```
with(plottools)
```

[annulus, arc, arrow, circle, cone, cuboid, curve, cutin, cutout, cylinder, disk, dodecahedron, 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] **(1)**

```
with(plots)
```

[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, 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] **(2)**

```
with(DifferentialGeometry)
```

[&algmult, &minus, &mult, &plus, &tensor, &wedge, Annihilator, ApplyTransformation, 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] **(3)**

```
with(VectorCalculus )
```

[&x, `*`, `+`, `-`, `.`, <,>, <|>, About, AddCoordinates, ArcLength, BasisFormat, Binormal, 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] **(4)**

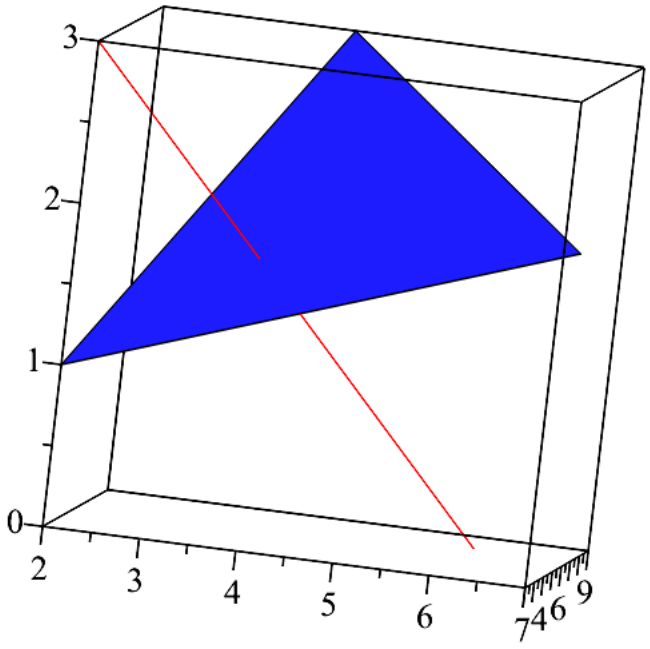
```
ext43_dual := proc(ext43)
    local S1, comps;
    S1 := DifferentialGeometry:-evalDG([DifferentialGeometry:-&wedge(DifferentialGeometry:-
    &wedge(dx, dy), dz), DifferentialGeometry:-&wedge(DifferentialGeometry:-&wedge(dx, dy), dw),
    DifferentialGeometry:-&wedge(DifferentialGeometry:-&wedge(dx, 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

ext42_dual := proc(ext42 )
 local S1, comps;
S1 := evalDG([dx &wedge dy , dx &wedge dz, dx &wedge dw, dy &wedge dz, dy &wedge dw, dz
    &wedge dw ]) ;
comps := GetComponents(ext42, S1);
 return _DG([["form", M, 2], [[[1, 2], comps[6]], [[1, 3], -comps[5]], [[1, 4], comps[4]], [[2, 3],
    comps[3]], [[2, 4], -comps[2]], [[3, 4], comps[1]]]]);
 end proc;
```

```
ext42_dual := proc(ext42)                                                    (7)
    local S1, comps;
    S1 := DifferentialGeometry:-evalDG([DifferentialGeometry:-`&wedge`(dx, dy),
    DifferentialGeometry:-`&wedge`(dx, dz), DifferentialGeometry:-`&wedge`(dx, dw),
    DifferentialGeometry:-`&wedge`(dy, dz), DifferentialGeometry:-`&wedge`(dy, dw),
    DifferentialGeometry:-`&wedge`(dz, dw)]);
    comps := DifferentialGeometry:-GetComponents(ext42, S1);
    return _DG([["form", M, 2], [[[1, 2], comps[6]], [[1, 3], VectorCalculus:-`-`(comps[5
    ])], [[1, 4], comps[4]], [[2, 3], comps[3]], [[2, 4], VectorCalculus:-`-`(comps[2])], [[3,
    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)                                                (8)
    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
```

Use Differential Geometry to compute triangle-edge instersections. Below is a plot of the triangle and edge for the example.

```
display({polygonplot3d( [ [2, 2, 1], [7, 4, 2], [4, 9, 3] ], color=blue), line([2, 2, 3], [6, 7, 0], color
    =red) })
```




*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*]

*frame name: M* **(9)**

Define functions for the three points. We will use all 3 for the triangle and two for the edge.

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

$a := e1\_0\, dx + e1\_1\, dy + e1\_2\, dz + e1\_3\, dw$ **(10)**

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

$b := e2\_0\, dx + e2\_1\, dy + e2\_2\, dz + e2\_3\, dw$ **(11)**

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

$c := e3\_0\, dx + e3\_1\, dy + e3\_2\, dz + e3\_3\, dw$ **(12)**

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

```
t_ext3 := ( a &wedge b ) &wedge c
```

$$t\_ext3 := (e1\_0\,e2\_1\,e3\_2 - e1\_0\,e2\_2\,e3\_1 - e1\_1\,e2\_0\,e3\_2 + e1\_1\,e2\_2\,e3\_0 + e1\_2\,e2\_0\,e3\_1 - e1\_2\,e2\_1\,e3\_0)\,dx \wedge dy \wedge dz + (e1\_0\,e2\_1\,e3\_3 - e1\_0\,e2\_3\,e3\_1 - e1\_1\,e2\_0\,e3\_3 + e1\_1\,e2\_3\,e3\_0 + e1\_3\,e2\_0\,e3\_1 - e1\_3\,e2\_1\,e3\_0)\,dx \wedge dy \wedge dw + (e1\_0\,e2\_2\,e3\_3 - e1\_0\,e2\_3\,e3\_2 - e1\_2\,e2\_0\,e3\_3 + e1\_2\,e2\_3\,e3\_0 + e1\_3\,e2\_0\,e3\_2 - e1\_3\,e2\_2\,e3\_0)\,dx \wedge dz \wedge dw + (e1\_1\,e2\_2\,e3\_3 - e1\_1\,e2\_3\,e3\_2 - e1\_2\,e2\_1\,e3\_3 + e1\_2\,e2\_3\,e3\_1 + e1\_3\,e2\_1\,e3\_2 - e1\_3\,e2\_2\,e3\_1)\,dy \wedge dz \wedge dw \quad \textbf{(13)}$$

Use the wedge product to join two points to yield an ext4_2 for the edge.

```
e_ext2 := a &wedge b;
```

$$e\_ext2 := (e1\_0\,e2\_1 - e1\_1\,e2\_0)\,dx \wedge dy + (e1\_0\,e2\_2 - e1\_2\,e2\_0)\,dx \wedge dz + (e1\_0\,e2\_3 - e1\_3\,e2\_0)\,dx \wedge dw + (e1\_1\,e2\_2 - e1\_2\,e2\_1)\,dy \wedge dz + (e1\_1\,e2\_3 - e1\_3\,e2\_1)\,dy \wedge dw + (e1\_2\,e2\_3 - e1\_3\,e2\_2)\,dz \wedge dw \quad \textbf{(14)}$$

Get instances for the triangle and edge

```
t1_ext3 := subs( {e1_0 = 2, e1_1 = 2, e1_2 = 1, e1_3 = 1, e2_0 = 7, e2_1 = 4, e2_2 = 2, e2_3 = 1, e3_0
    = 4, e3_1 = 9, e3_2 = 3, e3_3 = 1}, t_ext3 )
```

$$t1\_ext3 := 9\,dx \wedge dy \wedge dz + 31\,dx \wedge dy \wedge dw + 8\,dx \wedge dz \wedge dw - 3\,dy \wedge dz \wedge dw \quad \textbf{(15)}$$

```
e1_ext2 := subs( {e1_0 = 2, e1_1 = 2, e1_2 = 3, e1_3 = 1, e2_0 = 6, e2_1 = 7, e2_2 = 0, e2_3 = 1},
    e_ext2 )
```

$$e1\_ext2 := 2\,dx \wedge dy - 18\,dx \wedge dz - 4\,dx \wedge dw - 21\,dy \wedge dz - 5\,dy \wedge dw + 3\,dz \wedge dw \quad \textbf{(16)}$$

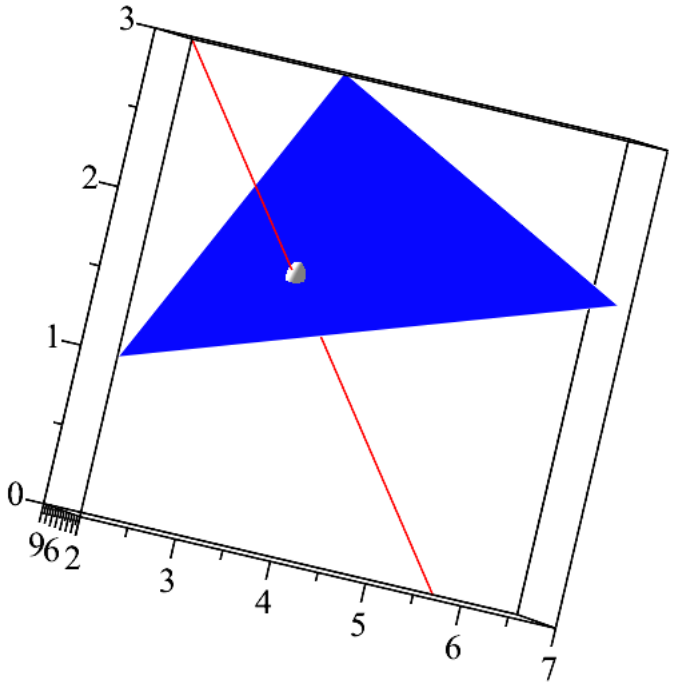
Compute the intersection of the edge and triangle. This is a combination of duals and wedges with a reverse dual and coversion to a point.

```
intersection := ext41_to_point( ext43_reverse_dual( ext42_dual(e1_ext2 )
    &wedge ext43_dual(t1_ext3) ) )
```

$$intersection := \left[\frac{538}{145}, \frac{120}{29}, \frac{249}{145}\right] \quad \textbf{(17)}$$

The Plot below has a sphere (squished due to the axis scaling) at the point of intersection.

```
display( {polygonplot3d( [ [2, 2, 1], [7, 4, 2], [4, 9, 3] ], color = blue), line( [2, 2, 3], [6, 7, 0], color
    = red), sphere( intersection, 0.1, color = white, style = patchnogrid ) } )
```


3
2
1
0
96
2
3
4
5
6
7