Pensieve header: The full list of w equations with the unitary V gauged by C to get the simplest buckle equation.

```
SetDirectory["C:\\drorbn\\AcademicPensieve\\2012-05\\beta5.1"];
<< betaCalculus.m
Clear[ħ]; Unprotect[C];
$PerturbativeDegree = 4;
βSimplify[expr_] := Replace[
   Series[Normal[expr], {ħ, 0, $PerturbativeDegree}],
   sd_SeriesData :> MapAt[Expand, sd, 3]
  ];
βCollect[B[ω_, μ_]] := B[βSimplify[ω], βSimplify[μ]];
{V0, C0, sol} = Get[Switch[$PerturbativeDegree,
    4, "SolutionToDegree4-120523.m",
    6, "SolutionToDegree6-120523.m",
    8, "SolutionToDegree8-120524.m"
   ]];
C = C0 /. κ1 → 0;
ν = B[Series[Sinh[c1 ħ / 2] / (c1 ħ / 2), {ħ, 0, $PerturbativeDegree}], 0];
Φ0 =
  (Inverse[V0] // dP[12, 3]) ** Inverse[V0] ** (V0 // dP[2, 3]) ** (V0 // dP[1, 23]);
V = Inverse[C // dP[12]] ** V0 ** (C ** (C // dP[2]));
Φ = (Inverse[V] // dP[12, 3]) ** Inverse[V] ** (V // dP[2, 3]) ** (V // dP[1, 23]);
CC = C ** C;
Clear[C];
```

```
Φ == Φ0
```

```
True
```

```
DeleteCases[{
  "Test" → xxx == yyy,
  "R4" → R[2, 3] ** R[1, 3] ** V == V ** (R[1, 3] // dΔ[1, 1, 2]),
  "TwistEq" → V ** Θ[1, 2] == R[1, 2] ** (V // dP[2, 1]),
  "Unitarity" → V ** (CC // dP[12]) ** (V // dA[1] // dA[2]) == CC ** (CC // dP[2]),
  "VerticalFlipForV" →
   V ** (CC // dP[12]) ** (V // dS[1] // dS[2]) == R[1, 2] ** CC ** (CC // dP[2]),
  "CapEquation" → ((V ** (CC // dP[12])) // dcap[1] // dcap[2]) ==
    CC ** (CC // dP[2]),
  "VSidesDelete" → (V // dη[1]) == B[1, 0] && (V // dη[2]) == B[1, 0],
  "CapsAndCups" → CC == (CC // dS[1]),
  "Pentagon" → Φ ** (Φ // dP[1, 23, 4]) ** (Φ // dP[2, 3, 4]) ==
    (Φ // dP[12, 3, 4]) ** (Φ // dP[1, 2, 34]),
  "PositiveHexagon" → (Θ[1, 2, +1] // dP[12, 3]) ==
    (Φ ** Θ[2, 3, +1] ** Inverse[Φ // dP[1, 3, 2]] ** Θ[1, 3, +1] ** (Φ // dP[3, 1, 2])),
  "NegativeHexagon" → (Θ[1, 2, -1] // dP[12, 3]) ==
    (Φ ** Θ[2, 3, -1] ** Inverse[Φ // dP[1, 3, 2]] ** Θ[1, 3, -1] ** (Φ // dP[3, 1, 2])),
  "HorizontalFlipForΦ" → Φ ** (Φ // dP[3, 2, 1]) == B[1, 0],
  "VerticalFlipForΦ" → Φ ** (Φ // dS[1] // dS[2] // dS[3]) == B[1, 0],
  "OverhandEquation" →
   (Φ // dΔ[1, 0, 1] // dS[2] // dS[3] // dm[0, 3, 0] // dm[1, 2, 1]) == B[1, 0],
  "ValueOfν" → (Φ // dS[2] // dm[3, 2, 2] // dm[2, 1, 1]) == ν,
  "ValueOfCC" → CC ** CC == Inverse[ν],
  "VTopDelete" → (V // dS[1] // dm[2, 1, 1]) == R[1, 1, -1 / 2],
  "EKTopCapLeftPuncture" → (V // tη[1] // dS[2] // hm[1, 2, 1]) == B[1, 0],
  "EKRightCupLeftPuncture" → (V // hη[2] // tη[1] // dm[1, 2, 1]) == B[1, 0],
  "EKRightCupTopPuncture" → (V // hη[2] // dS[1] // dm[2, 1, 1]) == B[1, 0],
  "EKTopCapRightPuncture" → (V // tη[2] // dS[1] // dm[2, 1, 1]) == R[1, 1, -1 / 2],
  "EKLeftCupRightPuncture" → (V // hη[1] // tη[2] // dm[2, 1, 1]) == R[1, 1, 1 / 2],
  "EKLeftCupTopPuncture" → (V // hη[1] // dS[1] // dm[2, 1, 1]) == R[1, 1, -1 / 2],
  "EKLeftCupTopPuncture-2" →
   (V // hη[1] // dS[2] // dm[1, 2, 1]) == (R[1, 1, -1 / 2] // dS[1]),
  "BuckleEquation" → (
    buckle = (Inverse[Φ] // dP[13, 2, 4]) **
      (Φ // dP[1, 3, 2]) ** Θ[3, 2] ** Inverse[Φ] ** (Φ // dP[12, 3, 4]);
    LuckyV = buckle // tη[1] // hη[2] // dm[1, 2, 1] // tη[3] // hη[4] //
      dm[3, 4, 2];
    V == LuckyV
   )
 }, _ → True]
```

```
{Test → xxx == yyy}
```

```
{V // dcap[1] // tη[2],
  V // dcap[2] // tη[1]} // ColumnForm
```

$$\begin{pmatrix} 1 & h[2] \\ t[1] & \frac{1}{2}+\frac{c_1\hbar}{8}+\frac{1}{48}c_1^2\hbar^2+\frac{1}{384}c_1^3\hbar^3+\frac{c_1^4\hbar^4}{3840}+O[\hbar]^5 \end{pmatrix}$$

```
(1)
```