from fenics import\*

mesh = UnitSquareMesh(8,8)

V = FunctionSpace(mesh,'P', 1)

u\_D = Expression ('1 +x[0]\*x[0]+2\*x[1]\*x[1]', degree = 2)

def boundary (x, on\_boundary):

return on\_boundary

bc = DirichletBC(V,u\_D,boundary)

u = TrialFunction (V)

v = TestFunction(V)

f = Constant(-6.0)

a = dot(grad(u), grad(v))\*dx

L = f\*v\*dx

u = Function(V)

solve (a == L, u, bc)

vtkfile = File("poisson/solution.pvd")

vtkfile << u

error\_L2 = errornorm(u\_D,u,"L2")

vertex\_values\_u\_D = u\_D.compute\_vertex\_values(mesh)

vertex\_values\_u = u.compute\_vertex\_values(mesh)

import numpy as np

error\_max = np.max(np.abs(vertex\_values\_u\_D - vertex\_values\_u))

print("error\_L2 =", error\_L2)

print("error\_max =", error\_max)