

# Notes on fractal harmonic measure problem

Qile Yan

December 10, 2023

## 1 Result on unit disk

In this section, we consider the domain  $\Omega$  is a unit disk (see Fig. 1).

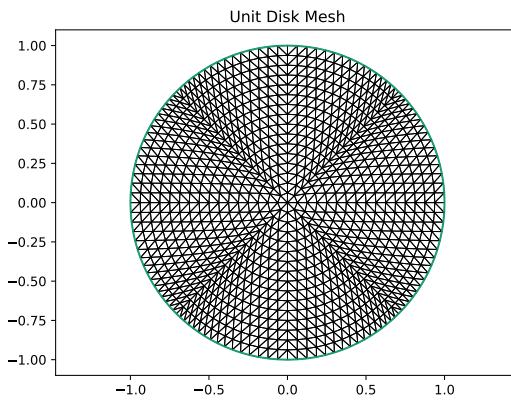


Figure 1: Unit disk.

We first test the firedrake solver on the PDE:

$$-\Delta u = f \quad \text{on } \Omega \tag{1.1}$$

with  $u = g$  on  $\partial\Omega$ .

In Fig. 2 and Fig. 3, we plot the error in  $L^2$  and  $H^1$  norm with respect to mesh size  $h$  and degree of freedom. In Fig. 2, the manufactured solution is  $u(x, y) = 2 + x + 3y$ . In Fig. 3, the manufactured solution is  $u(x, y, z) = 2 + x^2 + y$ . The Lagrange linear element is used. The uniform refinement is done by built-in function `MeshHierarchy`.

(a) Error vs mesh size



(b) Error vs degree of freedom

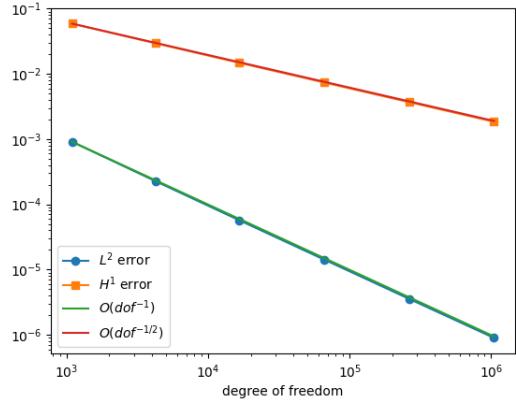


Figure 2: Unit disk: manufactured solution  $u = x + 3y + 2$ . Note that only round off error appears here since the solution is in the linear finite element space. Remark: Here, we use the Krylov subspace method with BoomerAMG in preconditioner in the solver. If we change the solver to be LU factorisation, the error here would be around  $10^{-14} \sim 10^{-15}$ .

(a) Error vs mesh size



(b) Error vs degree of freedom

Figure 3: Unit disk: manufactured solution  $u = x^2 + y + 2$ .

Next, we solve the PDE

$$-\Delta u = f \quad \text{on } \Omega \quad (1.2)$$

with  $u = 0$  on the unit disk and evaluate the solution at points from center to boundary. In particular, we evaluate the solutions at

$$\mathbf{x}_i = (0, 1 - 1/2^i), \quad 0 \leq i \leq n_{max}. \quad (1.3)$$

In Fig. 4, we show the results of the PDE (1.2) with a different  $f$ . That is, we define  $f$  as

$$f(x, y) = e^{-20(x^2+y^2)}.$$



Figure 4: On unit disk. (a). forcing term  $f$ . (b) solution  $u$ . (c). Evaluation of solution at a sequence points  $\{\mathbf{x}_i\}_{i=1}^{15}$  where  $\mathbf{x}_i$  is defined as in (1.3). Notice that the slope is almost equal to  $\frac{\partial u}{\partial n} \approx 0.025$ .

$$\begin{aligned} - \int_{\partial\Omega} \frac{\partial u}{\partial n} &= \int_{\Omega} f = 2\pi \int_0^1 e^{-20r^2} r dr = \frac{-\pi}{20}(e^{-20} - 1) \\ \Rightarrow \frac{\partial u}{\partial n} &= \frac{1}{40}(e^{-20} - 1) \approx -0.025 \end{aligned}$$

## 2 On snowflake boundary domain

In this note we solve the PDE

$$-\Delta u = f \quad \text{on } \Omega \tag{2.1}$$

with  $u = 0$  on  $\partial\Omega$ . Here,  $\Omega$  is a unit square where each edges are replaced by the Koch snowflake (See Fig. 5). In this note, we use the FEM python package firedrake solving the PDE.

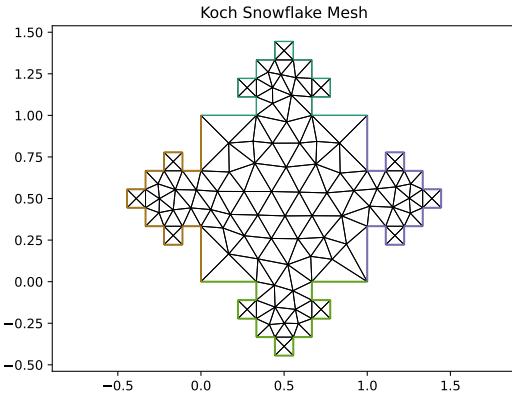
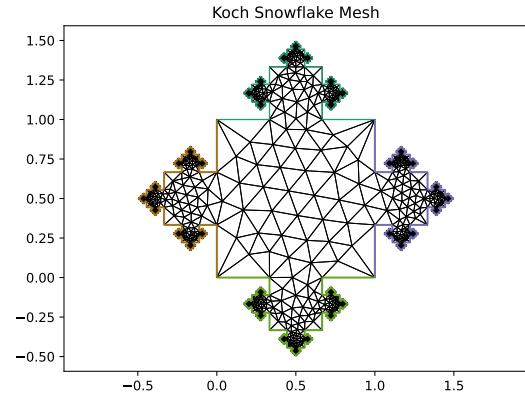
(a)  $n = 2$ (b)  $n = 4$ 

Figure 5: Unit square with each edges replaced by a Koch snowflake with  $n$  iterations. Meshsize is 0.5 for each vertices in .geo file.

In Fig. 6 and Fig. 7, we plot the error in  $L^2$  and  $H^1$  norm with respect to mesh size  $h$  and degree of freedom where domain is unit square with snowflake ( $n = 4$  iterations). In Fig. 6, the manufactured solution is  $u(x, y) = 2 + x + 3y$ . In Fig. 7, the manufactured solution is  $u(x, y, z) = 2 + x^2 + y$ . The Lagrange linear element is used. The uniform refinement is done by built-in function MeshHierarchy.

(a) Error vs mesh size

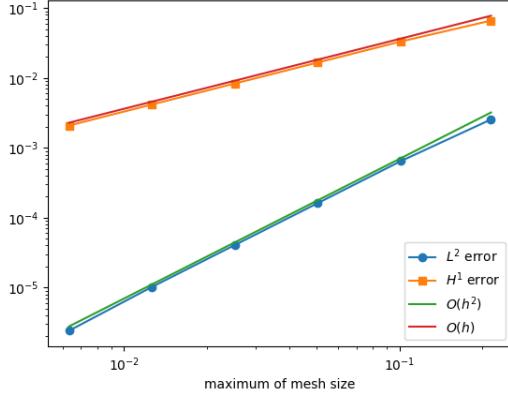


(b) Error vs degree of freedom

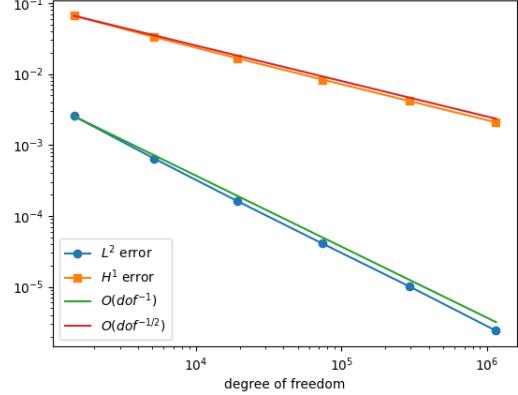


Figure 6: Unit square with  $n = 8$  snowflake iterations: manufactured solution  $u = x + 3y + 2$ . Note that only round off error appears here since the solution is in the linear finite element space.

(a) Error vs mesh size

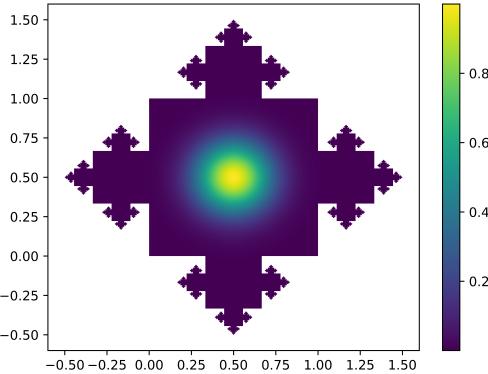
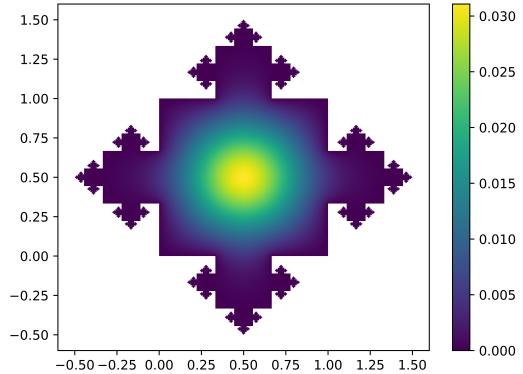


(b) Error vs degree of freedom

Figure 7: Unit square with  $n = 8$  snowflake iterations: manufactured solution  $u = x^2 + y + 2$ .

In Fig. 8(b), we plot the solution of the PDE (2.1) where  $\Omega$  is the square with 4 snowflake iterations as in Fig. 5(b) and  $f$  is defined as

$$f(x, y) = e^{-20((x-0.5)^2 + (y-0.5)^2)}.$$

(a)  $f$ (b) Solution  $u$ Figure 8: On unit square with  $n = 8$  snowflake iterations. (a). forcing term  $f$ . (b) solution of the PDE (2.1).

In Fig. 10, we evaluate the solution along a random path with each points  $\mathbf{x}_i, 0 \leq i \leq n$  is a center of the square from  $i$ -th snowflake iteration. In particular, we create the sequence of the points along the path as followings:

- 1). Set  $\mathbf{x}_0 = (0.5, 0.5)$
- 2). Pick a random integer from 0 to 3. Each integer corresponds to one of four normal directions  $\vec{v}_1$ . (upward, downward, left, right).
- 3). Set  $\mathbf{x}_1 = \mathbf{x}_0 + h_0 \vec{v}_1$  with  $h_0 = \frac{2}{3}$ .

- 4). For  $i = 2, \dots, n$ , Pick a random integer from 0 to 2. Each integer corresponds to one of three normal directions  $\vec{v}_i$  (forward, left, right). Set

$$\mathbf{x}_i = \mathbf{x}_{i-1} + h_{i-1} \vec{v}_i.$$

where  $h_{i-1} = \frac{2}{3^i}$ .



Figure 9: Two random path on the unit square with  $n = 8$  snowflake iterations.

Let  $dx_i = (1/3)^i$ . Then we would like to approximate  $u(\mathbf{x}_i)$  as a function of  $dx_i$  in the form

$$u(\mathbf{x}_i) = c(dx_i)^\alpha$$

By taking the log on both ends,

$$\log u(\mathbf{x}_i) = \log c + \alpha \log(dx_i), \quad 0 \leq i \leq n.$$

The coefficients  $c$  and  $\alpha$  are found by least squares approximation.

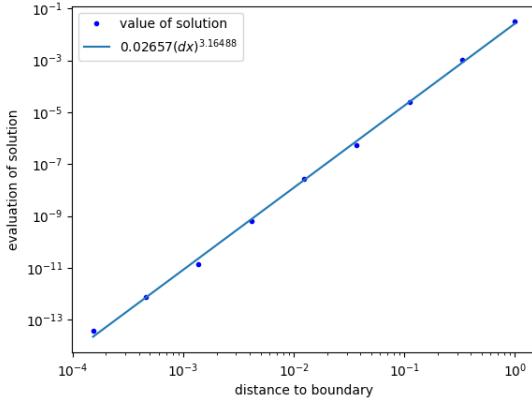
Running over 500 random path for two times, we get:

	Mean	Standard deviation
$c$	0.02598	0.005612
$\alpha$	2.89311	0.16436

and

	Mean	Standard deviation
$c$	0.02567	0.005760
$\alpha$	2.90044	0.16815

(a)  $c = 0.02657, \alpha = 3.16488$



(b)  $c = 0.03466, \alpha = 3.05545$

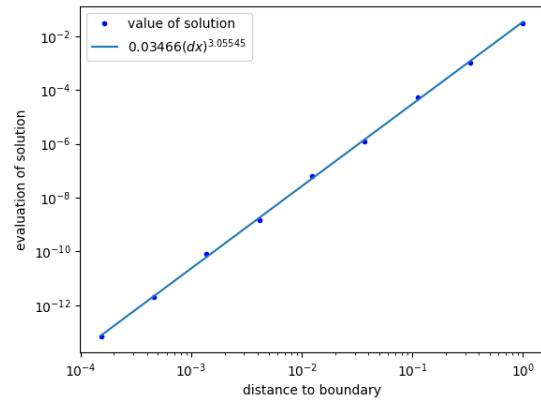
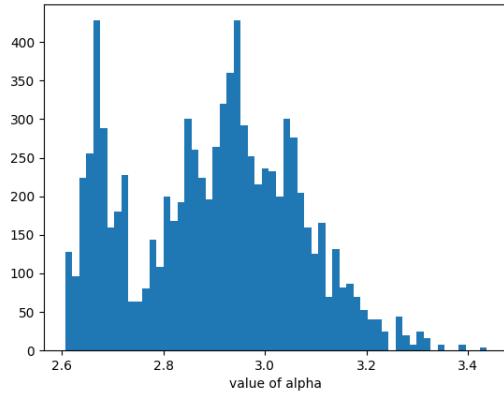


Figure 10: On unit square with  $n = 8$  snowflake iterations. Result of different random path

(a) distribution of  $\alpha$



(b) distribution of  $c$ .

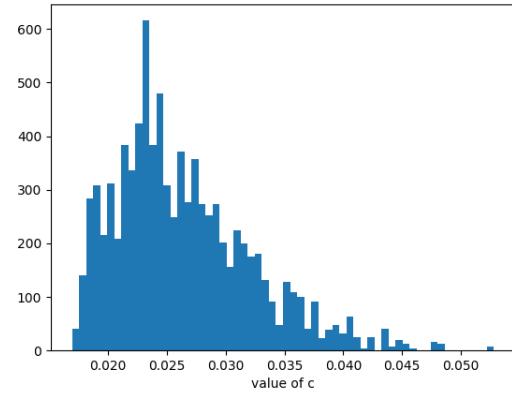


Figure 11: On unit square with  $n = 8$  snowflake iterations. Result over all possible path. Total number of path is  $4 * 3^{8-1} = 8748$  (a). distribution of  $\alpha$ . (b). distribution of  $c$ .