```
Magma V2.12-BETA-4   Fri Aug 24 2005 12:50:46 on fb07-2go [Seed = 17408]
Type ? for help.  Type <Ctrl>-D to quit.

> iload "warwick_interactive.mag";
Interactive-loading "warwick_interactive.mag"
> //
> //   Magma Workshop on Group Theory and Algebraic Geometry
> //   University of Warwick, Coventry, UK
> //   August 22--26, 2005
> //
> //   Twisted forms of linear algebraic groups: Relative root subgroups
> //   Sergei Haller
> //
> //   warwick_interactive.mag
> //
> //   ran on Magma V2.12-BETA-4
> //
> //
> import "/usr/local/lib/MAGMA/2.11/package/LieThry/GrpLieAuto.m" : extendPerm;
> //
> function extendFldAut( E, sigma )
>      F := BaseRing(E);
>      sigmaFE := hom< F->E | x:->sigma(x) >;
>      aut := hom< E->E | sigmaFE, [ E.i : i in [1..Rank(E)] ] >;
>      return iso< E->E | x:->aut(x), x:->aut(x) >;
> end function;
> //
> k := Rationals();
> //
> P<x> := PolynomialRing( k );
> K<i> := ext< k | x^2+1 >;
> //
> A,_,m := AutomorphismGroup(K);
> sigma := m(A.1); // complex conjugation
> //
> E<s,u,v,w,y,z> := RationalFunctionField( K, 6 );
> //
> G := GroupOfLieType( "A3", E );
> R := RootDatum(G); W := WeylGroup(G);
> //
> sigmaE := extendFldAut(BaseRing(G),sigma);
> fld := FieldAutomorphism( G, sigmaE );
> //
> DynkinDiagram(G);

A3    1 - 2 - 3
> //
> tau := Sym(3)!(1,3); // Dynkin diagram Symmetry
> perm := extendPerm(R,tau) * W.2;
> //
> Orbits(sub<Sym(#Roots(R))|perm>);
[
     GSet{ 6 },
     GSet{ 12 },
     GSet{ 1, 5 },
     GSet{ 2, 8 },
     GSet{ 3, 4 },
     GSet{ 7, 11 },
     GSet{ 9, 10 }
]
> //
> rts := Roots(R);
> X  := RootSpace(R);
> X0 := sub< X | rts[2], rts[1]-rts[3] >;
```

```
> Xbar, pi := quo<X|X0>;
> //
> pi(rts);
{@
     Xbar.1,
     0,
     2*Xbar.1,
     -Xbar.1,
     -2*Xbar.1
@}
> { j : j in [1..#rts] | pi(rts[j]) eq   Xbar.1 };
{ 1, 3, 4, 5 }
> { j : j in [1..#rts] | pi(rts[j]) eq 2*Xbar.1 };
{ 6 }
> //
> J_delta  := { 1, 3 };
> J_2delta := { 6 };
> //
> h := VectorSpace(K,3)![1,1,-1];
> coc := GraphAutomorphism(G, tau)
>      * InnerAutomorphism(G, elt<G|W.2> * elt<G|h>);
> //
> aut := fld*coc;
> // check that we have a cocycle
> forall{ r : r in [1..#Roots(R)] | x @ (aut^2) eq x     where x is elt<G|<r,i>> };
true
> //
> // ROOT ELEMENTS
> // 2delta
> x_2delta := func< t | x6 * x6@aut     where x6 is elt<G| <6, t> > >;
> x_2delta(v+i*w);
x6(2*i*w)
> x_2delta(v+i*w) eq x_2delta(v+i*w) @ aut;
true
> // delta
> u_delta := func< t | x1 * x1@aut * x3 * x3@aut
>            where x1 is elt<G| <1,t[1]> >
>            where x3 is elt<G| <3,t[2]> >>;
> t := [s+i*u, y+i*z];
> u_delta( t ); u_delta( t ) @ aut;
x1(s + i*u) x3(y + i*z) x4(y - i*z) x5(s - i*u)
x1(s + i*u) x3(y + i*z) x4(y - i*z) x5(s - i*u) x6(-s^2 - u^2 + y^2 + z^2)
> // now with the correction term
> x_delta := func< t | u_delta(t) * elt<G| <6, 1/2 * (-t[1]*t[1]@sigmaE + t[2]*t[2]@sigmaE ) >> >;
> x_delta( t );   x_delta( t ) @ aut;
x1(s + i*u) x3(y + i*z) x4(y - i*z) x5(s - i*u) x6(-1/2*s^2 - 1/2*u^2 + 1/2*y^2 + 1/2*z^2)
x1(s + i*u) x3(y + i*z) x4(y - i*z) x5(s - i*u) x6(-1/2*s^2 - 1/2*u^2 + 1/2*y^2 + 1/2*z^2)
> x_delta( t ) eq x_delta( t ) @ aut;
true
> x_delta( t ) * x_2delta( v+i*w );
x1(s + i*u) x3(y + i*z) x4(y - i*z) x5(s - i*u) x6(-1/2*s^2 - 1/2*u^2 + 2*i*w + 1/2*y^2 + 1/2*z^2)
> // end
>

Total time: 4.129 seconds, Total memory usage: 5.92MB
```