```
> restart :
```

```
> K :=proc(x) local y; y := op(1, x) : if (type(x, `+`) = true) then add(K(op(i, x)), i = 1
    .. nops(x)) elif
        (type(x, `*`) = true) then expand(K(y)·x/y + y·K(x/y)) elif
        (type(x, `^`) = true) then op(2, x)·y^(op(2, x) - 1)·K(y) elif
        (type(x, function) = true) then Kfunction(x) elif
        (type(x, symbol) = true) then Ksymbol(x) else 0 fi end proc:
```

```
> Ksymbol :=proc(x); if x = k# then 0 elif x = P then (-L(k)·P - L(L(k))) elif x = P1 then (
    -L#(k)·P - L(L#(k)) - 2·I·Tau(k)) else 'K'(x) fi end proc:
```

```
> recherche :=proc(x) local y; y := op(1, x); while (type(y, function) = true) do y := op(1, y)
    od; y; end proc:
```

```
> Kfunction :=proc(x) local y; y := recherche(x); if y = k then 'K'(x) else Kfunction1(x) fi
    end proc:
```

```
> Kfunction1 :=proc(x) local y, J : if x = k# then 0 elif x = P then (-L(k)·P - L(L(k))) elif x
    = P1 then (-L#(k)·P - L(L#(k)) - 2·I·Tau(k)) else J := op(0, x) : y := op(1, x) : if J
    = L then L(Kfunction1(y)) - 'L(k)'·L(y) elif
                                                                        J = L#
    then L#(Kfunction1(y)) - 'L#'(k)·L(y) elif
                                                                        J = K#
    then K#(Kfunction1(y)) elif J = Tau then Tau(Kfunction1(y)) - 'L(k)'·Tau(y) -'Tau(k)'
    ·L(y) fi fi end proc:
```

```
> L :=proc(x) local y; y := op(1, x) : if (type(x, `+`) = true) then add(L(op(i, x)), i = 1
    .. nops(x)) elif
        (type(x, `*`) = true) then expand(L(y)·x/y + y·L(x/y)) elif
        (type(x, `^`) = true) then op(2, x)·y^(op(2, x) - 1)·L(y) elif
        (type(x, function) = true) then 'L'(x) elif
        (type(x, symbol) = true) then 'L'(x) else 0 fi end proc:
```

```
> L# :=proc(x) local y; y := op(1, x) : if (type(x, `+`) = true) then add(L#(op(i, x)), i = 1
    .. nops(x)) elif
        (type(x, `*`) = true) then expand(L#(y)·x/y + y·L#(x/y)) elif
        (type(x, `^`) = true) then op(2, x)·y^(op(2, x) - 1)·L#(y) elif
        (type(x, function) = true) then 'L#(x)' elif
        (type(x, symbol) = true) then 'L#'(x) else 0 fi end proc:
```

```
> Tau :=proc(x) local y; y := op(1, x) : if (type(x, `+`) = true) then add(Tau(op(i, x)), i = 1
    .. nops(x)) elif
        (type(x, `*`) = true) then expand(Tau(y)·x/y + y·Tau(x/y)) elif
        (type(x, `^`) = true) then op(2, x)·y^(op(2, x) - 1)·Tau(y) elif
        (type(x, function) = true) then 'Tau'(x) elif
        (type(x, symbol) = true) then 'Tau'(x) else 0 fi end proc:
```

```
> K# :=proc(x) local y; y := op(1, x) : if (type(x, `+`) = true) then add(K#(op(i, x)), i = 1
      .. nops(x)) elif
                  (type(x, `*`) = true) then expand(K#(y)·(x/y) + y·K#(x/y))elif
                  (type(x, `^`) = true) then op(2, x)·y^(op(2, x) − 1) ·K#(y) elif
                  (type(x, function) = true) then Kfunction#(x) elif
                  (type(x, symbol) = true) then Ksymbol#(x) else 0 fi end proc:
```

```
> Kfunction# :=proc(x) local y; y := recherche(x); if y = k# then 'K#'(x) else Kfunction1#(x)
      fi end proc:
```

```
> Kfunction1# :=proc(x) local y, J : if x = k then 0 elif x = P1 then (−L#(k#)·P1
      − L#(L#(k#))) elif x = P then (−L(k#)·P1 − L#(L(k#)) + 2·I·Tau(k#))else J
      := op(0, x) : y := op(1, x) : if J = L then L(Kfunction1#(y)) − 'L(k#)'·L#(y) elif
                                                                              J = L#
      then L#(Kfunction1#(y)) − 'L#'(k#)·L#(y) elif
                                                                              J = K
      then K(Kfunction1#(y)) elif J = Tau then Tau(Kfunction1#(y)) − 'L#(k#)'·Tau(y) −
      'Tau(k#)'·L#(y)fi fi end proc:
  Ksymbol# :=proc(x); if x = k then 0 elif x = P1 then (−L#(k#)·P1 − L#(L#(k#))) elif x = P
      then (−L(k#)·P1 − L#(L(k#)) + 2·I·Tau(k#) )else 'K#(x)' fi end proc:
```

```
>
  Der :=proc(x) local y; y := op(1, x) : if (type(x, `+`) = true) then add(Der(op(i, x)), i = 1
      .. nops(x)) elif
                  (type(x, `*`) = true) then expand((x/y)·Der(y) + y·Der(x/y))elif
                  (type(x, `^`) = true) then op(2, x)·y^(op(2, x) − 1) ·Der(y) elif
                  ( (type(x, function) = true) or (type(x, symbol) = true)) then Tau(x)·W[1]
      + L(x)·W[2] + K(x)·W[3] + L#(x)·W[4] + K#(x)·W[5]
                  else 0 fi end proc:
  derivation :=proc(x) : collect( Der(x), [W[1], W[2], W[3], W[4], W[5]]) : end proc:
```

```
> with(DifferentialGeometry) :
> with(Tools) : with(LinearAlgebra) :
> DGsetup([x, y, z, y1, z1], [b, b1, c, c1, d, d1, e, e1], M, verbose);
```

The following coordinates have been protected:

[x, y, z, y1, z1, b, b1, c, c1, d, d1, e, e1]

The following vector fields have been defined and protected:

[D_x, D_y, D_z, D_y1, D_z1, D_b, D_b1, D_c, D_c1, D_d, D_d1, D_e, D_e1]

The following differential 1-forms have been defined and protected:

[dx, dy, dz, dy1, dz1, db, db1, dc, dc1, dd, dd1, de, de1]

frame name: M **(1)**

```
M > g := Matrix([[c·c1, 0, 0, 0, 0], [b, c, 0, 0, 0], [d, e, c/c1, 0, 0], [b1, 0, 0, c1, 0], [d1, 0,
```

$0, e1, \frac{c1}{c}]]]);$

$$g := \begin{bmatrix} c\,c1 & 0 & 0 & 0 & 0 \\ b & c & 0 & 0 & 0 \\ d & e & \frac{c}{c1} & 0 & 0 \\ b1 & 0 & 0 & c1 & 0 \\ d1 & 0 & 0 & e1 & \frac{c1}{c} \end{bmatrix} \quad (2)$$

M > $h := MatrixInverse(g):$

M > $A := map(evalDG, (ExteriorDerivative(g).h));$

$$A := \left[\left[\frac{dc}{c} + \frac{dc1}{c1}, 0\,dx, 0\,dx, 0\,dx, 0\,dx\right], \right. \quad (3)$$

$$\left[\frac{db}{c\,c1} - \frac{b\,dc}{c^2\,c1}, \frac{dc}{c}, 0\,dx, 0\,dx, 0\,dx\right],$$

$$\left[\frac{(-d\,c + e\,b)\,dc}{c^3\,c1} - \frac{(-d\,c + e\,b)\,dc1}{c^2\,c1^2} + \frac{dd}{c\,c1} - \frac{b\,de}{c^2\,c1}, -\frac{e\,dc}{c^2} + \frac{e\,dc1}{c1\,c} + \frac{de}{c}, \frac{dc}{c} - \frac{dc1}{c1}, 0\,dx, 0\,dx\right],$$

$$\left[\frac{db1}{c\,c1} - \frac{b1\,dc1}{c\,c1^2}, 0\,dx, 0\,dx, \frac{dc1}{c1}, 0\,dx\right],$$

$$\left.\left[-\frac{(-d1\,c1 + e1\,b1)\,dc}{c1^2\,c^2} + \frac{(-d1\,c1 + e1\,b1)\,dc1}{c1^3\,c} + \frac{dd1}{c\,c1} - \frac{b1\,de1}{c\,c1^2}, 0\,dx, 0\,dx, \frac{e1\,dc}{c\,c1} - \frac{e1\,dc1}{c1^2} + \frac{de1}{c1}, -\frac{dc}{c} + \frac{dc1}{c1}\right]\right]$$

M > $t[1] := \frac{dc}{c}:$

M > $t[2] := \frac{db}{c\,c1} - \frac{b\,dc}{c^2\,c1}:$

M > $t[3] := \frac{(-d\,c + e\,b)\,dc}{c^3\,c1} - \frac{(-d\,c + e\,b)\,dc1}{c^2\,c1^2} + \frac{dd}{c\,c1} - \frac{b\,de}{c^2\,c1}:$

M > $t[4] := -\frac{e\,dc}{c^2} + \frac{e\,dc1}{c1\,c} + \frac{de}{c}:$

M > $t[5] := \frac{dc1}{c1}:$

M > $t[6] := \frac{db1}{c\,c1} - \frac{b1\,dc1}{c\,c1^2}:$

M > $t[7] := -\frac{(-d1\,c1 + e1\,b1)\,dc}{c1^2\,c^2} + \frac{(-d1\,c1 + e1\,b1)\,dc1}{c1^3\,c} + \frac{dd1}{c\,c1} - \frac{b1\,de1}{c\,c1^2}:$

M > $t[8] := \frac{e1\, dc}{c\, c1} - \frac{e1\, dc1}{c1^2} + \frac{de1}{c1}$:

M > $FD := FrameData([t[1], t[2], t[3], t[4], t[5], t[6], t[7], t[8], dx, dy, dz, dy1, dz1], N)$:

M > $DGsetup\left(FD, [E], \left[\text{alpha}[1], \text{alpha}[2], \text{alpha}[3], \text{alpha}[4], \alpha^{\#}[1], \alpha^{\#}[2], \alpha^{\#}[3], \alpha^{\#}[4], \text{rho}, \text{kappa}, \text{zeta}, \kappa^{\#}, \zeta^{\#}\right]\right)$;

frame name: N **(4)**

N > $T := Vector\left(\left[\text{rho}, \text{kappa}, \text{zeta}, \kappa^{\#}, \zeta^{\#}\right]\right)$:

N > $X := Der(L^{\#}(k)); X1 := Der(L(k^{\#})); XX := subs\left(\left\{W[3] = \frac{W[3]}{L^{\#}(k)}, W[5] = \frac{W[5]}{L(k^{\#})}\right\}, X\right) : XX1 := subs\left(\left\{W[3] = \frac{W[3]}{L^{\#}(k)}, W[5] = \frac{W[5]}{L(k^{\#})}\right\}, X1\right)$:

$$X := \mathrm{T}(L^{\#}(k))\, W_1 + L(L^{\#}(k))\, W_2 + K(L^{\#}(k))\, W_3 + L^{\#}(L^{\#}(k))\, W_4 - L^{\#}(k^{\#})\, L^{\#}(k)\, W_5$$

$$X1 := \mathrm{T}(L(k^{\#}))\, W_1 + L(L(k^{\#}))\, W_2 - L(k)\, L(k^{\#})\, W_3 + L^{\#}(L(k^{\#}))\, W_4 + K^{\#}(L(k^{\#}))\, W_5 \qquad \textbf{(5)}$$

N > $W := h.T$:

$Eq1 := P\cdot(W[1]\&wedge\, W[2]) - \frac{L(k)}{L^{\#}(k)}\cdot(W[1]\&wedge\, W[3]) + P1\cdot(W[1]\&wedge\, W[4]) - \frac{L^{\#}(k^{\#})}{L(k^{\#})}\cdot(W[1]\&wedge\, W[5]) + I\cdot(W[2]\&wedge\, W[4])$:

N > $Eq2 := -\frac{L(k)}{L^{\#}(k)}\cdot(W[2]\&wedge\, W[3]) + (W[3]\&wedge\, W[4]) - \frac{\mathrm{Tau}(k)}{L^{\#}(k)}\cdot(W[1]\&wedge\, W[3])$:

N > $Eq3 := \left(\frac{XX}{L^{\#}(k)}\,\&wedge\, W[3]\right)$:

N > $Eq4 := -\frac{L^{\#}(k^{\#})}{L(k^{\#})}\cdot(W[4]\&wedge\, W[5]) + (W[5]\&wedge\, W[2]) - \frac{\mathrm{Tau}(k^{\#})}{L(k^{\#})}\cdot(W[1]\&wedge\, W[5])$:

N > $Eq5 := \left(\frac{XX1}{L(k^{\#})}\,\&wedge\, W[5]\right)$:

omega := $Vector([Eq1, Eq2, Eq3, Eq4, Eq5])$:

Omega := $map(evalDG, g.\text{omega})$:

N > $A := map(evalDG, (ExteriorDerivative(g).h))$:

$B := A\ \&MatrixWedge\ T$:

N > $SE := map(evalDG, (B\ \&MatrixPlus\ \text{Omega}))$:

N > $List := GenerateForms\left(\left[\text{alpha}[1], \text{alpha}[2], \text{alpha}[3], \text{alpha}[4], \alpha^{\#}[1], \alpha^{\#}[2], \alpha^{\#}[3], \alpha^{\#}[4], \text{rho}, \text{kappa}, \text{zeta}, \kappa^{\#}, \zeta^{\#}\right], 2\right)$:

N > *result* :=**proc**(*l*) **local** *k, t, X*; *X* := 0 : *t* := *expand*(*GetComponents*(*l, List*)) : **for** *k* **from** 1 **to** 78 **do** *X* := *X* + *t*[*k*]·*List*[*k*] **od**; *X*; **end proc**:

N > $Res1 := result(SE[1])$;

(6)

$$Res1 := \alpha_1 \wedge \rho + \alpha^{\#}_1 \wedge \rho + \left(\frac{\mathrm{I}\, b1}{c\, c1} + \frac{c1\, L(k)\, e}{c^2\, L^{\#}(k)} + \frac{P}{c} \right) \rho \wedge \kappa - \frac{L(k)\, c1\, \rho \wedge \zeta}{L^{\#}(k)\, c} + \left(-\frac{\mathrm{I}\, b}{c\, c1} + \frac{c\, L^{\#}(k^{\#})\, e1}{c1^2\, L(k^{\#})} + \frac{P1}{c1} \right) \rho \wedge \kappa^{\#} - \frac{L^{\#}(k^{\#})\, c\, \rho \wedge \zeta^{\#}}{L(k^{\#})\, c1} + \mathrm{I}\, \kappa \wedge \kappa^{\#} \quad (6)$$

N > $Res2 := result(SE[2]);$

$$Res2 := \alpha_1 \wedge \kappa + \alpha_2 \wedge \rho + \left(\frac{\mathrm{T}(k)\, e}{c^2\, L^{\#}(k)} - \frac{e\, b1}{c^2\, c1} - \frac{L(k)\, d}{c^2\, L^{\#}(k)} + \frac{\mathrm{I}\, b\, b1}{c^2\, c1^2} + \frac{L(k)\, e\, b}{c^3\, L^{\#}(k)} + \frac{b\, P}{c^2\, c1} \right) \rho \wedge \kappa + \left(-\frac{\mathrm{T}(k)}{c\, L^{\#}(k)} + \frac{b1}{c\, c1} \right) \rho \wedge \zeta + \left(-\frac{d}{c\, c1} + \frac{e\, b}{c1\, c^2} - \frac{\mathrm{I}\, b^2}{c^2\, c1^2} + \frac{b\, L^{\#}(k^{\#})\, e1}{c1^3\, L(k^{\#})} + \frac{b\, P1}{c\, c1^2} \right) \rho \wedge \kappa^{\#} - \frac{b\, L^{\#}(k^{\#})\, \rho \wedge \zeta^{\#}}{L(k^{\#})\, c1^2} - \frac{L(k)\, c1\, \kappa \wedge \zeta}{L^{\#}(k)\, c} + \left(-\frac{e}{c} + \frac{\mathrm{I}\, b}{c\, c1} \right) \kappa \wedge \kappa^{\#} + \zeta \wedge \kappa^{\#} \quad (7)$$

N > $Res3 := result(SE[3]);$

$$Res3 := \alpha_1 \wedge \zeta + \alpha_3 \wedge \rho + \alpha_4 \wedge \kappa - \alpha^{\#}_1 \wedge \zeta + \left(\frac{L(L^{\#}(k))\, d}{c1\, c^2\, L^{\#}(k)} - \frac{e\, L^{\#}(k^{\#})\, d1}{L(k^{\#})\, c1^2\, c} + \frac{e\, L^{\#}(k^{\#})\, e1\, b1}{L(k^{\#})\, c1^3\, c} + \frac{e\, L^{\#}(L^{\#}(k))\, b1}{c1^2\, c^2\, L^{\#}(k)} - \frac{e\, \mathrm{T}(L^{\#}(k))}{c1\, c^2\, L^{\#}(k)} + \frac{e^2\, \mathrm{T}(k)}{c^3\, L^{\#}(k)} - \frac{e^2\, b1}{c1\, c^3} + \frac{\mathrm{I}\, d\, b1}{c1^2\, c^2} + \frac{d\, P}{c1\, c^2} \right) \rho \wedge \kappa + \left(\frac{d1\, L^{\#}(k^{\#})}{L(k^{\#})\, c1^2} - \frac{e1\, L^{\#}(k^{\#})\, b1}{L(k^{\#})\, c1^3} - \frac{L^{\#}(L^{\#}(k))\, b1}{c1^2\, c\, L^{\#}(k)} - \frac{L(L^{\#}(k))\, b}{c1\, c^2\, L^{\#}(k)} + \frac{\mathrm{T}(L^{\#}(k))}{c1\, c\, L^{\#}(k)} - \frac{\mathrm{T}(k)\, e}{c^2\, L^{\#}(k)} + \frac{e\, b1}{c^2\, c1} + \frac{L(k)\, e\, b}{c^3\, L^{\#}(k)} - \frac{L(k)\, d}{c^2\, L^{\#}(k)} \right) \rho \wedge \zeta + \left(\frac{2\, L^{\#}(k^{\#})\, e1\, d}{L(k^{\#})\, c1^3} - \frac{L^{\#}(k^{\#})\, e1\, e\, b}{L(k^{\#})\, c1^3\, c} + \frac{L^{\#}(L^{\#}(k))\, d}{c1^2\, c\, L^{\#}(k)} - \frac{L^{\#}(L^{\#}(k))\, e\, b}{c1^2\, c^2\, L^{\#}(k)} - \frac{e\, d}{c1\, c^2} + \frac{e^2\, b}{c1\, c^3} - \frac{\mathrm{I}\, d\, b}{c1^2\, c^2} + \frac{d\, P1}{c1^2\, c} \right) \rho \wedge \kappa^{\#} + \left(-\frac{2\, L^{\#}(k^{\#})\, d}{c1^2\, L(k^{\#})} + \frac{L^{\#}(k^{\#})\, e\, b}{c1^2\, L(k^{\#})\, c} \right) \rho \wedge \zeta^{\#} + \left(\frac{L(L^{\#}(k))}{L^{\#}(k)\, c} - \frac{c1\, L(k)\, e}{c^2\, L^{\#}(k)} \right) \kappa \wedge \zeta + \left(\frac{e\, L^{\#}(k^{\#})\, e1}{L(k^{\#})\, c1^2} + \frac{e\, L^{\#}(L^{\#}(k))}{L^{\#}(k)\, c1\, c} - \frac{e^2}{c^2} + \frac{\mathrm{I}\, d}{c1\, c} \right) \kappa \wedge \kappa^{\#} - \frac{L^{\#}(k^{\#})\, e\, \kappa \wedge \zeta^{\#}}{c1\, L(k^{\#})} + \left(-\frac{c\, L^{\#}(k^{\#})\, e1}{c1^2\, L(k^{\#})} - \frac{L^{\#}(L^{\#}(k))}{L^{\#}(k)\, c1} + \frac{e}{c} \right) \zeta \wedge \kappa^{\#} + \frac{L^{\#}(k^{\#})\, c\, \zeta \wedge \zeta^{\#}}{L(k^{\#})\, c1} \quad (8)$$

N > $Torsion := \mathbf{proc}(S, i, j)\ \mathbf{local}\ k, X;\ k := 13 \cdot (i-1) - \frac{i \cdot (i-1)}{2} + j - i;\ X := GetComponents(S, List);\ X[k];\ \mathbf{end\ proc}:$

N > $eq := expand(2 \cdot Torsion(SE[2], 10, 12) - Torsion(SE[3], 11, 12) - Torsion(SE[1], 9, 12));$

$$eq := -\frac{3\, e}{c} + \frac{3\, \mathrm{I}\, b}{c\, c1} + \frac{L^{\#}(L^{\#}(k))}{L^{\#}(k)\, c1} - \frac{P1}{c1} \quad (9)$$

N > $Normalisation := expand(solve(eq, b));$

$$Normalisation := -\mathrm{I}\, e\, c1 + \frac{\frac{1}{3}\, \mathrm{I}\, L^{\#}\left(L^{\#}(k)\right)\, c}{L^{\#}(k)} - \frac{1}{3}\, \mathrm{I}\, P1\, c \tag{10}$$

N > $g := expand\left(\frac{(Normalisation + I \cdot e \cdot c1)}{c}\right);$

$$g := \frac{\frac{1}{3}\, \mathrm{I}\, L^{\#}\left(L^{\#}(k)\right)}{L^{\#}(k)} - \frac{1}{3}\, \mathrm{I}\, P1 \tag{11}$$