

```
// example 3 : poisson-error.edp [slide page 20]
// error estimation for finite element solution of Poisson equation
// with P2 element
// for tutorial by Japan SIAM, Tokyo, 11-12 Feb.2016, Atsushi Suzuki
```

```
int n1, n2, n3;
n1 = 20;
n2 = n1 * 2;
n3 = n2 * 2;
mesh Th1=square(n1,n1);
mesh Th2=square(n2,n2);
mesh Th3=square(n3,n3);
fespace Vh1(Th1,P2);
fespace Vh2(Th2,P2);
fespace Vh3(Th3,P2);

Vh1 u1,v1;
Vh2 u2,v2;
Vh3 u3,v3;
real err1, err2, err3, hh1, hh2, hh3;

func f = 5.0/4.0 * pi * pi * sin(pi * x) * sin(pi * y / 2.0);
func h = (-pi)/2.0 * sin(pi * x);
func g = sin(pi * x) * sin(pi * y / 2.0);
// for error estimation
func sol = sin(pi * x) * sin(pi * y / 2.0);
func solx = pi * cos(pi * x) * sin(pi * y / 2.0);
func soly = (pi / 2.0) * sin(pi * x) * cos(pi * y / 2.0);

solve poisson1(u1,v1) =
int2d(Th1)( dx(u1)*dx(v1)+dy(u1)*dy(v1) )
- int2d(Th1)( f*v1 )
- int1d(Th1,1) ( h*v1 )
+ on(2,3,4,u1=g);

solve poisson2(u2,v2) =
int2d(Th2)( dx(u2)*dx(v2)+dy(u2)*dy(v2) )
- int2d(Th2)( f*v2 )
- int1d(Th2,1) ( h*v2 )
+ on(2,3,4,u2=g);

solve poisson3(u3,v3) =
int2d(Th3)( dx(u3)*dx(v3)+dy(u3)*dy(v3) )
- int2d(Th3)( f*v3 )
- int1d(Th3,1) ( h*v3 )
+ on(2,3,4,u3=g);

hh1 = 1.0 / n1 * sqrt(2.0);
hh2 = 1.0 / n2 * sqrt(2.0);
hh3 = 1.0 / n3 * sqrt(2.0);
// int2d uses qf5pT : 5th order integration quadrature
err1 = int2d(Th1)( (dx(u1) - solx) * (dx(u1) - solx) +
(dy(u1) - soly) * (dy(u1) - soly) +
(u1 - sol) * (u1 - sol));
err1 = sqrt(err1);

err2 = int2d(Th2)( (dx(u2) - solx) * (dx(u2) - solx) +
(dy(u2) - soly) * (dy(u2) - soly) +
(u2 - sol) * (u2 - sol));
err2 = sqrt(err2);

err3 = int2d(Th3)( (dx(u3) - solx) * (dx(u3) - solx) +
(dy(u3) - soly) * (dy(u3) - soly) +
(u3 - sol) * (u3 - sol));
err3 = sqrt(err3);
cout << "DOF=" << u1[.].n << "\t h=" << hh1 << " err-H1=" << err1 << endl;
cout << "DOF=" << u2[.].n << "\t h=" << hh2 << " err-H1=" << err2 << endl;
cout << "DOF=" << u3[.].n << "\t h=" << hh3 << " err-H1=" << err3 << endl;
cout << "O(h^2)=" << log(err1/err2)/log(hh1/hh2) << endl;
cout << "O(h^2)=" << log(err2/err3)/log(hh2/hh3) << endl;
```