `qflll` applies to a lattice (historical bug). It returns the transformation matrix.

```
? M = R~*R
%21 = [14,16,22;16,24,24;22,24,38]
? qfsign(M)
%22 = [3, 0]
? T = qflllgram(M)
%23 = [-1,-2,-1;1,0,0;0,1,1]
? T~*M*T
%24 = [6,-2,0;-2,6,0;0,0,8]
```

LLL

PARI/GP LLL uses a mix of fplll and flatter algorithms.

```
? L = qfcholesky(M);
? S = qflll(L)
%26 = [-1,-2,-1;1,0,0;0,1,1]
? S == T
%27 = 1
? round((L*S)~*(L*S))
%28 = [6,-2,0;-2,6,0;0,0,8]
```

lindep

`lindep` uses LLL to find small relations between real numbers.

```
? z7 = zeta(7); z25=zeta(2)*zeta(5);
? z43 = zeta(4)*zeta(3);
? lindep([zetamult([3,2,2]),z7,z25,z43])
%31 = [16,-157,120,-36]~
? (157*z7-120*z25+36*z43)/16 - zetamult([3,2,2])
%32 = 5.078502937536913999E-38
```

Minimal vectors

`qfminim` computes the minimal vectors of a definite positive quadratic form.

```
? qfminim(Q)
%33 = [6,2,[0,1,1;1,-1,0]]
```

The output is `[N,B,M]` where N is the total number of minimal vectors, B is their norm, and M is a matrix with $N/2$ columns which are representatives of the minimal vectors modulo $\pm id$.
It is possible to specify an upper bound for the norm:

```
? qfminim(Q,10)
%34 = [18,8,[0,1,2,1,0,1,2,2,1;2,-2,-2,1,1,-1,-1,0,
```

qfcvp

`qfcvp(Q,v)` solves the closest vector problem, that is, finds integral vectors x that minimizes $Q(x - v)$.

```
? qfcvp(Q,[10/3,10/3])
%35 = [3,0.66666666666666664076,[3,4,3;4,3,3]]
```

Theta function

```
? qfrep(Q,20)
%36 = Vecsmall([0,3,0,0,0,3,0,3,0,0,0,0,0,6,0,0,0,3
? L = lfunqf(Q); lfunan(L,20)
%37 = [6,0,6,6,0,0,12,0,6,0,0,6,12,0,0,6,0,0,12,0]
? lfunparams(L)
%38 = [3,1,[0,1]]
? lfun(L,1) - 2*Pi/sqrt(3)/x
%39 = 0.E-37*x^-1+O(x^0)
? [mf,F,v] = mffromqf(Q); mfcoefs(F,20)
%40 = [1,6,0,6,6,0,0,12,0,6,0,0,6,12,0,0,6,0,0,12,0
? mfparams(F)
%41 = [3,1,-3,y,t+1]
```

Theta function for spherical polynomials

```
? p = x^6+6*y*x^5-20*y^3*x^3-15*y^4*x^2+y^6;
? p'' - deriv(p',y) + derivn(p,2,y)
%43 = 0 \\ spherical
? [mf2,F2,v2] = mffromqf(Q, p); mfcoefs(F2,20)
%44 = [0,6,0,-162,384,0,0,-1716,0,4374,0,0,-10368,3
? mfparams(F2)
%45 = [3,7,-3,y,t+1]
```

Perfection

`qfperfection` return the rank of perfection (the rank of the space spanned by the $v^t v$ for all minimal vectors v). A form is perfect if its rank of perfection is $\frac{n(n+1)}{2}$ where n is the dimension.

```
? qfperfection(Q)
%46 = 3
```

Hence Q is perfect.

qfsolve

GP includes a port of the program `qfsolve` by Denis Simon to solve integral quadratic forms.
Find a solution of $x^2 + 3y^2 - 21z^2 = 0$:

```
? qfsolve([1,0,0;0,3,0;0,0,-21])
%47 = [3,2,1]~
```

For dimension 3: Find parametric solution of
$x^2 + 3y^2 - 21z^2 = 0$:

```
? M = qfparam([1,0,0;0,3,0;0,0,-21],[3,2,1]~)
%48 = [-3,-12,9;2,-6,-6;1,0,3]
? v = y^2 * M*[1,x/y,(x/y)^2]~
%49 = [9*x^2-12*y*x-3*y^2,-6*x^2-6*y*x+2*y^2,3*x^2+
? v[1]^2+3*v[2]^2-21*v[3]^2
%50 = 0
```

qfminimize

Given a square symmetric matrix G with rational coefficients, and non-zero determinant, return $[H, U]$ such that $H = c^t UGU$ for some rational c, and H integral with minimal determinant. The coefficients of U are usually nonintegral.

```
? G = matdiagonal([650, -104329, -104329]);
? [H,U]=qfminimize(G); H
%52 = [-1,0,0;0,-1,0;0,0,1]
? U
%53 = [0,0,1/5;5/323,-1/323,0;-1/323,-5/323,0]
? U~*G*U
%54 = [-26,0,0;0,-26,0;0,0,26]
```

Hence $c = 26$ in this example.

qfauto

GP includes a port of the program ISOM by Bernt Souvignier for computation of automorphisms and isomorphisms of lattices.

- `qfauto`: compute the automorphism group of a lattice.
- `qfisom`: compute an isomorphism between two lattices.
- `qfautoexport`: export the group to GAP or MAGMA format.
- `qfisominit`: precompute invariants for qfisom.

```
? qfauto(matid(3))
%55 = [48,[[-1,0,0;0,-1,0;0,0,-1],
%     [0,0,1;0,1,0;1,0,0] ,[0,0,1;-1,0,0;0,1,0]]]
? K=nfinit(x^3-3*x+1); L=round(K.t2)
%56 = [3,0,0;0,6,3;0,3,6]
? qfauto(L)
%57 = [24,[[-1,0,0;0,-1,0;0,0,-1],
%     [1,0,0;0,0,1;0,1,0] ,[1,0,0;0,1,0;0,1,-1]]]
? T=qflllgram(L); M = T~*L*T; qfisom(L,M)
%58 = [1,0,0;0,0,-1;0,1,1]
? Q=qfisominit(L); qfisom(Q,M)
%59 = [1,0,0;0,0,-1;0,1,1]
```

qforbits

`qforbits` returns the orbits of V under the action of the group of linear transformation generated by the set G.

```
?  Q=matid(6); G=qfauto(Q); V=qfminim(Q,3);
?  apply(x->[x[1],#x],qforbits(G,V))
%61 = [[[0,0,0,0,0,1]~,6],[[0,0,0,0,1,-1]~,30],
%      [[0,0,0,1,-1,-1]~,80]]
```

We see there is only one orbit for each norm ≤ 3.

The E_8 lattice

Example: the Gram matrix of the E_8 lattice

```
? E8 = matrix(8,8,i,j,if(i==1&&j==1,4, \
       i==j || (i==1 && j<8) || (j==1 && i<8),2,1))
? E8==E8~ \\ symmetric
%63 = 1
? matdet(E8) \\ unimodular
%64 = 1
? qfsign(E8) \\ signature
%65 = [8,0]
? L = qfminim(E8); L[1..2] \\ 240 minimal vectors
%66 = [240,2]
? V = L[3][,1] \\ first minimal vector
%67 = [3, -1, -1, -1, -1, -1, -2, 2]~
? qfeval(E8,V) \\ the norm is 2
%68 = 2
```

The E_8 lattice

```
? qfperfection(E8) \\ perfection rank
%69 = 36
? G=qfauto(E8); G[1] \\ number of isometries
%70 = 696729600
? A=G[2][1] \\ one isomorphim
%71 = [0,-1,0,-2,0,-1,1,-1;0,1,0,1,0,0,0,1;1,1,0,1,
? A~*E8*A==E8
%72 = 1
```

The E_8 lattice

```
? [mf,F,C]=mffromqf(E8);
? mfparams(F)
%74 = [1,4,1,y,t-1]
? mfcoefs(F,10)
%75 = [1,240,2160,6720,17520,30240,60480,82560,1404
? mfcoef(F,100003)
%76 = 240021600648006720
? L = lfunqf(E8);
? lfunparams(L)
%78 = [1,4,[0,1]]
? lfun(L,0)
%79 = -1
```

The Leech lattice

```
? V=concat([vector(23,i,2*i+1),51,145]);
? K=matkerint(Mat(V));
? M=matdiagonal(vector(25,i,if(i==25,-1,1)));
? L24 = K~*M*K; \\ Leech lattice
? matdet(L24) \\ unimodular
%84 = 1
? L = qfminim(L24); L[1..2]
%85 = [196560,4]
? qfperfection(L24)
%86 = 300
```

The Leech lattice

```
? G=qfauto(L24); G[1] \\ number of isometries
%87 = 8315553613086720000
? [mf,F,C]=mffromqf(L24);
? mfparams(F)
%89 = [1,12,1,y,t-1]
? mfcoef(F,100003)
%90 = 948503977475136166631426756753238862
%     021155110936326533120
```