CHAPITRE 9: UTILISATION DE MAPLE EN GÉOMÉTRIE

GEOMETRIE PLANE:

```
> restart;
```

Le package ***geometry*** contient des outils pour la géométrie plane :

```
> with(geometry);
```

[Apollonius, Appolonius, AreCollinear, AreConcurrent, AreConcyclic, AreConjugate, AreHarmonic, AreOrthogonal, AreParallel, ArePerpendicular, AreSimilar, AreTangent, CircleOfSimilitude, CrossProduct, CrossRatio, DefinedAs, Equation, EulerCircle, EulerLine, ExteriorAngle, ExternalBisector, FindAngle, GergonnePoint, GlideReflection, HorizontalCoord, HorizontalName, InteriorAngle, IsEquilateral, IsOnCircle, IsOnLine, IsRightTriangle, MajorAxis, MakeSquare, MinorAxis, NagelPoint, OnSegment, ParallelLine, PedalTriangle, PerpenBisector, PerpendicularLine, Polar, Pole, RadicalAxis, RadicalCenter, RegularPolygon, RegularStarPolygon, SensedMagnitude, SimsonLine, SpiralRotation, StretchReflection, StretchRotation, TangentLine, VerticalCoord, VerticalName, altitude, apothem, area, asymptotes, bisector, center, centroid, circle, circumcircle, conic, convexhull, coordinates, detail, diagonal, diameter, dilatation, directrix, distance, draw, dsegment, ellipse, excircle, expansion, foci, focus, form, homology, homothety, hyperbola, incircle, inradius, intersection, inversion, line, medial, median, method, midpoint, orthocenter, parabola, perimeter, point, powerpc, projection, radius, randpoint, reciprocation, reflection, rotation, segment, sides, similitude, slope, square, stretch, tangentpc, translation, triangle, vertex, vertices]

La fonction ***point*** permet de définir un point du plan:
Syntaxes équivalentes: point(P, Px, Py) ou point(P, [Px, Py]).

<u>Exemple:</u> pour définir les points O(0,0) , A(-1,3) et B(4,2).

```
> point(O,[0,0]):point(A,[-1,3]):point(B,[4,2]):
```

La fonction ***detail*** permet d'obtenir des informations sur un objet géométrique:

```
> detail(A);
```

name of the object: A
form of the object: point2d
coordinates of the point: [-1, 3]

```
> point(M,X,Y):
```

La fonction ***line*** permet de définir une droite, dont on peut avoir l'équation.

```
> line(AB,[A,B]):Equation(AB,[x,y]);
```

$$-14+x+5\,y=0$$

La fonction ***triangle*** permet de définir un triangle , dont on peut avoir l'orthocentre , ou le centre de gravité.

```
> triangle(OAB,[O,A,B]):orthocenter(H,OAB):coordinates(H);
```

$$\left[\frac{1}{7},\frac{5}{7}\right]$$

```
> centroid(K,OAB):coordinates(K);
```

$$\left[1,\frac{5}{3}\right]$$

La fonction ***circle*** permet de définir un cercle, dont on peut avoir le centre, ou le rayon :

```
> circle(C,[O,A,B]):omega:=center(C):coordinates(omega);r:=radius(
  C);
```

$$\left[\frac{10}{7},\frac{15}{7}\right]$$

$$r:=\frac{1}{49}\sqrt{325}\,\sqrt{49}$$

```
> detail(C);
assume that the names of the horizontal and vertical axes are _x and _y, respec
tively
```

name of the object: C

form of the object: circle2d

name of the center: center_C

coordinates of the center: [10/7, 15/7]

*radius of the circle: 1/49*325^(1/2)*49^(1/2)*

*equation of the circle: _x^2+_y^2-20/7*_x-30/7*_y = 0*

Calcul de la distance entre 2 points ou de la distance d'un point à une droite:

```
> distance(A,B);
```

$$\sqrt{26}$$

```
> distance(M,AB);
```

$$\frac{1}{26}\left|-14+X+5\,Y\right|\sqrt{26}$$

Projeté d'un point sur une droite

projeté de M sur (AB):

```
> projection(P,M,AB):coordinates(P);
```

$$\left[\frac{7}{13}-\frac{5}{26}\,Y+\frac{25}{26}\,X,\ \frac{1}{26}\,Y-\frac{5}{26}\,X+\frac{35}{13}\right]$$

Equation de la parallèle ou de la perpendiculaire à une droite donnée passant par un point donné
d1 est la parallèle à (AB) passant par K, d2 est la perpendiculaire à (AB) passant par K:

```
> ParallelLine(d1,K,AB):Equation(d1,[x,y]);
```

$$-\frac{28}{3}+x+5\,y=0$$

```
> PerpendicularLine(d2,K,AB):Equation(d2,[x,y]);
```

$$-\frac{10}{3}+5\,x-y=0$$

```
> ArePerpendicular(d1,d2);
```

true

Intersection de 2 droites , d'une droite et d'un cercle , de 2 cercles

```
> intersection(i1,d1,d2):detail(i1);coordinates(i1);
```

name of the object: i1
form of the object: point2d
coordinates of the point: [1, 5/3]

$$\left[1, \frac{5}{3}\right]$$

```
> intersection(i2,d1,C):detail(i2);
```

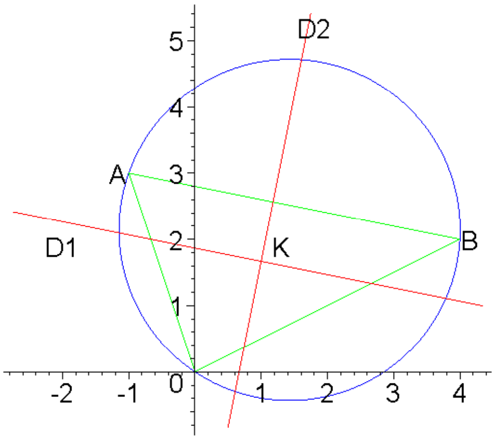
[*name of the object: d1_intersect1_C*
form of the object: point2d
*coordinates of the point: [103/78-5/78*1481^(1/2), 125/78+1/78*1481^(1/2)]* , *name of the*
object: d1_intersect2_C
form of the object: point2d
*coordinates of the point: [103/78+5/78*1481^(1/2), 125/78-1/78*1481^(1/2)]*]

Tracé de la figure:
on remarquera le paramétrage des 2 droites d1 et d2 et du cercle C .
La fonction ***display*** du package ***plots*** permet de représenter sur un même graphique différents objets graphiques précédemment définis , tels ici OAB , texte , etc ...

```
> with(plots):

> OAB:=plot([[HorizontalCoord(O),VerticalCoord(O)],[HorizontalCoor
  d(A),VerticalCoord(A)],[HorizontalCoord(B),VerticalCoord(B)],[Ho
  rizontalCoord(O),VerticalCoord(O)]],color=green):
> d1:=plot([HorizontalCoord(K)-5*t,VerticalCoord(K)+t,t=-2/3..3/4]
  ,numpoints=100,color=red):
> d2:=plot([HorizontalCoord(K)+t,VerticalCoord(K)+5*t,t=-0.5..0.75
  ],numpoints=100,color=red):
> cercleC:=plot([HorizontalCoord(omega)+r*cos(t),VerticalCoord(ome
  ga)+r*sin(t),t=0..2*Pi],numpoints=100,color=blue):
> texte:=textplot([[-1.15,3,`A`],[4+0.15,2,`B`],[1.8,5.2,`D2`],[-2
```

```
,1.9,`D1`],[1.3,1.9,`K`]],color=black):
> display({OAB,d1,d2,cercleC,texte},scaling=constrained);
```



AUTRES OUTILS DU PACKAGE LINALG POUR LA GÉOMÉTRIE:

Le package ***linalg*** d'algèbre linéaire contient également quelques outils utiles en géométrie euclidienne, notamment:

dotprod(v1,v2) calcule le produit scalaire(hermitien) des vecteurs v1 et v2.
norm(v) calcule la norme du vecteur v.
crossprod(v1,v2) calcule le produit vectoriel des vecteurs v1 et v2 (en dimension 3).
GramSchmidt({v1,v2,...,vn}) retourne la base orthogonale déduite de la famille {v1,v2,...,vn} par le procédé de Gram-Schmidt.

```
> with(linalg):
  v1:=vector([x1,y1,z1]);v2:=vector([x2,y2,z2]);
```

$$v1 := [x1, y1, z1]$$

$$v2 := [x2, y2, z2]$$

```
> dotprod(v1,v2);
```

$$x1\,\overline{x2} + y1\,\overline{y2} + z1\,\overline{z2}$$

```
> norm(v1);
```

$$\max(|x1|, |y1|, |z1|)$$

```
> crossprod(v1,v2);
```

$$[y1\,z2 - z1\,y2,\ z1\,x2 - x1\,z2,\ x1\,y2 - y1\,x2]$$

```
> v1:=vector([1,2,4]);v2:=vector([-1,6,5]);v3:=vector([2,0,3]);
  GramSchmidt({v1,v2,v3});
```

$$v1 := [1, 2, 4]$$

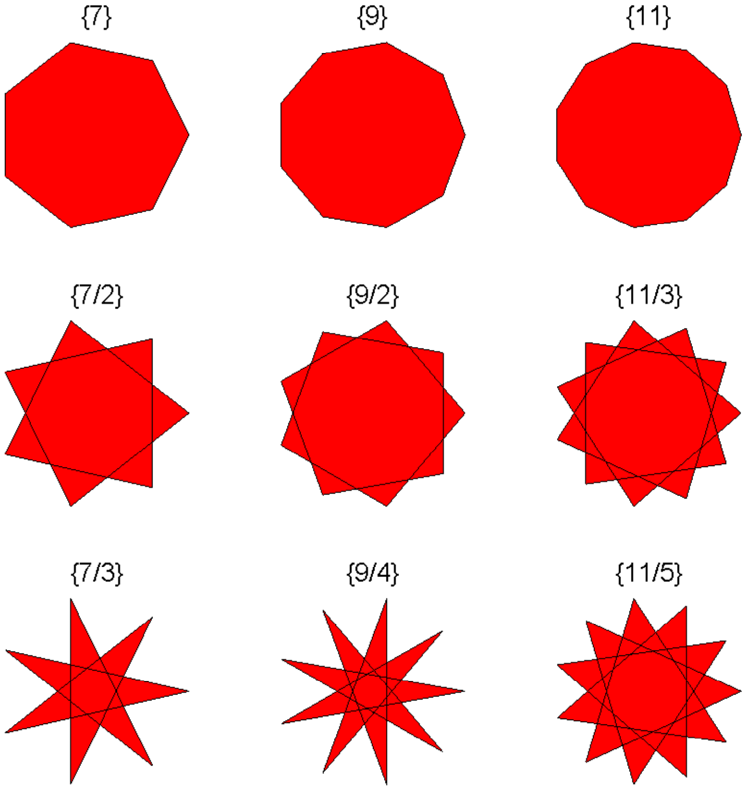
$$v2 := [-1, 6, 5]$$

$$v3 := [2, 0, 3]$$

$$\{[1, 2, 4], \left[\frac{-52}{21}, \frac{64}{21}, \frac{-19}{21}\right], \left[\frac{56}{341}, \frac{36}{341}, \frac{-32}{341}\right]\}$$

Représentation de polygônes réguliers, et polygônes étoilés:

```
> opts := filled=true, style=patch, axes=none:
  point(o,0,0):
  s1 := draw( RegularPolygon(_r7, 7, o, 3), opts, title="{7}" ):
  s2 := draw( RegularStarPolygon(_r7_2, 7/2, o, 3), opts,
  title="{7/2}" ):
  s3 := draw( RegularStarPolygon(_r7_3, 7/3, o, 3), opts,
  title="{7/3}" ):
  s4 := draw( RegularPolygon(_r9, 9, o, 3), opts, title="{9}" ):
  s5 := draw( RegularStarPolygon(_r9_2, 9/2, o, 3), opts,
  title="{9/2}" ):
  s6 := draw( RegularStarPolygon(_r9_4, 9/4, o, 3), opts,
  title="{9/4}" ):
  s7 := draw( RegularPolygon(_r11, 11, o, 3),opts, title="{11}"
  ):
  s8 := draw( RegularStarPolygon(_r11_3, 11/3, o, 3), opts,
  title="{11/3}" ):
  s9 := draw( RegularStarPolygon(_r11_5, 11/4, o, 3), opts,
  title="{11/5}" ):
  plots[display](array(1..3,1..3,[[s1,s4,s7],[s2,s5,s8],[s3,s6,s9]
  ]));
```



GEOMETRIE DE L'ESPACE:

Le package ***geom3d*** contient des outils pour travailler la géométrie dans l'espace :

```
> with(geom3d);
```

[*Archimedean, AreCollinear, AreConcurrent, AreConjugate, AreCoplanar, AreDistinct, AreParallel, ArePerpendicular, AreSameObjects, AreSamePlane, AreSkewLines, DefinedAs, DirectionRatios, Equation, FindAngle, FixedPoint, GlideReflect, GlideReflection, GreatDodecahedron, GreatIcosahedron, GreatRhombicuboctahedron, GreatRhombiicosidodecahedron, GreatStellatedDodecahedron, HarmonicConjugate, HexakisIcosahedron, HexakisOctahedron, InRadius, IsArchimedean, IsEquilateral, IsFacetted, IsOnObject, IsQuasi, IsRegular, IsRightTriangle, IsStellated, IsTangent, MidRadius, NormalVector, OnSegment, ParallelVector, PentagonalHexacontahedron, PentagonalIcositetrahedron, PentakisDodecahedron, QuasiRegularPolyhedron, RadicalCenter, RadicalLine, RadicalPlane, RegularPolyhedron, RhombicDodecahedron, RhombicTriacontahedron, RotatoryReflect, RotatoryReflection, ScrewDisplace, ScrewDisplacement, SmallRhombicuboctahedron, SmallRhombiicosidodecahedron, SmallStellatedDodecahedron, SnubCube, SnubDodecahedron, StereographicProjection, StretchRotate, TangentPlane, TetrakisHexahedron, TrapezoidalHexecontahedron, TrapezoidalIcositetrahedron, TriakisIcosahedron, TriakisOctahedron, TriakisTetrahedron, TruncatedCuboctahedron, TruncatedDodecahedron, TruncatedHexahedron, TruncatedIcosahedron, TruncatedIcosidodecahedron, TruncatedOctahedron, TruncatedTetrahedron, altitude, area, center, centroid, circle, coordinates, cube, cuboctahedron, detail, dilate, distance, dodecahedron, draw, dsegment, duality, faces, facet, form, gtetrahedron, hexahedron, homology, homothety, icosahedron, icosidodecahedron, identity, incident, intersection, inverse, inversion, line, midpoint, octahedron, parallel, parallelepiped, parallelpiped, plane, point, polar, pole, powerps, projection, radius, randpoint, reflect, reflection, rotate, rotation, schlafli, segment, sides, sphere, stellate, tetrahedron, tname, transform, translate, translation, transprod, triangle, unit, valuesubs, vertices, volume, xcoord, xname, ycoord, yname, zcoord, zname*]

La définition de points , de droites , de plans , se fait comme en dimension 2 par les fonctions ***point* , *line* ,** et ***plane***.

On définit 4 points O,A,B,C:

```
> point(O,[0,0,0]):point(A,[-1,2,0]):point(B,[0,1,1]):point(C,[-1,2,1]):
```

```
> detail(A);
```

name of the object: A
form of the object: point3d

coordinates of the point: [-1, 2, 0]

d est la droite passant par A dirigée par U(1,2,-1)

```
> U:=[1,2,-1]:line(d,[A,U]):
> plane(P,[A,B,C]):Equation(P,[x,y,z]);
  # équation du plan P=(ABC)
```

$$1 - x - y = 0$$

Un plan a une structure que l'on peut visualiser par ***detail*** :

```
> detail(P);
```

name of the object: P

form of the object: plane3d

equation of the plane: 1-x-y = 0

```
> parallel(Q, O, P):detail(Q);
  # Q est le plan passant par O et parallèle à P
```

name of the object: Q

form of the object: plane3d

equation of the plane: -x-y = 0

```
> plane(P1,[O,A,B]):
  intersection(delta,P1,P):detail(delta);
  # delta est l'intersection des plans P1 et P , définie par un
  système paramétrique :
Warning, assume that the parameter in the parametric equations is _t
```

name of the object: delta

form of the object: line3d

equation of the line: [x = -1-_t, y = 2+_t, z = -_t]

```
> ArePerpendicular(P,P1);
```

false

```
> distance(O,P);
```

$$\frac{1}{2}\sqrt{2}$$

```
> sphere(S,[O,A,B,C]):detail(S);
Warning, assume that the name of the axes are _x, _y and _z
```

name of the object: S

form of the object: sphere3d

name of the center: center_S_1

coordinates of the center: [-3/2, 1/2, 1/2]

*radius of the sphere: 1/2*11^(1/2)*

*surface area of the sphere: 11*Pi*

*volume of the sphere: 11/6*Pi*11^(1/2)* *equation of the sphere: _x^2+_y^2+_z^2+3*_x-_y-_z = 0*

```
> omega:=center(S):coordinates(omega);
  # centre omega de la sphère S
```

$$\left[\frac{-3}{2}, \frac{1}{2}, \frac{1}{2}\right]$$

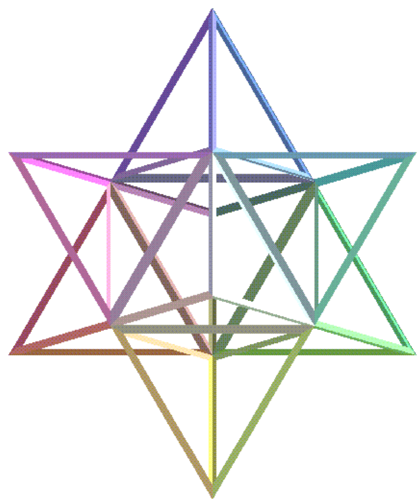
```
> projection(proj,O,P):coordinates(proj);
  # projeté de O sur le plan P
```

$$\left[\frac{1}{2}, \frac{1}{2}, 0\right]$$

AUTRES OUTILS DU PACKAGE GEOM3D:

Des outils existent pour représenter des polyèdres, solides de l'espace éventuellement étoilés:

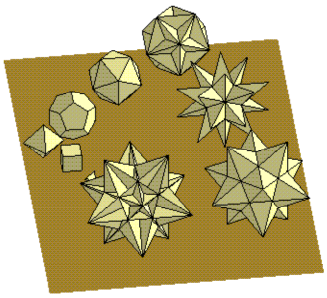
```
> octahedron(Oct,point(O,0,0,0),1.):
  stellate(StOct,Oct,1):
  draw( StOct, cutout=7/8, lightmodel=light4);
```



```
> point(e1,10,15,0), point(e2,-10,15,0),
  point(e3,-10,-15,0),point(e4,10,-15,0):
  r1 := 1.: point(o1,-2,2,r1):tetrahedron(p1,o1,r1):
  r2 := 2.: point(o2,-4,5,r2):cube(p2,o2,r2):
  r3 := 5./2: point(o3,-7,8,r3):octahedron(p3,o3,r3):
  r4 := 3.: point(o4,-3,12,r4):dodecahedron(p4,o4,r4):
  r5 := 7./2: point(o5,3,18,r5):icosahedron(p5,o5,r5):
  r6 := 6.:
  point(o6,15,9,r6):GreatStellatedDodecahedron(p6,o6,r6):
  r7 := 9./2: point(o7,11,23,r7):GreatDodecahedron(p7,o7,r7):
  r8 := 7.:
  point(o8,17,-9,r8):SmallStellatedDodecahedron(p8,o8,r8):
  r9 := 15./2: point(o9,2,-11,r9):GreatIcosahedron(p9,o9,r9):
  fl := [[25,30,0],[-10,30,0],[-10,-20,0],[25,-20,0]]:
```

```
pic1 := plots[polygonplot3d](fl,color=sienna):
pic2 :=
draw([seq(p||i,i=1..9)],color=COLOR(RGB,.9335294125,.9129411760,
.5205882350)):
plots[display]([pic2,pic1],style=patch,orientation=[-97,52],
title="Les neuf polyèdres réguliers", lightmodel=light4);
```

Les neuf polyèdres réguliers



Exercice corrigé 9:

Déterminer et représenter la surface S, ensemble des points M dont la somme des carrés des distances aux droites d1: $y=x$, $z=1$ et d2 : $y=-x$, $z=-1$ est égale à 3.

```
> restart:with(geom3d):
```

d1 est l'intersection de 2 plans p1 et q1:

```
> plane(p1,y-x=0,[x,y,z]):plane(q1,z-1=0,[x,y,z]):intersection(d1,
  p1,q1):
```

d2 est l'intersection de 2 plans p2 et q2:

```
> plane(p2,y+x=0,[x,y,z]):plane(q2,z+1=0,[x,y,z]):intersection(d2,
  p2,q2):
```

Définition d'un point quelconque M:

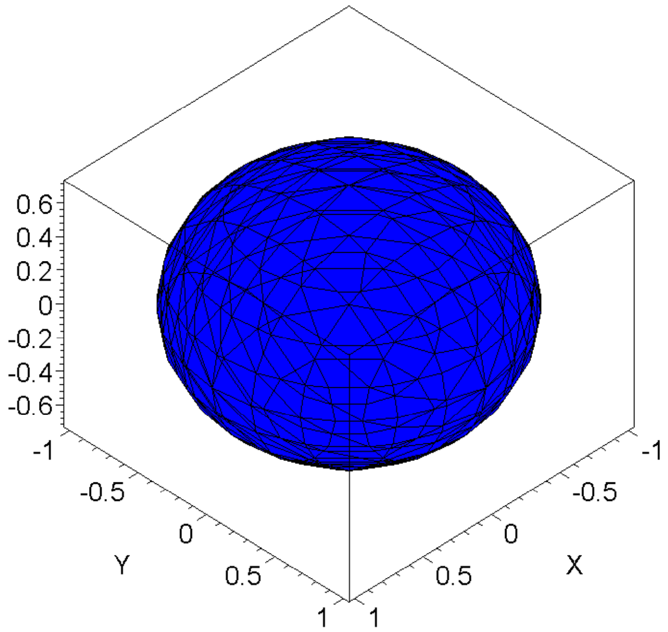
```
> point(M,X,Y,Z):
```

Equation et représentation de S:

```
> S:=distance(M,d1)^2+distance(M,d2)^2-3;
```

$$S := -1 + 2\,Z^2 + X^2 + Y^2$$

```
> with(plots):
> implicitplot3d(S,X=-1..1,Y=-1..1,Z=-1/sqrt(2)..1/sqrt(2),numpoints=300,color=blue,axes=BOXED);
```



Il s'agit d'un *ellipsoide* d'équation $X^2 + Y^2 + 2\,Z^2 = 1$ de centre O.

Travail dirigé 9:

Soit ABC un triangle du plan affine euclidien , M un point quelconque que l'on projette orthogonalement en H1,H2,H3 sur (BC) , (CA) , (AB) respectivement .
Montrer que M est sur le cercle circonscrit à ABC si et seulement si H1,H2,H3 sont alignés, sur la droite dite de ***Simson*** associée à M:

On travaillera dans un repère orthonormal tel que A(0,0) , B(1,0) , C(-2,3).