

# Lire des coordonnées de vecteurs

```
import graph;
import geometry;
defaultpen(fontsize(13pt));

unitsize(0.7cm);

//show(defaultcoordsys);

pair u = (-2, 2), v = (3, 1);

// créer le réseau à partir de u et v

int xmin = -2, xmax = 1;
int ymin = -2, ymax = 1;
real kmin = 0.2, kmax = 1, kinc = 0.2;
draw(origin--u, L=Label("$\vec{u}$",
↪ Relative(0.5), S), Arrow(TeXHead),
↪ p=2bp+darkred);
draw(origin--v, L=Label("$\vec{v}$",
↪ Relative(0.5), S), Arrow(TeXHead),
↪ p=2bp+darkgreen);
pair[] reseau ;

int indiceLabel = 0;

void placerVecteur (real[] a,real[] b, pair
↪ ptOrigin = origin)
{
    string[] labelvecteur =
↪ array("abcdefghijklmnopqrstwxyz"); //pas du u
↪ ni v
    int n = a.length;
    int m = b.length;
    for (int i = 0; i <n ; ++i) {
        for (int j = 0; j <m ; ++j) {
            draw(ptOrigin--shift(ptOrigin)*(a[i]*u+b[j]
↪ *v),
↪ L=Label("$\vec{"+labelvecteur[indiceLabel]
↪ +"}$", Relative(0.7)), Arrow());
            indiceLabel+=1;
        }
    }
}

real[] a = {0,-0.8, 1.8}, b = {-1.2, 1.2};
real[] ap = {1, -1}, autrep = {1};
real[] app = {-0.8, 0.8}, bpp = { 1, -1};

//création du réseau en fonction de u et v
for (real kx = kmin; kx <= kmax; kx = kx + kinc) {
    for (real ky = kmin; ky <= kmax; ky = ky +
↪ kinc) {
        for (int x = xmin; x <= xmax; ++x) {
            for (int y = ymin; y<= ymax; ++y) {
                reseau.push((kx + x)*u+(ky+y)*v);
            }
        }
    }
}

dot(reseau, p=grey+2bp);

pair[] origines={origin, 0.8*u+0.8*v, -u-0.8*v };
placerVecteur(a, b, 0);
placerVecteur(ap, autrep, 0.8*u+0.8*v );
placerVecteur(app, bpp, -u-0.8*v );

dot(origines, purple);

label(minipage("Donner la décompositon de chaque
↪ vecteur dans la base $(\vec{u}, \vec{v})$",
↪ 100), (8, 4));
//draw(origin--w, L=Label("$\vec{w}$",
↪ Relative(0.5), S), Arrow());

shipout(bbox(2mm, Fill(white)));
```