To complete the preprint titled "An affirmative answer to a conjecture for Metoki class by Kentaro Mikami at Akita University", we asked help of Groebner Basis Package of Maple, which is one of Symbol Calculus Softwares, in order to get bases of given cochain complexes and the cohomogoly groups.
In this note, we make use of Risa/Asir, which is another Symbol Calculus Software, and show the results we got by Maple and Risa/Asir are the same up to non-zero scalar multiples.
We remark that we added some line breaks so that we get better look.

Basis of $d_1\,(\mathbf{C}^6_{\mathrm{GF}}(\mathfrak{ham}^0_2, \mathfrak{sp}(2,\mathbb{R}))_{16}) \subset \mathbf{C}^7_{\mathrm{GF}}(\mathfrak{ham}^0_2, \mathfrak{sp}(2,\mathbb{R}))_{16}$**:**
Our source file for Risa/Asir is this:

```
/* On C^{6}_{wt=16} -> C^{7}_{wt=16}
output("exact_t1_wt16_6and7.txt")$
*/
YList = base_var_list(y, 1, 95) $
load("Mat_16_6and7_type1.rr")$
GList = [G1, G2, G3, G4, G5, G6, G7, G8, G9, G10, G11, G12, G13, G14, G15,
G16, G17, G18, G19, G20, G21, G22, G23, G24, G25, G26, G27, G28, G29, G30,
G31, G32, G33, G34, G35, G36, G37, G38, G39, G40, G41, G42, G43, G44, G45,
G46, G47, G48, G49, G50, G51, G52, G53, G54, G55, G56, G57, G58, G59, G60,
G61, G62, G63, G64, G65, G66, G67, G68, G69, G70, G71, G72, G73, G74, G75,
G76, G77, G78, G79, G80, G81, G82, G83, G84, G85, G86, G87, G88, G89, G90,
G91, G92, G93, G94, G95, G96, G97, G98, G99, G100, G101, G102, G103, G104,
G105, G106, G107, G108, G109, G110, G111, G112, G113, G114, G115, G116, G117,
G118, G119, G120, G121, G122, G123, G124, G125, G126, G127, G128, G129, G130,
G131, G132, G133, G134, G135, G136, G137, G138, G139, G140, G141, G142, G143,
G144, G145, G146, G147]$
ord( YList ) $          GB1 = nd_gr(GList, YList ,0,0) $
GB1 = reverse(GB1);     print(["GBe",GB1])$     output()$
end$
```

A part of the output of Groebner Basis is:

```
GBe=[-446227638468*y75+258371100400*y76-2677414594200*y77+2808720072600*y78
-483892450500*y79-838357655220*y80+871685530860*y81-1892343009627*y82+
2525687071848*y83+861370434243*y84+625187443152*y85+6093198421500*y86+
4546246681400*y87-2813196475270*y88+2152132471560*y89-15133158761840*y90+
9561265966665*y91+2198954966322*y92-9680559087150*y93-3770983597200*y94
-11367701561860*y95,
2231138192340*y74-3236925592800*y76+25986517801200*y77-19167959235840*y78
+7589254454700*y79+12083813535660*y80-9614968130520*y81+19627910409861*y82
-29219550942744*y83-2582960358429*y84-3902066256336*y85-48010669486950*y86
-34187627245100*y87+10980485213590*y88+6293554661580*y89+123735218701880*y90
-76475315472495*y91-27626805021798*y92+56696019952650*y93+53106035062800*y94
+101136816168220*y95,
68 more terms (sorry for skipping the rest and leaving heavy job to you)
]$
```

Kernel space of $d_1 : \mathbf{C}^7_{\mathrm{GF}}(\mathfrak{ham}^0_2, \mathfrak{sp}(2,\mathbb{R}))_{16} \to \mathbf{C}^8_{\mathrm{GF}}(\mathfrak{ham}^0_2, \mathfrak{sp}(2,\mathbb{R}))_{16}$**:**
Our source file for Risa/Asir is this:

```
/*
output("kerne_t1_wt16_7and8.txt")$
*/
WList = base_var_list( w, 1, 24) $
CList = base_var_list( c, 1, 95) $
YList = base_var_list( y, 1, 95) $
/* ###  On C^{7} -> C^{8} ### */
load("Mat_16_7and8_type1.rr")$
FList = [ F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15,
F16, F17, F18, F19, F20, F21, F22, F23, F24, F25, F26, F27, F28, F29, F30,
F31, F32, F33, F34, F35, F36, F37, F38, F39, F40, F41, F42, F43, F44, F45,
F46, F47, F48, F49, F50, F51, F52, F53, F54, F55, F56, F57, F58, F59, F60,
F61, F62, F63, F64, F65, F66, F67, F68, F69, F70, F71, F72, F73, F74, F75,
F76, F77, F78, F79, F80, F81, F82, F83, F84, F85, F86, F87, F88, F89, F90,
F91, F92, F93, F94, F95]$
/* ################################################################## */
NagasaW = length(WList)$    NagasaF = length(FList)$
for ( Uke = [], J=1; J <= NagasaW; J++ ) { MyA = WList[J-1];  Atai = 0;
    for (K=1 ; K <= NagasaF; K++ ){ MyB = FList[K-1];
    Atai += diff( MyB, MyA)* CList[K-1];}
    Uke = cons(Atai, Uke ); }
print("mark A: Got dual map TF of F")$  Uke = reverse( Uke );

print("mark B: Got Image(TR)")$     GBadj = nd_gr( Uke, CList,0, 0);

for (H=0, I=1; I <= NagasaF; I++){ H += CList[I-1]* YList[I-1]; }
Hnf = p_nf(H, GBadj, CList, 0)$
for( MyUkez = [], T=CList; T != []; T = cdr(T)){
    MyA = car(T); MyV = diff( Hnf, MyA);
    MyUkez = cons( MyV, MyUkez);}

print("mark C: Annihilators of Image(TF)")$
MyUkez = reverse(MyUkez);

print("mark D: Got Ker(F)")$    GBker = nd_gr( MyUkez, YList,0, 0);
output()$       end$
```

The outputs are the follows:

```
GBk=[368571103200*y1+1351427378400*y47-450475792800*y50-2083450541700*y51
+11274054788700*y77-12683683914000*y78+1590428838900*y79+7275101984700*y80
-10448587809000*y81+6410733790653*y82-13981567014072*y83-16098409761957*y84
-2603346695478*y85-30292840571400*y86-22358770705700*y87+10032702042550*y88
+883984609200*y89+5802437564276 0*y90-41010508144455*y91-15470397459450*y92
+40037017987050*y93+4390494333150*y94+55052825955540*y95,

22987779706584*y2-31015258334280*y57+65679370590240*y58-2333788839033300*y77
+2847547488345120*y78-708876435791700*y79-1175639802428520*y80
-517877711109000*y81-1971062567955153*y82+2643976210072392*y83
-367831348898103*y84+382136247747228*y85+3663823436752050*y86
+2800692629390700*y87-463769141240790*y88-888541532645580*y89
-9149932266616200*y90+7727929995111435*y91+2481073046817666*y92
-1146891126002550*y93-5041594682543400*y94-8618277247810580*y95,

69 terms more
]$
```

Basis of $\mathbf{H}^7_{\mathrm{GF}}(\mathfrak{ham}^0_2, \mathfrak{sp}(2,\mathbb{R}))_{16}$
The next is a source file for Risa/Asir. GBe and GBk are data gotten above.

```
/*
output("Betti_t1_wt16_7and7.txt")$
*/

load("exact_t1_wt16_6and7.txt")$
load("kerne_t1_wt16_7and8.txt")$
YList = base_var_list(y,1, 95) $
for(Uke=[], T= GBk; T != []; T = cdr(T)) {
    MyA = car(T); print(MyA); Atai = p_nf( MyA, GBe, YList , 0) ;
    Uke = cons(Atai, Uke);
    }$
Uke = reverse(Uke)$     ord( YList )$
GBb = nd_gr( Uke, YList , 0,0);
```

A basis of $\mathrm{H}^7_{\mathrm{GF}}(\mathfrak{ham}^0_2, \mathfrak{sp}(2,\mathbb{R}))_{16}$ is given

```
GBkmode = [-2027141067600*y76-6871115344500*y77+8293793595120*y78
-1593871052400*y79-3342315930030*y80-2188718191440*y81-6047944018587*y82
+7911486513648*y83-1366183084077*y84+1206881491512*y85+10895090886900*y86
+9572836551300*y87-1269138903120*y88-3867959161440*y89-28054435525860*y90
+23511502274085*y91+7468180349703*y92-2799062316375*y93-17517045194250*y94
-24368226519980*y95]
```