```
with(plottools)
[annulus, arc, arrow, circle, cone, cuboid, curve, cutin, cutout, cylinder, disk, dodecahedron,      (1)
    ellipse, ellipticArc, exportplot, extrude, getdata, hemisphere, hexahedron, homothety,
    hyperbola, icosahedron, importplot, line, octahedron, parallelepiped, pieslice, point, polygon,
    polygonbyname, prism, project, rectangle, reflect, rotate, scale, sector, semitorus, sphere,
    stellate, tetrahedron, torus, transform, translate, triangulate]
with(plots)
[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d,      (2)
    conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot,
    display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d,
    inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d,
    listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto,
    plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d,
    polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions,
    setoptions3d, shadebetween, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d,
    tubeplot]
with(DifferentialGeometry)
[&algmult, &minus, &mult, &plus, &tensor, &wedge, Annihilator, ApplyTransformation,      (3)
    ChangeFrame, ComplementaryBasis, ComposeTransformations, DGIm, DGImageSpace,
    DGNullSpace, DGRe, DGbasis, DGconjugate, DGsetup, DGsolve, DGzip,
    DeRhamHomotopy, DualBasis, ExteriorDerivative, ExteriorDifferentialSystems, Flow,
    FrameData, GetComponents, GroupActions, Hook, InfinitesimalTransformation,
    IntegrateForm, IntersectSubspaces, InverseTransformation, JetCalculus, Library,
    LieAlgebras, LieBracket, LieDerivative, Preferences, Pullback, PullbackVector,
    Pushforward, RemoveFrame, Tensor, Tools, Transformation, evalDG]
with(VectorCalculus)
[&x, `*`, `+`, `-`, `.`, <,>, <|>, About, AddCoordinates, ArcLength, BasisFormat, Binormal,      (4)
    ConvertVector, CrossProduct, Curl, Curvature, D, Del, DirectionalDiff, Divergence,
    DotProduct, Flux, GetCoordinateParameters, GetCoordinates, GetNames,
    GetPVDescription, GetRootPoint, GetSpace, Gradient, Hessian, IsPositionVector,
    IsRootedVector, IsVectorField, Jacobian, Laplacian, LineInt, MapToBasis,∇, Norm,
    Normalize, PathInt, PlotPositionVector, PlotVector, PositionVector, PrincipalNormal,
    RadiusOfCurvature, RootedVector, ScalarPotential, SetCoordinateParameters,
    SetCoordinates, SpaceCurve, SurfaceInt, TNBFrame, TangentLine, TangentPlane,
    TangentVector, Torsion, Vector, VectorField, VectorPotential, VectorSpace, Wronskian, diff,
    eval, evalVF, int, limit, series]
```

Define dual, reverse dual and 'to point' functions

```
ext43_dual := proc(ext43)
    local S1, comps;
    S1 := DifferentialGeometry:-evalDG([DifferentialGeometry:-&wedge(DifferentialGeometry:-
    &wedge(dx, dy), dz), DifferentialGeometry:-&wedge(DifferentialGeometry:-&wedge(dx, dy), dw),
    DifferentialGeometry:-&wedge(DifferentialGeometry:-&wedge(dx, dz), dw),
    DifferentialGeometry:-&wedge(DifferentialGeometry:-&wedge(dy, dz), dw)]);
    comps := DifferentialGeometry:-GetComponents(ext43, S1);
    return _DG([["form", M, 1], [[[1], comps[4]], [[2], -comps[3]], [[3], comps[2]], [[4],
    -comps[1]]]])
end proc
```

```
ext43_dual := proc(ext43)                                                              (5)
    local S1, comps;
    S1 := DifferentialGeometry:-evalDG([DifferentialGeometry:-
    `&wedge`(DifferentialGeometry:-`&wedge`(dx, dy), dz), DifferentialGeometry:-
    `&wedge`(DifferentialGeometry:-`&wedge`(dx, dy), dw), DifferentialGeometry:-
    `&wedge`(DifferentialGeometry:-`&wedge`(dx, dz), dw), DifferentialGeometry:-
    `&wedge`(DifferentialGeometry:-`&wedge`(dy, dz), dw)]);
    comps := DifferentialGeometry:-GetComponents(ext43, S1);
    return _DG([["form", M, 1], [[[1], comps[4]], [[2], VectorCalculus:-`-`(comps[3])],
    [[3], comps[2]], [[4], VectorCalculus:-`-`(comps[1])]]])
end proc
```

```
ext43_reverse_dual :=proc(ext43)
    local S1, comps;
    S1 := DifferentialGeometry:-evalDG([DifferentialGeometry:-&wedge(DifferentialGeometry:-
    &wedge(dx, dy), dz), DifferentialGeometry:-&wedge(DifferentialGeometry:-&wedge(dx, dy), dw),
    DifferentialGeometry:-&wedge(DifferentialGeometry:-&wedge(dx, dz), dw),
    DifferentialGeometry:-&wedge(DifferentialGeometry:-&wedge(dy, dz), dw)]);
    comps := DifferentialGeometry:-GetComponents(ext43, S1);
    return _DG([["form", M, 1], [[[1], -comps[4]], [[2], comps[3]], [[3], -comps[2]], [[4],
    comps[1]]]])
end proc
```

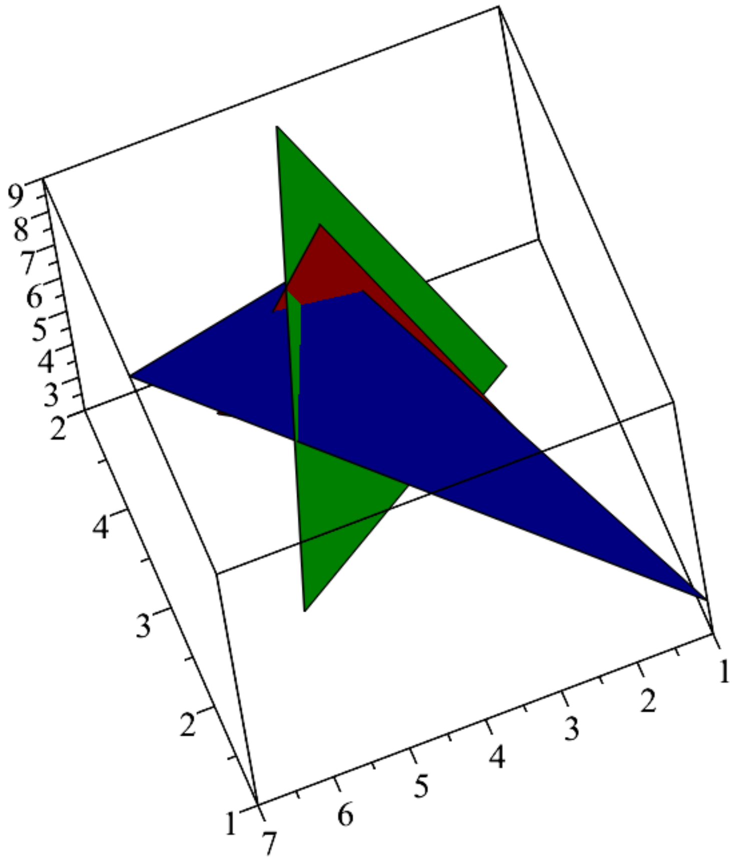
```
ext43_reverse_dual := proc(ext43)                                                      (6)
    local S1, comps;
    S1 := DifferentialGeometry:-evalDG([DifferentialGeometry:-
    `&wedge`(DifferentialGeometry:-`&wedge`(dx, dy), dz), DifferentialGeometry:-
    `&wedge`(DifferentialGeometry:-`&wedge`(dx, dy), dw), DifferentialGeometry:-
    `&wedge`(DifferentialGeometry:-`&wedge`(dx, dz), dw), DifferentialGeometry:-
    `&wedge`(DifferentialGeometry:-`&wedge`(dy, dz), dw)]);
    comps := DifferentialGeometry:-GetComponents(ext43, S1);
    return _DG([["form", M, 1], [[[1], VectorCalculus:-`-`(comps[4])], [[2], comps[3]],
    [[3], VectorCalculus:-`-`(comps[2])], [[4], comps[1]]]])
end proc
```

```
ext41_to_point :=proc(ext41)
    local S1, comps;
    S1 := DifferentialGeometry:-evalDG([dx, dy, dz, dw]);
    comps := DifferentialGeometry:-GetComponents(ext41, S1);
    return [VectorCalculus:-`*`(comps[1], comps[4]^VectorCalculus:-`-`(1)), VectorCalculus:-
    `*`(comps[2], comps[4]^VectorCalculus:-`-`(1)), VectorCalculus:-`*`(comps[3], comps[4]
    ^VectorCalculus:-`-`(1))]
```

```
end proc
ext41_to_point := proc(ext41)                                                        (7)
    local S1, comps;
    S1 := DifferentialGeometry:-evalDG([dx, dy, dz, dw]);
    comps := DifferentialGeometry:-GetComponents(ext41, S1);
    return [VectorCalculus:-`*`(comps[1], comps[4]^VectorCalculus:-`-`(1)),
    VectorCalculus:-`*`(comps[2], comps[4]^VectorCalculus:-`-`(1)), VectorCalculus:-
    `*`(comps[3], comps[4]^VectorCalculus:-`-`(1))]
end proc
```

Plot the three triangles to be intersected

```
display({polygonplot3d( [ [1, 3, 1], [4, 7, 4], [7, 9, 3] ], color = blue), polygonplot3d([[2, 2, 3], [6,
    7, 3], [4, 8, 4]], color = red), polygonplot3d([[2, 2, 4], [6, 7, 1], [4, 8, 5]], color = green) })
```



```
DGsetup([x, y, z, w], M, verbose)
```

The following coordinates have been protected:

$[x, y, z, w]$

The following vector fields have been defined and protected:

$[D\_x, D\_y, D\_z, D\_w]$

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

$$[dx, dy, dz, dw]$$

$$\textit{frame name: } M \tag{8}$$

Define functions for the three triangle points

$a := e1\_0 \cdot dx + e1\_1 \cdot dy + e1\_2 \cdot dz + e1\_3 \cdot dw$

$$a := e1\_0\, dx + e1\_1\, dy + e1\_2\, dz + e1\_3\, dw \tag{9}$$

$b := e2\_0 \cdot dx + e2\_1 \cdot dy + e2\_2 \cdot dz + e2\_3 \cdot dw$

$$b := e2\_0\, dx + e2\_1\, dy + e2\_2\, dz + e2\_3\, dw \tag{10}$$

$c := e3\_0 \cdot dx + e3\_1 \cdot dy + e3\_2 \cdot dz + e3\_3 \cdot dw$

$$c := e3\_0\, dx + e3\_1\, dy + e3\_2\, dz + e3\_3\, dw \tag{11}$$

Use the wedge product to join the three points to yield an ext4_3 for a triangle.

$t\_ext3 := (\ a \ \&wedge\ b\ )\ \&wedge\ c$

$$\begin{aligned} t\_ext3 := {} & (e1\_0\, e2\_1\, e3\_2 - e1\_0\, e2\_2\, e3\_1 - e1\_1\, e2\_0\, e3\_2 + e1\_1\, e2\_2\, e3\_0 \\ & + e1\_2\, e2\_0\, e3\_1 - e1\_2\, e2\_1\, e3\_0)\ dx \wedge dy \wedge dz + (e1\_0\, e2\_1\, e3\_3 - e1\_0\, e2\_3\, e3\_1 \\ & - e1\_1\, e2\_0\, e3\_3 + e1\_1\, e2\_3\, e3\_0 + e1\_3\, e2\_0\, e3\_1 - e1\_3\, e2\_1\, e3\_0)\ dx \wedge dy \wedge dw \\ & + (e1\_0\, e2\_2\, e3\_3 - e1\_0\, e2\_3\, e3\_2 - e1\_2\, e2\_0\, e3\_3 + e1\_2\, e2\_3\, e3\_0 \\ & + e1\_3\, e2\_0\, e3\_2 - e1\_3\, e2\_2\, e3\_0)\ dx \wedge dz \wedge dw + (e1\_1\, e2\_2\, e3\_3 - e1\_1\, e2\_3\, e3\_2 \\ & - e1\_2\, e2\_1\, e3\_3 + e1\_2\, e2\_3\, e3\_1 + e1\_3\, e2\_1\, e3\_2 - e1\_3\, e2\_2\, e3\_1)\ dy \wedge dz \wedge dw \end{aligned} \tag{12}$$

Get specific ext4_3s for each of the three triangles.

$t1\_ext3 := subs(\{e1\_0 = 1, e1\_1 = 3, e1\_2 = 1, e1\_3 = 1, e2\_0 = 4, e2\_1 = 7, e2\_2 = 4, e2\_3 = 1, e3\_0 = 7, e3\_1 = 9, e3\_2 = 3, e3\_3 = 1\}, t\_ext3\ )$

$$t1\_ext3 := 20\ dx \wedge dy \wedge dz - 6\ dx \wedge dy \wedge dw - 12\ dx \wedge dz \wedge dw - 10\ dy \wedge dz \wedge dw \tag{13}$$

$t2\_ext3 := subs(\{e1\_0 = 2, e1\_1 = 2, e1\_2 = 3, e1\_3 = 1, e2\_0 = 6, e2\_1 = 7, e2\_2 = 3, e2\_3 = 1, e3\_0 = 4, e3\_1 = 8, e3\_2 = 4, e3\_3 = 1\}, t\_ext3\ )$

$$t2\_ext3 := 44\ dx \wedge dy \wedge dz + 14\ dx \wedge dy \wedge dw + 4\ dx \wedge dz \wedge dw + 5\ dy \wedge dz \wedge dw \tag{14}$$

$t3\_ext3 := subs(\{e1\_0 = 2, e1\_1 = 2, e1\_2 = 4, e1\_3 = 1, e2\_0 = 6, e2\_1 = 7, e2\_2 = 1, e2\_3 = 1, e3\_0 = 4, e3\_1 = 8, e3\_2 = 5, e3\_3 = 1\}, t\_ext3\ )$

$$t3\_ext3 := 82\ dx \wedge dy \wedge dz + 14\ dx \wedge dy \wedge dw + 10\ dx \wedge dz \wedge dw + 23\ dy \wedge dz \wedge dw \tag{15}$$

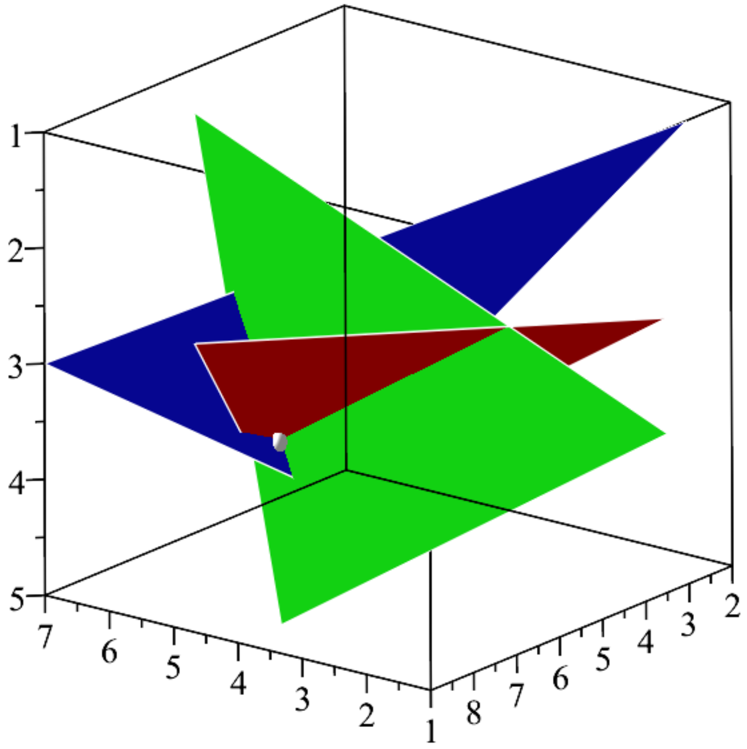
Compute the point of intersection for all three triangles. This is a combination of wedges and duals. Convert the ext4_1 to a point.

$intersection := ext41\_to\_point(\ ext43\_reverse\_dual(\ (ext43\_dual(t1\_ext3)\ \&wedge\ ext43\_dual(t2\_ext3))\ \&wedge\ ext43\_dual(t3\_ext3)\ )\ )$

$$intersection := \left[\frac{104}{23}, \frac{499}{69}, \frac{248}{69}\right] \tag{16}$$

Plot the triangles again but with the intersection point highlighted.

```
display( {polygonplot3d( [ [1, 3, 1], [4, 7, 4], [7, 9, 3] ], color = blue), polygonplot3d( [ [2, 2, 3], [6,
   7, 3], [4, 8, 4] ] , color = red), polygonplot3d( [ [2, 2, 4], [6, 7, 1], [4, 8, 5] ] , color = green),
    sphere(intersection, 0.1, color = white, style = patchnogrid) })
```


1
2
3
4
5
7
6
5
4
3
2
1
8
7
6
5
4
3
2