Algèbre linéaire et géométrie

(201-NYC)

Résolution de problèmes

RP 4

Déroulement Vous devez prendre connaissance du texte intitulé « Session Maple pour l'exercice RP-4». Ce texte vous donne toute l'information pertinente quant à la manière de tracer des graphiques de droites et de plans dans l'espace. Ce texte vous informe aussi sur les macro-commandes qu'il faut utiliser pour effectuer des multiplications scalaires et vectorielles.

Le texte dont il est question constitue en lui-même un modèle à suivre quant à la manière de documenter les développements mathématiques de vos solutions. Ce texte, que l'on retrouve à la fin du présent document, est présenté sous la forme d'une session Maple.

Objectif
— Développer la résolution de problèmes de géométrie vectorielle de la droite et du plan dans l'espace.
— S'initier aux tracés de droites et de plans dans l'espace afin d'être en mesure d'illustrer graphiquement des éléments de solutions à des problèmes de géométrie vectorielle.

Énoncé des problèmes

No. 1 Soit les trois plans d'équations

$$\begin{aligned} \Pi_1 &: \; 2\,x+y+z=1 \\ \Pi_2 &: \; 6\,x+y+z=3 \\ \Pi_3 &: \; x+y-z=0 \end{aligned}$$

a) Ces trois plans se coupant en un point P, trouver les coordonnées de ce point.

b) Illustrer graphiquement ces trois plans dans un même graphique en précisant les options `lightmodel=none` et `style=patchnogrid`. Choisir une couleur différente pour chacun. Superposer à ce graphique le tracé d'une petite sphère illustrant le point d'intersection des trois plans. Utiliser la macro-commande `sphere` de la bibliothèque `plottools`. Ajuster la valeur du rayon afin que le tracé illustre correctement celui d'un point de l'espace. Produire un graphique montrant clairement la position relative de ces trois plans en utilisant judicieusement les options `orientation` et `view`.

c) Trouver des équations paramétriques des droites d'intersection des plans deux-à-deux.

d) Sur un même graphique, superposer seulement ces droites et une petite sphère illustrant le point d'intersection. Tracer ces droites avec des couleurs différentes mais de même épaisseur (`thickness=3`). Ajuster judicieusement la valeur du rayon de la sphère.

e) Finalement, dans un dernier graphique, superposer tous ces objets : plans, point et droites. Produire un graphique bien orienté afin de montrer clairement toutes les caractéristiques de vos tracés.

No. 2 Soit le plan $\Pi: 2x-3y+6z-4=0$ et la droite $\Delta: \dfrac{x-2}{4}=\dfrac{y+6}{-1}=\dfrac{z-4}{3}$.

a) La droite Δ coupant le plan Π, trouver les coordonnées du point de percée.

b) Dans un même graphique, illustrer les tracés du plan Π, de la droite Δ et du point de percée avec une petite sphère. Ajuster judicieusement la valeur du rayon de la sphère. Produire un graphique bien orienté afin de montrer clairement la position relative de la droite Δ et du plan Π.

c) Trouver l'angle de percée de la droite Δ et du plan Π. Convertir votre résultat en degrés. Consulter l'aide Maple de `convert`.

No. 3 Soit le point $P(6,3,-2)$ et la droite $\Delta : \dfrac{x-2}{5}=\dfrac{y+1}{2}=z-3$

a) Calculer la distance **exacte** (pas d'approximation décimale) du point P à la droite Δ.

b) Illustrer cette distance en superposant, dans un même graphique, les tracés
— de la droite Δ
— du segment abaissé perpendiculairement du point P à la droite Δ
— de deux petites sphères illustrant le point P ainsi que le point d'intersection de la droite et de la perpendiculaire. Ajuster évidemment la taille des deux sphères avec un même rayon.

Encore ici, produire un graphique bien orienté afin de montrer clairement toutes les caractéristiques de vos tracés.

Session Maple pour l'exercice RP-4

Initialisation

```
> restart;
> with(plots,display,setoptions,setoptions3d):
  setoptions(labels=[x,y],axesfont=[TIMES,ROMAN,14],
             labelfont=[TIMES,ROMAN,14]);
  setoptions3d(labels=[x,y,z],axis[3]=[color="Niagara 1"],
               tickmarks=[4,4,4],lightmodel=none,axesfont=[TIMES,ROMAN,12],
               labelfont=[TIMES,ROMAN,12]);
> with(Physics[Vectors]):
  with(LinearAlgebra);
```

Tracé d'une droite dans le plan et dans l'espace

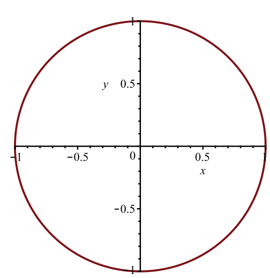
Rappelons la syntaxe paramétrique du tracé d'un lieu géométrique dans le plan cartésien :

`plot([x(t),y(t),t=a..b])`

L'abscisse $x(t)$ et l'ordonnée $y(t)$ de chaque point du lieu géométrique en cause sont obtenues en donnant différentes valeurs au paramètre t.

Par exemple, soit le lieu géométrique définit par $\mathrm{x}(t) = \cos(t)$ et $\mathrm{y}(t) = \sin(t)$. Le tracé qui sera obtenu est celui d'un cercle de rayon 1 centré à l'origine puisque les coordonnées de chaque point vérifient l'équation $x(t)^2 + y(t)^2 = 1$.

```
> plot([cos(t),sin(t),t=-Pi..Pi],scaling=constrained);
```

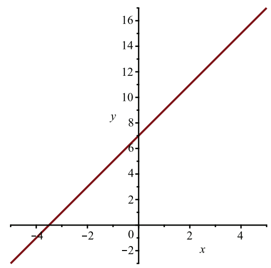


Pour utiliser la syntaxe paramétrique dans le cas d'une fonction définie par $y = \mathrm{f}(x)$, il suffit de considérer $\mathrm{x}(t) = t$. Donc, dans ce cas, nous pouvons poser $y(t) = \mathrm{f}(x(t)) = \mathrm{f}(t)$:

`plot([t,f(t),t=a..b])`

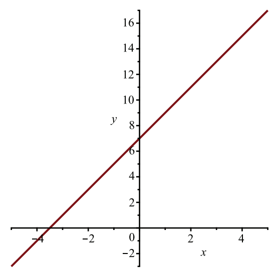
De sorte que, en considérant la fonction linéaire définie par $y = \mathrm{f}(x) = 2\,x + 7$ sur l'intervalle $[-5, 5]$, on formulera son tracé comme suit :

```
> plot([t,2*t+7,t=-5..5]);
```



Ce qui revient au même, pour Maple, en formulant à nouveau cette requête comme suit :

```
> plot([x,2*x+7,x=-5..5]);
```



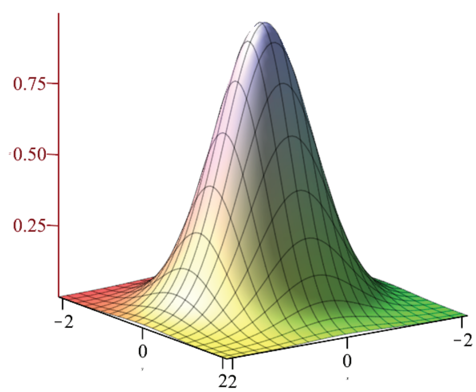
Pour réaliser le tracé d'un lieu géométrique dans l'espace, nous utiliserons la macro-commande `plot3d`. Nous allons également privilégier la syntaxe paramétrique de cette macro-commande :

```
plot3d([x(r,s),y(r,s),z(r,s)],r=a..b,s=c..d)
```

L'abscisse $x(r,s)$, l'ordonnée $y(r,s)$ et la cote $z(r,s)$ de chaque point du lieu géométrique de l'espace sont obtenues en donnant différentes valeurs aux paramètres r et s.

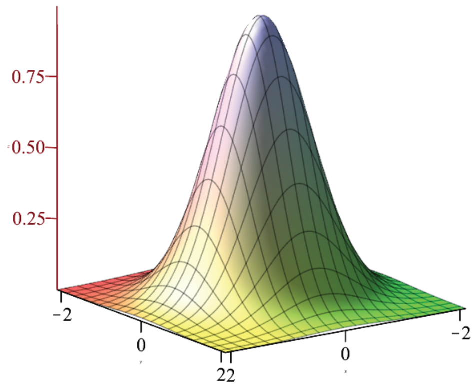
Par exemple, considérons le lieu géométrique définit par $\left\{\begin{array}{lcl} \mathrm{x}(r,\, s) & = & \cos(r) \\ \mathrm{y}(r,\, s) & = & \sin(s) \\ \mathrm{z}(r,\, s) & = & \mathbf{e}^{-r^2-s^2}, \quad r,\, s \in \mathbb{R} \end{array}\right.$

```
> plot3d([r,s,exp(-r^2-s^2)],r=-2.1..2.1,s=-2.1..2.1,
         grid=[60,60],axes=framed);
```



Ce qui revient au même, pour Maple, en formulant à nouveau cette requête comme suit :

```
> plot3d([x,y,exp(-x^2-y^2)],x=-2.1..2.1,y=-2.1..2.1,
         grid=[60,60],axes=framed);
```



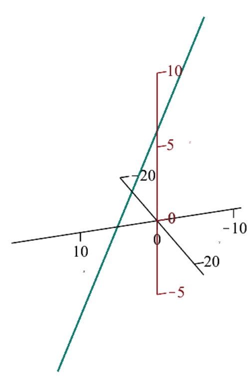
Par défaut, tout comme l'affichage 2D, l'affichage 3D est fait avec des axes de coordonnées qui ne sont pas dans le rapport 1 :1 (Échelle contrainte). C'est l'option `scaling=unconstrained` qui est validée par défaut. De plus, par défaut, le graphique 3D est affiché avec un système d'axes de style en boîte. Les types d'axes de coordonnées sont `boxed`, `normal`, `frame` ou `none`. Pour avoir un affichage avec un style d'axes autre que celui par défaut,il faudra spécifier explicitement, en options, `axes=type` dans la macro-commande plot3d.

Pour réaliser des tracés de droites dans l'espace, il suffit d'utiliser telles quelles leurs équations paramétriques.

Soit la droite d'équations paramétriques $\left\{\begin{array}{lcl} x & = & 9-2\,t \\ y & = & 10-3\,t \\ z & = & t, \qquad\quad t\in\mathbb{R}\end{array}\right.$

Bien que ce lieu géométrique (la droite) ne soit définit qu'à l'aide d'un seul paramètre, on doit quand même préciser, dans la macro-commande `plot3d`, un intervalle pour le paramètre muet. Dans un tel cas, la macro-commande `plot3d` considère les coefficients du paramètre muet dans les équations paramétriques comme étant tous nuls.

```
> plot3d([9-2*t,10-3*t,t],t=-5..10,s=-10..10,thickness=2,
         axes=normal,style=line,color="Niagara 6",
         orientation=[70,65]);
```



Nous avons modifié l'orientation par défaut de l'affichage (orientation=[45,45], en fait orientation=[45,45,0]) en spécifiant l'option `orientation=[70,65]` et imposer l'affichage des axes de coordonnées habituels en spécifiant l'option `axes=normal`. Si le troisième paramètre ψ n'est pas spécifié, le système le considère comme étant nul.

L'orientation $\theta = 70$, $\phi = 65$ et $\psi = 0$ a été obtenue en modifiant manuellement, avec la souris, l'orientation spatiale du graphique. Ces valeurs d'angles ont été lues dans la partie gauche de la barre contextuelle qui s'installe lorsque l'on clique sur le graphique 3D pour modifier son orientation.

Remarque : L'orientation par défaut du graphique 3D est $\theta = 45$ et $\phi = 45$. Il est possible, par contre, de modifier cette orientation par défaut durant une session Maple en utilisant la macro-commande `setoptions3d` de l'extension `plots`. Mais, le plus souvent, il faudra trouver manuellement une orientation mettant en évidence certaines caractéristiques désirées qui n'auront pas été misent automatiquement en évidence avec une orientation par défaut, quelle qu'elle soit. De façon pratique, il n'est donc pas utile de modifier l'orientation par défaut [45,45,0] avec la macro-commande `setoptions3d`.

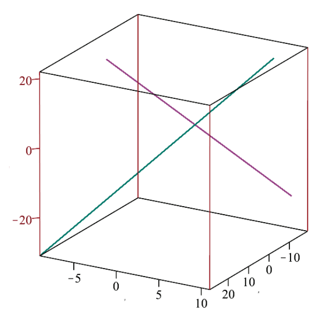
Il est, bien sûr, possible de superposer dans un même graphique les tracés de plusieurs droites dans l'espace.

Soit les droites d'équations paramétriques :

$$\Delta_1 : \left\{ \begin{array}{lcl} x & = & -1+2\,t \\ y & = & -1-6\,t \\ z & = & t, \qquad t \in \mathbb{R} \end{array} \right. \quad \text{et } \Delta_2 : \left\{ \begin{array}{lcl} x & = & -2+3\,t \\ y & = & 5-2\,t \\ z & = & -3+5\,t, \quad t \in \mathbb{R}. \end{array} \right.$$

```
> Delta_1:=plot3d([-2+3*t,5-2*t,-3+5*t],t=-3..5,s=-5..10,
          style=line,color="Niagara 14",thickness=2):
  Delta_2:=plot3d([-1+6*t,1-2*t,-1-6*t],t=-3..5,s=-6..6,
          style=line,color="Niagara 6",thickness=2):
```

```
> display({Delta_1,Delta_2},axes=boxed,orientation=[30,70,0]);
```



Le graphique précédent montre clairement que les deux droites ne sont pas parallèles. Mais, sont-elles concourantes ? Après avoir reproduit ces requêtes sur une feuille Maple, sélectionner le graphique avec un clic gauche maintenu enfoncé et modifier l'orientation spatiale de celui-ci. Que concluez-vous ? Droites concourantes ? Droites gauches ?

Vecteurs et norme euclidienne d'un vecteur

Les opérations qu'on effectuent sur des objets mathématiques dépendent de la nature de ces objets, c'est-à-dire de leur(s) type(s). Pour Maple, une liste et un vecteur sont deux objets différents. L'utilisation de crochets sert à créer une liste tandis que pour créer un vecteur on utilisera la macro-commande *Vector* de la bibliothèque de base.

```
> u:=[1,2,3];
  v:=Vector[row]([1,2,-3]);
```

$$u := [1, 2, 3]$$

$$v := \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

Quoique l'affichage des deux objets précédents « u » et « v » est presque le même : remarquez l'absence de virgules dans l'objet v. Seul « v » est reconnu par MAPLE comme un vecteur. En effet,

```
> type(u,Vector);
  type(v,Vector);
```

$$false$$

$$true$$

Pour créer un vecteur dont le type est compatible avec les macro-commandes de l'extension `LinearAlgebra`, on utilisera la macro-commande `Vector` de la bibliothèque de base. Notons, par défaut, que c'est un vecteur colonne qui est créé avec la macro-commande `Vector`. Pour créer un vecteur ligne, il faut spécifier l'option « row » : `Vector[row]`.

Il est bien sûr possible d'utiliser la syntaxe « $< \ >$ » pour la création de vecteurs compatibles avec les macro-commandes de l'extension `LinearAlgebra`.

```
> u:=<1|2|3>;      # vecteur ligne
  v:=<1,2,-3>;     # vecteur colonne
```

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

$$u := \begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix}$$

On obtient la norme d'un vecteur d'un espace euclidien orthonormé à l'aide de la macro-commande `VectorNorm` de l'extension `LinearAlgebra`. Dans ce cas, la norme du vecteur correspond à sa longueur.

```
> '||v||'=VectorNorm(v,Euclidean);
```

$$||v|| = \sqrt{14}$$

Avec la macro-commande `VectorNorm`, il existe plusieurs manières de calculer la norme d'un vecteur. Ici, en spécifiant l'option *Euclidean*, on demande alors simplement à MAPLE de calculer la racine carrée de la somme des carrés des composantes du vecteur, soit la norme euclidienne.

Tracé dans l'espace d'un segment de droite reliant deux points

Pour tracer un segment de droite entre deux points A et B de l'espace, il faudra d'abord obtenir des équations paramétriques de la droite passant par ces deux points. Utilisons le vecteur $\overrightarrow{AB}$ comme vecteur directeur et le vecteur $\overrightarrow{OA}$ pour obtenir des équations paramétriques.

Ainsi, avec l'équation vectorielle $(x, y, z) = \overrightarrow{OA} + t\overrightarrow{AB}$, il suffira alors de faire varier le paramètre $t \in [0,1]$ pour tracer un segment de droite reliant les points A et B.

Par exemple, traçons le segment de droite reliant les points A(1,1,1) et B(-1,3,0).

```
> A:=[1,1,1];  # A est un point (type liste)
  B:=[-1,3,0]; # B est un point (type liste)
```

$$A := [1,\ 1,\ 1]$$

$$B := [-1,\ 3,\ 0]$$

Créons les vecteurs $\overrightarrow{AB}$ et $\overrightarrow{OA}$.

```
> AB:=<B[1]-A[1],B[2]-A[2],B[3]-A[3]>;
  OA:=convert(A,Vector);
```

$$AB := \begin{bmatrix} -2 \\ 2 \\ -1 \end{bmatrix}$$

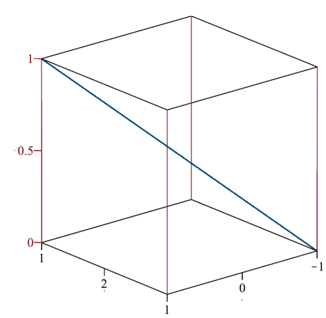
$$OA := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Assignons le calcul $\overrightarrow{OA} + t\overrightarrow{AB}$ à la variable «Droite» et traçons ce lieu.

```
> Droite:=OA+t*AB;
```

$$Lieu := \begin{bmatrix} 1-2\,t \\ 1+2\,t \\ 1-t \end{bmatrix}$$

```
> plot3d(Droite,t=0..1,s=-5..5,thickness=2,style=line,
         color="Niagara 8",orientation=[50,70,0]);
```



La macro-commande `line` de l'extension `plottools` est utile pour le tracé d'un segment de droite reliant deux points de l'espace. Mais, dans le cadre de ce RP, son utilisation n'est pas autorisée.

Comme second exemple, illustrons un segment de droite reliant deux points quelconques des deux droites Δ_1 et Δ_2 de la section précédente.

Par exemple, soit les points A(-1,1,-1) et B(-2,5,-3) appartenant respectivement aux droites Δ_1 et Δ_2.

```
> A:=[-1, 1, -1];
  B:=[-2, 5, -3];
```

$$A := [-1,\ 1,\ -1]$$

$$B := [-2,\ 5,\ -3]$$

```
> AB:=<B[1]-A[1]|B[2]-A[2]|B[3]-A[3]>;
  OA:=convert(A,Vector[row]);
```

$$AB := \begin{bmatrix} -1 & 4 & -2 \end{bmatrix}$$

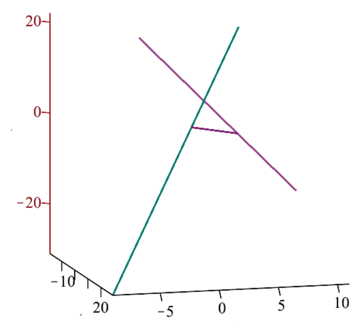
$$OA := \begin{bmatrix} -1 & 1 & -1 \end{bmatrix}$$

```
> Droite:=OA+t*AB;
```

$$Lieu := [-1-t,\ 1+4\,t,\ -1-2\,t]$$

```
> Segment:=plot3d(Droite,t=0..1,s=-5..5,thickness=2,
          style=line,color="Niagara 5"):
```

```
> display(Delta_1,Delta_2,Segment,axes=framed,
          orientation=[-15,80,0],viewpoint="circleright");
```



L'option `viewpoint="circleright"` permet de créer une animation du graphique 3D. Cela va nous permettre de mieux voir les caractéristiques du graphique 3D. Cliquer sur le graphique puis lancer l'animation avec le bouton Démarrer de la barre du menu contextuelle.

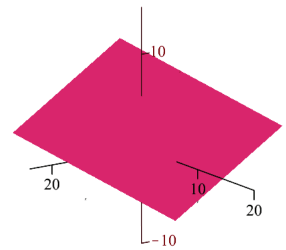
Tracé d'un plan de l'espace

Pour tracer un plan dans l'espace, on utilisera, bien sûr, la macro-commande `plot3d`. Si le plan à tracer est donné avec des équations paramétriques, on utilisera ces équations qui définissent directement l'abscisse $x(r,s)$, l'ordonnée $y(r,s)$ et la cote $z=(r,s)$ de chaque point du plan.

Par exemple, soit le plan Π : $\begin{cases} x \ = \ 1-3\,r-7s \\ y \ = \ -2+6\,s \\ z \ = \ 7+r-3\,s, \quad r,s \ \in \mathbb{R} \end{cases}$

Chaque paire de valeurs assignées aux paramètres r et s déterminent un point (x,y,z) de l'espace. L'ensemble de ces points constitueront une surface de l'espace. Dans ce cas-ci, cette surface est un plan.

```
> plot3d([1-3*r-7*s,-2+6*s,7+r-3*s],r=-6..2,s=-2..2,
          style=patch,grid=[2,2],
          axes=normal,color="Bright 15",
          view=[-20..25,-5..20,-10..15]);
```



La macro-commande `plot3d` autorise plusieurs options qui permettent de contrôler l'affichage du tracé des surfaces de l'espace. Pour avoir une idée de ce que l'utilisateur peut contrôler, consulter l'aide Maple en tapant la requête «`?plot3d[options]`». Puisque le style par défaut `style=patch` inclut un quadrillage de la surface, remarquer, dans l'exemple précédent, l'utilisation de l'option `grid` qui a été utilisée pour «régulariser» le quadrillage du tracé. Si, pour un tracé d'une surface, le quadrillage n'est pas requis, on précisera l'option `style=patchnogrid` (ou bien style=surface).

Lorsque le plan à tracer est donné avec une équation cartésienne, il suffira de considérer deux des variables libres et de lier la troisième à celles-ci afin de déduire des équations paramétriques du plan.

Superposons dans un même graphique, les tracés des deux plans Π_1 et Π_2 d'équations cartésiennes

$$\Pi_1 : 3\,x + 2\,y + 4\,z - 11 = 0$$
$$\Pi_2 : 2\,x + y - 3\,z - 1 = 0$$

Désignons deux des trois variables de l'équation comme variables libres et l'autre comme variable liée à celles-ci. Choisissons ici les variables x et y comme variables libres dans chaque équation. Ainsi, c'est donc z qui sera la variable liée.

Il s'agit ensuite, d'isoler la variable liée. Avec les équations données ici, il est facile d'isoler mentalement z. De cette manière, les équations suivantes sont des équations paramétriques des deux plans :

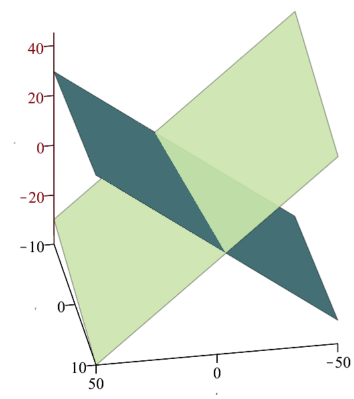
$$\Pi_1 : \begin{cases} x = r \\ y = s \\ z = \dfrac{3\,r + 2\,s - 11}{-4} \end{cases} \quad r, s \in \mathbb{R} \qquad \text{et } \Pi_2 : \begin{cases} x = r \\ y = s \\ z = \dfrac{2\,r + s - 1}{3} \end{cases} \quad r, s \in \mathbb{R}.$$

```
> PI_1:=plot3d([r,s,(3*r+2*s-11)/(-4)],r=-50..50,s=-10..10,
        grid=[2,2],style=patch,color="Nautical 4",transparency=0.1):
  PI_2:=plot3d([r,s,(2*r+s-1)/3],r=-50..50,s=-10..10,
        grid=[2,2],style=patch,color="Nautical 12"):
```

Sans manquer de rigueur mathématique, on pourrait très bien évoquer directement les variables x et y dans la macro-commande `plot3d` en lieu et place des paramètres r et s. Pour Maple, il n'y a aucun problème pourvu que les intervalles de traçage correspondent aux choix des noms x et y. Ainsi, formulons de nouveau les requêtes précédentes.

```
> PI_1:=plot3d([x,y,(3*x+2*y-11)/(-4)],x=-50..50,y=-10..10,
        grid=[2,2],style=patch,color="Nautical 4",transparency=0.1):
  PI_2:=plot3d([x,y,(2*x+y-1)/3],x=-50..50,y=-10..10,
        grid=[2,2],style=patch,color="Nautical 12"):
```

```
> display({PI_1,PI_2},axes=framed,orientation=[80,60,0]);
```



Puisque ces deux plans sont sécants, voyons comment obtenir l'équation de la droite d'intersection puis de superposer son tracé à ceux des deux plans.

```
> Éq_1:=3*x+2*y+4*z-11=0;
  Éq_2:=2*x+y-3*z-1=0;
```

$$\acute{E}q\_1 := 3\,x + 2\,y + 4\,z - 11 = 0$$

$$\acute{E}q\_2 := 2\,x + y - 3\,z - 1 = 0$$

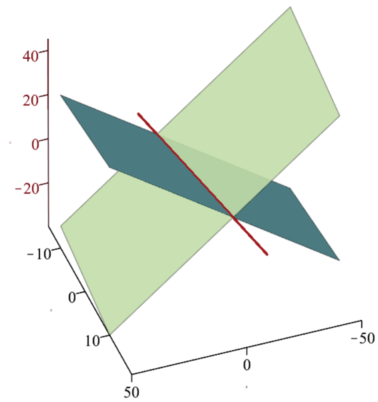
```
> Solution:=solve({Éq_1,Éq_2}) assuming z::real;
```

$$Solution := \{x = -9 + 10\,z,\; y = 19 - 17\,z,\; z = z\}$$

En accord avec le résultat précédent, désignons z comme variable libre.

```
> Droite_inter:==plot3d([-9+10*t,19-17*t,t],t=0..2,s=-2..2,
                  thickness=3,style=line,color="Bright 10"):
```

```
> display([PI_1,PI_2,Droite_inter],axes=framed,orientation=[70,50,0]);
```



Tracé d'un faisceau de plans

Illustrons quelques plans du faisceau de plans engendrés par les plans d'équations

$$\Pi_{g1} : 3\,x - y + 2\,z - 5 = 0$$
$$\Pi_{g2} : 2\,x + y - z + 4 = 0$$

```
> PI_g1:=3*x-y+2*z-5=0;
  PI_g2:=2*x+y-z+4=0;
```

$$PI\_g1 := 3\,x - y + 2\,z - 5 = 0$$

$$PI\_g2 := 2\,x + y - z + 4 = 0$$

Par commodité, créons une fonction de deux variables $k1$ et $k2$ qui servira à générer des équations cartésiennes de plans de ce faisceau.

```
> Faisceau:=(k1,k2)->k1*PI_g1+k2*PI_g2;
```

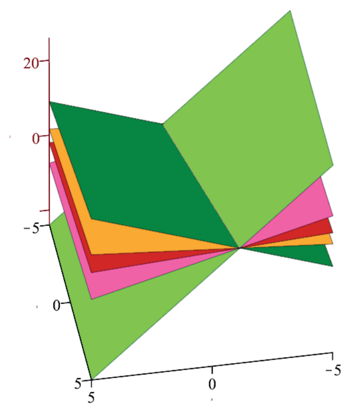
$$Faisceau := (k1,\, k2) \rightarrow k1\; PI\_g1 + k2\; PI\_g2$$

Obtenons maintenant les tracés de cinq plans quelconques de ce faisceau. Désignons z comme variable liée. Ainsi, pour chaque plan du faisceau, nous devrons isoler z par calcul et ce calcul sera poser directement dans la macro-commande `plot3d` pour lier la valeur de la cote aux valeurs de x et de y.

```
> PI_1:=plot3d([x,y,solve(Faisceau(1,0),z)],x=-5..5,y=-5..5,grid=[2,2],style=patch,
              color="Bright 15",thickness=0)assuming z::real:
  PI_2:=plot3d([x,y,solve(Faisceau(0,1),z)],x=-5..5,y=-5..5,grid=[2,2],style=patch,
              color="Bright 3",thickness=0)assuming z::real:
  PI_3:=plot3d([x,y,solve(Faisceau(1,1),z)],x=-5..5,y=-5..5,grid=[2,2],style=patch,
              color="Bright 16",thickness=0)assuming z::real:
  PI_4:=plot3d([x,y,solve(Faisceau(-1,1),z)],x=-5..5,y=-5..5,grid=[2,2],style=patch,
              color="Bright 10",thickness=0)assuming z::real:
  PI_5:=plot3d([x,y,solve(Faisceau(2,-5),z)],x=-5..5,y=-5..5,grid=[2,2],style=patch,
              color="Bright 6",thickness=0)assuming z::real:
```

Superposons ces tracés dans un même graphique.

```
> display({seq(PI_||(1..5)},axes=framed,orientation=[80,50,0]);
```



Obtenons ensuite l'équation de l'arête de ce faisceau.

```
> Solution:=solve({PI_g1,PI_g2})  assuming z::real;
```

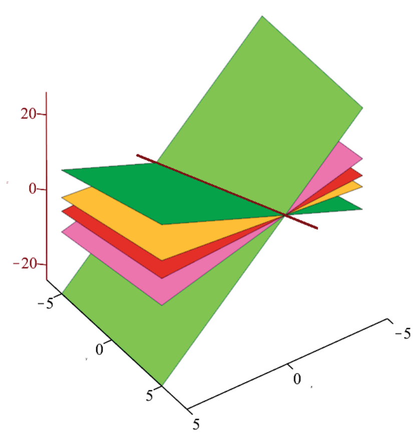
$$Solution := \{x = x,\ y = -7\,x - 3,\ z = -5\,x + 1\}$$

En accord avec la solution donnée, désignons la variable x comme étant la variable libre.

```
> Arête:=plot3d([t,-7*t-3,-5*t+1],t=-1.5..0.5,s=-5..10,
                 thickness=3,style=line,color="Niagara 1"):
```

Finalement, dans un même graphique, superposons l'arête à ces cinq plans.

```
> display({seq(PI_||(1..5),Arête},axes=framed,orientation=[55,50,0]);
```



Multiplication scalaire et vectorielle

Dans une démarche de résolution de problèmes, il peut être nécessaire, par exemples, de trouver un vecteur normal à un plan pour lequel on connaît un point et les vecteurs directeurs ou encore de trouver les composantes du vecteur projection orthogonale $\text{proj}_{\vec{v}}(\vec{u})$. Un vecteur normal au plan sera obtenu avec la multiplication vectorielle de deux vecteurs directeurs de ce plan.

Pour effectuer la multiplication vectorielle, nous pouvons utiliser la macro-commande `CrossProduct` et pour la multiplication scalaire, nous pouvons utiliser la macro-commande `Dotproduct`. Ces deux macro-commandes appartiennent à l'extension `LinearAlgebra`.

Dans l'initialisation, nous avons chargé les macro-commandes de la sous-bibliothèque `Physics[Vectors]`. Cela nous permets de bien documenter la zone des résultats. Il faut charger d'abord Physics[Vectors] puis `LinearAlgebra` afin de ne pas avoir un conflict avec la forme *infixée* `&x` (multiplication vectorielle) de chacune de ces bibliothèques.

Obtenons le produit vectoriel des vecteurs $\vec{u} = (2, -1, 3)$ et $\vec{v} = (6, 2, 7)$.

```
> u_:=<2|-1|3>;
  v_:=<6|2|7>;
```

$$\vec{u} := \begin{bmatrix} 2 & -1 & 3 \end{bmatrix}$$

$$\vec{v} := \begin{bmatrix} 6 & 2 & 7 \end{bmatrix}$$

```
> w_ := u_ &x v_;
```

$$\vec{w} = \begin{bmatrix} -13 & 4 & 10 \end{bmatrix}$$

Vérifions si, effectivement, le vecteur $\vec{w}$ est bien perpendiculaire aux deux vecteurs $\vec{u}$ et $\vec{v}$.

```
> 'u_ . w_' =u_ . w_;
  'v_ . w_' =v_ . w_;
```

$$\vec{u} \bullet \vec{w} = 0$$

$$\vec{v} \bullet \vec{w} = 0$$

Autre exemple de calcul. Obtenons les composantes du vecteur projection orthogonale de $\vec{u}$ sur $\vec{v}$, soit $\text{proj}_{\vec{v}}(\vec{u}) = \dfrac{\vec{u} \cdot \vec{v}}{\vec{v} \cdot \vec{v}}\,\vec{v}$

```
>'proj[v_](u_)'=(u_.v_)/(v_.v_) * v_
```

$$proj_{\vec{v}}(\vec{u}) = \begin{bmatrix} \frac{186}{89} & \frac{62}{89} & \frac{217}{89} \end{bmatrix}$$

Comme dernier exemple, déterminons si les points A(6,3,3), B(-2,1,1), C(1,-1,0) et D(3,-2,-1) sont coplanaires. Appliquons le critère de coplanarité de trois vecteurs de l'espace. Calculons le produit mixte des vecteurs $\overrightarrow{AB}$, $\overrightarrow{AC}$ et $\overrightarrow{AD}$.

Posons directement les vecteurs lignes $\overrightarrow{AB}$, $\overrightarrow{AC}$ et $\overrightarrow{AD}$.

```
> AB_:=<-8|-2|-2>;
  AC_:=<-5|-4|-3>;
  AD_:=<-3|-5|-4>;
```

$$\overrightarrow{AB} := \begin{bmatrix} -8 & -2 & -2 \end{bmatrix}$$

$$\overrightarrow{AC} := \begin{bmatrix} -5 & -4 & -3 \end{bmatrix}$$

$$\overrightarrow{AD} := \begin{bmatrix} -3 & -5 & -4 \end{bmatrix}$$

Appliquons le critère de coplanarité, calculons le produit mixte $\overrightarrow{AB} \cdot \overrightarrow{AC} \times \overrightarrow{AD}$.

```
> 'AB_ . AC_ &x AD_'=AB_ . AC_ &x AD_;
```

$$\overrightarrow{AB} \bullet \left(\&x\left(\overrightarrow{AC}, \overrightarrow{AD}\right)\right) = -12$$

Puisque le produit mixte n'est pas nul, cela montre que les vecteurs $\overrightarrow{AB}$, $\overrightarrow{AC}$ et $\overrightarrow{AD}$ ne sont pas coplanaires, et donc que les points A, B, C et D ne le sont pas non plus.