$U_q(\mathfrak{sl}_2) \leftrightarrow F_q(SL_2)$ duality

理論背景

$Generator(U_q(\mathfrak{sl}_2^+)) = \{E, F, H\}$

$Generator(F_q(SL_2^+)) = \{a, b, c, d\}$

について，E, F, H の双対基底を構成したい．両者はE, F, H および$\xi := a - 1, b, c$の次数に関するフィルターが定まるので，双対基底をフィルターから定まる次数 deg で打ち切って，deg を上げることで双対基底の詳細度を上げていくことで具体的に双対基底が計算できるのではないか？と考えた．すなわち，$G := \{E, F, H\} = \{e_1, e_2, e_3\}, G^\vee := \{b, c, \xi\} = \{f^1, f^2, f^3\}$としてある係数$x^{(X)}_{d_1,\cdots,d_n}$があり，双対基底がつぎのようにかけるとする．

$$e_n^\vee = \sum_d x_d^{(n)} \prod_i f^{d_i}$$

ここで次の仮定(A)のもと，詳細度を上げることにする．

(A)
次数 deg で打ち切ったものが双対基底をなす，すなわち

$$e_n^\vee \approx \sum_{|d| \le deg} x_d^{(n)} \prod_i f^{d_i}$$

と近似して，任意の長さ 2 以上 deg 以下の添字配列D $(2 \le |D| \le deg)$　にたいして

$$\left\langle \prod_i e_{D_i}, \sum_{|d| \le deg} x_d^{(n)} \prod_i f^{d_i} \right\rangle = 0$$

となる．

$deg = 1$のとき$E^\vee \approx b, F^\vee \approx c, H^\vee \approx \xi$という近似できる．

実際，$X, Y \in G$について$\langle X, Y^\vee \rangle = \delta_{X,Y}$が成立する．

$deg = 2, 3, \ldots$と段階的に精度を上げることを考える．

まず古典的な場合$q = 1$で考える．

$q = 1 \wedge deg = 2$の考察

結論としてある仮定のもとで，$q = 1 \wedge deg = 2$の場合の双対基底として有り得る解を一意に絞ることに成功した．

具体的には次の 2 次の近似を得た．

$E^\vee \approx b + b\xi, F^\vee \approx c - c\xi, H^\vee \approx \xi - bc - \frac{1}{2}\xi^2$

2 次の係数のつくる対称行列$(x_{ij}^{(n)})$の 3 次元配列を以後計算する.

生成元の定義

```
[~,E,F,H]=Usl2Full.getGenerator
```

```
E = 
    coeff    base
    _____    ____
      1       E  
F = 
    coeff    base
    _____    ____
      1       F  
H = 
    coeff    base
    _____    ____
      1       H  
```

```
[~,M,L]=FSL2Full.getGenerator
```

```
M = 
    coeff    base
    _____    ____
      1      m11 
    coeff    base
    _____    ____
      1      m21 
    coeff    base
    _____    ____
      1      m12 
    coeff    base
    _____    ____
      1      m22 
L = 
    coeff    base
    _____    ____
      1       d1 
    coeff    base
    _____    ____
      1       d2 
```

```
a=M(1,1);
b=M(1,2);
c=M(2,1);
d=M(2,2);
xi=M(1,1)-1;
```

検算

```
HP(M(1,2)*M(2,2),E(1))
```

ans = 1

```
HP(M(1,2)*M(2,2),H(1)^1*E(1))
```

```
ans = 0
```

```
HP(M(1,2)*M(2,2),H(1)^2*E(1))
```

```
ans = 0
```

```
HP(M(1,2)*M(2,2),H(1))
```

```
ans = 0
```

symbolic 方程式による解法

```
X = sym('X', [3 3]);
eqn1 = X(1,1) == X(1,2)^2;
eqn2 = X(2,2) + X(3,3) == 0;
eqns = [eqn1, eqn2];
S = solve(eqns, X, 'ReturnConditions', true);
disp(S);
```

```
          X1_1: z2^2
          X2_1: z1
          X3_1: z
          X1_2: z2
          X2_2: -z6
          X3_2: z3
          X1_3: z4
          X2_3: z5
          X3_3: z6
    parameters: [z    z1    z2    z3    z4    z5    z6]
    conditions: symtrue
```

symbolic の solve を使えば解の探索がラクに行うことができるだろう

dual basis の探索

$a-1,b,c$ で位相的に生成される代数においてフィルターの次数を上げていくことで, E,F,H の双対基底を探索する.

係数の決定に symbolic equation を用いる.

変数の宣言

X:2 次の係数

```
X = sym('X', [3 3 3]);
cond=fold(@and,reshape(permute(X, [1, 3, 2])==X,1,[]))
```

cond = $X_{3,3,2} = X_{3,2,3} \land X_{3,2,3} = X_{3,3,2} \land X_{3,3,1} = X_{3,1,3} \land X_{3,2,2} = X_{3,2,2} \land X_{3,1,3} = X_{3,3,1} \land X_{3,2,1} = X_{3,1,2} \land X_{3,1,2} = X_{3,2}$

```
G2=[M(1,2),M(2,1),M(1,1)-1];
G1=G2;
for ii=1:3
    G2(ii)=G2(ii)+sum(sum( ...
        arrayfun(@(x,y,z)x*y*z, ...
```

```
        reshape(X(ii,:,:),3,3,1),repmat(G1,3,1),repmat(G1,3,1).') ...
        ),2);
end
```

方程式の立式

仮定(A)のもと，条件式を立式する．

```
G3=[E,F,H];
G4=[E*E,E*F,E*H,F*F,F*H,H*H];
cond1=symtrue(3,9);
for ii=1:3
   for jj=1:3
       cond1(ii,jj)=HP(G2(ii),G3(jj))==(ii==jj)+0;
   end
   for jj=1:6
       cond1(ii,3+jj)=HP(G2(ii),G4(jj))==0;
   end
end
```

方程式を解く

```
S = solve([cond;cond1(:)], X, 'ReturnConditions', true);
disp(S);
```

```
        X1_1_1: 0
        X2_1_1: 0
        X3_1_1: 0
        X1_2_1: 0
        X2_2_1: 0
        X3_2_1: -1/2
        X1_3_1: 1/2
        X2_3_1: 0
        X3_3_1: 0
        X1_1_2: 0
        X2_1_2: 0
        X3_1_2: -1/2
        X1_2_2: 0
        X2_2_2: 0
        X3_2_2: 0
        X1_3_2: 0
        X2_3_2: -1/2
        X3_3_2: 0
        X1_1_3: 1/2
        X2_1_3: 0
        X3_1_3: 0
        X1_2_3: 0
        X2_2_3: -1/2
        X3_2_3: 0
        X1_3_3: 0
        X2_3_3: 0
        X3_3_3: -1/2
    parameters: [1×0 sym]
    conditions: symtrue
```

解が唯一存在し，それが出力された．結果として次を得た．

$E^\vee \approx b + b\xi, F^\vee \approx c - c\xi, H^\vee \approx \xi - bc - \frac{1}{2}\xi^2$

$q = 1 \wedge deg = 3$の考察

上記と同様にして，やはり一意的な解を得た.

$E^\vee \approx b + b\xi - b^2c,$
$F^\vee \approx c - c\xi + bc^2 + c\xi^2,$
$H^\vee \approx \xi - bc - \frac{1}{2}\xi^2 + \frac{1}{3}\xi^3$

```
%% 1) Known degree-2 solution
% From your PDF, the solve(...) output gave unique values for X(i,j,k).
% Below we embed them in a numeric array. (Double-check your indexing!)

Cdeg2 = zeros(3,3,3);

% Each line matches "X(i_j_k) = value" from your PDF's solution listing.
% For readability, we group them in the same order they appeared:

Cdeg2(1,1,1) = 0;
Cdeg2(2,1,1) = 0;
Cdeg2(3,1,1) = 0;
Cdeg2(1,2,1) = 0;
Cdeg2(2,2,1) = 0;
Cdeg2(3,2,1) = 0;
Cdeg2(1,3,1) = 0;
Cdeg2(2,3,1) = 0;
Cdeg2(3,3,1) = 0;

Cdeg2(1,1,2) = 0;
Cdeg2(2,1,2) = 0;
Cdeg2(3,1,2) = -1;
Cdeg2(1,2,2) = 0;
Cdeg2(2,2,2) = 0;
Cdeg2(3,2,2) = 0;
Cdeg2(1,3,2) = 0;
Cdeg2(2,3,2) = 0;
Cdeg2(3,3,2) = 0;

Cdeg2(1,1,3) = 1;
Cdeg2(2,1,3) = 0;
Cdeg2(3,1,3) = 0;
Cdeg2(1,2,3) = 0;
Cdeg2(2,2,3) = -1;
Cdeg2(3,2,3) = 0;
```

```
Cdeg2(1,3,3) = 0;
Cdeg2(2,3,3) = 0;
Cdeg2(3,3,3) = -1/2;

% If you need symmetry checks, apply them; but these were already solved
and are final.

%% 2) Declare NEW symbolic variables for degree-3 only
% We'll call them "Cdeg3" for clarity, with shape [3,3,3,3].
% There's no need to solve for degree-2 again, so let's keep Cdeg2_known
numeric.
Cdeg3=sym("Cdeg3",[3 3 3 3],"real");

% (Optional) If you want symmetry in the triple indices (j,k,l), impose it.
% The code below checks all permutations. You can skip it if not needed.
condSymCdeg3 = true;
for ii = 1:3
    for jj = 1:3
        for kk = 1:3
            for ll = 1:3
                if ~(jj <= kk && kk <= ll)
                    Cdeg3(ii,jj,kk,ll) = 0;
                end
            end
        end
    end
end

%% 3) Build your candidate dual basis Gdual(i) up to degree-3
% G1 = [m12, m21, m11-1] (the standard building blocks)
G1 = [M(1,2), M(2,1), M(1,1) - 1];

Gdual = G1;  % store the 3 dual-basis elements

for ii = 1:3
    % Start with the linear piece:
    gdTemp = G1(ii);

    % Add the KNOWN degree-2 solution:
    for jj = 1:3
        for kk = 1:3
            gdTemp = gdTemp + Cdeg2(ii,jj,kk)*G1(jj)*G1(kk);
        end
    end

    % Add the UNKNOWN degree-3 piece:
    for jj = 1:3
        for kk = 1:3
            for ll = 1:3
                gdTemp = gdTemp + Cdeg3(ii,jj,kk,ll)*G1(jj)*G1(kk)*G1(ll);
```

```
            end
        end
    end

    Gdual(ii) = gdTemp;
end
```

```
%% 4) Define monomials in E, F, H up to degree 3
%  - Gsl2Linear = [E, F, H]
%  - Gsl2Quad   = [E^2, E F, E H, F^2, F H, H^2]
%  - Gsl2Cubic  = [E^3, E^2F, E^2H, E F^2, E F H, E H^2, F^3, F^2H, F H^2,
H^3]

Gsl2Linear = [E, F, H];
Gsl2Quad   = [E^2, E*F, E*H, F^2, F*H, H^2];
Gsl2Cubic  = [ ...
    E^3,     E^2*F,   E^2*H, ...
    E*F^2,   E*F*H,   E*H^2, ...
    F^3,     F^2*H,   F*H^2, H^3 ...
];

%% 5) Impose dual-basis conditions up to degree 3
% The deg=2 portion is already satisfied by Cdeg2_known.
% But we still re-state the full conditions:
%   HP(Gdual(i), E/F/H) = delta(i,j)
%   HP(Gdual(i), any monomial of degree ≥ 2) = 0
% This enforces the "full" orthogonality conditions with the newly
introduced deg=3 terms.

cond1 = symtrue;
cond2 = symtrue;
cond3 = symtrue;

% (a) Linear: i-th dual basis vs j-th basis element => delta(i,j)
for ii = 1:3
    for jj = 1:3
        if ii == jj
            cond1(end+1)= (HP(Gdual(ii), Gsl2Linear(jj)) == 1);
        else
            cond1(end+1)= (HP(Gdual(ii), Gsl2Linear(jj)) == 0);
        end
        fprintf('process (i=%d, j=%d)\n', ii, jj);
    end
end
```

```
process (i=1, j=1)
process (i=1, j=2)
process (i=1, j=3)
process (i=2, j=1)
```

```
process (i=2, j=2)
process (i=2, j=3)
process (i=3, j=1)
process (i=3, j=2)
process (i=3, j=3)
```

```
% (b) Quadratic: must be 0
for ii = 1:3
    for jj = 1:numel(Gsl2Quad)
        cond2(end+1)=  (HP(Gdual(ii), Gsl2Quad(jj)) == 0);
        fprintf('process (i=%d, j=%d)\n', ii, jj);
    end
end
```

```
process (i=1, j=1)
process (i=1, j=2)
process (i=1, j=3)
process (i=1, j=4)
process (i=1, j=5)
process (i=1, j=6)
process (i=2, j=1)
process (i=2, j=2)
process (i=2, j=3)
process (i=2, j=4)
process (i=2, j=5)
process (i=2, j=6)
process (i=3, j=1)
process (i=3, j=2)
process (i=3, j=3)
process (i=3, j=4)
process (i=3, j=5)
process (i=3, j=6)
```

```
% (c) Cubic: must be 0
for ii = 1:3
    for jj = 1:numel(Gsl2Cubic)
        cond3(end+1)= (HP(Gdual(ii), Gsl2Cubic(jj)) == 0);
        fprintf('process (i=%d, j=%d)\n', ii, jj);
    end
end
```

```
process (i=1, j=1)
process (i=1, j=2)
process (i=1, j=3)
process (i=1, j=4)
process (i=1, j=5)
process (i=1, j=6)
process (i=1, j=7)
process (i=1, j=8)
process (i=1, j=9)
process (i=1, j=10)
process (i=2, j=1)
process (i=2, j=2)
process (i=2, j=3)
process (i=2, j=4)
process (i=2, j=5)
```

```
process (i=2, j=6)
process (i=2, j=7)
process (i=2, j=8)
process (i=2, j=9)
process (i=2, j=10)
process (i=3, j=1)
process (i=3, j=2)
process (i=3, j=3)
process (i=3, j=4)
process (i=3, j=5)
process (i=3, j=6)
process (i=3, j=7)
process (i=3, j=8)
process (i=3, j=9)
process (i=3, j=10)
```

```
%% Combine conditions & solve
% - condSymCdeg3: ensures symmetry in the triple indices if desired
% - condDualBasis: ensures dual-basis orthogonality up to deg=3
allConditions = [ cond1(:);cond2(:);cond3(:)];
Vars=Cdeg3(~isAlways(Cdeg3==0,"Unknown","false"));
% Solve for the unknown Cdeg3 only.
% (Cdeg2_known is numeric and not part of the solve.)
solution = solve(allConditions, Vars(:), 'ReturnConditions', true);

disp(solution);
```

```
    Cdeg31_1_1_1: 0
    Cdeg32_1_1_1: 0
    Cdeg33_1_1_1: 0
    Cdeg31_1_1_2: -1
    Cdeg32_1_1_2: 0
    Cdeg33_1_1_2: 0
    Cdeg31_1_2_2: 0
    Cdeg32_1_2_2: 1
    Cdeg33_1_2_2: 0
    Cdeg31_2_2_2: 0
    Cdeg32_2_2_2: 0
    Cdeg33_2_2_2: 0
    Cdeg31_1_1_3: 0
    Cdeg32_1_1_3: 0
    Cdeg33_1_1_3: 0
    Cdeg31_1_2_3: 0
    Cdeg32_1_2_3: 0
    Cdeg33_1_2_3: 0
    Cdeg31_2_2_3: 0
    Cdeg32_2_2_3: 0
    Cdeg33_2_2_3: 0
    Cdeg31_1_3_3: 0
    Cdeg32_1_3_3: 0
    Cdeg33_1_3_3: 0
    Cdeg31_2_3_3: 0
    Cdeg32_2_3_3: 1
    Cdeg33_2_3_3: 0
    Cdeg31_3_3_3: 0
    Cdeg32_3_3_3: 0
    Cdeg33_3_3_3: 1/3
      parameters: [1×0 sym]
      conditions: symtrue
```

上記の実行結果により， $deg=3$ の場合も一意的な解を得ることができた.

$$E^\vee \approx b+b\xi-b^2c,$$
$$F^\vee \approx c-c\xi+bc^2+c\xi^2,$$
$$H^\vee \approx \xi-bc-\frac{1}{2}\xi^2+\frac{1}{3}\xi^3$$

```
Ec=b+b*xi-b^2*c
```

```
Ec =
    coeff       base

    _____    ___________
     1       m11 m12
     -1      m12 m12 m21
```

```
Fc=c-c*xi+b*c^2+c*xi^2
```

```
Fc =
    coeff       base

    _____    ___________
     3       m21
     -3      m11 m21
     1       m11 m11 m21
     1       m12 m21 m21
```

```
Ec=xi-b*c-(1/2)*xi^2+(1/3)*xi^3
```

```
Ec =
    coeff       base

    _____    ___________
    -11/6    1
    3        m11
    -3/2     m11 m11
    -1       m12 m21
    1/3      m11 m11 m11
```

```
HP(E^2*H,Fc)
```

ans = 0