GAP 4 Package Forms

Sesquilinear and Quadratic

1.0

May 2007

John Bamberg
Jan De Beule

John Bamberg — Email: bamberg@cage.ugent.be
— Homepage: http://cage.ugent.be/~bamberg
— Address: Department of Pure Mathematics, Ghent University, Galglaan 2, 9000 Ghent, Belgium

Jan De Beule — Email: jdebeule@cage.ugent.be
— Homepage: http://cage.ugent.be/~jdebeule
— Address: Department of Pure Mathematics, Ghent University, Galglaan 2, 9000 Ghent, Belgium

Copyright

Contents

Chapter 1

Introduction

1.1 Philosophy

Forms is a package for computation with sesquilinear and quadratic forms on finite vector spaces. It provides users with the basic tools to work with classical groups and polar geometry, and enables one to specify a form and its corresponding geometry. Also included in the functionality of the package are operations which allow a user to change coordinates; or to “change form” and work in an isometric (or similar) polar space.

1.2 Overview over this manual

The next chapter (2) gives some basic examples of the use of this package. In ”Background Theory of Forms” (Chapter 3) we revise the basic notions of the theory of sesquilinear and quadratic forms, where we also set the notation and conventions adopted by the this package. In Sections 4.1, 4.2, and 4.3 we provide the details of the operations, functions, and attributes entailed in Forms.

Chapter 2

Examples

Here we give some simple examples that display some of the functionality of Forms.

2.1 A conic of PG(2,8)

Consider the three-dimensional vector space $V = GF(8)^3$ over $GF(8)$, and consider the following quadratic polynomial in 3 variables:

$$x_1^2 + x_2 x_3.$$

Then this polynomial defines a quadratic form in V and the zeros form a *conic* of the associated projective plane. So in particular, our quadratic form defines a degenerate parabolic quadric of Witt Index 1. We will see now how we can use Forms to view this example.

Example

```
gap> gf := GF(8);
GF(2^3)
gap> vec := gf^3;
( GF(2^3)^3 )
gap> r := PolynomialRing( gf, 3 );
GF(2^3)[x_1,x_2,x_3]
gap> poly := r.1^2 + r.2 * r.3;
x_1^2+x_2*x_3
gap> form := QuadraticFormByPolynomial( poly, r );
< quadratic form >
gap> Display( form );
Quadratic form
Gram Matrix:
 1 . .
 . . 1
 . . .
Polynomial: x_1^2+x_2*x_3
gap> IsDegenerateForm( form );
true
gap> WittIndex( form );
1
gap> IsParabolicForm( form );
true
gap> RadicalOfForm( form );
<vector space of dimension 1 over GF(2^3)>
```

Now our conic is stabilised by $GO(3,8)$, but not the same $GO(3,8)$ that is installed in GAP. However, our conic is the canonical conic given in Forms.

Example
```
gap> canonical := IsometricCanonicalForm( form );
< quadratic form >
gap> form = canonical;
true
```

So we "change forms"...

Example
```
gap> go := GO(3,8);
GO(0,3,8)
gap> mat := InvariantQuadraticForm( go )!.matrix;
[ [ Z(2)^0, 0*Z(2), 0*Z(2) ], [ 0*Z(2), 0*Z(2), 0*Z(2) ],
[ 0*Z(2), Z(2)^0, 0*Z(2) ] ]
gap> gapform := QuadraticFormByMatrix( mat, GF(8) );
< quadratic form >
gap> b := BaseChangeToCanonical( gapform );
[ [ Z(2)^0, 0*Z(2), 0*Z(2) ], [ 0*Z(2), Z(2)^0, 0*Z(2) ],
[ 0*Z(2), 0*Z(2), Z(2)^0 ]  ]
gap> hom := BaseChangeHomomorphism( b, GF(8) );
^[ [ Z(2)^0, 0*Z(2), 0*Z(2) ], [ 0*Z(2), Z(2)^0, 0*Z(2) ],
[ 0*Z(2), 0*Z(2), Z(2)^0 ] ]
gap> newgo := Image(hom, go);
Group([ [ [ Z(2)^0, 0*Z(2), 0*Z(2) ], [ 0*Z(2), Z(2^3), 0*Z(2) ],
[ 0*Z(2), 0*Z(2), Z(2^3)^6 ] ],  [ [ Z(2)^0, 0*Z(2), 0*Z(2) ],
[ Z(2)^0, Z(2)^0, Z(2)^0 ], [ 0*Z(2), Z(2)^0, 0*Z(2) ] ] ])
```

Now we look at the action of our new $GO(3,8)$ on the conic.

Example
```
gap> conic := Filtered(vec, x -> IsZero( x^form ));;
gap> Size( conic );
64
gap> orbs := Orbits(newgo, conic, OnRight);;
gap> List(orbs, Size);
[ 1, 63 ]
```

So we see that there is a fixed point, which is actually the *nucleus* of the conic, or in other words, the radical of the form.

2.2 A form for W(5,3)

The symplectic polar space $W(5,q)$ is defined by an alternating reflexive bilinear form on the six-dimensional vector space $GF(q)^6$. Any invertible 6×6 matrix A which satisfies $A+A^T=0$ is a candidate for the Gram matrix of a symplectic polarity. The canonical form we adopt in Forms for an alternating form is

$$f(x,y)=x_1y_2-x_2y_1+x_3y_4-x_4y_3\cdots+x_{2n-1}y_{2n}-x_{2n}y_{2n-1}.$$

Example

```
gap> f := GF(3);
GF(3)
gap> gram := [
[0,0,0,1,0,0],
[0,0,0,0,1,0],
[0,0,0,0,0,1],
[-1,0,0,0,0,0],
[0,-1,0,0,0,0],
[0,0,-1,0,0,0]] * One(f);;
gap> form := BilinearFormByMatrix( gram, f );
< bilinear form >
gap> IsSymplecticForm( form );
true
gap> Display( form );
Bilinear form
Gram Matrix:
 . . . 1 . .
 . . . . 1 .
 . . . . . 1
 2 . . . . .
 . 2 . . . .
 . . 2 . . .
gap> b := BaseChangeToCanonical( form );;
gap> Display( b );
 . . . . . 1
 . . 2 . . .
 . . . . 1 .
 . 2 . . . .
 . . . 1 . .
 2 . . . . .
gap> Display( b * gram * TransposedMat(b) );
 . 1 . . . .
 2 . . . . .
 . . . 1 . .
 . . 2 . . .
 . . . . . 1
 . . . . 2 .
```

Chapter 3

Background Theory on Forms

In this section, we give a very brief overview on the theory of sesquilinear and quadratic forms. The reader can find more in the texts: Cameron [Cam00], Taylor [Tay92], Aschbacher [Asc00], or Kleidman and Liebeck [KL90].

3.1 Sesquilinear forms, dualities, and polarities

A *sesquilinear form* on a vector space V over a field F, is a map f from $V \times V$ to F which is linear in the first coordinate, but semilinear in the second coordinate; that is, there is a field automorphism α (the *companion automorphism* of f) such that $f(v, \lambda w) = \lambda^\alpha f(v,w)$ for all $v, w \in V$ and $\lambda \in F$. If α is the identity, then f is *bilinear*. Two vectors v and w are *orthogonal* (w.r.t. f) if $f(v,w) = 0$. The *radical* of f is the subspace consisting of vectors which are orthogonal to every vector, and we say that f is *non-degenerate* if its radical is trivial (and *degenerate* otherwise). A *duality* δ of a projective space $\mathcal{P}$ is an incidence reversing permutation of the subspaces of $\mathcal{P}$, and a *polarity* of $\mathcal{P}$ is a duality of order 2. An example of such arises from a non-degenerate sesquilinear form; given a subspace W, we let $W^\perp$ be the set of points which are orthogonal with every element of W. We say that a subspace W is *totally isotropic* with respect to a polarity if W contains or is contained in $W^\perp$. The Birkhoff-von Neumann Theorem states that every duality of the projective space $PG(n,q)$ arises from a non-degenerate sesquilinear form (up to a scalar). Such a duality is a polarity if it is *reflexive*, i.e., $f(v,w) = 0$ implies $f(w,v) = 0$. Now a sesquilinear form f is *hermitian* if $f(v,w) = f(w,v)^\alpha$ holds where α is the companion automorphism of f and α has order 2. But if α is trivial then f is *symmetric*. If f inturn satisfies $f(v,v) = 0$ (for all v) then f is *alternating*. It is a well-known theorem of polar geometry that a non-degenerate reflexive sesquilinear form is either alternating, symmetric, or similar to an hermitian form. The associated polarity is called *symplectic*, *orthogonal*, and *unitary* respectively (though there are some other conventions for the characteristic 2 case).

3.1.1 Example

Let M be an invertible 4-dimensional square matrix over F and consider the following map on pairs of elements of the 4-dimensional vector space V over F:

$$f(v,w) = vMw^T.$$

Then f is a sesquilinear form of V and M is the *Gram matrix* of f.

3.2 Quadratic forms

We have seen that a polar space can arise from a reflexive sesquilinear form, but there are other polar spaces which do not arise this way, but instead have an associated quadratic form. A map Q from a vector space V to a field F is a *quadratic form* if it satisfies $Q(\lambda v) = \lambda^2 Q(v)$ for all $v \in V$ and $\lambda \in F$. We say that a subspace W is *totally singular* if the restriction of Q to W is trivial. Note that a subspace is totally isotropic (with respect to the associated polarity) if it is totally singular, but the converse is not always true.

3.2.1 Example

Let M be an invertible 4-dimensional square matrix over F and consider the following map on elements of the 4-dimensional vector space V over F:

$$f(v) = vMv^T.$$

Then f is a quadratic form of V and M is the *Gram matrix* of f.

Given a quadratic form Q, there is an associated sesquilinear form f (which may not be reflexive) defined as follows

$$f(v,w) = Q(v+w) - Q(v) - Q(w).$$

For characteristic not 2, the quadratic form and its associated sesquilinear form f determine one another, as $2Q(v) = f(v,v)$ (for all v).

3.3 Morphisms of forms

An *isometry* from a formed space (V, f) to a formed space (W, f') is a bijection ϕ such that for all v, w in V we have

$$f(v,w) = f'(\phi(v), \phi(w)).$$

The weaker notions of *similarity* and *semi-similarity* are also important in polar geometry. If there exists a scalar λ such that for all v, w in V we have

$$f(v,w) = \lambda f'(\phi(v), \phi(w))$$

then we say that ϕ is a similarity. If we also have a fixed field automorphism α such that

$$f(v,w) = \lambda f'(\phi(v), \phi(w))^{\alpha},$$

then ϕ is a semi-similarity. Naturally, we say that the formed spaces (V, f) and (W, f') are *isometric* (resp. *similar*) if there exists an isometry (resp. similarity) between them. Every non-degenerate reflexive sesquilinear form is alternating, symmetric, or similar to an hermitian form. Thus, up to similarity, the non-degenerate polar spaces come in five flavours: symplectic, unitary, orthogonal-elliptic, orthogonal-hyperbolic, and orthogonal-parabolic. In the case of the orthogonal spaces, they are distinguished by their Witt Index (the common dimension of their maximal totally singular/isotropic subspaces).

3.4 An important convention

In Forms, we have stipulated a convention on the creation of forms so as to cause as little confusion as possible. The hermitian forms will simply be those with the Frobenius Automorphism is the companion automorphism. We should also caution the user on what information is ”enough” to specify a form as problems can arise in even characteristic.

3.4.1 Example

Let F be a finite field of square order and let M be the following 4×4 matrix over F:

$$\begin{array}{|cccc|}\hline 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \hline\end{array}$$

Let α be the unique automorphism of F of order 2. Then the form

$$(u,v) := uM(v^T)^\alpha$$

defines a non-degenerate hermitian sesquilinear form. If F has odd characteristic, then the form

$$(u,v) := uMv^T$$

defines a non-degenerate orthogonal form, but if F has even characteristic, then this form is both:

1. a symplectic bilinear form, and
2. the associated bilinear form arising from a quadratic form.

In the latter, case we see that the bilinear form does not *define* the quadratic form, but rather that the quadratic form is necessary in order to define the polar geometry.

3.5 Canonical forms

Every nondegenerate polar space has a direct decomposition into a sum

$$L_1 \perp L_2 \perp \cdots L_n \perp U$$

where each of the L_i are hyperbolic lines and U is an anisotropic subspace of dimension at most 2. Thus if the given polar space is defined by a sesquilinear form f, then there is an isometric polar space defined by a Gram Matrix of the form

$$\begin{array}{|c|cc|cc|cc|cc|}\hline \mathrm{U} & & & & & & & & \\ \hline & 0 & 1 & & & & & & \\ & \varepsilon & 0 & & & & & & \\ \hline & & & 0 & 1 & & & & \\ & & & \varepsilon & 0 & & & & \\ \hline & & & & & * & & & \\ & & & & & & * & & \\ \hline & & & & & & & 0 & 1 \\ & & & & & & & \varepsilon & 0 \\ \hline\end{array}$$

were the top left hand corner represents the anisotropic part, and there are zeros everywhere else. The value of ε is -1 if the form is alternating, otherwise it is 1.

Chapter 4

Functionality

4.1 Functions for creating forms

4.1.1 BilinearFormByMatrix

◊ `BilinearFormByMatrix( m, f )` (operation)

Returns: a bilinear form

The argument `m` must be a square matrix over the finite field `f`. The field must be specified, as only the characteristic of the underlying field can be determined by the entries of `m`. The output will be a record rec(matrix, basefield, type).

Example
```
gap> gf := GF(3^2);
GF(3^2)
gap> mat := IdentityMat(4, gf);
[ [ Z(3)^0, 0*Z(3), 0*Z(3), 0*Z(3) ], [ 0*Z(3), Z(3)^0, 0*Z(3), 0*Z(3) ],
  [ 0*Z(3), 0*Z(3), Z(3)^0, 0*Z(3) ], [ 0*Z(3), 0*Z(3), 0*Z(3), Z(3)^0 ] ]
gap> f := BilinearFormByMatrix( mat, gf );
< bilinear form >
gap> Display(f);
Bilinear form
Gram Matrix:
 1 . . .
 . 1 . .
 . . 1 .
 . . . 1
```

4.1.2 QuadraticFormByMatrix

◊ `QuadraticFormByMatrix( m, f )` (operation)

Returns: a quadratic form

The argument `m` must be a square matrix over the finite field `f`. The field must be specified, as only the characteristic of the underlying field can be determined by the entries of `m`. The output will be a record rec(matrix, basefield, type).

Example
```
gap> gf := GF(2^2);
GF(2^2)
gap> mat := InvariantQuadraticForm( SO(-1, 4, 4) )!.matrix;
```

```
[ [ 0*Z(2), Z(2)^0, 0*Z(2), 0*Z(2) ], [ 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2) ],
  [ 0*Z(2), 0*Z(2), Z(2^2)^2, Z(2)^0 ], [ 0*Z(2), 0*Z(2), 0*Z(2), Z(2^2)^2 ] ]
gap> f := QuadraticFormByMatrix( mat, gf );
< quadratic form >
gap> Display(f);
Quadratic form
Gram Matrix:
z = Z(4)
   .   1   .   .
   .   .   .   .
   .   . z^2   1
   .   .   . z^2
```

4.1.3 HermitianFormByMatrix

◊ `HermitianFormByMatrix( m, f )` (operation)

Returns: a quadratic form

The argument `m` must be a square matrix over the finite field `f` of square order. The field must be specified, as only the characteristic of the underlying field can be determined by the entries of `m`. The output will be a record rec(matrix, basefield, type).

Example

```
gap> gf := GF(3^2);
gap> mat := IdentityMat(4, gf);
gap> f := HermitianFormByMatrix( mat, gf );
< hermitian form >
gap> Display(f);
Hermitian form
Gram Matrix:
 1 . . .
 . 1 . .
 . . 1 .
 . . . 1
Witt Index: 2
```

4.1.4 BilinearFormByPolynomial

◊ `BilinearFormByPolynomial( poly, r, n )` (operation)

◊ `BilinearFormByPolynomial( poly, r )` (operation)

Returns: a bilinear form

The argument `poly` must be a polynomial in the polynomial ring `r`. The (optional) last argument is the dimension for the underlying vector space of the resulting form, which by default is the number of indeterminates specified by `poly`.

Example

```
gap> r := PolynomialRing( GF(11), 4);
GF(11)[x_1,x_2,x_3,x_4]
gap> vars := IndeterminatesOfPolynomialRing( r );
[ x_1, x_2, x_3, x_4 ]
gap> pol := vars[1]*vars[2]+vars[3]*vars[4];
x_1*x_2+x_3*x_4
gap> form := BilinearFormByPolynomial(pol, r, 4);
```

```
< bilinear form >
gap> Display(form);
Bilinear form
Gram Matrix:
 . 6 . .
 6 . . .
 . . . 6
 . . 6 .
Polynomial: x_1*x_2+x_3*x_4
gap> ## Projective Points...
gap> projpoints := List(Subspaces( GF(11)^4, 1 ), Representative);;
gap> ## Number of totally isotropic points
gap> Number(projpoints, t -> IsZero( [t, t]^form ));
144
```

4.1.5 QuadraticFormByPolynomial

◊ `QuadraticFormByPolynomial( poly, r, n )` (operation)
◊ `QuadraticFormByPolynomial( poly, r )` (operation)

Returns: a quadratic form

The argument `poly` must be a polynomial in the polynomial ring `r`. The (optional) last argument is the dimension for the underlying vector space of the resulting form, which by default is the number of indeterminates specified by `poly`.

Example
```
gap> r := PolynomialRing( GF(8), 3);
GF(2^3)[x_1,x_2,x_3]
gap> poly := r.1^2 + r.2^2 + r.3^2;
x_1^2+x_2^2+x_3^2
gap> form := QuadraticFormByPolynomial(poly, r);
< quadratic form >
gap> RadicalOfForm(form);
<vector space over GF(2^3), with 3 generators>
```

4.1.6 HermitianFormByPolynomial

◊ `HermitianFormByPolynomial( poly, r, n )` (operation)
◊ `HermitianFormByPolynomial( poly, r )` (operation)

Returns: an hermitian form

The argument `poly` must be a polynomial in the polynomial ring `r` (defined over a finite field of square order). The (optional) last argument is the dimension for the underlying vector space of the resulting form, which by default is the number of indeterminates specified by `poly`.

Example
```
gap> r := PolynomialRing( GF(9), 4);
GF(3^2)[x_1,x_2,x_3,x_4]
gap> vars := IndeterminatesOfPolynomialRing( r );
[ x_1, x_2, x_3, x_4 ]
gap> poly := vars[1]*vars[2]^3+vars[1]^3*vars[2]+vars[3]*vars[4]^3+vars[3]^3*vars[4];
x_1^3*x_2+x_1*x_2^3+x_3^3*x_4+x_3*x_4^3
gap> form := HermitianFormByPolynomial(poly,r);
< hermitian form >
```

```
gap> Display(form);
Hermitian form
Gram Matrix:
 . 1 . .
 1 . . .
 . . . 1
 . . 1 .
Polynomial: x_1^3*x_2+x_1*x_2^3+x_3^3*x_4+x_3*x_4^3
```

4.2 Attributes and properties of forms

4.2.1 IsReflexiveForm

◊ `IsReflexiveForm( f )` (attribute)

Returns: true or false.

A sesquilinear form f is *reflexive* if whenever we have $f(u,v) = 0$, then we also have $f(v,u) = 0$. This operation simply returns `true` or `false`.

4.2.2 IsAlternatingForm

◊ `IsAlternatingForm( f )` (attribute)

Returns: true or false.

A bilinear form f is *alternating* if $f(v,v) = 0$ for all v. This operation simply returns `true` or `false`.

4.2.3 IsSymmetricForm

◊ `IsSymmetricForm( f )` (attribute)

Returns: true or false.

A bilinear form f is *symmetric* if $f(u,v) = f(v,u)$ for all pairs of vectors u and v. This operation simply returns `true` or `false`.

4.2.4 IsDegenerateForm

◊ `IsDegenerateForm( f )` (attribute)

Returns: true or false.

A sesquilinear form f is *degenerate* if there exists a nonzero vector v which is orthogonal to every other vector. That is, $f(v,w) = 0$ for all w. Likewise, a quadratic form Q is degenerate if there is a nonzero vector v such that $Q(v) = 0$. This operation simply returns `true` or `false`.

4.2.5 BaseField

◊ `BaseField( f )` (attribute)

Returns: the underlying field of `f`.

4.2.6 GramMatrix

◊ `GramMatrix( f )` (attribute)

Returns: the associated Gram matrix of `f`.

4.2.7 WittIndex

◊ `WittIndex( f )` (attribute)

Returns: the Witt index of `f`.

The Witt index is the maximum dimension of a totally singular subspace. So for example, if `f` is a symplectic form and d is the dimension of its underlying vector space, then the Witt index of `f` is $d/2$.

4.2.8 RadicalOfForm

◊ `RadicalOfForm( f )` (attribute)

Returns: a subspace, the radical, of the vectors space associated with `f`.

Example
```
gap> r := PolynomialRing( GF(8), 3 );
GF(2^3)[x_1,x_2,x_3]
gap> poly := r.1^2 + r.2 * r.3;
x_1^2+x_2*x_3
gap> form := QuadraticFormByPolynomial( poly, r );
< quadratic form >
gap> RadicalOfForm( form );
<vector space of dimension 1 over GF(2^3)>
```

4.2.9 PolynomialOfForm

◊ `PolynomialOfForm( f )` (attribute)

Returns: the polynomial associated with `f`.

Example
```
gap> mat := [ [ Z(8) , 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2) ],
 [ 0*Z(2), Z(2)^0, Z(2^3)^5, 0*Z(2), 0*Z(2) ],
 [ 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2) ],
 [ 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), Z(2)^0 ],
 [ 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2) ] ];;
gap> form := QuadraticFormByMatrix(mat,GF(8));
< quadratic form >
gap> PolynomialOfForm(form);
Z(2^3)*x_1^2+x_2^2+Z(2^3)^5*x_2*x_3+x_4*x_5
```

4.2.10 DiscriminantOfForm

◊ `DiscriminantOfForm( f )` (attribute)

Returns: a string

Given a quadratic or sesquilinear form `f`, this operation returns a string: “square” or “nonsquare”. Discriminants can be used to delineate the isometry type of an orthogonal form in even (algebraic) dimension.

Example

```
gap> gram := InvariantQuadraticForm(GO(-1,4,5))!.matrix;;
gap> f := QuadraticFormByMatrix(gram, GF(5));
< quadratic form >
gap> DiscriminantOfForm(f);
"nonsquare"
```

4.3 Functions for changing forms

4.3.1 BaseChangeToCanonical

◊ `BaseChangeToCanonical( f )` (operation)

Returns: a base-transition matrix

The argument `f` is a sesquilinear or quadratic form. For every isometry class of forms, there is a canonical representative, which is in block diagonal form. If `M` is the Gram matrix of the form `f`, then `b * M * TransposedMat(b)` is the Gram matrix of the canonical representative.

Example

```
gap> f := GF(3);
GF(3)
gap> gram := [
[0,0,0,1,0,0],
[0,0,0,0,1,0],
[0,0,0,0,0,1],
[-1,0,0,0,0,0],
[0,-1,0,0,0,0],
[0,0,-1,0,0,0]] * One(f);;
gap> form := BilinearFormByMatrix( gram, f );
< bilinear form >
gap> b := BaseChangeToCanonical( form );;
gap> Display( b * gram * TransposedMat(b) );
 . 1 . . . .
 2 . . . . .
 . . . 1 . .
 . . 2 . . .
 . . . . . 1
 . . . . 2 .
```

4.3.2 IsometricCanonicalForm

◊ `IsometricCanonicalForm( f )` (attribute)

Returns: the canonical form isometric to the sesquilinear or quadratic form `f`.

For every isometry type of sesquilinear or quadratic form, there is a canonical one. In Forms, the canonical form of each class is that which preserves the natural hyperbolic line decomposition (see Secton 3.5).

Example

```
gap> mat := [ [ Z(8) , 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2) ],
[ 0*Z(2), Z(2)^0, Z(2^3)^5, 0*Z(2), 0*Z(2) ],
[ 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2) ],
[ 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), Z(2)^0 ],
[ 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2), 0*Z(2) ] ];;
```

```
gap> form := QuadraticFormByMatrix(mat,GF(8));
< quadratic form >
gap> iso := IsometricCanonicalForm(form);
< quadratic form >
gap> Display(form);
Quadratic form
Gram Matrix:
z = Z(8)
 z^1   .   .   .   .
   .   1 z^5   .   .
   .   .   .   .   .
   .   .   .   .   1
   .   .   .   .   .
Witt Index: 2
gap> Display(iso);
Quadratic form
Gram Matrix:
 1 . . . .
 . . 1 . .
 . . . . .
 . . . . 1
 . . . . .
Witt Index: 2
```

4.4 Operations on forms

4.4.1 BaseChangeHomomorphism

◊ `BaseChangeHomomorphism( b, gf )` (operation)

Returns: the inner automorphism of GL(d,q) associated to the base-transition `b`.

The argument `b` must be an invertible matrix over the finite field `gf`.

Example

```
gap> gl:=GL(3,3);
GL(3,3)
gap> go:=GO(3,3);
GO(0,3,3)
gap> gram:=InvariantBilinearForm(go)!.matrix;
[ [ 0*Z(3), Z(3)^0, 0*Z(3) ], [ Z(3)^0, 0*Z(3), 0*Z(3) ], [ 0*Z(3), 0*Z(3), Z(3) ] ]
gap> f:=FormByMatrix(gram,GF(3),"parabolic");
< bilinear form >
gap> b:=BaseChangeToCanonical(f);;
gap> hom := BaseChangeHomomorphism(b, GF(3));
^[ [ 0*Z(3), 0*Z(3), Z(3) ], [ Z(3), Z(3), Z(3)^0 ], [ Z(3), 0*Z(3), Z(3) ] ]
gap> newgo := Image(hom, go);
Group(
[ [ [ Z(3)^0, 0*Z(3), Z(3) ], [ Z(3)^0, Z(3), Z(3)^0 ], [ 0*Z(3), 0*Z(3), Z(3) ] ],
  [ [ Z(3), Z(3)^0, 0*Z(3) ], [ 0*Z(3), Z(3), 0*Z(3) ], [ Z(3)^0, Z(3)^0, Z(3) ] ] ])
gap> gens:=GeneratorsOfGroup(newgo);;
gap> canonical := b * gram * TransposedMat(b);
[ [ Z(3), 0*Z(3), 0*Z(3) ], [ 0*Z(3), 0*Z(3), Z(3)^0 ], [ 0*Z(3), Z(3)^0, 0*Z(3) ] ]
gap> ForAll(gens, y -> y * canonical * TransposedMat(y) = canonical);
```

```
true
```

4.4.2 EvaluateForm

◊ EvaluateForm(f, u, v) (operation)

◊ EvaluateForm(f, u) (operation)

Returns: a finite field element

The argument f is either a sesquilinear or quadratic form defined over a finite field $GF(q)$. The other argument is a pair of vectors or matrices, or a single vector or matrix, which represent the bases of given subspaces of $GF(q)^d$. There is also an overloading of the operation ^ which we show in the following example:

Example
```
gap> mat := [[Z(8),0,0,0],[0,0,Z(8)^4,0],[0,0,0,1],[0,0,0,0]]*Z(8)^0;;
gap> form := QuadraticFormByMatrix(mat,GF(8));
< quadratic form >
gap> u := [ Z(2^3)^4, Z(2^3)^4, Z(2)^0, Z(2^3)^3 ];
[ Z(2^3)^4, Z(2^3)^4, Z(2)^0, Z(2^3)^3 ]
gap> EvaluateForm( form, u );
Z(2^3)^6
gap> u^form;
Z(2^3)^6
```

Here is an example using sesquilinear forms...

Example
```
gap> gram := [[0,0,0,0,0,2],[0,0,0,0,2,0],[0,0,0,1,0,0],[0,0,1,0,0,0],
 [0,2,0,0,0,0],[2,0,0,0,0,0]]*Z(3)^0;;
gap> form := BilinearFormByMatrix(gram,GF(3));
< bilinear form >
gap> u := [ [ Z(3)^0, 0*Z(3), 0*Z(3), Z(3)^0, 0*Z(3), Z(3)^0 ],
  [ 0*Z(3), 0*Z(3), Z(3)^0, Z(3)^0, Z(3), 0*Z(3) ] ];;
gap> v := [ [ Z(3)^0, 0*Z(3), Z(3)^0, Z(3), 0*Z(3), Z(3) ],
  [ 0*Z(3), Z(3)^0, 0*Z(3), Z(3), Z(3), Z(3) ] ];;
gap> EvaluateForm( form, u, v);
[ [ Z(3)^0, Z(3)^0 ], [ 0*Z(3), 0*Z(3) ] ]
gap> [u,v]^form;
[ [ Z(3)^0, Z(3)^0 ], [ 0*Z(3), 0*Z(3) ] ]
```

References

[Asc00] M. Aschbacher. *Finite group theory*, volume 10 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2000. 9

[Cam00] Peter J. Cameron. *Projective and Polar Spaces*. Online notes, http://www.maths.qmul.ac.uk/~pjc/pps/, 2000. 9

[KL90] Peter Kleidman and Martin Liebeck. *The subgroup structure of the finite classical groups*, volume 129 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1990. 9

[Tay92] Donald E. Taylor. *The geometry of the classical groups*, volume 9 of *Sigma Series in Pure Mathematics*. Heldermann Verlag, Berlin, 1992. 9

Index