Calculs géométriques 3D sous matlab

D. Legland

24 juin 2009

Résumé

Au cours de mes travaux de recherches j'ai souvent eu à travailler avec des objets géométriques dans le plan ou dans l'espace. Au fil des besoins, j'ai implémenté un certain nombre de fonctions que j'ai essayé de rendre accessibles. Je développe ici les fonctions pour des primitives géométriques en 3D, un autre document fait référence aux calculs pour les figures du plan.

Table des matières

1 Introduction et conventions

Ce document présente une collection de fonctions écrites pour des calculs géométriques en 3 dimensions, telles que les coordonnées du point d'intersection d'un plan avec une droite, ou des tests de parallélisme de vecteurs. Cette bibliothèque fait suite et est complémentaire de son équivalent pour le 2D.

Le principe est de cacher le plus possible la représentation interne des données, et le détail des calculs à l'utilisateur. Le but est de faciliter la mise au point des programmes, et la compréhension des sources. Idéalement, outre l'initialisation des données, on ne devrait pas manipuler directement les valeurs numériques, mais uniquement des résultats de fonctions géométriques.

La suite de l'introduction détaille les conventions utilisées, puis les différentes fonctions sont présentées, classées en fonctions des primitives sur lesquelles elles opèrent.

1.1 Système de coordonnées

On se place dans un repère cartésien orthonormé $(O, \overrightarrow{i}, \overrightarrow{j}, \overrightarrow{k})$. On utilise aussi Ox, Oy et Oz pour identifier les 3 axes principaux.

Les unités de distances ne sont pas spécifiées.

L'unité d'angle est le radian. On considère des angles compris entre 0 et 2π.

1.2 Coordonnées sphériques

Pour les angles 3D, on considère d'une part la colatitude, notée θ, d'autre part la longitude/l'azimut, notée φ. La colatitude varie entre 0 (vers le nord) et π (vers le sud), et vaut $\pi/2$ pour les directions horizontales. La longitude varie entre 0 (direction de l'axe Ox) et 2π, la valeur $\pi/2$ correspondant à la direction de l'axe Oy.

Les coordonnées sphériques en 3D sont constituées d'un angle 3D, et de la distance à l'origine, notée ρ. Elles sont notées dans un vecteur ligne :

$$[\theta, \varphi, \rho]$$

On peut aussi considérer un angle de rotation autour de la normale, noté ψ. Cele permet de représenter toutes les rotations possibles d'un objet autour de l'origine :

$$[\theta, \varphi, \psi]$$

1.3 Conventions de programmation

1.3.1 Noms des fonctions

Les noms de fonction sont nommés selon une convention 'à la Java' : les mots sont collés, écrits en minuscules, la première lettre de chaque mot est en majuscule sauf le premier mot. Exemple : « nomDeLaFonction ».

Le nom de chaque fonction est choisi en fonction de l'opération effectuée, et du ou des types de primitive sur lequele elle opère. En général, les noms des fonctions commencent par un verbe :

```
p2 = transformPoint3d(p1, trans);
l1 = intersectPlanes(plane1, plane2);
```

1.3.2 Vectorisation

Dans les cas où la fonction accepte un couple d'arguments (par exemple, pour tester si des vecteurs sont parallèles), j'essaie dans la mesure du possible de « vectoriser » la fonction. Ainsi, si le premier argument est un vecteur, le deuxième un tableau de vecteur, alors la fonction sera appelée pour chaque couple de vecteurs.

2 Objets utilitaires

2.1 Boîte

La boîte est un rectangle dont les bords sont parallèles aux axes considérés. Cette structure de donnée est surtout utilisée pour définir les coordonnées minimales et maximales d'une primitive, ou bien pour sélectionner les primitives dans une zone donnée.

Une boîte est définie par les 6 valeurs [xmin, xmax, ymin, ymax, zmin, zmax].

2.2 Vecteurs

Un vecteur 3d est représenté par le triplet des ses coordonnées $[v_x, v_y, v_z]$. Exemple :

```
v1 = [10 20 30]; % declare un vecteur
vs = [10 20 30;20 30 10;30 10 20]; % déclare une liste de 3 vecteurs
```

2.3 Opérations sur les vecteurs

2.3.1 vectorNorm3d

Calcule la norme d'un vecteur.

entrée les coordonnées du vecteur, ou un groupe de coordonnées.

sortie la norme du vecteur, ou un ensemble de normes.

2.3.2 normalizeVector3d

Calcule le vecteur normalisé, c'est à dire tel que sa norme vaille 1.

entrée un vecteur ou un ensemble de vecteurs.

sortie les vecteurs normalisés, avec une norme égale à 1.

2.3.3 transformVector3d

Transforme un vecteur par une transformation affine.

entrée la représentation d'un vecteur, la matrice d'une transformation affine

sortie la représentation du vecteur transformé

2.3.4 isParallel3d

Teste si deux vecteurs 3d sont parallèles.

entrée les coordonnées de deux vecteurs ou groupes de vecteurs.

sortie la valeur 1 pour chaque couple de vecteurs parallèles.

2.3.5 isPerpendicular3d

Teste si deux vecteurs 3d sont perpendiculaires.

entrée les coordonnées de deux vecteurs ou groupes de vecteurs.

sortie la valeur 1 pour chaque couple de vecteurs perpendiculaires.

3 Points

Un point est représenté par le triplet de ses coordonnées cartésiennes. Un groupe de n point est donné par un tableau $[n \times 3]$.

3.1 Opérations sur les points

3.1.1 isCoplanar

Teste si un groupe de points est coplanaire. Pour tenir compte des erreurs numériques, la tolérance peut être spécifiée.

entrée les coordonnées de chaque point, éventuellement la tolérance.

sortie la valeur 1 si les points sont coplanaires, 0 sinon.

3.1.2 distancePoints3d

Calcule la distance entre des points. La métrique euclidienne est utilisée par défaut, mais d'autres métriques peuvent être spécifiées.

entrée les coordonnées du premier point, les coordonnées du deuxième point, éventuellement la métrique (euclidienne par défaut).

sortie la distance entre les points.

3.1.3 transformPoint3d

Calcule les coordonnées de points transformés par une transformation affine.

entrée les coordonnées des points, la matrice de transformation.

sortie les coordonnées des points transformés.

3.2 Conversions de coordonnées

3.2.1 cart2sph2

Convertit les coordonnées cartésiennes 3D en coordonnées sphériques. La représentation des coordonnées sphériques est différente de celle de Matlab.

entrée les coordonnées cartésiennes.

sortie les coordonnées sphériques.

3.2.2 sph2cart2

Convertit les coordonnées sphériques en coordonnées cartésiennes $[x, y, z]$.

entrée les coordonnées sphériques.

sortie les coordonnées cartésiennes.

3.2.3 cart2cyl

Convertit les coordonnées cartésiennes $[x, y, z]$ en coordonnées cylindriques : $[\varphi, r, z]$.

entrée les coordonnées cartésiennes.

sortie les coordonnées cylindriques.

3.2.4 cyl2cart

Convertit les coordonnés cylindriques $[\varphi, r, z]$ en coordonnées cartésiennes $[x, y, z]$.

entrée les coordonnées cylindriques.

sortie les coordonées cartésiennes.

3.3 Calculs sur les angles

3.3.1 anglePoints3d

Calcule l'angle entre deux points, ou plutôt entre 2 vecteurs. Le résultat est compris entre 0 et π.

entrée les coordonnées de deux vecteurs ou de deux points.

sortie l'angle entre les deux vecteurs.

3.3.2 sphericalAngle

Calcule l'angle de 3 points à la surface d'une sphère.

entrée les coordonnées cartésiennes de 3 points A, B et C.

sortie l'angle sphérique $(\overrightarrow{BA}; \overrightarrow{BC})$.

3.3.3 angleSort3d

Trie des points coplanaires en fonction de leur angle par rapport à l'origine.

entrée une liste de points, et éventuellement le point de référence pour le calcul des angles.

sortie les points triés en fonction de l'angle par rapport au point de référence.

3.3.4 randomAngle3d

Renvoie un angle 3D uniformément distribué sur la sphère. La longitude est choisie de manière uniforme entre 0 et 2π, la colatitude est choisie entre 0 et π, selon une distribution en sinus.

sortie un angle 3d aléatoire.

4 Droites, segments de droite et rayons

4.1 Représentations

Une droite est représentée par une origine, et un vecteur directeur :

$$[x_0, y_0, z_0, d_x, d_y, d_z]$$

On considère parfois aussi des segments de droites en 3D. Ils sont représentés par les coordonnées du premier point suivies des coordonnées du deuxième point :

$$[x_1, y_1, z_1, x_2, y_2, z_2]$$

4.2 Création

4.2.1 createLine3d

Crée une droite 3D à partir de deux points. Si plusieurs couples de points sont fournis, plusieurs droites sont créées.

entrée les coordonnées de chaque point.

sortie la représentation d'une droite.

4.3 Fonctions spécifiques

4.3.1 distancePointLine3d

Calcule la distance entre un point et une droite en 3D.

entrée les coordonnées du point, les coordonnées de la droite.

sortie la distance entre le point et la droite.

4.3.2 clipLine3d

Sélectionne la partie d'une droite visible dans une boîte, et renvoie le segment de droite 3D correspondant.

entrée la représentation d'une droite, les limites de la boîte

sortie la représentation d'un segment de droite

4.3.3 transformLine3d

Transforme une droite par une transformation affine.

entrée la représentation d'une droite, la matrice d'une transformation affine

sortie la représentation de la droite transformée

4.3.4 linePosition3d

Calcule la position d'un point sur une droite en 3D, c'est à dire le paramètre t tel que les coordonnées du point vérifient

$$\begin{aligned} x_p &= x_0 + td_x \\ y_p &= y_0 + td_y \\ z_p &= z_0 + td_z \end{aligned}$$

entrée les coordonnées du point, les coordonnées de la droite.

sortie le paramètre t représentant la position sur la droite.

5 Plans

Un plan P est représenté par une origine et deux vecteur directeurs :

$$[x_0, y_0, z_0, d_{x1}, d_{y1}, d_{z1}, d_{x2}, d_{y2}, d_{z2}]$$

On considère que la représentation d'un plan est normalisée si les deux vecteurs directeurs sont orthogonaux, et que leurs normes sont égales à l'unité.

5.1 Création

5.1.1 createPlane

Crée un plan à partir de 3 points.

entrée les coordonnées de 3 points.

sortie la représentation normalisée du plan contenant ces trois points.

La fonction permet aussi de créer un plan à partir d'une origione et de l'angle normal du plan.

5.1.2 medianPlane

Calcule l'angle médian entre 2 points.

entrée les coordonnées de deux points.

sortie une représentation du plan contenant tous les points équidistants des deux points donnés en argument.

5.1.3 normalizePlane

Normalise la représentation d'un plan.

entrée la représentation d'un plan.

sortie une représentation normalisée du plan (vecteurs directeurs orthogonaux et normalisés).

5.2 Mesure sur plans

5.2.1 planeNormal

Calcule le vecteur normal d'un plan.

entrée la représentation d'un plan.

sortie le vecteur normal du plan. Si le plan est normalisé, la norme du vecteur normal vaut 1.

5.2.2 dihedralAngle

Calcule l'angle dihédral formé par deux plans.

entrée les représentations de deux plans.

sortie l'angle dihédral entre les deux plans, donné entre 0 et π.

5.3 Interactions avec des points

5.3.1 planePosition

Calcule la position d'un point sur un plan. Les vecteurs directeurs du plan doivent être orthogonaux.

entrée les coordonnées d'un point, la représentation du plan.

sortie les coordonnées du point dans le système de coordonnées du plan.

5.3.2 isBelowPlane

Teste la position d'un point par rapport à un plan. On dit qu'un point est sous un plan si le produit cartésien du vecteur liant l'origine du plan au point considérer par le vecteur normal du plan est négatif.

entrée les coordonnées du point, la représentation du plan.

sortie la valeur 1 si le point est sous le plan.

5.3.3 projPointOnPlane

Calcule les coordonnées 3D de la projection d'un point sur un plan.

entrée les coordonnées du point, la représentation du plan.

sortie les coordonnées 3D du point projeté.

5.3.4 distancePointPlane

Calcule la distance d'un point à un plan.

entrée les coordonnées du point, la représentation du plan.

sortie la distance euclidienne entre le point et le plan.

5.4 Interactions avec des droites

5.4.1 intersectPlanes

Calcule une représentation de la droite située à l'intersection de deux plans.

entrée la représentation de deux plans.

sortie la représentation de la droite commune aux deux plans. Si les plans sont parallèles, renvoie la valeur [NaN, NaN, NaN, NaN, NaN, NaN].

5.4.2 intersectLinePlane

Calcule le point d'intersection entre une droite et un plan.

entrée la paramétrisation de la droite ainsi que celle du plan.

sortie les coordonnées (3D) du point d'intersection. Si la droite est parallèle au plan, renvoie [NaN, NaN, NaN].

5.4.3 intersectEdgePlane

Calcule le point d'intersection entre un segment de droite et un plan.

entrée la paramétrisation du segment ainsi que celle du plan.

sortie les coordonnées (3D) du point d'intersection. Si le segment est parallèle au plan, renvoie [NaN, NaN, NaN].

6 Polygones et polyèdres

6.1 Polygones

Un polygone de n côtés en 3D est représenté par les coordonnées de ses sommets, dans un tableau $[n \times 3]$. On suppose que les sommets sont coplanaires.

6.1.1 polygonCentroid3d

Calcule le centrïde (centre de gravité) d'un polygone 3D.

entrée les coordonnées des sommets du polygone

sortie le centroïde du polygone

6.1.2 polygon3dNormalAngle

Calcule l'angle normal en 3D d'un polygone.

entrée les coordonnées des sommets du polygone, l'indice du ou des sommets pour lesquels on veut calculer l'angle normal.

sortie l'angle normal des sommets du polygone. La somme des angles normaux de tous les sommets d'un polygone connexe vaut 4π.

6.1.3 clipConvexPolygon3dPlane

Découpe un polygone 3D convexe par un plan. La fonction ne garde que les sommets situés sous le plan, et ajoute les points d'intersection des arêtes avec le plan.

entrée les coordonnées des sommets du polygone, la représentation du plan.

sortie les sommets du polygone découpé par le plan.

6.2 Représentation des polyèdres

Les polyèdres sont représentés à l'aide de deux structures de données. Un premier tableau contient les coordonnées des sommets du polyèdre. Un deuxième tableau contient les indices des sommets de chaque face. Dans le cas d'un polyèdre où toutes les faces ont le même nombre de sommets, cela peut être un tableau avec autant de lignes de que de faces, et autant de colonnes que le

nombre de sommets par face. Sinon, on utilise un tableau de cellules, et chaque cellule contient un tableau d'indice, donné par un vecteur ligne.

Certaines fonctions utilisent aussi les arêtes du polyèdre. Les arêtes sont représentées par un tableau à deux colonnes, contenant les indices des sommets origine et destination de chaque arête.

6.3 Polyèdres réguliers

La plupart des fonctions suivantes renvoient un polyèdre type, sans pouvoir spécifier d'option sur la position ou la taille du polyèdre. Pour manipuler d'autres positions ou d'autres tailles, il suffit de transformer les coordonnées des sommets par la transformation affine appropriée (voir la section 8).

6.3.1 createCube

Crée un cube unité, avec un coin situé en $[0, 0, 0]$ et le coin opposé situé en $[1, 1, 1]$.

6.3.2 createCubeOctahedron

Crée un cube-octaèdre, qui peut être vu comme un carré auquel on enlève les coins, ou comme un octaèdre auquel on enlève les coins. Le polèdre a 6 faces carrées et 8 faces triangulaire.

6.3.3 createIcosahedron

Crée un icosaèdre, figure constituée de 20 faces triangulaires. Le polyèdre dual est le dodécaèdre.

6.3.4 createOctahedron

Crée un octaèdre, composé de 8 faces triangulaires.

6.3.5 createRhombododecahedron

Crée un rhombododécaèdre, constitué de 12 faces en forme de losange. Ce polyèdre peut être utilisé pour paver l'espace. Il correspond à un système de voisinage selon 12 voisins, dirigés vers les milieux des arêtes d'un cube unité.

6.3.6 createTetraedron

Crée un tétraèdre, une pyramide régulière à base triangulaire.

6.3.7 createTetrakaidecahedron

Crée un tétrakaidécaèdre (il a peut-être d'autres noms moins barbares...). Il est constitué de 8 faces hexagonales et 6 faces carrées. Il permet de paver l'espace, et correspond à un voisinage selon le 14 plus proches voisins.

6.3.8 createSoccerBall

Crée un ballon de football. Ce polyèdre, qui comprend 12 faces pentagonales et 20 faces hexagonales, correspond à un icosaèdre tronqué ou à un dodécaèdre tronqué.

6.4 Création de polyèdres

6.4.1 minConvexHull

Calcule l'enveloppe convexe 3D d'une série de points, et formate le résultat de manière à fusionner les faces coplanaires. Pour des enveloppes convexes de points situés sur une grille, cela permet de visualiser le résultat de manière beaucoup plus parlante.

6.4.2 steinerPolytope

Calcule le polyèdre de Steiner d'un ensemble de points. Il est obtenu appliquant des dilatations successives par des segments d'orientations différentes. Les segments utilisés sont les segments reliant l'origine aux points passés en paramètre.

entrée un ensemble de points, ou de vecteurs.

sortie le polyèdre de Steiner résultant.

6.5 Mesures sur des polyèdres

6.5.1 faceNormal

Calcule le vecteur normal d'une face d'un polyèdre.

entrée les coordonnées des sommets, les indices des sommets de chaque face (soit dans un tableau, soit dans une liste de cellules).

sortie le vecteur normal de chaque face.

6.5.2 faceCentroids

Calcule le centroïde (centre de gravité) de chaque face d'un polyèdre.

entrée les coordonnées des sommets, les indices des sommets de chaque face (soit dans un tableau, soit dans une liste de cellules).

sortie le centroïde de chaque face.

6.5.3 polyhedronNormalAngle

Calcule l'angle normal au(x) sommet(s) d'un polyèdre convexe.

entrée les coordonnées des sommets, les indices des sommets de chaque arête (optionnel, peut être déduit des indices des sommets des faces), les indices des sommets de chaque face, les indices des sommets pour lesquels on veut calculer l'angle normal.

sortie l'angle normal pour caque sommet demandé.

7 Surfaces et courbes

7.1 Cercles

Un cercle en 3D est représenté par son centre, son rayon, et un angle 3D correspondant à la normale du plan dans lequel il est contenu. De plus, on peut spécifier le paramètre ψ qui correspond à l'angle de rotation du cercle autour de la normale (permet de repérer les points sur le cercle, et de dessiner des arcs de cercle). Un cercle est donc représenté par :

$$[x_0, y_0, z_0, r, \theta, \varphi, \psi]$$

Note : ceci correspond à ce qui devrait être le cas pour être cohérent avec le reste des fonctions. Cependant, l'ordre des paramètres θ et φ est inversé dans les deux fonctions suivantes...

7.1.1 circle3dPosition

Calcule la position d'un point sur un cercle 3D. Le résultat correspond à l'angle du point par rapport à l'origine du cercle, calculé dans le plan contenant le cercle.

entrée les coordonnées du point, la réprésentation du cercle.

sortie la position angulaire du point sur le cercle, en radians entre 0 et 2π.

7.1.2 circle3dOrigin

Calcule les coordonnées du point correspondant au premier point du cercle, de position angulaire 0.

entrée la représentation du cercle.

sortie les coordonnées du premier point du cercle.

7.2 Sphère

Une sphère est définie par les coordonnées de son centre et son rayon :

$$[x_c, y_c, z_c, r]$$

7.2.1 createSphere

Calcule la sphère qui contient 4 points.

entrée les coordonnées de 4 points.

sortie la représentation de la sphère contenant ces points.

7.2.2 intersectLineSphere

Calcule les points d'intersection d'une droite avec une sphère.

entrée la représentation de la droite, et celle de la sphère.

sortie un tableau 2×3 contenant les coordonnées des points d'intersection.

7.2.3 intersectPlaneSphere

Calcule une représentation du cercle d'intersection entre une sphère et un plan.

entrée la représentation du plan, celle de la sphère.

sortie la représentation du cercle intersection.

7.3 Autres surfaces

7.3.1 revolutionSurface

Crée une surface de révolution à partir d'une courbe génératrice. On suppose que la courbe est paramétrée dans le repère (Oxz), et que la révolution se fait autour de l'axe Oz.

entrée les coordonnées des points de la courbe, le nombre de points de révolution, et éventuellement l'axe de révolution dans le plan (Oxz).

sortie les coordonnées des sommets du maillage corresondant à la surface de révolution créée, en trois tableaux séparés.

7.3.2 surfaceCurvature

Calcule la courbure locale sur une suface lisse dans une direction donnée, en fonction des deux courbures principales de la surface. Il s'agit de l'application numérique de la relation :

$$\kappa(\theta) = \kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta$$

entrée les courbures principales au point considéré, l'angle de la direction considérée par rapport à la direction de la première courbure principale.

sortie la courbure dans la direction considérée.

8 Transformations affines 3D

8.1 Représentation

Une transformation affine en 3 dimensions peut être représentée par une matrice 4×4 de la forme :

$$\begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La dernière colonne de la matrice correspond au vecteur de translation. Les 9 coefficients du coin supérieur gauche définissent les facteurs de taille, de cisaillement, de rotation.

8.2 Création

8.2.1 createTranslation3d

Crée la matrice correspondant à une translation par un vecteur donné.

entrée le vecteur de translation.

sortie une matrice de translation.

8.2.2 createRotationOx

Crée une matrice de rotation autour de l'axe Ox. Une rotation par un angle de $\pi/2$ transforme le vecteur $\overrightarrow{j}$ en le vecteur $\overrightarrow{k}$.

entrée l'angle de la rotation

sortie la matrice de rotation.

8.2.3 createRotationOy

Crée une matrice de rotation autour de l'axe Oy. Une rotation par un angle de $\pi/2$ transforme le vecteur $\overrightarrow{k}$ en le vecteur $\overrightarrow{i}$.

entrée l'angle de la rotation

sortie la matrice de rotation.

8.2.4 createRotationOz

Crée une matrice de rotation autour de l'axe Oz. Une rotation par un angle de $\pi/2$ transforme le vecteur $\overrightarrow{i}$ en le vecteur $\overrightarrow{j}$.

entrée l'angle de la rotation

sortie la matrice de rotation.

8.2.5 createScaling3d

Crée une matrice d'agrandissement par rapport à l'origine.

entrée le facteur d'agrandissement, ou les 3 facteurs si ils sont différents selon les axes

sortie la matrice d'agrandissement

8.3 Opérations sur les transformations

8.3.1 composeTransforms3d

Calcule la matrice de transformation équivalente à l'application successive de plusieurs transformations. Il s'agit d'un simple produit matriciel, mais l'ordre du produit n'étant pas l'ordre d'application des transformation, la fonction évite les ambiguïtés.

entrée une série de matrice de transformations, dans l'ordre de leur application.

sortie la matrice de transformation équivalente.

8.3.2 localToGlobal3d

Calcule une matrice de transformation 3D correspondant à une succession de transformations élémentaires :

- une rotation autour de l'axe Oz d'un angle psi correspondant à une rotation autour de l'axe normal de l'objet
- une rotation autour de l'axe Oy d'un angle theta correspondant à la colatitude
- une rotation autour de l'axe Oz d'un angle phi correspondant à la longitude
- une translation d'un vecteur v correspondant au replacement d'un objet dans le référentiel global

Cette fonction facilite les changements de coordonnées depuis un repère local (repère d'un objet), au repère global de référence.

entrée les paramètres v, θ, φ, et ψ

sortie la matrice de transformation équivalente.

9 Fonctions de dessin

La bibliothèque fournit quelques fonction de dessin pour certaines formes géométriques. Pour les formes non bornées (lignes droites), la forme est tronquée à la zone visible de la fenêtre courante.

9.1 Primitives de base

drawPoint3d Dessine un point ou un groupe de points.

drawLine3d Dessine une ligne découpée à l'intérieur de l'axe courant.

drawEdge3d Dessine un segment de droite à l'intérieur de l'axe courant.

drawPlane3d Dessine le polygone 3D correspondant à l'intersection d'un plan avec l'axe courant.

fillPolygon3d remplit un polygone en 3D.

drawPolyhedra dessine une polyèdre, défini par un ensemble de sommets et une liste d'indices pour chaque face.

9.2 Objets courbes

drawCurve3d Dessine une courbe définie par une série de points (polyligne).

drawCircle3d Dessine un cercle en 3 dimensions.

drawCircleArc3d Dessine un arc de cercle en 3 dimensions.

drawSphere Dessine une sphère, ou plus exactement un polyèdre qui approche une sphère.

drawCylinder Dessine un mailage qui approche un cylindre.

drawSphericalTriangle Dessine un triangle sphérique, composé d'arcs de grand cercle.

drawSphericalPatch Dessine un morceau de surface de sphère, délimité par des arcs de cercles 3D.

drawGrid3d Dessine un ensemble de points disposés en grille.

Index