poisson.edp

```
load "gmsh";
//load "Element_P3" // [P3] piecewise P3 continuous finite element(2d)

int N = 100;
// Define mesh boundary
border Gamma1(t=0,2*pi) { x=cos(t); y=sin(t); label=111; }
border Gamma2(t=0,2*pi) { x=0.5*cos(t); y=0.5*sin(t); label=222; }

// The triangulated domain Th is on the left side of its boundary
mesh Th = buildmesh(Gamma1(N)+Gamma2(-N));
plot(Th,wait=true); // plot(Th,wait=true,ps="Th.eps");

fespace Vh(Th, P1);
// Define u and v as piecewise-P1 continuous functions
Vh u,v;

// Poisson Eq: -△ u = f
// $-\nabla u = f$
func f = x*y;

// Define the PDE
solve Poisson(u,v) = int2d(Th)(dx(u)*dx(v) + dy(u)*dy(v)) - int2d(Th)(f*v) + on(111, 222, u=0);

plot(u,dim=3, ps="outputs/poisson.eps", fill=true, wait=true);

ofstream sol("outputs/u.csv");
for(int j=0; j<Th.nv; j++) {
        sol << Th(j).x << "," << Th(j).y << "," << u[][j] << endl;
}

ofstream tri("outputs/Th.csv");
for(int i=0;i<Th.nt;i++){
    tri << Th[i][0] << "," << Th[i][1] << "," << Th[i][2] << endl;
}

ofstream gp("outputs/gnuplot.gp");
```

```
for (int i = 0; i < Th.nt ; i++){
    for (int j = 0; j < 3; j++){
      gp << Th[i][j].x << " "<< Th[i][j].y << " " << u[][Vh(i,j)] << endl;
    }
    gp << Th[i][0].x << " " << Th[i][0].y << " " << u[][Vh(i,0)] << endl << endl << endl;
}
```

FreeFem++: poisson.eps

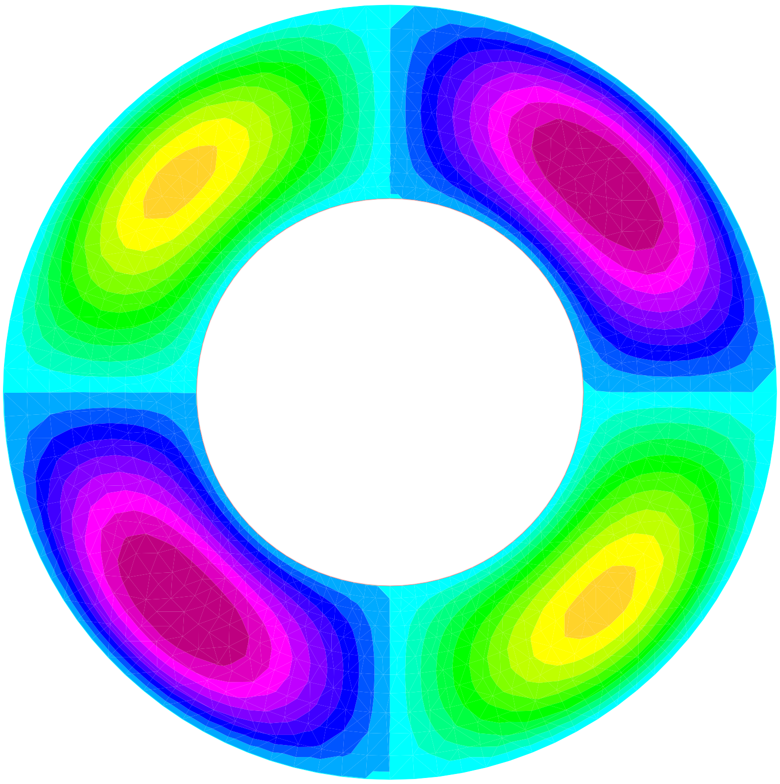

図 1　poisson.eps

Gnuplot: gnuplot.eps

Python: python.eps

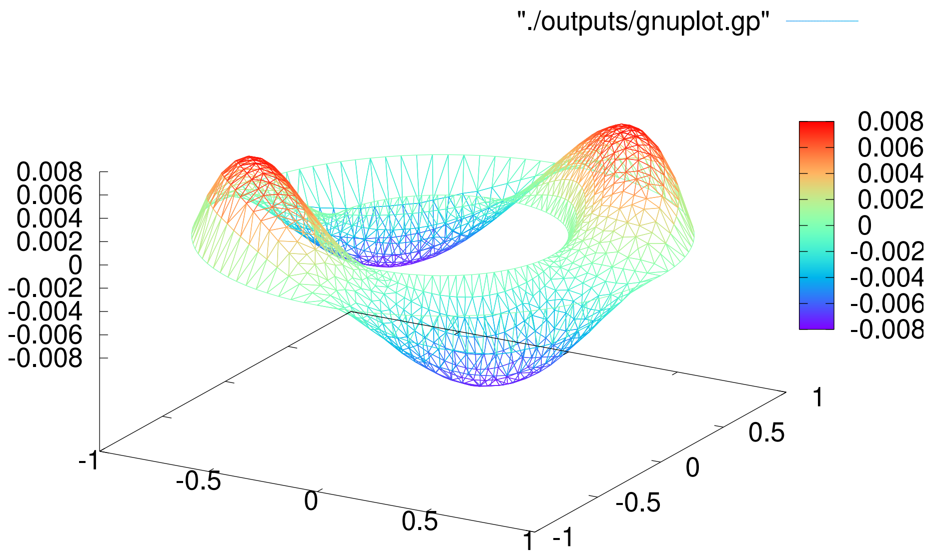

図 2　gnuplot.eps

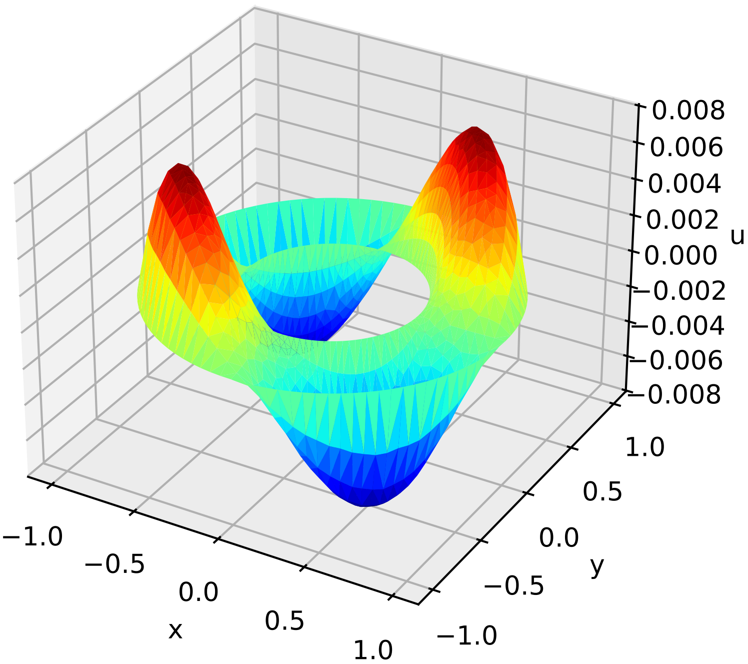

図 3　python.eps