Content and output of StreitsCriterion.gap

Nicolas Müller, Richard Pink

November 17, 2017

This is a formatted version of the GAP code in the file StreitsCriterion.gap, which can be found at [1], together with its output. This worksheet belongs to [2].

```
# Date: November 17, 2017

# This function calculates a fixed point P of an automorphism T of P^1
# (given as a matrix), and the eigenvalue of T on the tangent space of P.
getZetaAndP:=function(F, T)
    local S, Si, diag, zeta, P;

    # We change coordinates, such that T acts on P^1 as T(x) = zeta*x
    S := Eigenvectors(F, T);
    if Length(S) < 2 then
        Print("The eigenvectors of T are not defined over F");
        return -1;
    fi;
    Si:=S^(-1);
    diag:=S*T*Si;

    # The matrix diag=[[a,0],[0,b]] yields
    # the Moebius transformation  az/b.
    zeta:=diag[1][1]/diag[2][2];

    # Which fixed point P of T did we map to 0 with coordinate
    # transformation? Since the coordinate transform is given by the
    # Moebius transformation induced by S, we have P=S^(-1)(0),
    # when we view S^(-1) as a Moebius transformation.
    if Si[2][2] = 0 then
        P := infinity;
    else
        P := Si[1][2]/Si[2][2];
    fi;

    return [zeta, P];
end;

# Let X be a hyperelliptic curve with affine equation y^2=f(x)
# for some separable polynomial f.
# INPUT:
#  - Gt is the lift of the reduced automorphism group G of X to SL_2(C).
#  - f is a separable polynomial such that y^2=f(x) is an affine equation
```

```
#    for X in the same coordinates as those given by Gt.
#  - F is the field in which the calculations take place.
# OUTPUT:
# This function returns the character that Sym^2 chi_X induces on
# the reduced automorphism group.
getSymmetricSquareCharacterOnReducedAutomorphismGroup:=function(Gt, f, F)
    local G, p, cgsG, values, zeta, P, temp, pT, genus, k, T, x;

    x := IndeterminateOfUnivariateRationalFunction(f);

    if IsOddInt(Degree(f)) then
        genus := (Degree(f)-1)/2;
    else
        genus := (Degree(f)-2)/2;
    fi;

    G := Gt/Group(-IdentityMat(2));
    p := NaturalHomomorphism(G); # The quotient map Gt -> G.

    # For each conjugacy class of G we take a representative in SL_2(C).
    cgsG := List(ConjugacyClasses(G),
            c->PreImagesRepresentative(p, Representative(c)));
    values := [];
    # For each conjugacy class of G we calculate the value of
    # the character induced on G by Sym^2 chi_X
    for T in cgsG do
        pT := Image(p, T);
        if Order(pT) = 1 then
            # The character Sym^2 chi_X has degree (genus+genus^2)/2
            Add(values, (genus+genus^2)/2);
        else
            temp := getZetaAndP(F,T);
            if not IsList(temp) then
                Print("Did␣not␣get␣a␣list!");
                return -1;
            fi;
            zeta := temp[1];
            P := temp[2];

            # We define k as in the Proposition 4.2
            if P = infinity then
                if IsOddInt(Degree(f)) then
                    k := 1;
                else
                    k := 0;
                fi;
            else
                if Value(f,[x],[P])=0 then
                    k := 1;
                else
                    k := 0;
                fi;
            fi;
            # Formulas from Proposition 4.2
```

```
            if Order(pT) = 2 then
                Add(values, (-1)^k*(1+(-1)^(genus+1)+2*genus)/4);
            else
                Add(values, zeta^(2-k)*(zeta^genus-1)*
                             (zeta^(genus+1)-1)/((zeta-1)*(zeta^2-1)));
            fi;
        fi;
    od;
    return ClassFunction(G, values);
end;


x   := X(Rationals,"x");;

# Define the polynomials
t4  := x*(x^4-1);;

p4  := x^4+2*E(4)*Sqrt(3)*x^2+1;;

r4  := x^12-33*x^8-33*x^4+1;;

s4  := x^8+14*x^4+1;;

r5  := x^20-228*x^15+494*x^10+228*x^5+1;;

s5  := x*(x^10+11*x^5-1);;

t5  := x^30+522*x^25-10005*x^20-10005*x^10-522*x^5+1;;

# We define the lifts of the reduced automorphism groups to SL_2(C).
# Define the lift of A4 to SL_2(C)

M1  := [[-E(4),0],[0,E(4)]];;
M2  := [[1,E(4)],[1,-E(4)]];;
M2  := M2 / RootsOfPolynomial(CF(4), x^2 - Determinant(M2))[1];;
A4l := Group(M1,M2);;

# Define the lift of S4 to SL_2(C)

M1 := [[E(4),0],[0,1]]/E(8);;
M2 := -[[1,-1],[1,1]];;
M2 := M2 / RootsOfPolynomial(CF(8), x^2 - Determinant(M2))[1];;
S4l := Group(M1,M2);;

# Define the lift of A5 to SL_2(C)

om := (-1+Sqrt(5))/2;;
M1 := [[E(5),0],[0,1]]/E(10);;
M2 := [[om,1],[1,-om]];;
x  := X(Rationals,"x");;
M2 := M2 / RootsOfPolynomial(CF(5), x^2 - Determinant(M2))[1];;
A5l := Group(M1,M2);;

# For each curve X, we calculate the character of the reduced
```

```
# automorphism group induced by the symmetric square of the character
# of the representation of Aut(X) on the holomorphic differentials and
# evaluate the condition given by Streit.

Print("X4:\n");
f4 := t4*p4;
symX4 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(A4l, f4,
                                                                CF(12));
Print("<chiSymX4,chiTriv>␣=␣");
Print(ScalarProduct(symX4, TrivialCharacter(UnderlyingGroup(symX4))));
# Output: 0 -> X4 has CM
Print("\n");
```

Output:

```
X4:
<chiSymX4,chiTriv> = 0
```

```
Print("X5:\n");
f5 := t4;
symX5 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(S4l, f5,
                                                                CF(24));
Print("<chiSymX5,chiTriv>␣=␣");
Print(ScalarProduct(symX5, TrivialCharacter(UnderlyingGroup(symX5))));
# Output: 0 -> X5 has CM
Print("\n");
```

Output:

```
X5:
<chiSymX5,chiTriv> = 0
```

```
Print("X6:\n");
f6 := s4;
symX6 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(S4l, f6,
                                                                CF(24));
Print("<chiSymX6,chiTriv>␣=␣");
Print(ScalarProduct(symX6, TrivialCharacter(UnderlyingGroup(symX6))));
# Output: 1
Print("\n");
```

Output:

```
X6:
<chiSymX6,chiTriv> = 1
```

```
Print("X7:\n");
f7 := r4;
symX7 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(S4l, f7,
                                                                CF(24));
Print("<chiSymX7,chiTriv>␣=␣");
Print(ScalarProduct(symX7, TrivialCharacter(UnderlyingGroup(symX7))));
# Output: 0 -> X7 has CM
Print("\n");
```

Output:

```
X7:
<chiSymX7,chiTriv> = 0
```

```
Print("X8:\n");
f8 := s4*t4;
symX8 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(S4l, f8,
                                                                CF(24));
Print("<chiSymX8,chiTriv>␣=␣");
Print(ScalarProduct(symX8, TrivialCharacter(UnderlyingGroup(symX8))));
# Output: 1
Print("\n");
```

Output:

```
X8:
<chiSymX8,chiTriv> = 1
```

```
Print("X9:\n");
f9 := r4*t4;
symX9 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(S4l, f9,
                                                                CF(24));
Print("<chiSymX9,chiTriv>␣=␣");
Print(ScalarProduct(symX9, TrivialCharacter(UnderlyingGroup(symX9))));
# Output: 0 -> X9 has CM
Print("\n");
```

Output:

```
X9:
<chiSymX9,chiTriv> = 0
```

```
Print("X10:\n");
f10 := r4*s4;
symX10 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(S4l, f10,
                                                                 CF(24));
Print("<chiSymX10,chiTriv>␣=␣");
Print(ScalarProduct(symX10, TrivialCharacter(UnderlyingGroup(symX10))));
# Output: 1
Print("\n");
```

Output:

```
X10:
<chiSymX10,chiTriv> = 1
```

```
Print("X11:\n");
f11 := r4*s4*t4;
symX11 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(S4l, f11,
                                                                 CF(24));
Print("<chiSymX11,chiTriv>␣=␣");
Print(ScalarProduct(symX11, TrivialCharacter(UnderlyingGroup(symX11))));
# Output: 1
Print("\n");
```

Output:

```
X11:
<chiSymX11,chiTriv> = 1
```

```
Print("X12:\n");
f12 := s5;
symX12 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(A5l, f12,
                                                                CF(60));
Print("<chiSymX12,chiTriv>␣=␣");
Print(ScalarProduct(symX12, TrivialCharacter(UnderlyingGroup(symX12))));
# Output: 1
Print("\n");
```

Output:

```
X12:
<chiSymX12,chiTriv> = 1
```

```
Print("X13:\n");
f13 := r5;
symX13 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(A5l, f13,
                                                                CF(60));
Print("<chiSymX13,chiTriv>␣=␣");
Print(ScalarProduct(symX13, TrivialCharacter(UnderlyingGroup(symX13))));
# Output: 2
Print("\n");
```

Output:

```
X13:
<chiSymX13,chiTriv> = 2
```

```
Print("X14:\n");
f14 := t5;
symX14 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(A5l, f14,
                                                                CF(60));
Print("<chiSymX14,chiTriv>␣=␣");
Print(ScalarProduct(symX14, TrivialCharacter(UnderlyingGroup(symX14))));
# Output: 0 -> X14 has CM
Print("\n");
```

Output:

```
X14:
<chiSymX14,chiTriv> = 0
```

```
Print("X15:\n");
f15 := s5*r5;
symX15 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(A5l, f15,
                                                                CF(60));
Print("<chiSymX15,chiTriv>␣=␣");
Print(ScalarProduct(symX15, TrivialCharacter(UnderlyingGroup(symX15))));
# Output: 4
Print("\n");
```

Output:

```
X15:
<chiSymX15,chiTriv> = 4
```

```
Print("X16:\n");
f16 := s5*t5;
symX16 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(A5l, f16,
                                                                 CF(60));
Print("<chiSymX16,chiTriv>␣=␣");
Print(ScalarProduct(symX16, TrivialCharacter(UnderlyingGroup(symX16))));
# Output: 1
Print("\n");
```

Output:

```
X16:
<chiSymX16,chiTriv> = 1
```

```
Print("X17:\n");
f17 := r5*t5;
symX17 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(A5l, f17,
                                                                 CF(60));
Print("<chiSymX17,chiTriv>␣=␣");
Print(ScalarProduct(symX17, TrivialCharacter(UnderlyingGroup(symX17))));
# Output: 2
Print("\n");
```

Output:

```
X17:
<chiSymX17,chiTriv> = 2
```

```
Print("X18:\n");
f18 := s5*r5*t5;
symX18 := getSymmetricSquareCharacterOnReducedAutomorphismGroup(A5l, f18,
                                                                 CF(60));
Print("<chiSymX18,chiTriv>␣=␣");
Print(ScalarProduct(symX18, TrivialCharacter(UnderlyingGroup(symX18))));
# Output: 4
Print("\n");
```

Output:

```
X18:
<chiSymX18,chiTriv> = 4
```

References

[1] Worksheets for this paper: URL: `https://people.math.ethz.ch/~pink/ftp/MuellerPink2017/`

[2] Müller, N., Pink, R.: *Hyperelliptic Curves with Many Automorphisms*. Preprint 2017.