
PoissonFEM Documentation

Release 1.0.0

Rafał Białozor

October 13, 2013

CONTENTS

Contents:

RÓWNANIE POISSONA:

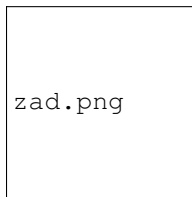
W ogólnym przypadku równanie Poissona ma postać:

$$\frac{\partial^2}{\partial x^2}u(x, y, z) + \frac{\partial^2}{\partial y^2}u(x, y, z) + \frac{\partial^2}{\partial z^2}u(x, y, z) = f(x, y, z)$$

lub zapisując krócej:

$$\nabla^2 u = f$$

PRZYKŁAD 1



Zdefiniujmy obszar w którym poszukiwać będziemy naszego rozwiązania.

```
int dlugosc=2, wysokosc=1;

// Definicja konturu

int lewy=1;
int dolny=2;
int prawy=3;
int gorny=4;

border left(t=wysokosc,0) {x=0; y=t;label=lewy;};
border bottom(t=0,dlugosc) {x=t; y=0;label=dolny;};
border right(t=0,wysokosc) {x=dlugosc; y=t;label=prawy;};
border top(t=dlugosc,0) {x=t; y=wysokosc;label=gorny;};
```

Brzeg obszaru określa się przy użyciu funkcji parametrycznych $x = f(t)$ i $y = g(t)$. Zdefiniowany brzeg zgodny jest z obiegiem w lewo. Definiowanie brzegów obszarów zgodnie z obiegiem, jest istotną sprawą przy definiowaniu otworów w zadanym obszarze, tym problemem zajmiemy się jednak później.

Generowanie siatki trójkątnej. Zdefiniujmy zmienne typu int określające liczbę węzłów zadanych na brzegach obszaru

```
int NpSzerokosc=10 ;
int NpDlugosc=10 ;

mesh siatka=buildmesh(left(NpSzerokosc)+top(NpDlugosc)
+right(NpSzerokosc)+bottom(NpDlugosc));
```

Podobnie jak w pythonie, wizualizację obiektów użykujemy poleceniem *plot*

```
plot(siatka);
fespace Vh(siatka,P1);
func f= 45;
int k=4;
Vh u,v;
problem Poisson(u,v,solver=LU) =
int2d(siatka)(k*dx(u)*dx(v) + k*dy(u)*dy(v))
-int2d(siatka)( f*v ) - int1d(siatka,left)((0*v))
```

```
- intld(siatka,bottom)((-30*v))
- intld(siatka,top)((-30*v))
+ on(right,u=10);
real cpu=clock();
Poisson;
plot(u, value=true, fill=true, cmm="Rozklad temperature");
```

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*