

Fiche TD avec le logiciel  : tdr17

Dessine moi une maison ...

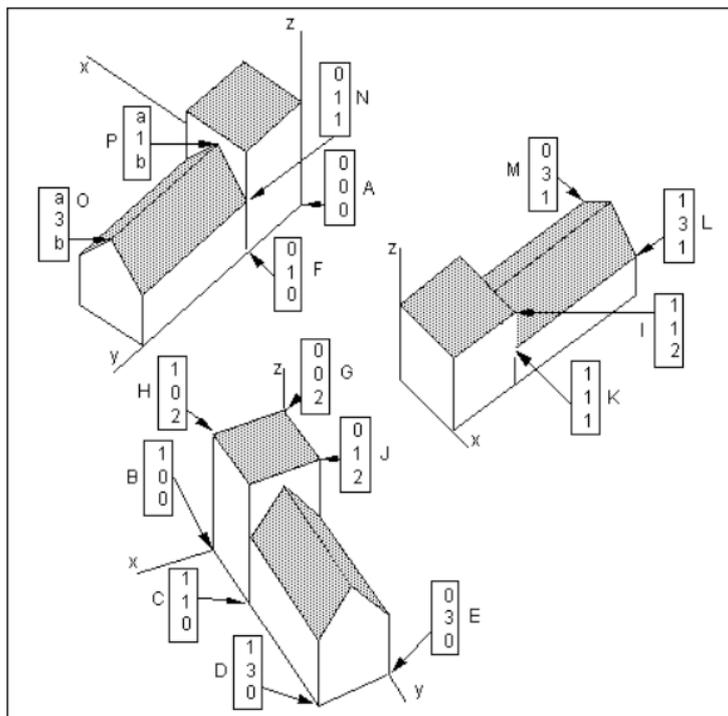
D. Chessel, A.B. Dufour & J.R. Lobry

Une petite expérience de représentation euclidienne sert d'introduction à la nature de la carte factorielle.

Table des matières

1 Dessiner une maison : le plan	2
1.1 Importer les données	3
1.2 Implanter la fonction	3
1.3 Exécuter	3
2 Modifier l'image pour lui donner une interprétation	4
3 Exercice : refaire ces dessins	6
4 Exercice : retrouver comment est fait ce schéma	7

1 Dessiner une maison : le plan



Considérons un bâtiment formé d'une tour carrée de base 1×1 (supposons que l'unité de mesure soit fixée à un mètre) et de hauteur 2 et d'une petite maison de largeur 1, longueur 2 et hauteur 1 surmontée d'un toit symétrique à deux pentes ($a = \frac{1}{2}$, $b = \sqrt{3}$). La totalité de l'information est contenue dans la table 1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
x	0	1	1	1	0	0	0	1	1	0	1	1	0	0	a	a
y	0	0	1	3	3	1	0	0	1	1	1	3	3	1	3	1
z	0	0	0	0	0	0	2	2	2	2	1	1	1	1	b	b

TABLE 1 – La matrice des coordonnées à trois dimensions des 16 points du bâtiment.

La question est : dessiner une quelconque des figures du type ci-dessus.

1.1 Importer les données

```
load(url("http://pbil.univ-lyon1.fr/R/donnees/maison.rda"))
maison
  x y z
A 0.0 0 0.000000
B 1.0 0 0.000000
C 1.0 1 0.000000
D 1.0 3 0.000000
E 0.0 3 0.000000
F 0.0 1 0.000000
G 0.0 0 2.000000
H 1.0 0 2.000000
I 1.0 1 2.000000
J 0.0 1 2.000000
K 1.0 1 1.000000
L 1.0 3 1.000000
M 0.0 3 1.000000
N 0.0 1 1.000000
O 0.5 3 1.732051
P 0.5 1 1.732051

options(digits = 3) # on suggère trois chiffres
t(maison)           # on veut transposer
  A B C D E F G H I J K L M N O P
x 0 1 1 1 0 0 0 1 1 0 1 1 0 0 0.50 0.50
y 0 0 1 3 3 1 0 0 1 1 1 3 3 1 3.00 1.00
z 0 0 0 0 0 0 2 2 2 2 1 1 1 1 1.73 1.73
```

1.2 Implanter la fonction

Donner un sens mathématique à cette petite procédure :

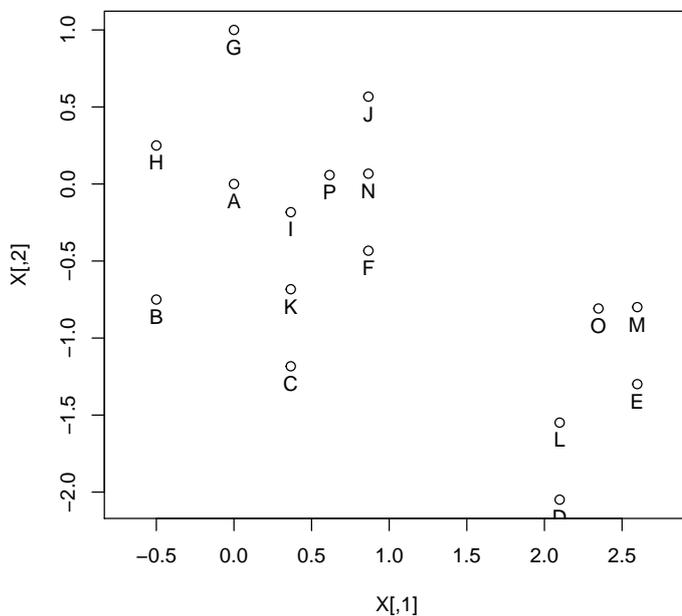
```
essai.3d <- function(X, alpha, beta, lab = F, ...)
{
  matrixh <- function(a = pi/6, b = pi/3) {
    h <- matrix(0, 2, 3)
    h[1, 1] <- -sin(a) ; h[1, 2] <- cos(a) ; h[1, 3] <- 0
    h[2, 1] <- -cos(a) * sin(b) ; h[2, 2] <- -sin(a) * sin(b)
    h[2, 3] <- cos(b)
    return(t(h))
  }
  alpha <- alpha * pi/180 ; beta <- beta * pi/180
  w <- matrixh(a=alpha, b=beta)
  X <- as.matrix(X)
  X <- X %*% w
  plot(X, ...)
  if (lab)
    text(X[, 1], X[, 2], rownames(X), pos = 1)
  return(invisible(X))
}
```

Noter :

- * la fonction dans la fonction,
- * le passage des paramètres par valeurs,
- * le paramètre ...

1.3 Exécuter

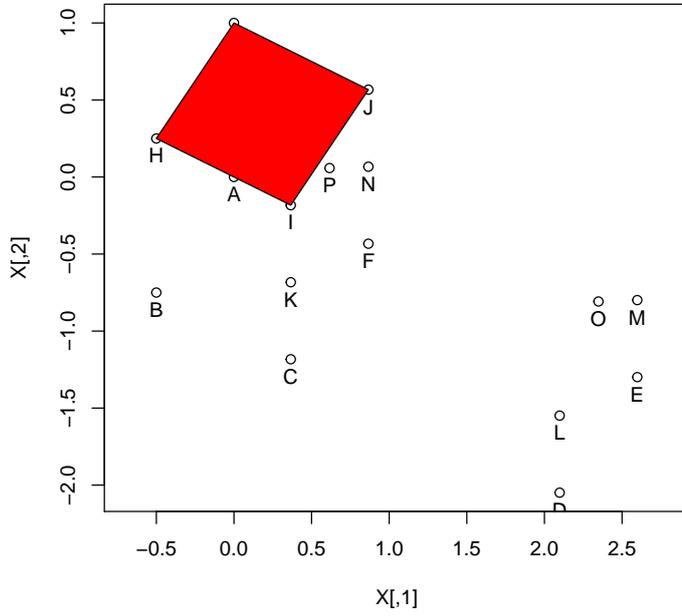
```
maison2d <- essai.3d(X = maison, alpha = 30, beta = 60, lab = TRUE, asp = TRUE)
```



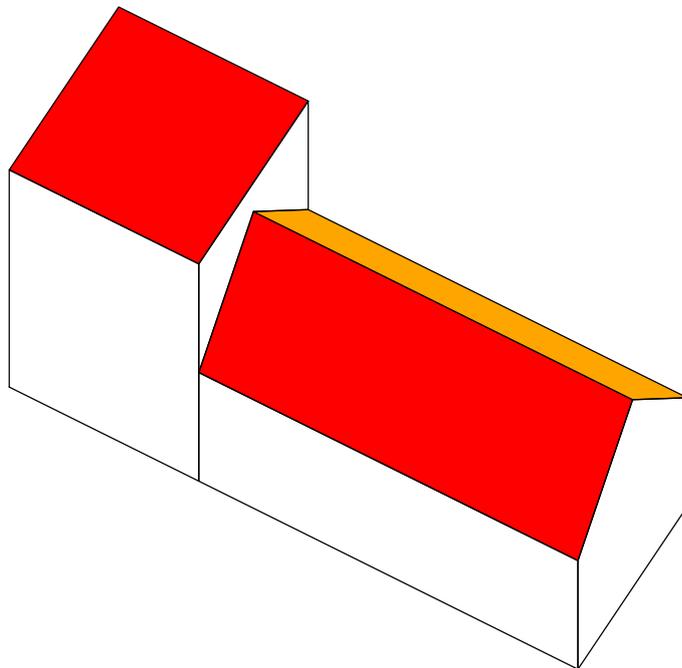
2 Modifier l'image pour lui donner une interprétation

Définir une fonction pour peindre un pan de la maison :

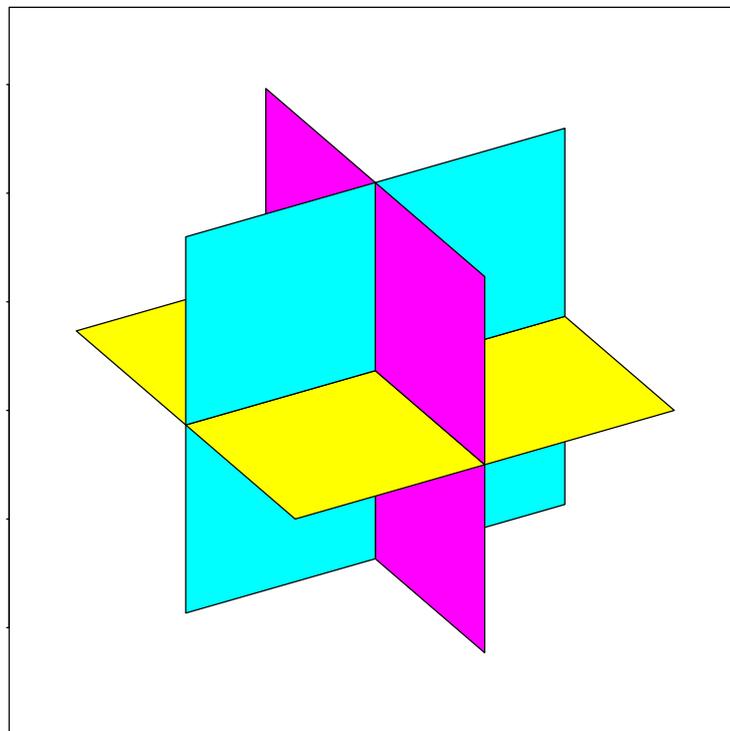
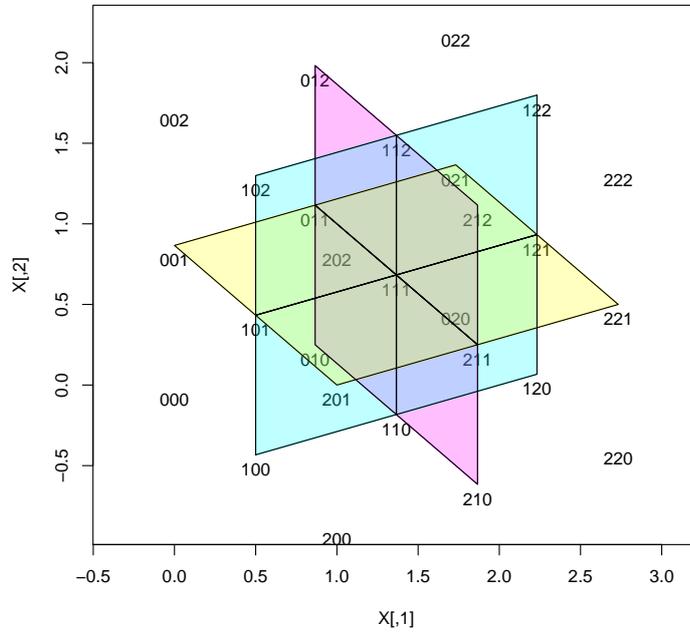
```
peindre <- function(data, points, col = "red", border = "black")
{
  x <- sapply(points, function(pt) data[which(rownames(data)==pt), 1])
  y <- sapply(points, function(pt) data[which(rownames(data)==pt), 2])
  polygon(x,y, col = col, border = border)
}
maison2d <- essai.3d(X = maison, alpha = 30, beta = 60, lab = TRUE, asp = TRUE)
peindre(maison2d, c("G","H","I","J"))
```



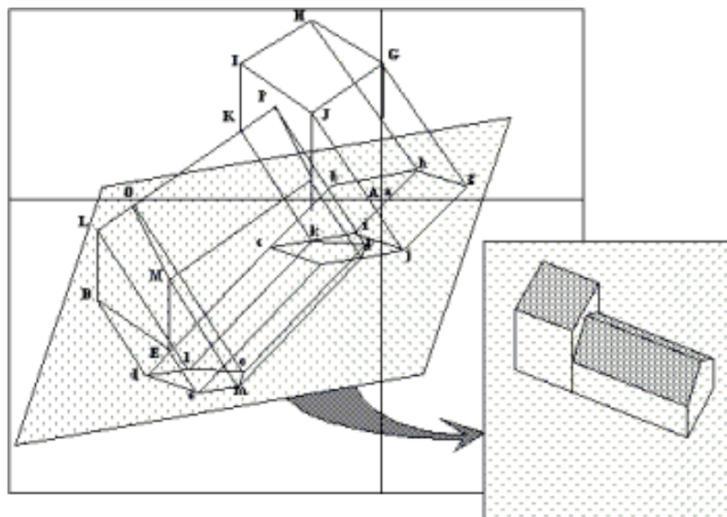
Peindre la maison :



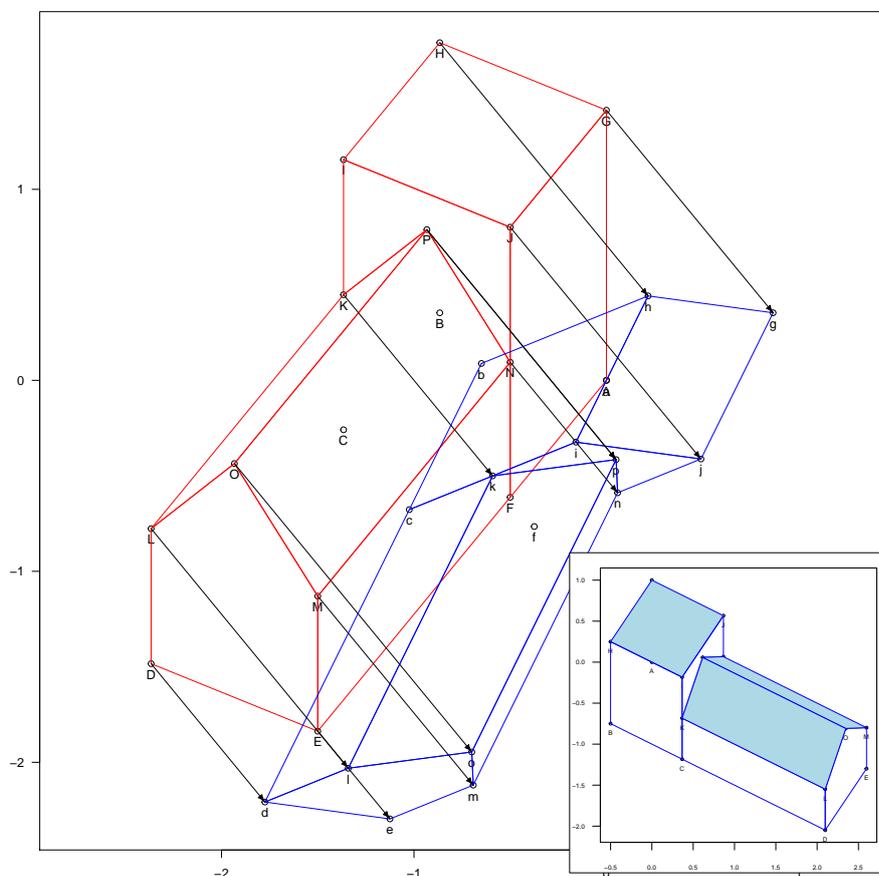
3 Exercice : refaire ces dessins



4 Exercice : retrouver comment est fait ce schéma



On utilise un double changement de base. En partant d'un point M de coordonnées (x, y, z) dans la base canonique, on calcule ses coordonnées (u, v, w) dans la base $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ associée à un couple d'angles (α, β) . On projette M en m de coordonnées $(0, v, w)$ dans cette base puis on calcule les nouvelles coordonnées de m (x', y', z') dans la base canonique. Le centre du dessin est fait par la représentation des points M et m dans une nouvelle direction associée à un couple d'angles (α', β') . On montre ainsi la projection. Le résultat de cette projection est en médaillon :



Les fonctions utilisées sont :

```
essai.3d <- function (X, alpha, beta, lab = F, ...) {
  matrixh <- function(a = pi/6, b = pi/3) {
    h <- matrix(0, 2, 3)
    h[1, 1] <- -sin(a) ; h[1, 2] <- cos(a) ; h[1, 3] <- 0
    h[2, 1] <- -cos(a) * sin(b) ; h[2, 2] <- -sin(a) * sin(b)
    h[2, 3] <- cos(b)
    return(t(h))
  }
  alpha <- alpha * pi/180 ; beta <- beta * pi/180
  w <- matrixh(a=alpha, b=beta)
  X <- as.matrix(X)
  X <- X %>% w
  plot(X, asp=1, ...)
  if (lab)
    text(X[, 1], X[, 2], row.names(X), pos = 1)
  return(invisible(X))
}
changebase <- function (X, alpha, beta) {
  uvw <- function(a = pi/6, b = pi/3) {
    h <- matrix(0, 3, 3)
    h[1, 1] <- cos(a)*cos(b) ; h[1,2] <- sin(a)*cos(b)
    h[1, 3] <- sin(b)
    h[2, 1] <- -sin(a) ; h[2, 2] <- cos(a) ; h[2, 3] <- 0
    h[3, 1] <- -cos(a) * sin(b) ; h[3, 2] <- -sin(a) * sin(b)
    h[3, 3] <- cos(b)
    h
  }
  return(t(h))
}
alpha <- alpha * pi/180 ; beta <- beta * pi/180
w <- uvw(a=alpha, b=beta)
x <- as.matrix(X)
```

```
y <- x %*% w
y[,1] <- 0
y <- y %*% t(w)
y <- as.data.frame(y)
names(y) <- names(X)
row.names(y) <- letters[1:nrow(y)]
X <- rbind.data.frame(X,y)
return(invisible(X))
}
```