computation

November 8, 2024

[1]:
```
load("methods.sage")
```

We calculate $\pi_i((\mathrm{THR}(\mathbb{Z}/4)^{\phi\mathbb{Z}/2})^{C_2})$ (via calculating $L_i^{(n)}F_{\mathbb{Z}/4}(\mathbb{Z}/2)$) and $\pi_i(\mathrm{THR}(\mathbb{Z}/4)^{\phi\mathbb{Z}/2})$, and the $(r-f)_i$ map between them.

[2]:
```
for i in range(0, 7):
    print(f"Computation of (r-f)_{i}:")
    r_minus_f(i)
    print("")
```

```
Computation of (r-f)_0:
L_0^(0)F_{Z/4}(Z/2)
C4
pi_0((THR(Z/4)^{\phi Z/2})^{C_2}):
C4
Basis labels:
{(0, 0): f1}
Abelian group invariants corresponding to basis labels:
[4]
pi_0(THR(Z/4)^{\phi Z/2}):
C2^1
Basis labels:
{(0, 0, 0): f1}
Images of basis elements under f map:
[f1]
Images of basis elements under r map:
[f1]
ker(r-f):
C4
coker(r-f):
C2

Computation of (r-f)_1:
L_1^(0)F_{Z/4}(Z/2)
C2 x C2
L_1^(1)F_{Z/4}(Z/2)
C2
pi_1((THR(Z/4)^{\phi Z/2})^{C_2}):
C2 x C2 x C2 x C2
```

```
Basis labels:
{(0, 0): f1, (0, 1): f2, (1, 0): f3, (0, 1, 0): f4}
Abelian group invariants corresponding to basis labels:
[2, 2, 2, 2]
pi_1(THR(Z/4)^{\phi Z/2}):
C2^3
Basis labels:
{(0, 0, 1): f1, (0, 1, 0): f2, (1, 0, 0): f3}
Images of basis elements under f map:
[<identity> of …, f1, <identity> of …, f2*f3]
Images of basis elements under r map:
[f1, <identity> of …, f3, <identity> of …]
ker(r-f):
C2
coker(r-f):
1

Computation of (r-f)_2:
L_2^(0)F_{Z/4}(Z/2)
C2 x C2
L_2^(1)F_{Z/4}(Z/2)
C2 x C2
L_2^(2)F_{Z/4}(Z/2)
C2
pi_2((THR(Z/4)^{\phi Z/2})^{C_2}):
C2 x C2 x C2 x C2 x C2 x C2 x C2
Basis labels:
{(0, 0): f1, (0, 1): f2, (1, 0): f3, (1, 1): f4, (2, 0): f5, (0, 1, 1): f6, (0,
2, 0): f7}
Abelian group invariants corresponding to basis labels:
[2, 2, 2, 2, 2, 2, 2]
pi_2(THR(Z/4)^{\phi Z/2}):
C2^6
Basis labels:
{(0, 0, 2): f1, (0, 1, 1): f2, (0, 2, 0): f3, (1, 0, 1): f4, (1, 1, 0): f5, (2,
0, 0): f6}
Images of basis elements under f map:
[<identity> of …, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, f2*f4, f3*f6]
Images of basis elements under r map:
[f1, <identity> of …, f4, <identity> of …, f6, <identity> of …, <identity>
of …]
ker(r-f):
C2 x C2
coker(r-f):
C2

Computation of (r-f)_3:
```

```
L_3^(0)F_{Z/4}(Z/2)
C2 x C2 x C2
L_3^(1)F_{Z/4}(Z/2)
C2 x C2 x C2
L_3^(2)F_{Z/4}(Z/2)
C2 x C2
L_3^(3)F_{Z/4}(Z/2)
C2
pi_3((THR(Z/4)^{\phi Z/2})^{C_2}):
C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2
Basis labels:
{(0, 0): f1, (0, 1): f2, (0, 2): f3, (1, 0): f4, (1, 1): f5, (1, 2): f6, (2, 0):
f7, (2, 1): f8, (3, 0): f9, (0, 1, 2): f10, (0, 2, 1): f11, (0, 3, 0): f12, (1,
2, 0): f13}
Abelian group invariants corresponding to basis labels:
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
pi_3(THR(Z/4)^{\phi Z/2}):
C2^10
Basis labels:
{(0, 0, 3): f1, (0, 1, 2): f2, (0, 2, 1): f3, (0, 3, 0): f4, (1, 0, 2): f5, (1,
1, 1): f6, (1, 2, 0): f7, (2, 0, 1): f8, (2, 1, 0): f9, (3, 0, 0): f10}
Images of basis elements under f map:
[<identity> of …, f1, <identity> of …, <identity> of …, f6, <identity> of
…, <identity> of …, <identity> of …, <identity> of …, f2*f5, f3*f8,
f4*f10, f7*f9]
Images of basis elements under r map:
[f1, <identity> of …, <identity> of …, f5, <identity> of …, <identity> of
…, f8, <identity> of …, f10, <identity> of …, <identity> of …,
<identity> of …, <identity> of …]
ker(r-f):
C2 x C2 x C2 x C2
coker(r-f):
C2

Computation of (r-f)_4:
L_4^(0)F_{Z/4}(Z/2)
C4 x C2 x C2
L_4^(1)F_{Z/4}(Z/2)
C4 x C2 x C2
L_4^(2)F_{Z/4}(Z/2)
C4 x C2 x C2
L_4^(3)F_{Z/4}(Z/2)
C2 x C2
L_4^(4)F_{Z/4}(Z/2)
C2
pi_4((THR(Z/4)^{\phi Z/2})^{C_2}):
C4 x C4 x C4 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x
C2 x C2
```

```
Basis labels:
{(0, 0): f1, (0, 1): f2, (0, 2): f3, (1, 0): f5, (1, 1): f6, (1, 2): f7, (2, 0):
f9, (2, 1): f10, (2, 2): f11, (3, 0): f13, (3, 1): f14, (4, 0): f15, (0, 1, 3):
f16, (0, 2, 2): f17, (0, 3, 1): f18, (0, 4, 0): f19, (1, 2, 1): f20, (1, 3, 0):
f21}
Abelian group invariants corresponding to basis labels:
[2, 2, 4, 2, 2, 4, 2, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2]
pi_4(THR(Z/4)^{\phi Z/2}):
C2^15
Basis labels:
{(0, 0, 4): f1, (0, 1, 3): f2, (0, 2, 2): f3, (0, 3, 1): f4, (0, 4, 0): f5, (1,
0, 3): f6, (1, 1, 2): f7, (1, 2, 1): f8, (1, 3, 0): f9, (2, 0, 2): f10, (2, 1,
1): f11, (2, 2, 0): f12, (3, 0, 1): f13, (3, 1, 0): f14, (4, 0, 0): f15}
Images of basis elements under f map:
[<identity> of …, <identity> of …, f1, <identity> of …, <identity> of …,
f7, <identity> of …, <identity> of …, f12, <identity> of …, <identity> of
…, <identity> of …, f2*f6, f3*f10, f4*f13, f5*f15, f8*f11, f9*f14]
Images of basis elements under r map:
[f1, <identity> of …, f1, f6, <identity> of …, f6, f10, <identity> of …,
f10, f13, <identity> of …, f15, <identity> of …, <identity> of …,
<identity> of …, <identity> of …, <identity> of …, <identity> of …]
ker(r-f):
C4 x C2 x C2 x C2 x C2 x C2 x C2
coker(r-f):
C2 x C2

Computation of (r-f)_5:
L_5^(0)F_{Z/4}(Z/2)
C2 x C2 x C2 x C2
L_5^(1)F_{Z/4}(Z/2)
C2 x C2 x C2 x C2
L_5^(2)F_{Z/4}(Z/2)
C2 x C2 x C2 x C2
L_5^(3)F_{Z/4}(Z/2)
C2 x C2 x C2
L_5^(4)F_{Z/4}(Z/2)
C2 x C2
L_5^(5)F_{Z/4}(Z/2)
C2
pi_5((THR(Z/4)^{\phi Z/2})^{C_2}):
C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x
C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2
Basis labels:
{(0, 0): f1, (0, 1): f2, (0, 2): f3, (0, 3): f4, (1, 0): f5, (1, 1): f6, (1, 2):
f7, (1, 3): f8, (2, 0): f9, (2, 1): f10, (2, 2): f11, (2, 3): f12, (3, 0): f13,
(3, 1): f14, (3, 2): f15, (4, 0): f16, (4, 1): f17, (5, 0): f18, (0, 1, 4): f19,
(0, 2, 3): f20, (0, 3, 2): f21, (0, 4, 1): f22, (0, 5, 0): f23, (1, 2, 2): f24,
(1, 3, 1): f25, (1, 4, 0): f26, (2, 3, 0): f27}
```

```
Abelian group invariants corresponding to basis labels:
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2]
pi_5(THR(Z/4)^{\phi Z/2}):
C2^21
Basis labels:
{(0, 0, 5): f1, (0, 1, 4): f2, (0, 2, 3): f3, (0, 3, 2): f4, (0, 4, 1): f5, (0,
5, 0): f6, (1, 0, 4): f7, (1, 1, 3): f8, (1, 2, 2): f9, (1, 3, 1): f10, (1, 4,
0): f11, (2, 0, 3): f12, (2, 1, 2): f13, (2, 2, 1): f14, (2, 3, 0): f15, (3, 0,
2): f16, (3, 1, 1): f17, (3, 2, 0): f18, (4, 0, 1): f19, (4, 1, 0): f20, (5, 0,
0): f21}
Images of basis elements under f map:
[<identity> of …, f1, <identity> of …, <identity> of …, <identity> of …,
f8, <identity> of …, <identity> of …, <identity> of …, f14, <identity> of
…, <identity> of …, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, <identity> of …, <identity> of …, f2*f7, f3*f12, f4*f16,
f5*f19, f6*f21, f9*f13, f10*f17, f11*f20, f15*f18]
Images of basis elements under r map:
[f1, <identity> of …, <identity> of …, <identity> of …, f7, <identity> of
…, <identity> of …, <identity> of …, f12, <identity> of …, <identity> of
…, <identity> of …, f16, <identity> of …, <identity> of …, f19,
<identity> of …, f21, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, <identity> of …]
ker(r-f):
C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2
coker(r-f):
C2 x C2 x C2 x C2

Computation of (r-f)_6:
L_6^(0)F_{Z/4}(Z/2)
C2 x C2 x C2 x C2
L_6^(1)F_{Z/4}(Z/2)
C2 x C2 x C2 x C2
L_6^(2)F_{Z/4}(Z/2)
C2 x C2 x C2 x C2
L_6^(3)F_{Z/4}(Z/2)
C2 x C2 x C2 x C2
L_6^(4)F_{Z/4}(Z/2)
C2 x C2 x C2
L_6^(5)F_{Z/4}(Z/2)
C2 x C2
L_6^(6)F_{Z/4}(Z/2)
C2
pi_6((THR(Z/4)^{\phi Z/2})^{C_2}):
C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x
C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x
C2 x C2
```

```
Basis labels:
{(0, 0): f1, (0, 1): f2, (0, 2): f3, (0, 3): f4, (1, 0): f5, (1, 1): f6, (1, 2):
f7, (1, 3): f8, (2, 0): f9, (2, 1): f10, (2, 2): f11, (2, 3): f12, (3, 0): f13,
(3, 1): f14, (3, 2): f15, (3, 3): f16, (4, 0): f17, (4, 1): f18, (4, 2): f19,
(5, 0): f20, (5, 1): f21, (6, 0): f22, (0, 1, 5): f23, (0, 2, 4): f24, (0, 3,
3): f25, (0, 4, 2): f26, (0, 5, 1): f27, (0, 6, 0): f28, (1, 2, 3): f29, (1, 3,
2): f30, (1, 4, 1): f31, (1, 5, 0): f32, (2, 3, 1): f33, (2, 4, 0): f34}
Abelian group invariants corresponding to basis labels:
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2]
pi_6(THR(Z/4)^{\phi Z/2}):
C2^28
Basis labels:
{(0, 0, 6): f1, (0, 1, 5): f2, (0, 2, 4): f3, (0, 3, 3): f4, (0, 4, 2): f5, (0,
5, 1): f6, (0, 6, 0): f7, (1, 0, 5): f8, (1, 1, 4): f9, (1, 2, 3): f10, (1, 3,
2): f11, (1, 4, 1): f12, (1, 5, 0): f13, (2, 0, 4): f14, (2, 1, 3): f15, (2, 2,
2): f16, (2, 3, 1): f17, (2, 4, 0): f18, (3, 0, 3): f19, (3, 1, 2): f20, (3, 2,
1): f21, (3, 3, 0): f22, (4, 0, 2): f23, (4, 1, 1): f24, (4, 2, 0): f25, (5, 0,
1): f26, (5, 1, 0): f27, (6, 0, 0): f28}
Images of basis elements under f map:
[<identity> of …, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, <identity> of …, f2*f8, f3*f14, f4*f19, f5*f23, f6*f26,
f7*f28, f10*f15, f11*f20, f12*f24, f13*f27, f17*f21, f18*f25]
Images of basis elements under r map:
[f1, <identity> of …, <identity> of …, <identity> of …, f8, <identity> of
…, <identity> of …, <identity> of …, f14, <identity> of …, <identity> of
…, <identity> of …, f19, <identity> of …, <identity> of …, <identity> of
…, f23, <identity> of …, <identity> of …, f26, <identity> of …, f28,
<identity> of …, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, <identity> of …, <identity> of …, <identity> of …,
<identity> of …, <identity> of …, <identity> of …, <identity> of …]
ker(r-f):
C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2
coker(r-f):
C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2 x C2
```

Recall that we can restrict $r - f$ to a map from the n-indexed diagonal component of $(\mathrm{THR}(\mathbb{Z}/4)^{\phi\mathbb{Z}/2})^{C_2}$ to the sum of the n-indexed components of $\mathrm{THR}(\mathbb{Z}/4)^{\phi\mathbb{Z}/2}$, which we denote γ_n. Below we output the action of γ_n on homotopy groups, for those parameters relevant to resolving the extension problem in degree 1.

[3]:
```
for i in range(1, 3):
    for n in range(0, 2):
        print(f"Computation of pi_{i}(gamma_{n})")
```

```
        r_minus_f_fixed_n(i, n)
        print("")
```

```
Computation of pi_1(gamma_0)
L_1^(0)F_{Z/4}(Z/2)
C2 x C2
Basis labels:
{(0, 0): f1, (0, 1): f2}
Abelian group invariants corresponding to basis labels:
[2, 2]
pi_1((0, m)-indexed part of THR(Z/4)^{\phi Z/2}):
C2^2
Basis labels:
{(0, 0, 1): f1, (0, 1, 0): f2}
Images of basis elements under f map:
[<identity> of …, f1]
Images of basis elements under r map:
[f1, <identity> of …]
ker(pi_1(gamma_0)):
C2
coker(pi_1(gamma_0)):
C2

Computation of pi_1(gamma_1)
L_1^(1)F_{Z/4}(Z/2)
C2
Basis labels:
{(1, 0): f1}
Abelian group invariants corresponding to basis labels:
[2]
pi_1((1, m)-indexed part of THR(Z/4)^{\phi Z/2}):
C2^1
Basis labels:
{(1, 0, 0): f1}
Images of basis elements under f map:
[<identity> of …]
Images of basis elements under r map:
[f1]
ker(pi_1(gamma_1)):
1
coker(pi_1(gamma_1)):
1

Computation of pi_2(gamma_0)
L_2^(0)F_{Z/4}(Z/2)
C2 x C2
Basis labels:
{(0, 0): f1, (0, 1): f2}
```

```
Abelian group invariants corresponding to basis labels:
[2, 2]
pi_2((0, m)-indexed part of THR(Z/4)^{\phi Z/2}):
C2^3
Basis labels:
{(0, 0, 2): f1, (0, 1, 1): f2, (0, 2, 0): f3}
Images of basis elements under f map:
[<identity> of …, <identity> of …]
Images of basis elements under r map:
[f1, <identity> of …]
ker(pi_2(gamma_0)):
C2
coker(pi_2(gamma_0)):
C2 x C2

Computation of pi_2(gamma_1)
L_2^(1)F_{Z/4}(Z/2)
C2 x C2
Basis labels:
{(1, 0): f1, (1, 1): f2}
Abelian group invariants corresponding to basis labels:
[2, 2]
pi_2((1, m)-indexed part of THR(Z/4)^{\phi Z/2}):
C2^2
Basis labels:
{(1, 0, 1): f1, (1, 1, 0): f2}
Images of basis elements under f map:
[<identity> of …, <identity> of …]
Images of basis elements under r map:
[f1, <identity> of …]
ker(pi_2(gamma_1)):
C2
coker(pi_2(gamma_1)):
C2
```

We can use our program to double check the lemma claiming

$\ker(\partial_i)/(\mathrm{im}(\partial_{i+1}) + \ker(\partial_i) \cap \bigoplus_{B_i^\circ} \mathbb{Z}/4 + 2\ker(\partial_i)) \cong \mathbb{Z}/2,$

where ∂_i is the differential of the Moore complex of the levelwise free simplicial abelian group $\sigma M_\bullet$ with geometric realisation $\Sigma^n H\mathbb{Z}/2$.

[4]:
```
for i in range(0, 7):
    for n in range(0, i+1):
        print(f"Checking lemma for i={i}, n={n}")
        check_quotient_lemma(i, n)
        print("")
```

```
Checking lemma for i=0, n=0
ker(delta_0)/(im(delta_1) + ker(delta_0) \cap off_diag_part + 2ker(delta_0)):
C2
Confirmed that quotient agrees with projecting onto (id_[0], id_[0]) component

Checking lemma for i=1, n=0
ker(delta_1)/(im(delta_2) + ker(delta_1) \cap off_diag_part + 2ker(delta_1)):
C2
Confirmed that quotient agrees with projecting onto (id_[1], id_[1]) component

Checking lemma for i=1, n=1
ker(delta_1)/(im(delta_2) + ker(delta_1) \cap off_diag_part + 2ker(delta_1)):
C2
Confirmed that quotient agrees with projecting onto (id_[1], id_[1]) component

Checking lemma for i=2, n=0
ker(delta_2)/(im(delta_3) + ker(delta_2) \cap off_diag_part + 2ker(delta_2)):
C2
Confirmed that quotient agrees with projecting onto (id_[2], id_[2]) component

Checking lemma for i=2, n=1
ker(delta_2)/(im(delta_3) + ker(delta_2) \cap off_diag_part + 2ker(delta_2)):
C2
Confirmed that quotient agrees with projecting onto (id_[2], id_[2]) component

Checking lemma for i=2, n=2
ker(delta_2)/(im(delta_3) + ker(delta_2) \cap off_diag_part + 2ker(delta_2)):
C2
Confirmed that quotient agrees with projecting onto (id_[2], id_[2]) component

Checking lemma for i=3, n=0
ker(delta_3)/(im(delta_4) + ker(delta_3) \cap off_diag_part + 2ker(delta_3)):
C2
Confirmed that quotient agrees with projecting onto (id_[3], id_[3]) component

Checking lemma for i=3, n=1
ker(delta_3)/(im(delta_4) + ker(delta_3) \cap off_diag_part + 2ker(delta_3)):
C2
Confirmed that quotient agrees with projecting onto (id_[3], id_[3]) component

Checking lemma for i=3, n=2
ker(delta_3)/(im(delta_4) + ker(delta_3) \cap off_diag_part + 2ker(delta_3)):
C2
Confirmed that quotient agrees with projecting onto (id_[3], id_[3]) component

Checking lemma for i=3, n=3
ker(delta_3)/(im(delta_4) + ker(delta_3) \cap off_diag_part + 2ker(delta_3)):
C2
```

```
Confirmed that quotient agrees with projecting onto (id_[3], id_[3]) component

Checking lemma for i=4, n=0
ker(delta_4)/(im(delta_5) + ker(delta_4) \cap off_diag_part + 2ker(delta_4)):
C2
Confirmed that quotient agrees with projecting onto (id_[4], id_[4]) component

Checking lemma for i=4, n=1
ker(delta_4)/(im(delta_5) + ker(delta_4) \cap off_diag_part + 2ker(delta_4)):
C2
Confirmed that quotient agrees with projecting onto (id_[4], id_[4]) component

Checking lemma for i=4, n=2
ker(delta_4)/(im(delta_5) + ker(delta_4) \cap off_diag_part + 2ker(delta_4)):
C2
Confirmed that quotient agrees with projecting onto (id_[4], id_[4]) component

Checking lemma for i=4, n=3
ker(delta_4)/(im(delta_5) + ker(delta_4) \cap off_diag_part + 2ker(delta_4)):
C2
Confirmed that quotient agrees with projecting onto (id_[4], id_[4]) component

Checking lemma for i=4, n=4
ker(delta_4)/(im(delta_5) + ker(delta_4) \cap off_diag_part + 2ker(delta_4)):
C2
Confirmed that quotient agrees with projecting onto (id_[4], id_[4]) component

Checking lemma for i=5, n=0
ker(delta_5)/(im(delta_6) + ker(delta_5) \cap off_diag_part + 2ker(delta_5)):
C2
Confirmed that quotient agrees with projecting onto (id_[5], id_[5]) component

Checking lemma for i=5, n=1
ker(delta_5)/(im(delta_6) + ker(delta_5) \cap off_diag_part + 2ker(delta_5)):
C2
Confirmed that quotient agrees with projecting onto (id_[5], id_[5]) component

Checking lemma for i=5, n=2
ker(delta_5)/(im(delta_6) + ker(delta_5) \cap off_diag_part + 2ker(delta_5)):
C2
Confirmed that quotient agrees with projecting onto (id_[5], id_[5]) component

Checking lemma for i=5, n=3
ker(delta_5)/(im(delta_6) + ker(delta_5) \cap off_diag_part + 2ker(delta_5)):
C2
Confirmed that quotient agrees with projecting onto (id_[5], id_[5]) component

Checking lemma for i=5, n=4
```

```
ker(delta_5)/(im(delta_6) + ker(delta_5) \cap off_diag_part + 2ker(delta_5)):
C2
Confirmed that quotient agrees with projecting onto (id_[5], id_[5]) component

Checking lemma for i=5, n=5
ker(delta_5)/(im(delta_6) + ker(delta_5) \cap off_diag_part + 2ker(delta_5)):
C2
Confirmed that quotient agrees with projecting onto (id_[5], id_[5]) component

Checking lemma for i=6, n=0
ker(delta_6)/(im(delta_7) + ker(delta_6) \cap off_diag_part + 2ker(delta_6)):
C2
Confirmed that quotient agrees with projecting onto (id_[6], id_[6]) component

Checking lemma for i=6, n=1
ker(delta_6)/(im(delta_7) + ker(delta_6) \cap off_diag_part + 2ker(delta_6)):
C2
Confirmed that quotient agrees with projecting onto (id_[6], id_[6]) component

Checking lemma for i=6, n=2
ker(delta_6)/(im(delta_7) + ker(delta_6) \cap off_diag_part + 2ker(delta_6)):
C2
Confirmed that quotient agrees with projecting onto (id_[6], id_[6]) component

Checking lemma for i=6, n=3
ker(delta_6)/(im(delta_7) + ker(delta_6) \cap off_diag_part + 2ker(delta_6)):
C2
Confirmed that quotient agrees with projecting onto (id_[6], id_[6]) component

Checking lemma for i=6, n=4
ker(delta_6)/(im(delta_7) + ker(delta_6) \cap off_diag_part + 2ker(delta_6)):
C2
Confirmed that quotient agrees with projecting onto (id_[6], id_[6]) component

Checking lemma for i=6, n=5
ker(delta_6)/(im(delta_7) + ker(delta_6) \cap off_diag_part + 2ker(delta_6)):
C2
Confirmed that quotient agrees with projecting onto (id_[6], id_[6]) component

Checking lemma for i=6, n=6
ker(delta_6)/(im(delta_7) + ker(delta_6) \cap off_diag_part + 2ker(delta_6)):
C2
Confirmed that quotient agrees with projecting onto (id_[6], id_[6]) component
```

Some extra sanity checks, confirming that $\pi_i(\Sigma^n H\mathbb{Z}/2 \otimes_{H\mathbb{Z}/4} \Sigma^n H\mathbb{Z}/2)$ computed using the non-abelian derived functor of the tensor square agrees with expectations.

[5]:
```python
for n in range(0, 6):
    Tensor2sM_test(n, 5)
    print("")
```

```
n=0, checking pi_i(|Tensor2(sM)|) for i from 0 to 5
Success

n=1, checking pi_i(|Tensor2(sM)|) for i from 0 to 5
Success

n=2, checking pi_i(|Tensor2(sM)|) for i from 0 to 5
Success

n=3, checking pi_i(|Tensor2(sM)|) for i from 0 to 5
Success

n=4, checking pi_i(|Tensor2(sM)|) for i from 0 to 5
Success

n=5, checking pi_i(|Tensor2(sM)|) for i from 0 to 5
Success
```