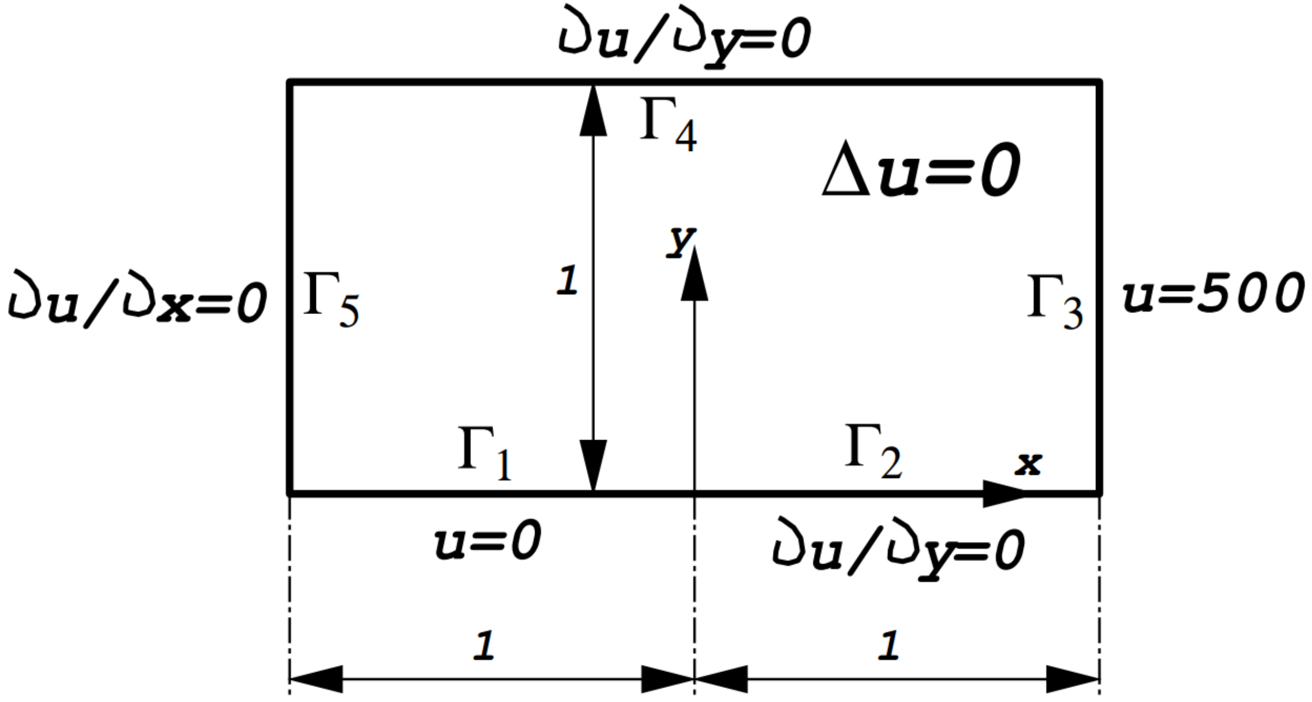
APMA 2560

2D Finite Elements: Singular Solutions

Siyuan Song
05/15/2018

Motz Problem – Problem Description



Governing Equation

$$\nabla^2 u = 0$$

Dirichlet boundary condition

$$u = 0 \quad \text{at } \Gamma_1$$

$$u = 500 \quad \text{at } \Gamma_3$$

Natural boundary condition

$$\partial u / \partial y = 0 \quad \text{at } \Gamma_2$$

$$\partial u / \partial y = 0 \quad \text{at } \Gamma_4$$

$$\partial u / \partial x = 0 \quad \text{at } \Gamma_5$$

Motz Problem – Mathematical background

Take the variation on both side

$$\delta I = \iint \left(\nabla^2 u\right) \delta v d\Omega = 0$$

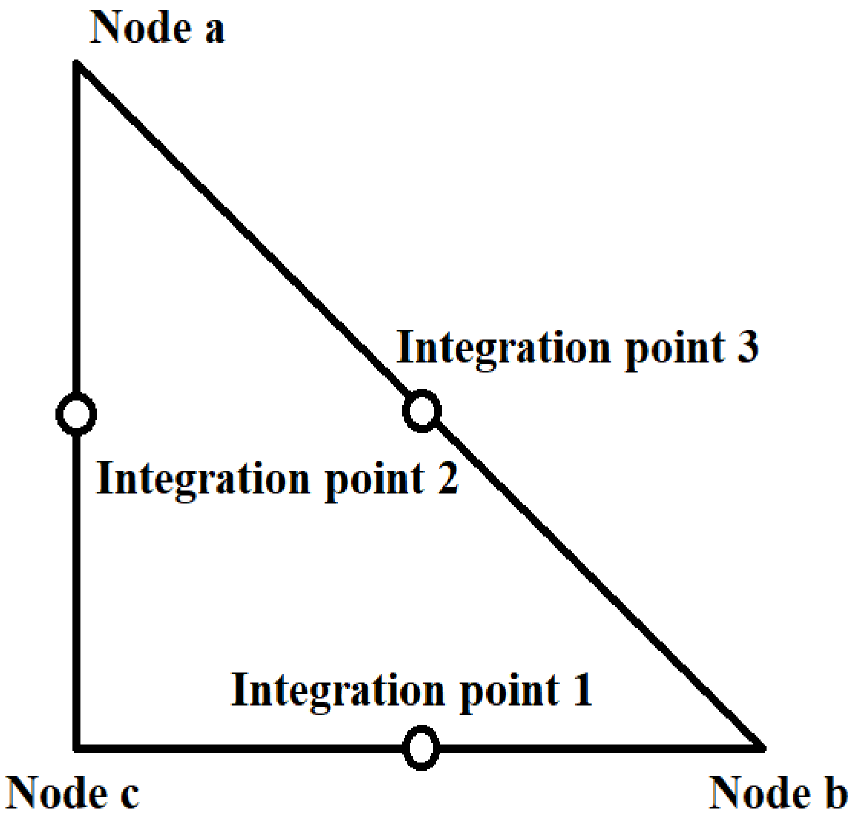
According to Divergence Theorem

$$\delta I = \int_\Gamma \left(\nabla u\right) \mathbf{n} \delta v d\left(\partial\Omega\right) - \iint \left(\nabla u\right) \delta\left(\nabla v\right) d\Omega = -\iint \left(\nabla u\right) \delta\left(\nabla v\right) d\Omega = -\frac{1}{2} \delta \iint \left|\nabla v\right|^2 d\Omega$$

The functional is

$$I = -\frac{1}{2} \iint \left|\nabla v\right|^2 d\Omega$$

Motz Problem – Finite Element Analysis – Shape Function



3-Node 3-Integration point
Linear Triangular Element

Shape function

$$N^a(x_1,x_2)=\frac{(x_2-x_2^b)(x_1^c-x_1^b)-(x_1-x_1^b)(x_2^c-x_2^b)}{(x_2^a-x_2^b)(x_1^c-x_1^b)-(x_1^a-x_1^b)(x_2^c-x_2^b)}$$

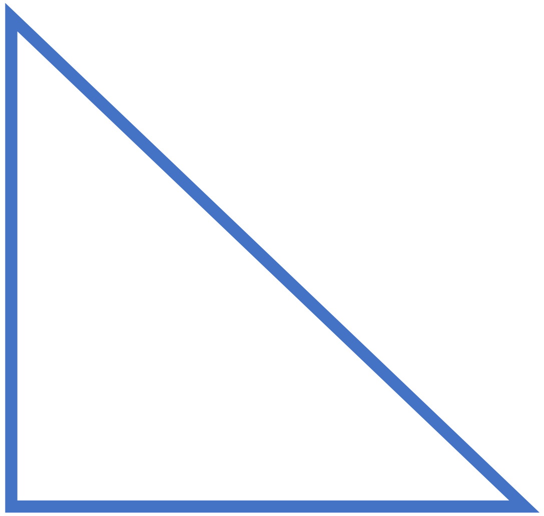
$$N^b(x_1,x_2)=\frac{(x_2-x_2^c)(x_1^a-x_1^c)-(x_1-x_1^c)(x_2^a-x_2^c)}{(x_2^b-x_2^c)(x_1^a-x_1^c)-(x_1^b-x_1^c)(x_2^a-x_2^c)}$$

$$N^c(x_1,x_2)=\frac{(x_2-x_2^a)(x_1^b-x_1^a)-(x_1-x_1^a)(x_2^b-x_2^a)}{(x_2^c-x_2^a)(x_1^b-x_1^a)-(x_1^c-x_1^a)(x_2^b-x_2^a)}$$

Point-value in a element

$$v_e(x_1,x_2)=N^a(x_1,x_2)v^a+N^b(x_1,x_2)v^b+N^c(x_1,x_2)v^c$$

Motz Problem – Finite Element Analysis – Why Triangle?



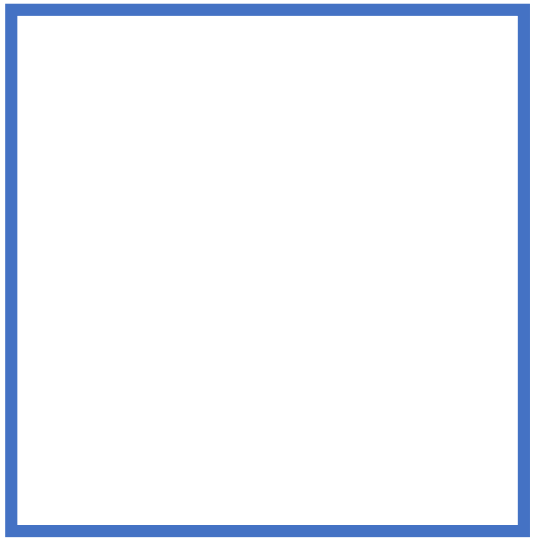
Shape function (Local Coordinate)

$$N^a = \xi_1, N^b = \xi_2, N^c = 1 - \xi_1 - \xi_2$$

$\partial N^a / \partial \xi_1 = 1, \partial N^a / \partial \xi_2 = 0,$ Easy to fit the boundary.

$\partial N^b / \partial \xi_1 = 0, \partial N^b / \partial \xi_2 = 1,$ Constant derivatives.

$\partial N^c / \partial \xi_1 = -1, \partial N^b / \partial \xi_2 = -1,$ Easy programing



Shape function (Local Coordinate)

$$N^a = 0.25(1 - \xi_1)(1 - \xi_2), N^b = 0.25(1 + \xi_1)(1 - \xi_2)$$

$$N^c = 0.25(1 + \xi_1)(1 + \xi_2), N^d = 0.25(1 - \xi_1)(1 + \xi_2)$$

Only suitable for uniform boundary

Linear derivatives. Difficult programing

Motz Problem – Finite Element Analysis – Integration points

The functional in a single element

$$I\left(v_e\right)=-\frac{1}{2}\sum_{i=a}^{c}\sum_{j=a}^{c}v^{(i)}v^{(j)}\iint \nabla N_i \nabla N_j d\Omega$$

We can define $K_{ij}=\iint \nabla N_i \nabla N_j d\Omega$

$$I\left(v_e\right)=-\frac{1}{2}v_e^T K_e v_e$$

Assemble all elements in the system

$$I\left(v\right)=-\frac{1}{2}v^T K v$$

To minimize the functional, we have

$$-Kv=0$$

In local coordinate, the three nodes for the elements are

$$\left(\xi_1,\xi_2\right)^a=\left(1,0\right)\quad \left(\xi_1,\xi_2\right)^b=\left(0,1\right)$$

$$\left(\xi_1,\xi_2\right)^c=\left(0,0\right)$$

The three shape functions for local coordinate are chosen as

$$N^a=\xi_1, N^b=\xi_2, N^c=1-\xi_1-\xi_2$$

The three integration points are chose as

$$\xi_1^1=0.5, \xi_2^1=0, w_1=1/6$$

$$\xi_1^2=0, \xi_2^2=0.5, w_2=1/6$$

$$\xi_1^3=0.5; \xi_2^3=0.5, w_3=1/6$$

Motz Problem – Finite Element Analysis – Galerkin Method

For a single point

$$v_e = \begin{bmatrix} v^a & v^b & v^c \end{bmatrix}^T$$

The Ke matrix can be determined by

$$\left(K_e\right)_{ij} = \sum_{k=1}^{3} \nabla N_k^{(i)} \nabla N_k^{(i)} w_k \eta_k$$

w_k is the weight for the integration point k,
η_k is the term that measure the area of the element,
i, j refer to the three nodes of the elements,

$\nabla N_k^{(i)}$ is the value of $\nabla N^{(i)}$ at the int_point k

For a single integration points

$$\tilde{B}_k = \begin{bmatrix} \partial N_k^a / \partial x_1 & \partial N_k^a / \partial x_2 \\ \partial N_k^b / \partial x_1 & \partial N_k^b / \partial x_2 \\ \partial N_k^c / \partial x_1 & \partial N_k^c / \partial x_2 \end{bmatrix}$$

Then Ke matrix can be expressed as

$$K_e = \sum_{k=1}^{3} \left(\tilde{B}_k \tilde{B}_k^T \right) w_k \eta_k \quad \eta_k = \begin{bmatrix} dx_1/d\xi_1 , dx_1/d\xi_2 \\ dx_2/d\xi_1 , dx_2/d\xi_2 \end{bmatrix}$$

Assemble the stiffness matrix, we shall have

Motz Problem – Finite Element Analysis – Galerkin Method

$$
\begin{bmatrix} k_{11} & k_{12} & \dots & k_{1N} \\ k_{21} & k_{22} & \dots & k_{2N} \\ \dots & \dots & \dots & \dots \\ k_{N1} & k_{N2} & \dots & k_{NN} \end{bmatrix} \begin{bmatrix} v^{(1)} \\ v^{(2)} \\ \dots \\ v^{(N)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}
$$

The boundary condition is set as

$$
\begin{bmatrix} k_{11} & 0 & \dots & k_{1N} \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ k_{N1} & 0 & \dots & k_{NN} \end{bmatrix} \begin{bmatrix} v^{(1)} \\ v^{(2)} \\ \dots \\ v^{(N)} \end{bmatrix} = \begin{bmatrix} 0 - k_{12} v_D^{(2)} \\ v_D^{(2)} \\ \dots \\ 0 - k_{N2} v_D^{(2)} \end{bmatrix}
$$

For the Natural boundary condition, we can just leave it free

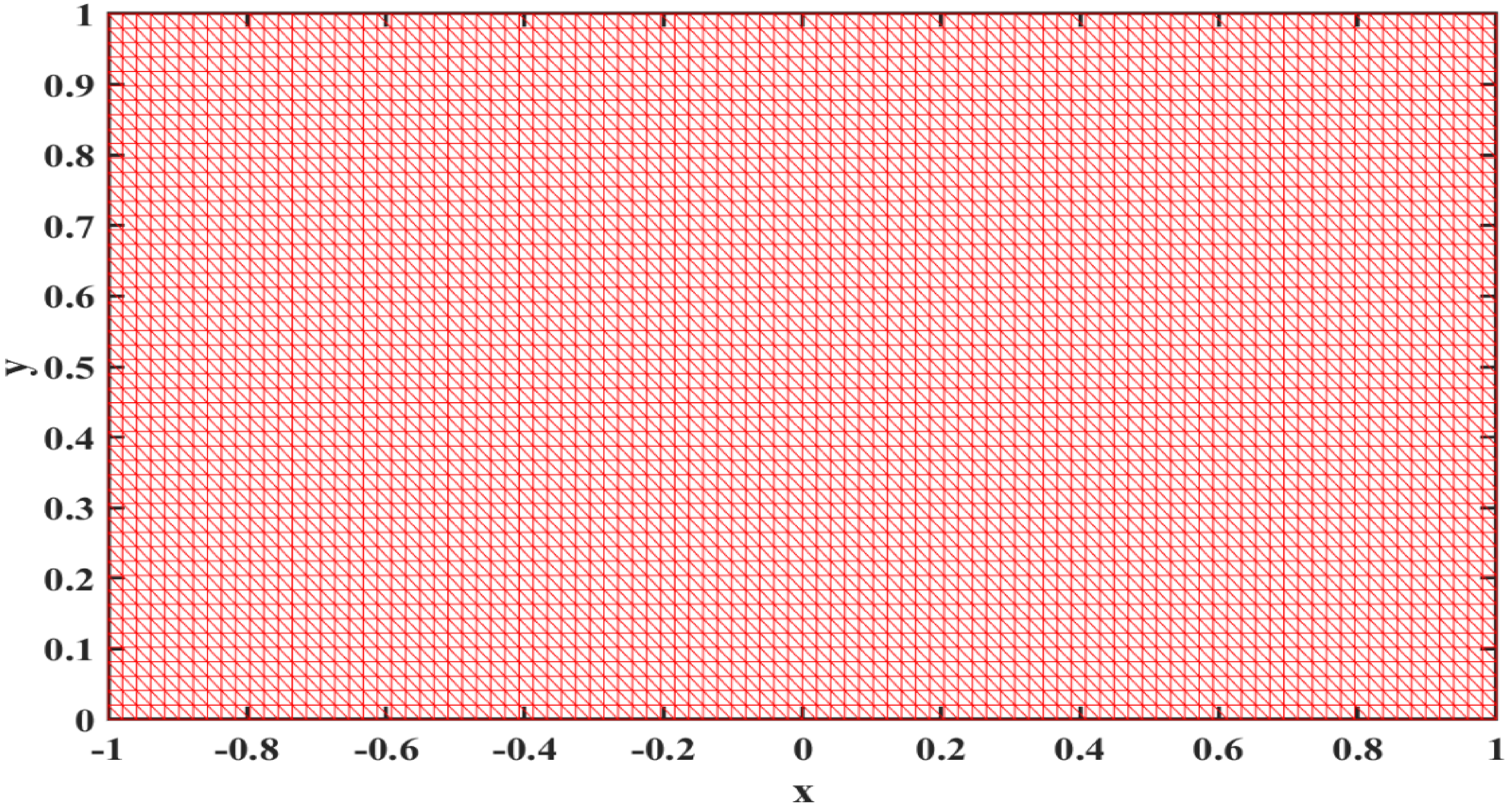
Therefore the solution is

$$
v = \tilde{r} / \tilde{K}
$$

Program organization

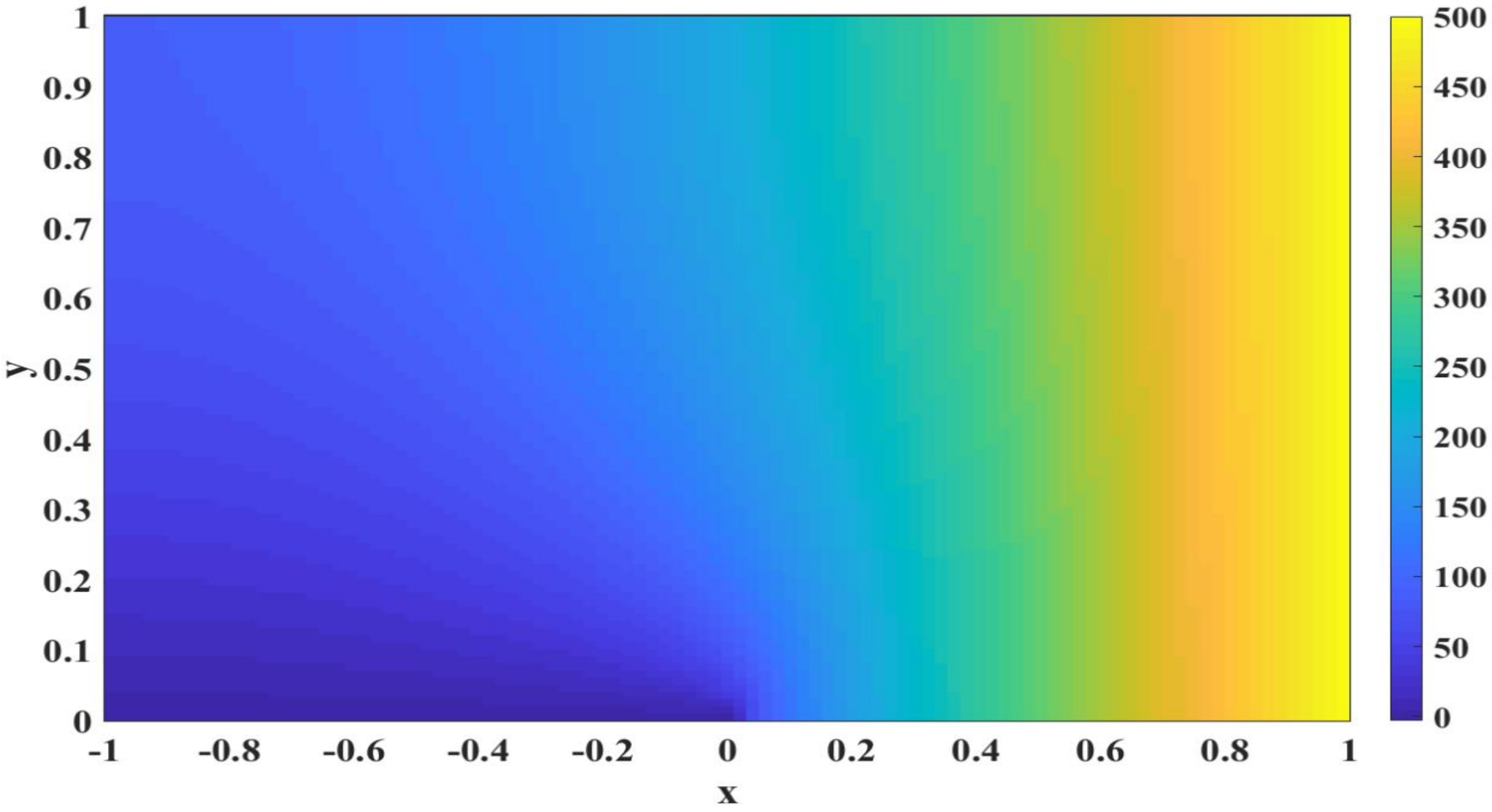
```
function PDE_Final_1()...
%=================Some Functions may be used in the Main function==========
%
% grid generation function. (wrong)
function [coord,connect,connect_number,nnode] = grid_generation(x_min,x_max,y_min,y_max,x_num
% grid generatoin iteractive function. (wrong)
function [coord,connect,connect_number,nnode] = grid_iteractive(x_min,x_max,y_min,y_max,r_num
% finite element program
function pressure = fem_Motz(coord,connect,connect_number,nnode,u_1,u_3,coeff)...
% contour of the fem solutions
function fem_contour(x_min,x_max,y_min,y_max,coord,pressure)...
% contour of the exact solutions
function exact_contour(x_min,x_max,y_min,y_max,coord,nnode)...
% compare the value of u on the position y = y_line
function compare_yline(x_min,x_max,y_min,y_max,x_number,y_number,coord,pressure,y_yline)...
% problem 1.1, pointwise error contours
function err_contour(x_min,x_max,y_min,y_max,coord,pressure,nnode)...
function derr_contour(x_min,x_max,y_min,y_max,coord,connect,connect_number,pressure)...
% problem 1.2, L2, L_inf, H1 norm
function [L2,L_inf,H1] = error_norm(coord,connect,connect_number,pressure)...
% area of a triangle
function area_element = triangle_area(edge_element)...
% function determine the value of u at the point (x1,x2)
function [u,u_x1,u_x2] = u_predict(x1,x2,x1_node,x2_node,u_node)...
% integrationpoints function for 2D, we can obtain the position and weight
function xi = abq_UEL_2D_integrationpoints(n_points, n_nodes)...
% shapefunction for 2D, we can obtain the N and dNdx.
function f = abq_UEL_2D_shapefunctions(xi,n_points,n_nodes)...
% stiffness function, we can obtain the stiffness matrix==
function kel = elstif(coord,coeff,n_nodes,n_points)...
% Exact solution function, we can obtain the exact solution at the point (x,y)
function s = Motz(x,y)...
%=================================end=====================================
```

Motz Problem –Results Discussion – Uniform Grid



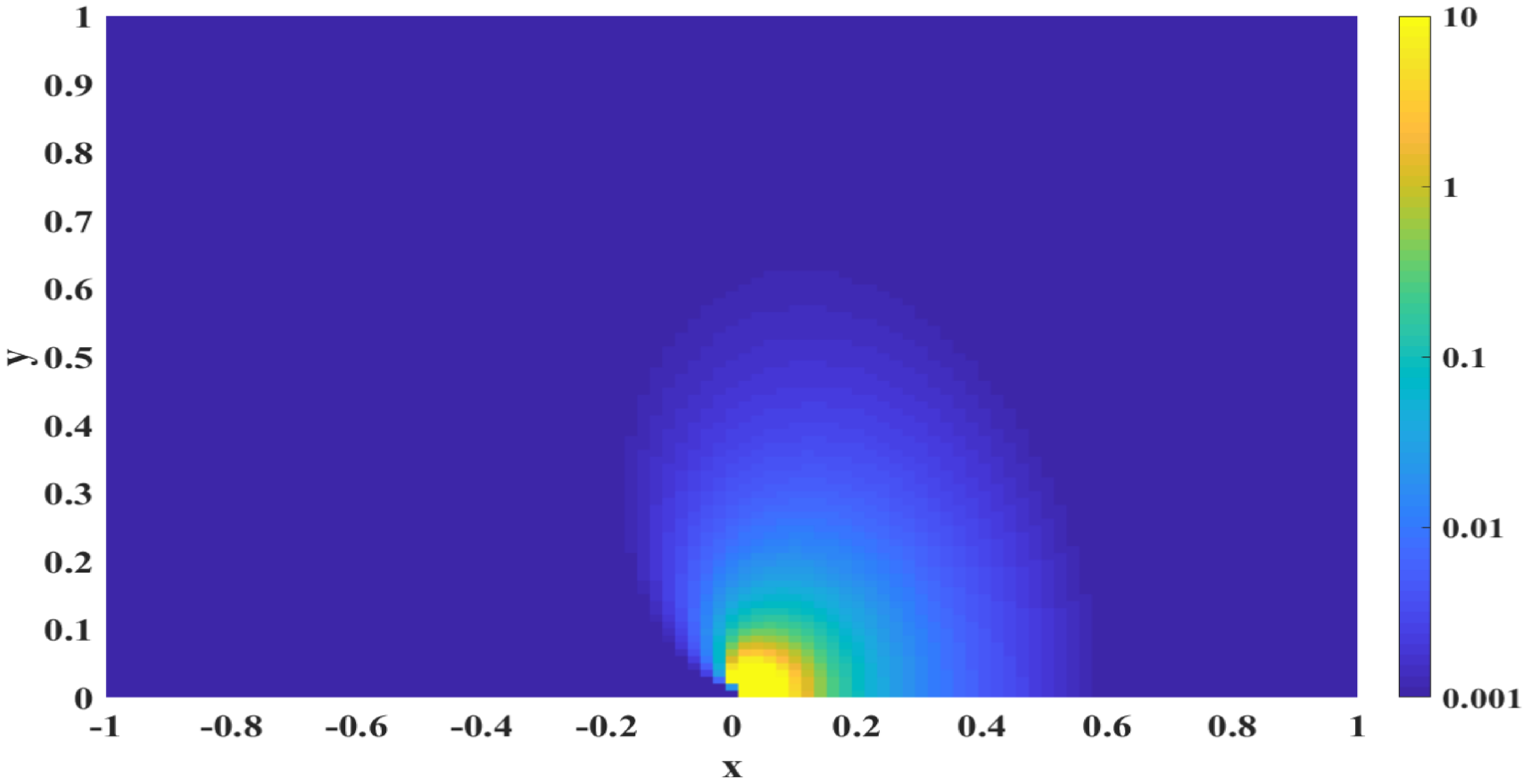
Uniform triangular grid (Delaunay in Matlab)

Motz Problem –Results Discussion – FEM Solution Contour



Contour of the exact solution

Motz Problem –Results Discussion – Error Contour



Contour of the pointwise error (Error level is high near the origin)

Motz Problem –Results Discussion

Denote

$$f = u - v$$

The L2 norm is defined as

$$e_{L2} = \sqrt{\int_\Omega |f|^2 \, d\Omega}$$

The L_inf norm is defined as

$$e_{L\infty} \equiv \inf \left\{ C \geq 0 : |f| \leq C \ \text{ for almost every point in } \Omega \right\}$$

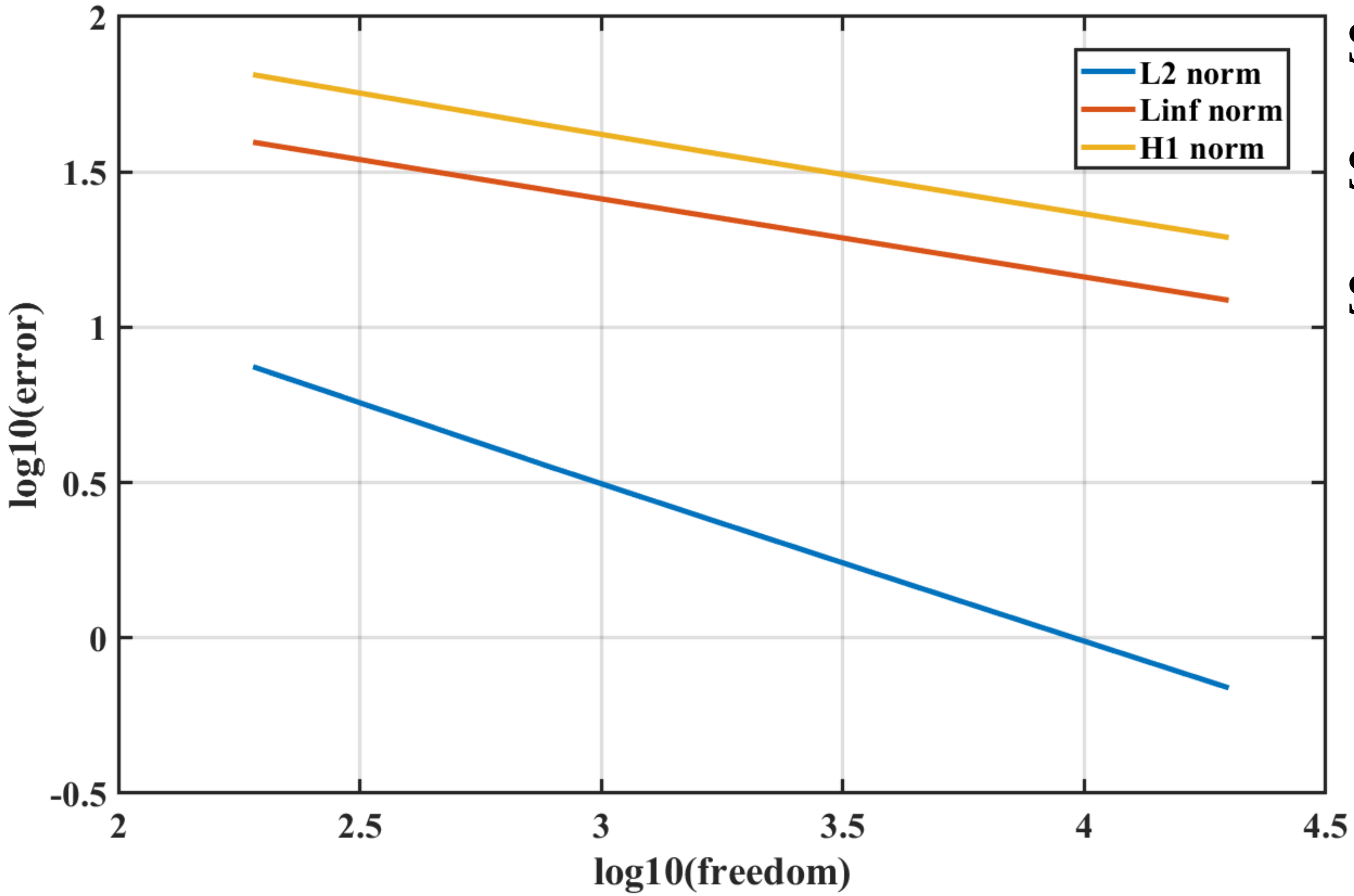
The H1 norm is defined as

$$e_{H1} = \sqrt{\int_\Omega \left[|f|^2 + |\nabla f|^2 \right] d\Omega} = \sqrt{\int_\Omega \left[f^2 + f_x^2 + f_y^2 \right] d\Omega}$$

The convergence order is defined

$$e \sim O\left(h^\alpha\right)$$

Motz Problem –Results Discussion – Uniform Grid



Error analysis for uniform refining (Motz Problem)

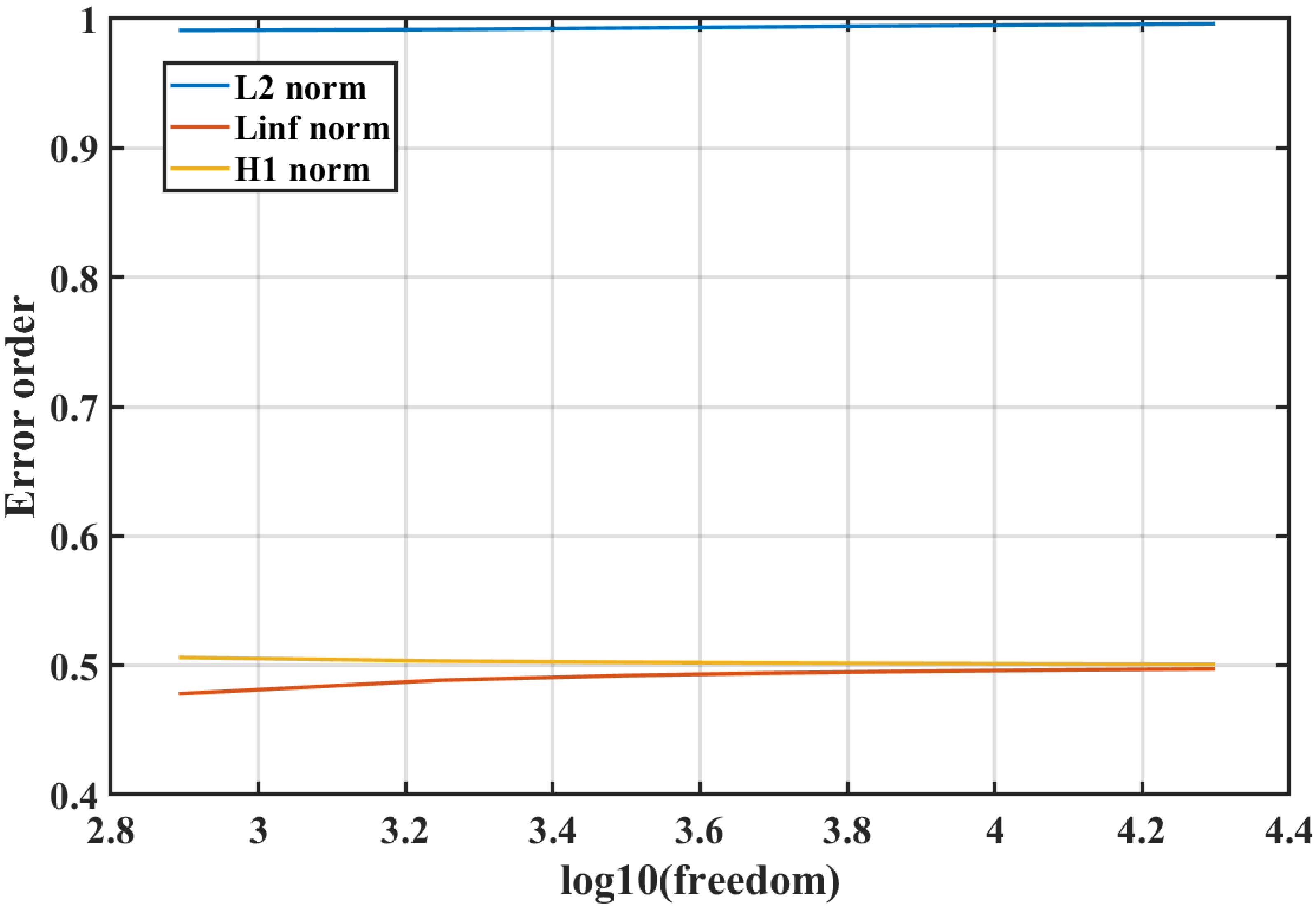
Slope of L2 is -0.5

Slope of Linf is -0.25

Slope of H1 is -0.25

- Freedom is the number of nodes in the system

Motz Problem –Results Discussion – Uniform Grid – Error Order



Error analysis for uniform refining (Motz Problem)

Order of L2 is 1

Order of Linf is 0.5

Order of H1 is 0.5

- Freedom is the number of nodes in the system

Motz Problem –Results Discussion –Local Refining

Assume the smallest increment of distance is h, the linear coarsen factor is α.

$$1 = h + \alpha h + \ldots + \alpha^{N-2} h = \frac{\alpha^{N-1} - 1}{\alpha - 1} h \Rightarrow h = \frac{\alpha - 1}{\alpha^{N-1} - 1}$$

The size of the maximum element is

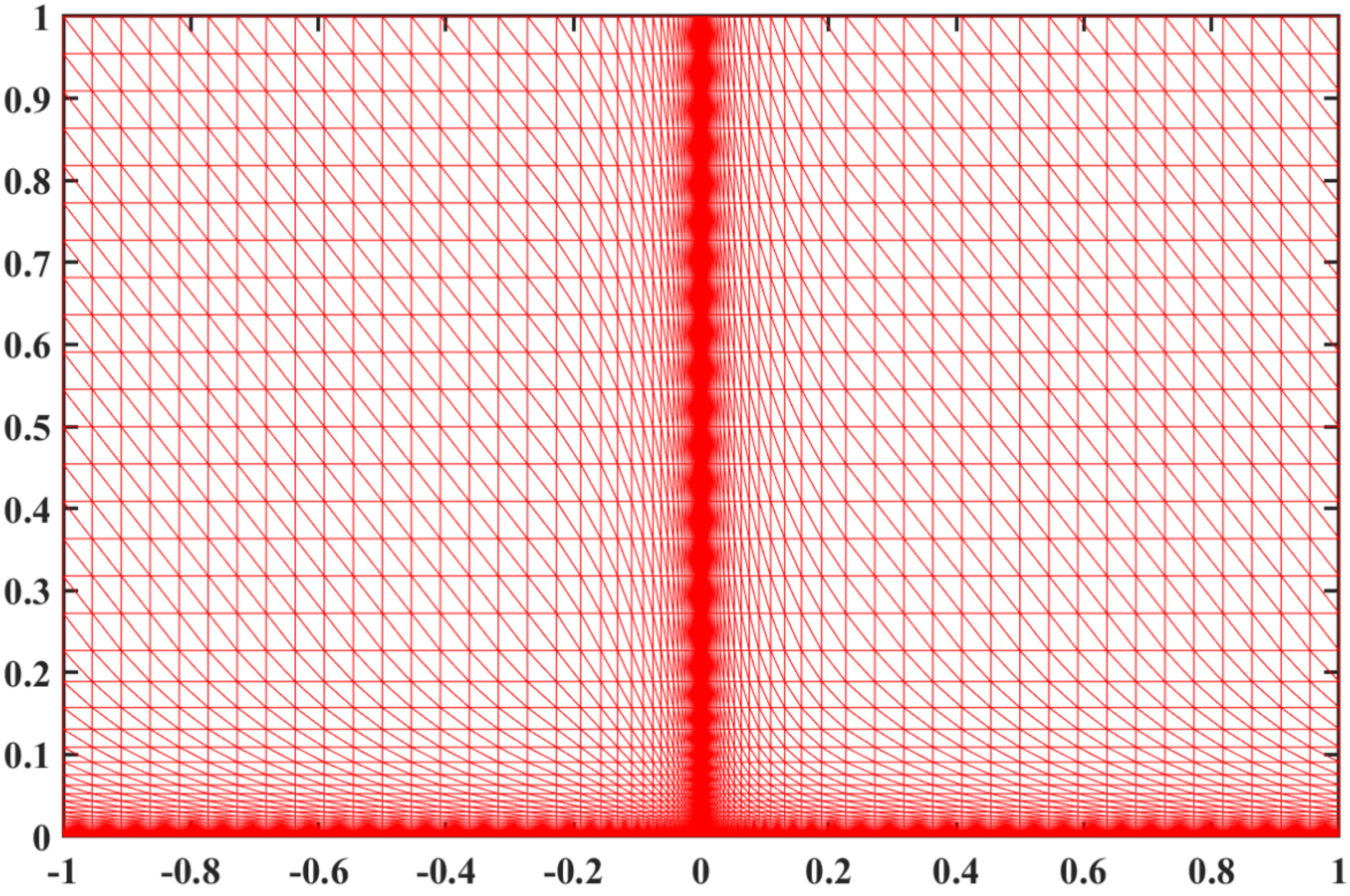
$$\lim\left(\alpha^{N-2} h\right) = \frac{\alpha^{N-1} - \alpha^{N-2}}{\alpha^{N-1} - 1} = 1 + \frac{1 - \alpha^{N-2}}{\alpha^{N-1} - 1} \approx 1 - \frac{1}{\alpha}$$

The accuracy has been “locked”

Only refine k layers

$$h + \alpha h + \ldots + \alpha^{k-1} h + \alpha^{k-1} h \cdot \left(N - 1 - k\right) = 1 \Rightarrow h = \frac{1}{\frac{\alpha^k - 1}{\alpha - 1} + \left(N - 1 - k\right)\alpha^{k-1}}$$

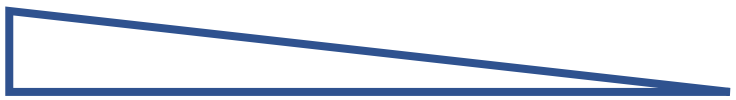
Motz Problem –Results Discussion –Local Refining –Grid



Local refine grid-1 of Motz problem

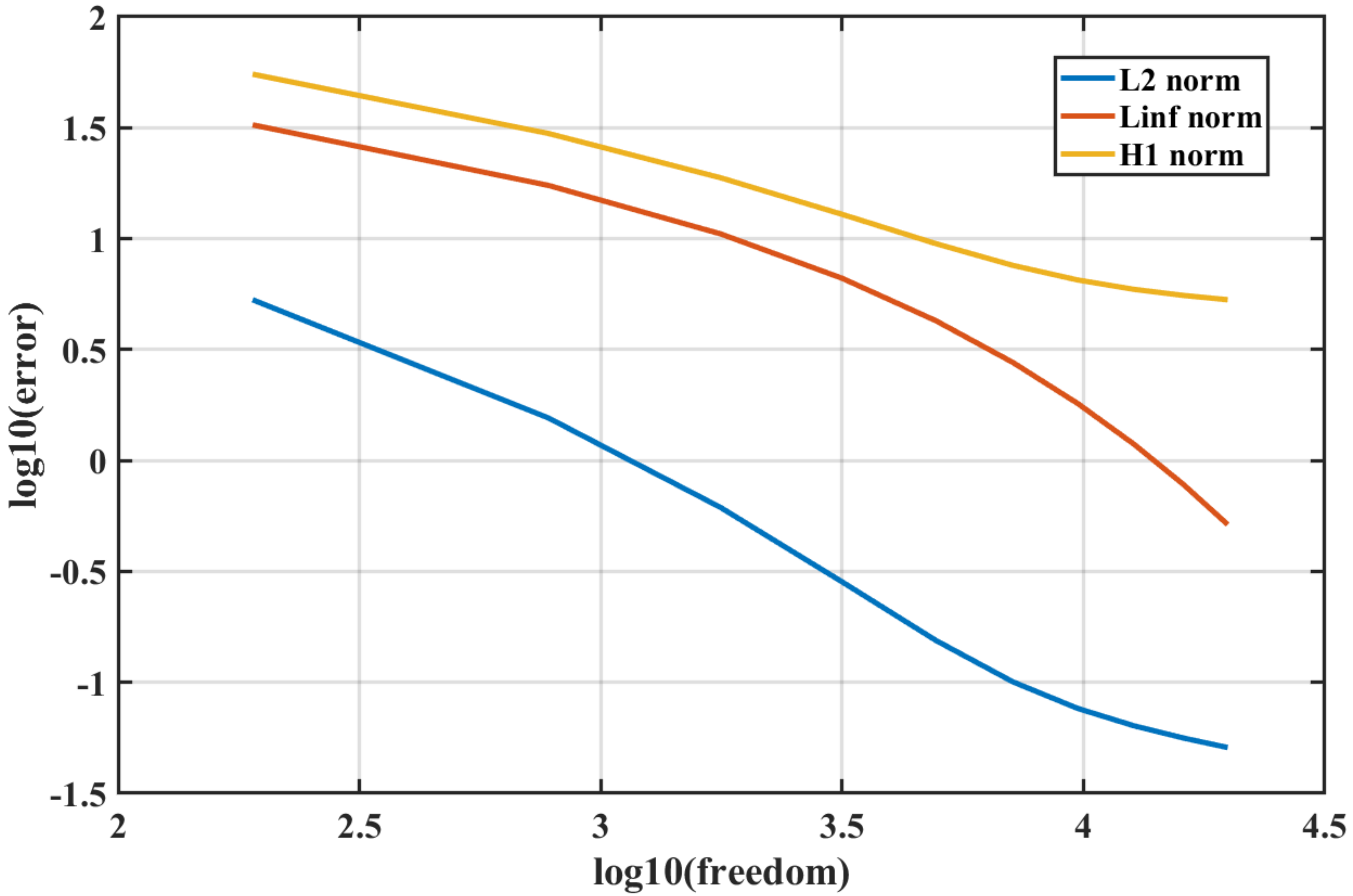
Easy to generate

Not very efficient



Accuracy depend on the longest edge

Motz Problem –Results Discussion –Local Refining



Error analysis for local refining-1 (Motz Problem)

Slope of L2 is -1

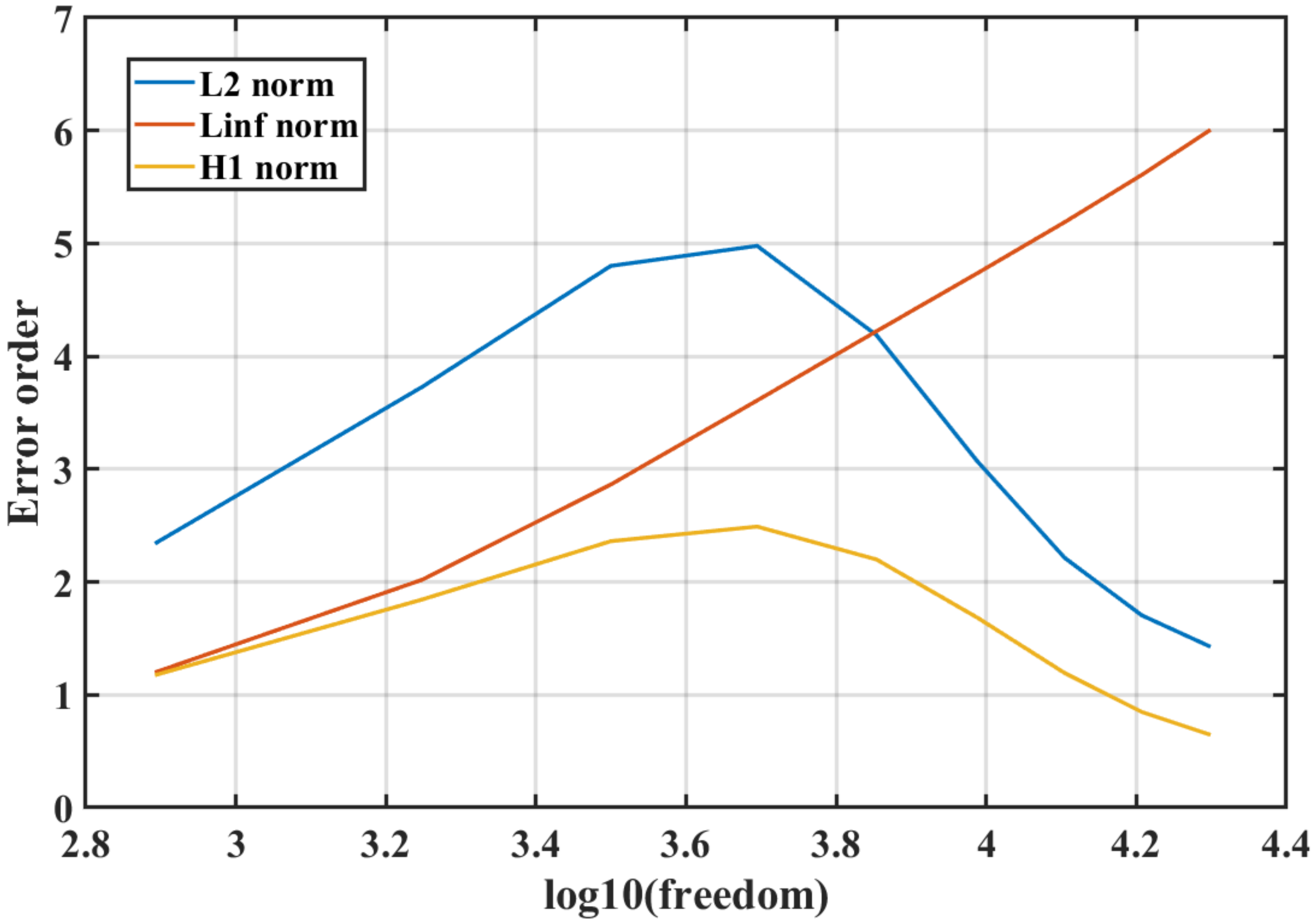
Slope of Linf is -1

Slope of H1 is -0.25

$$\alpha = 1.10, k = \frac{4N}{5}$$

Accuracy has been promoted when compared with the uniform refining

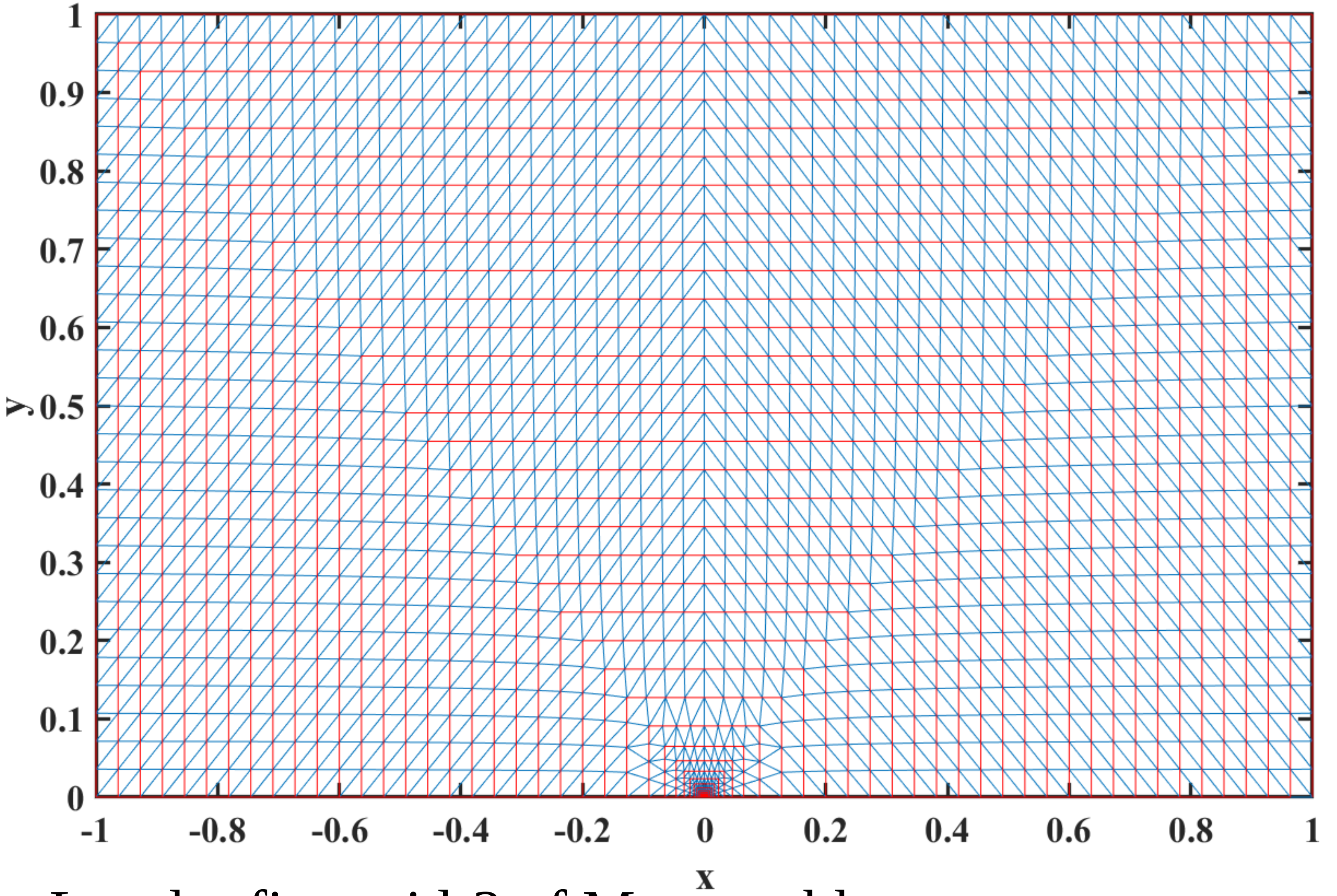
Motz Problem –Results Discussion –Local Refining – Error Order



Error analysis for local refining-1 (Motz Problem)

Error order is not constant. (Because of the local refining)
Error order has been promoted

Motz Problem –Results Discussion –Local Refining –Grid

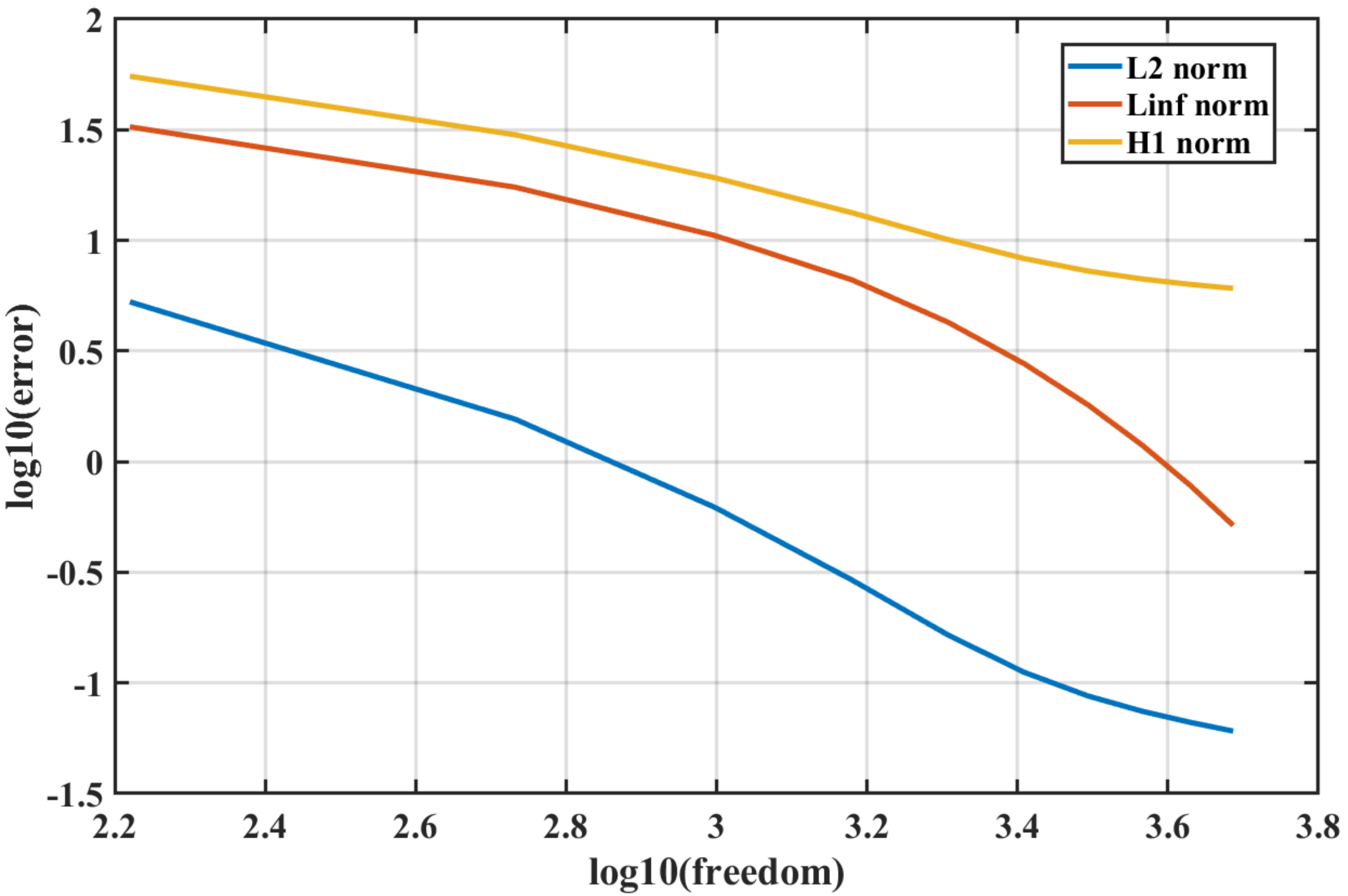


Local refine grid-2 of Motz problem

Not easy to generate

More efficient

Motz Problem –Results Discussion –Local Refining

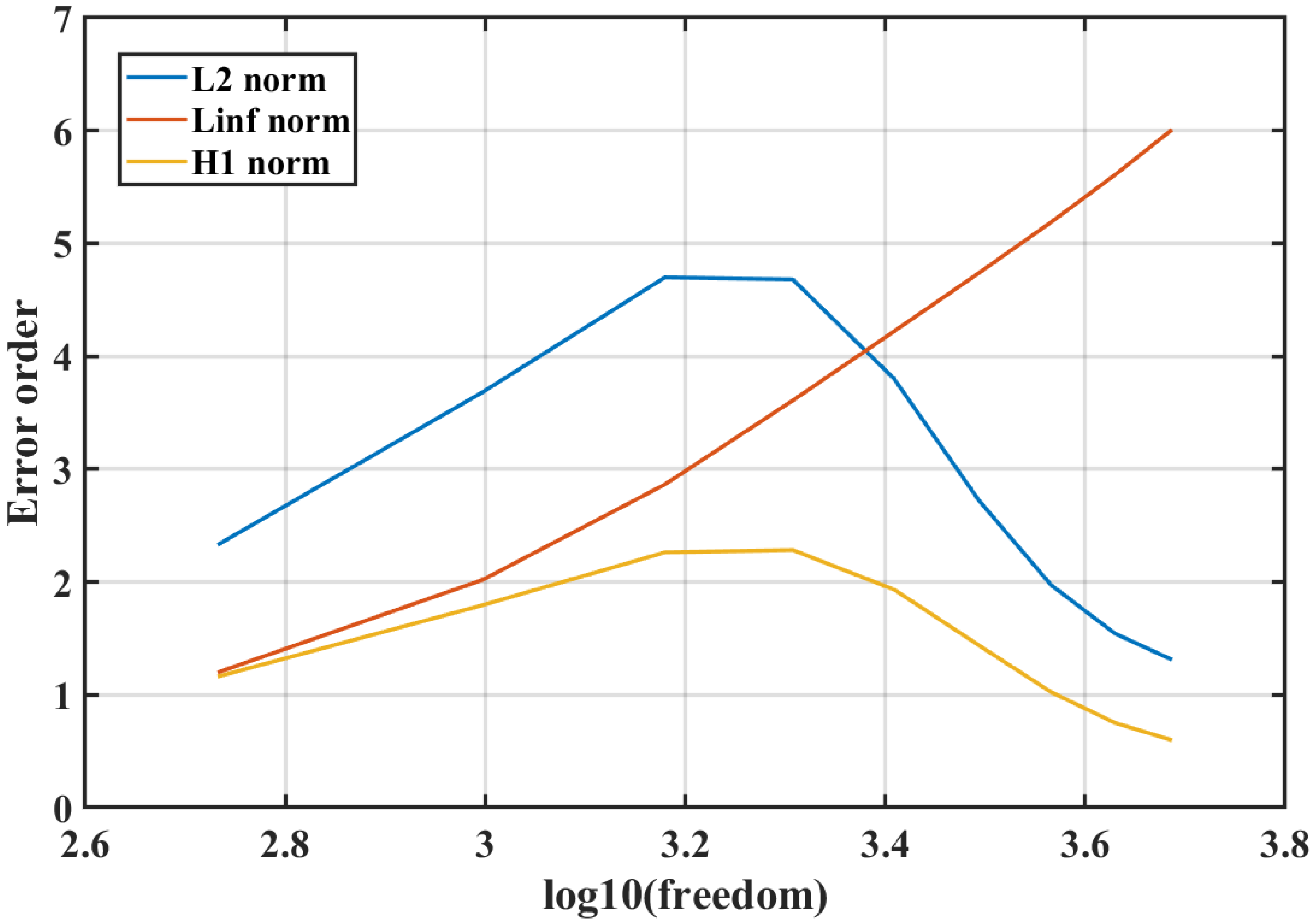


Error analysis for local refining-2 (Motz Problem)

$$\alpha = 1.10, k = \frac{4N}{5}$$

The second refinement method is better than the first one. The error of the system depends on the longest line of the elements. Local refining grid can help to promote the accuracy of the system.

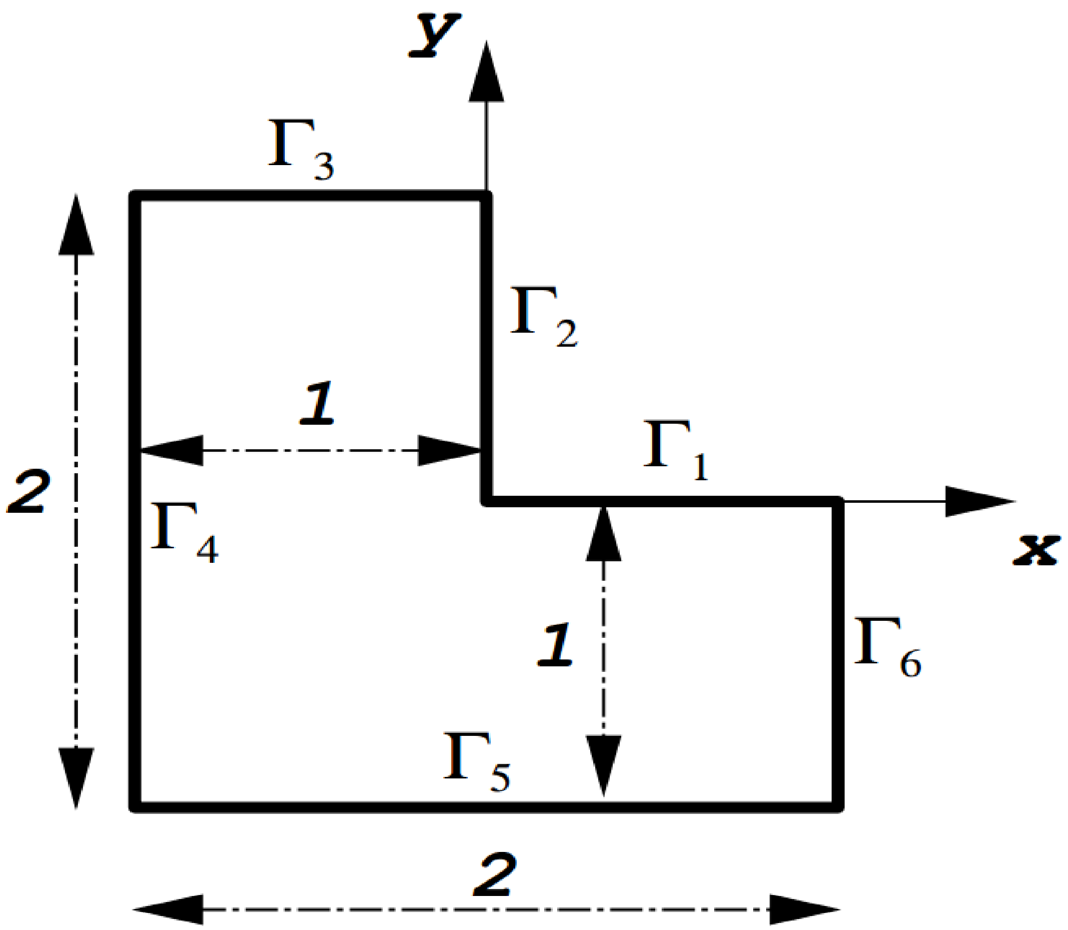
Motz Problem –Results Discussion –Local Refining – Error Order



Error analysis for local refining-2 (Motz Problem)

Error order is not constant. (Because of the local refining)
Error order has been promoted

L shape Problem – Problem Description



Governing Equation

$$\nabla^2 u = -1$$

Dirichlet boundary condition

$$u = 0 \quad \text{at } \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_5, \Gamma_6$$

L shape Problem – Mathematical background

Take the variation on both side

$$\delta I = \iint \left(\nabla^2 v + 1 \right) \delta v d\Omega = 0$$

According to Divergence Theorem

$$\delta I = -\frac{1}{2} \delta \iint \left| \nabla v \right|^2 d\Omega + \iint \delta v d\Omega = \delta \iint v - \frac{1}{2} \left| \nabla v \right|^2 d\Omega$$

The functional is

$$I = \iint \left(v - \frac{1}{2} \left| \nabla v \right|^2 \right) d\Omega$$

L shape Problem– Finite Element Analysis – Integration points

The functional in a single element

$$I\left(u_e\right)=\sum_{k=a}^{c} v^{(k)} \iint N^{(k)} d\Omega - \frac{1}{2}\sum_{i=a}^{c}\sum_{j=a}^{c} v^{(i)} v^{(j)} \iint \nabla N^{(i)} \nabla N^{(j)} d\Omega$$

We can define $K_{ij} = \iint \nabla N_i \nabla N_j d\Omega$ $\qquad r_k = \iint N^{(k)} d\Omega$

$$I\left(u_e\right)=r_e^T v_e - \frac{1}{2} v_e^T K_e v_e$$

Assemble all elements in the system

$$I\left(u\right)=r^T v - \frac{1}{2} v^T K v$$

To minimize the functional, we have

$$Kv = r^T$$

L shape Problem– Finite Element Analysis – Galerkin Method

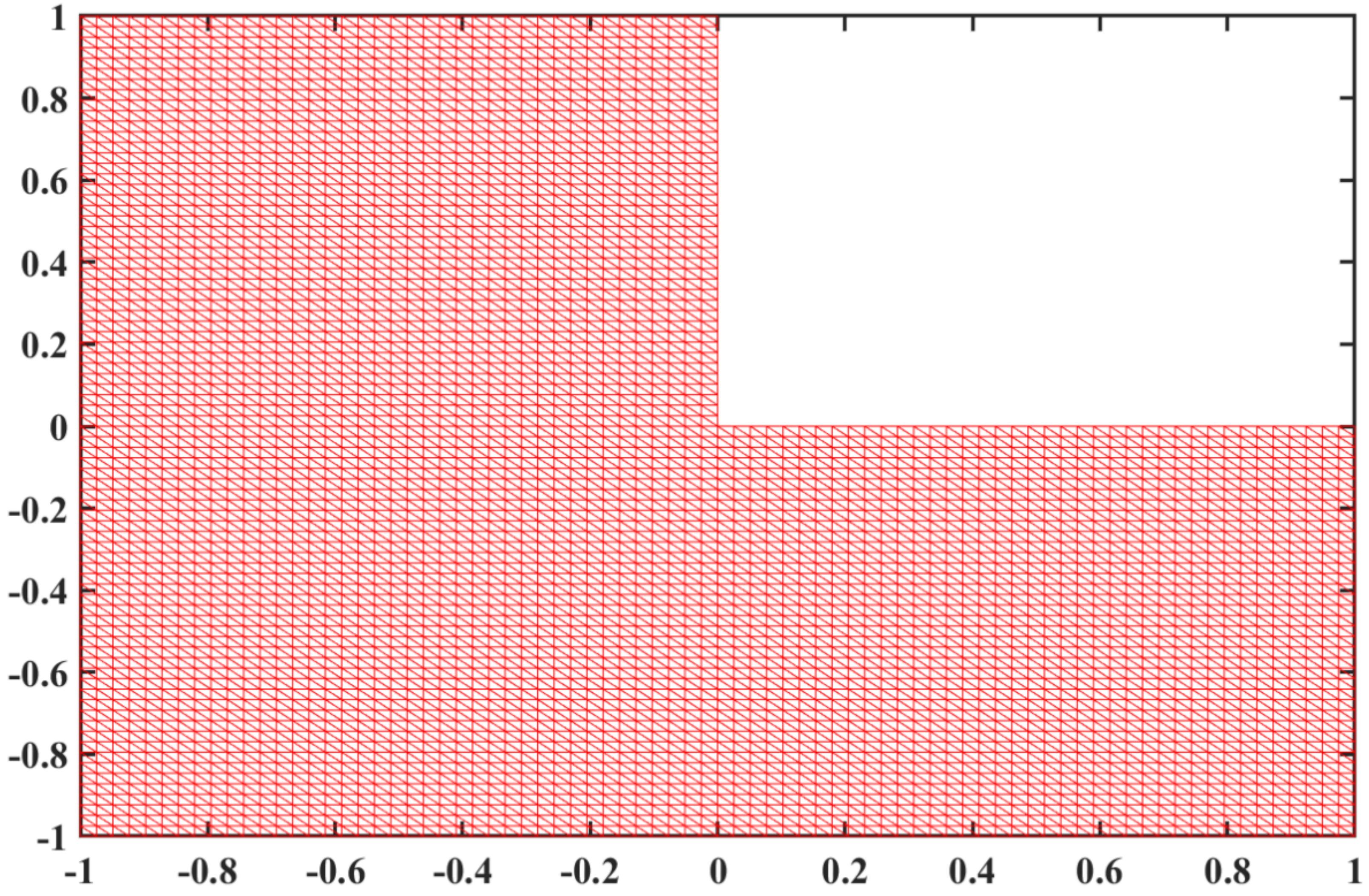
Denote $\widehat{B}_k = \begin{bmatrix} N_k^a \\ N_k^b \\ N_k^c \end{bmatrix}$

Therefore $r_e = \sum_{k=1}^{3} \widehat{B}_k w_k \eta_k$

Assemble the stiffness matrix and residual vector

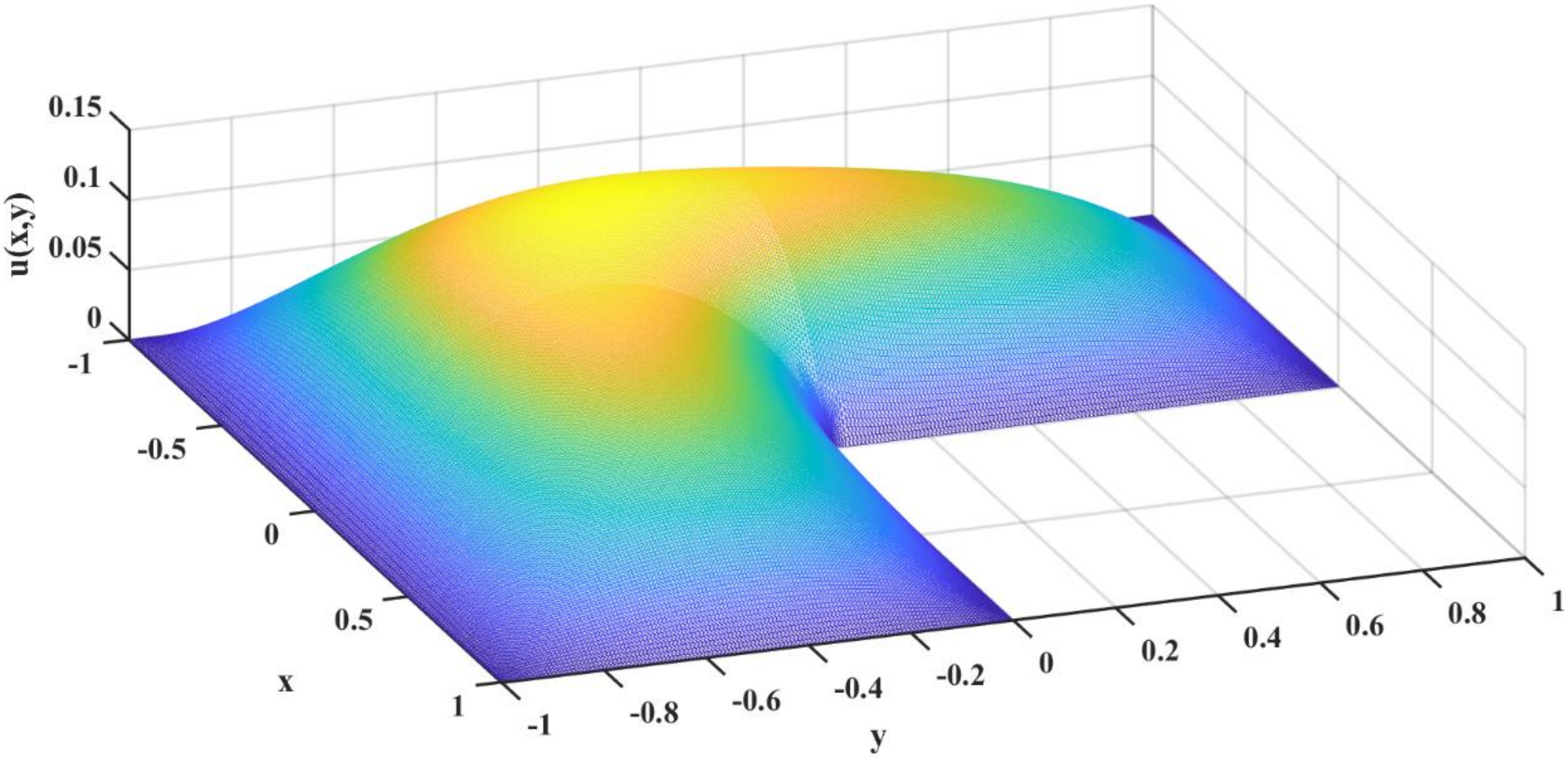
$$\begin{bmatrix} k_{11} & k_{12} & ... & k_{1N} \\ k_{21} & k_{22} & ... & k_{2N} \\ ... & ... & ... & ... \\ k_{N1} & k_{N2} & ... & k_{NN} \end{bmatrix} \begin{bmatrix} v^{(1)} \\ v^{(2)} \\ ... \\ v^{(N)} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ ... \\ r_k \end{bmatrix} \Rightarrow \begin{bmatrix} k_{11} & 0 & ... & k_{1N} \\ 0 & 1 & ... & 0 \\ ... & ... & ... & ... \\ k_{N1} & 0 & ... & k_{NN} \end{bmatrix} \begin{bmatrix} v^{(1)} \\ v^{(2)} \\ ... \\ v^{(N)} \end{bmatrix} = \begin{bmatrix} r_1 - k_{12} v_D^{(2)} \\ v_D^{(2)} \\ ... \\ r_N - k_{N2} v_D^{(2)} \end{bmatrix}$$

L shape Problem –Results Discussion –Uniform Grid



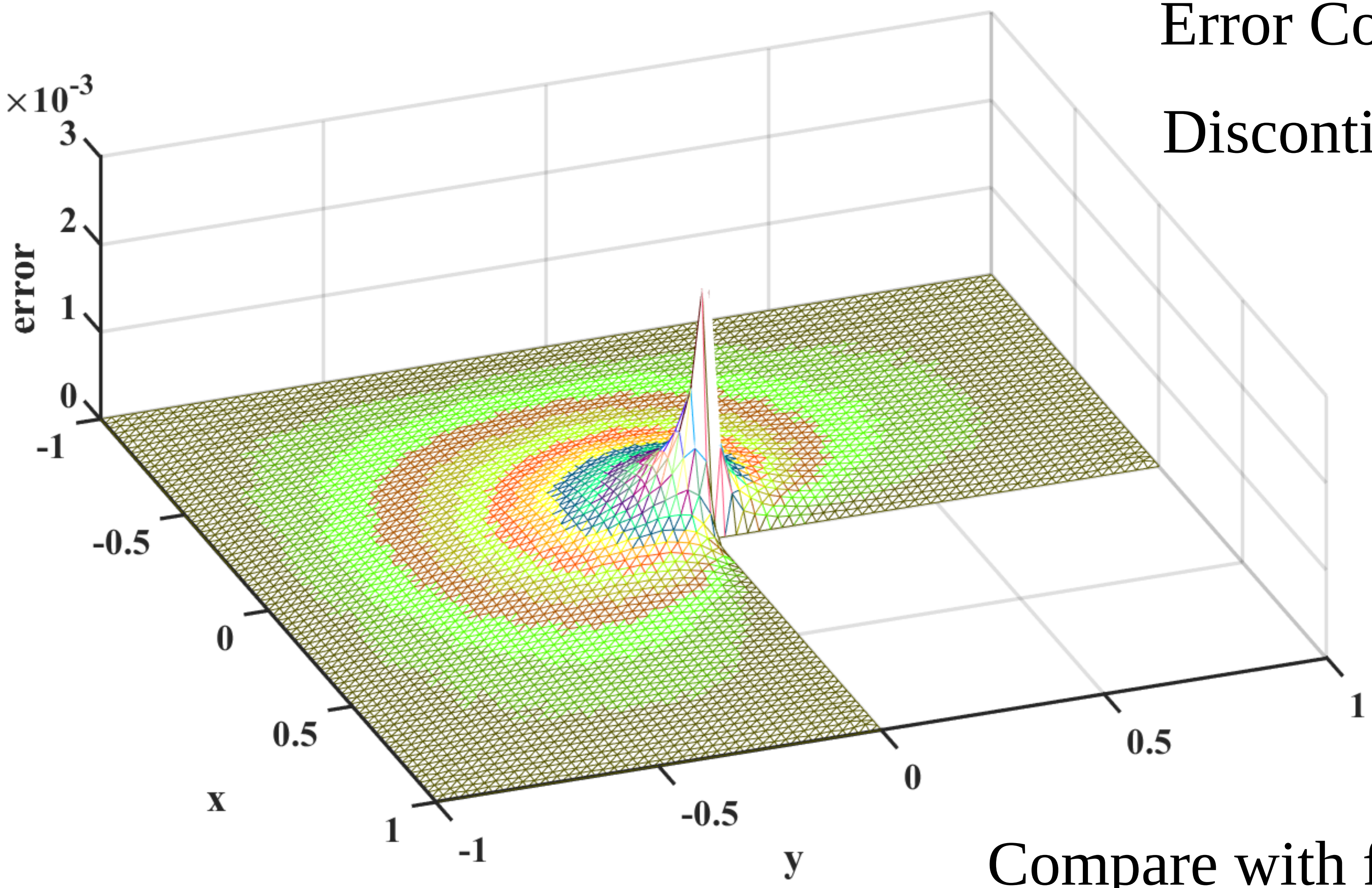
Uniform gird for L shape problem

L shape Problem –Results Discussion –FEM Solution Contour



Contour of the exact solution (55588 nodes)

L shape Problem –Results Discussion –Error Contour –1



Error Contour

Discontinuous point: max error

Compare with fine discretization data

Contour of the pointwise error – (Method 1 – direct comparing)

L shape Problem –Results Discussion – Exact Solution

Exact solution

$$u_H\left(r,\theta\right)=\sum_{k=1,3,\ldots} A_k r^{2k/3}\sin\left(\left(2\theta-\pi\right)k/3\right)$$

$$u_p\left(r,\theta\right)=\frac{-r^2}{6\pi}\left[\frac{3\pi}{2}-2\ln r\sin\left(2\theta\right)-\left(2\theta-\frac{5\pi}{2}\right)\cos\left(2\theta\right)\right],\theta\in\left[\frac{1}{2}\pi,2\pi\right]$$

$$u=u_H+u_p$$

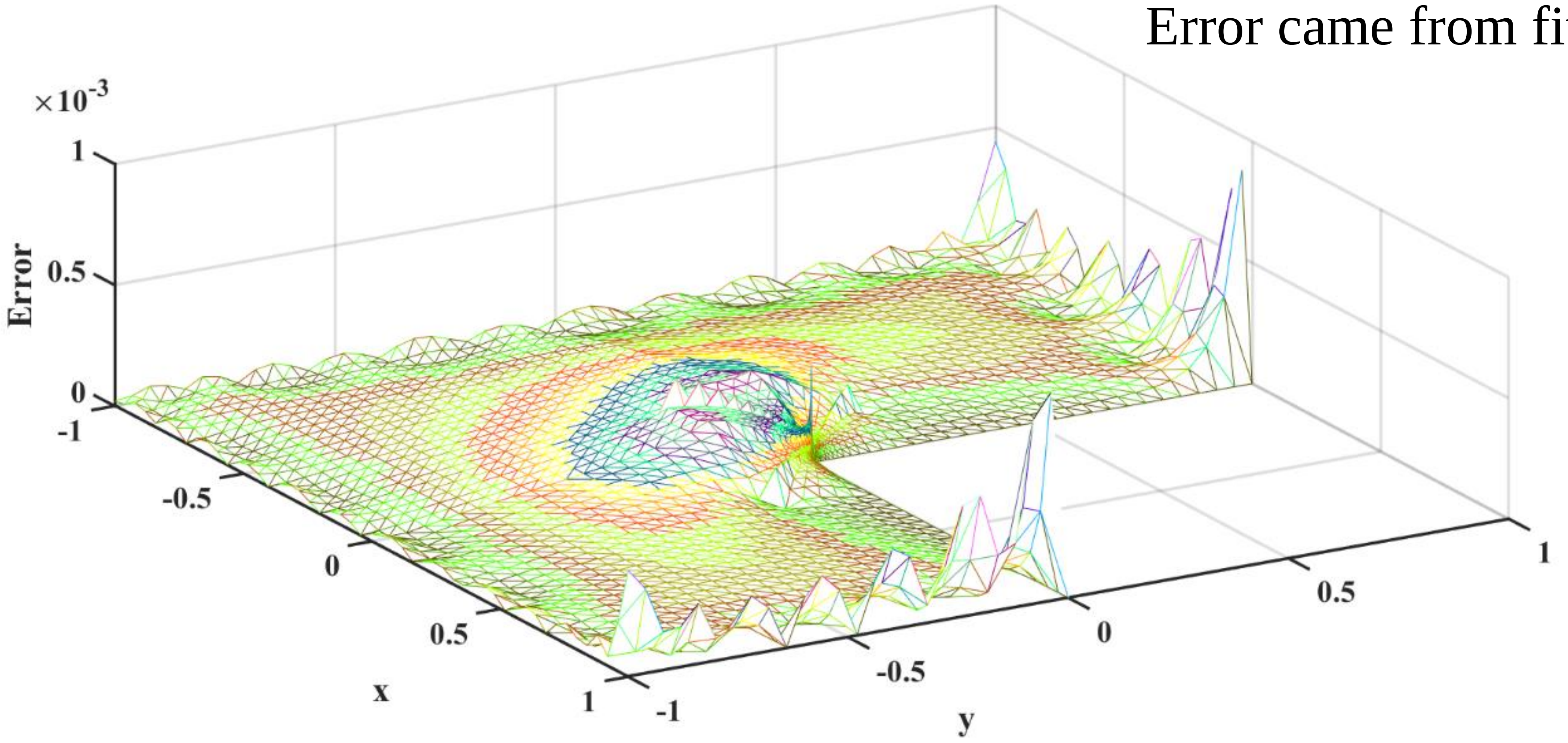
We can obtain the coefficient by linear regression, the coefficient matrix is

L shape Problem –Results Discussion –Coefficient Matrix

1	2	3
0.401924749945678	0.093647322392899	-0.009382266873999
4	5	6
-0.029886646782791	-0.008358994236376	-0.004727929057437
7	8	9
-0.001541324424179	-0.001092134923554	-0.000697767541724
10	11	12
-0.000538726404143	-0.000366251120773	-0.000232513632145
13	14	15
-0.000154344089805	-0.000112407286977	-0.000051702524179
16	17	18
-0.000033989886018	-0.000023728176110	-0.000005727204713
19	20	
-0.000003045494283	-0.000002103173041	

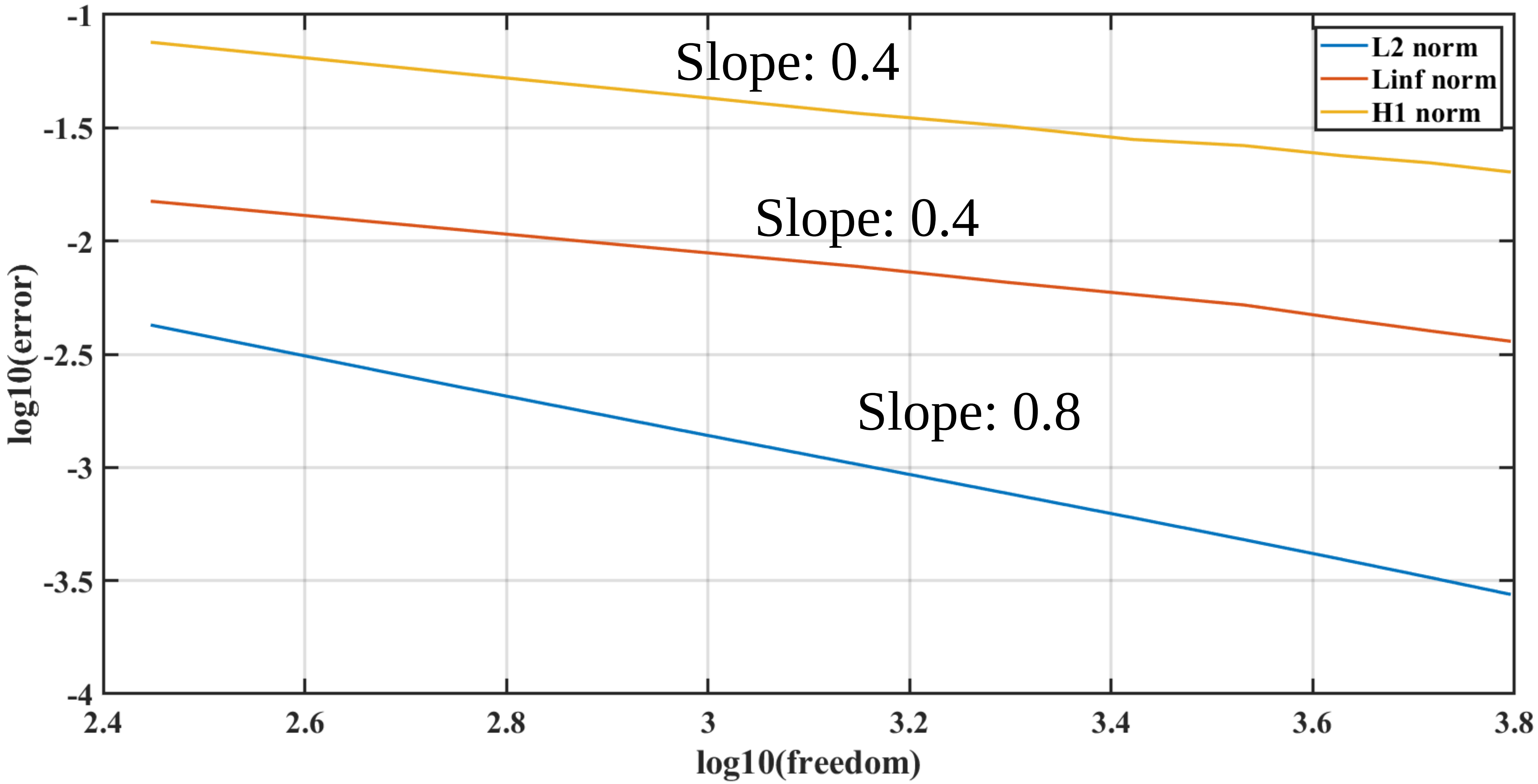
L shape Problem –Results Discussion – Error Contour

Error came from fitting



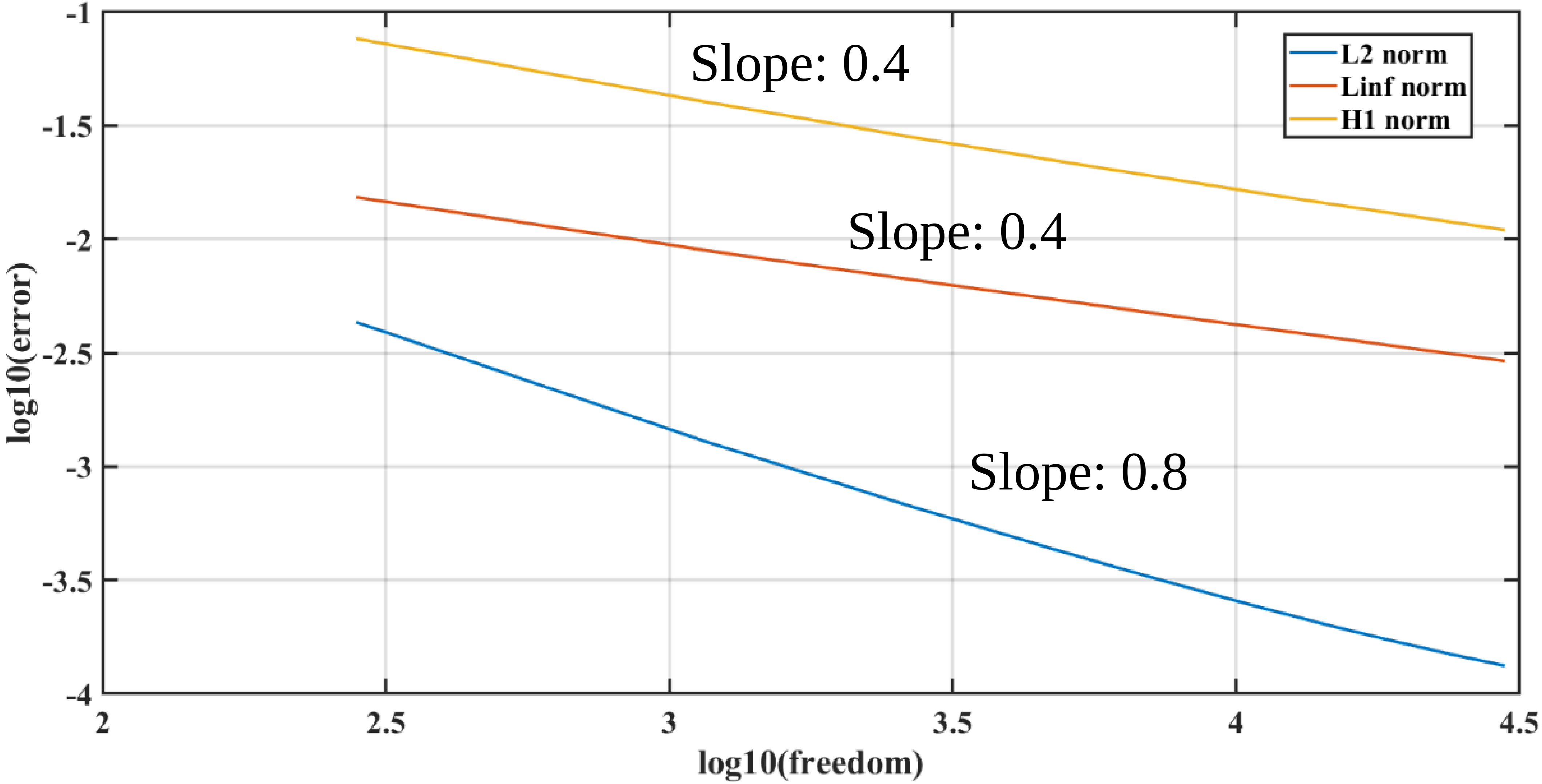
Contour of the pointwise error – (Method 2 – coefficient matrix)

L shape Problem –Results Discussion – Uniform Grid



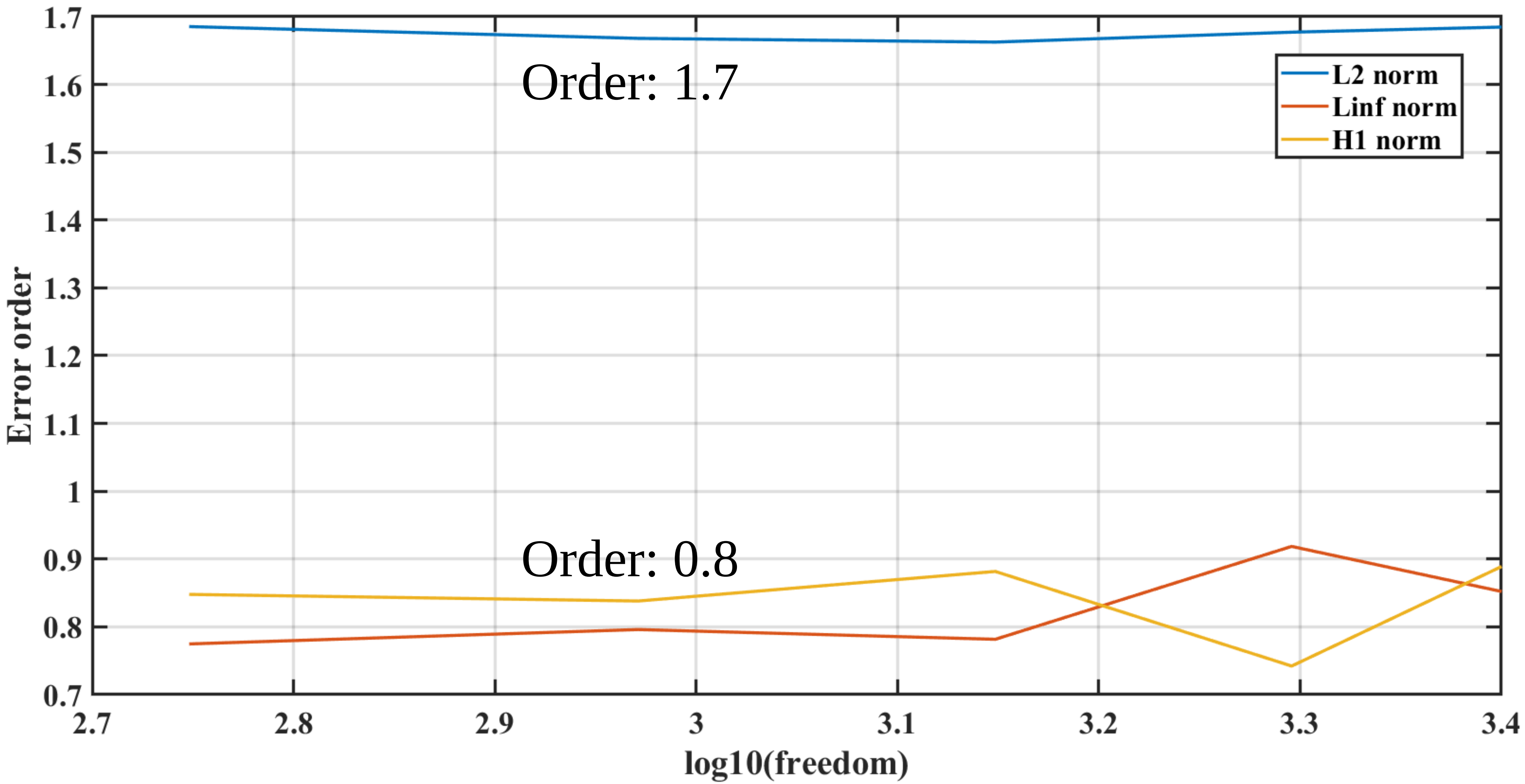
Error analysis for uniform refining (Method - 1)

L shape Problem –Results Discussion – Uniform Grid



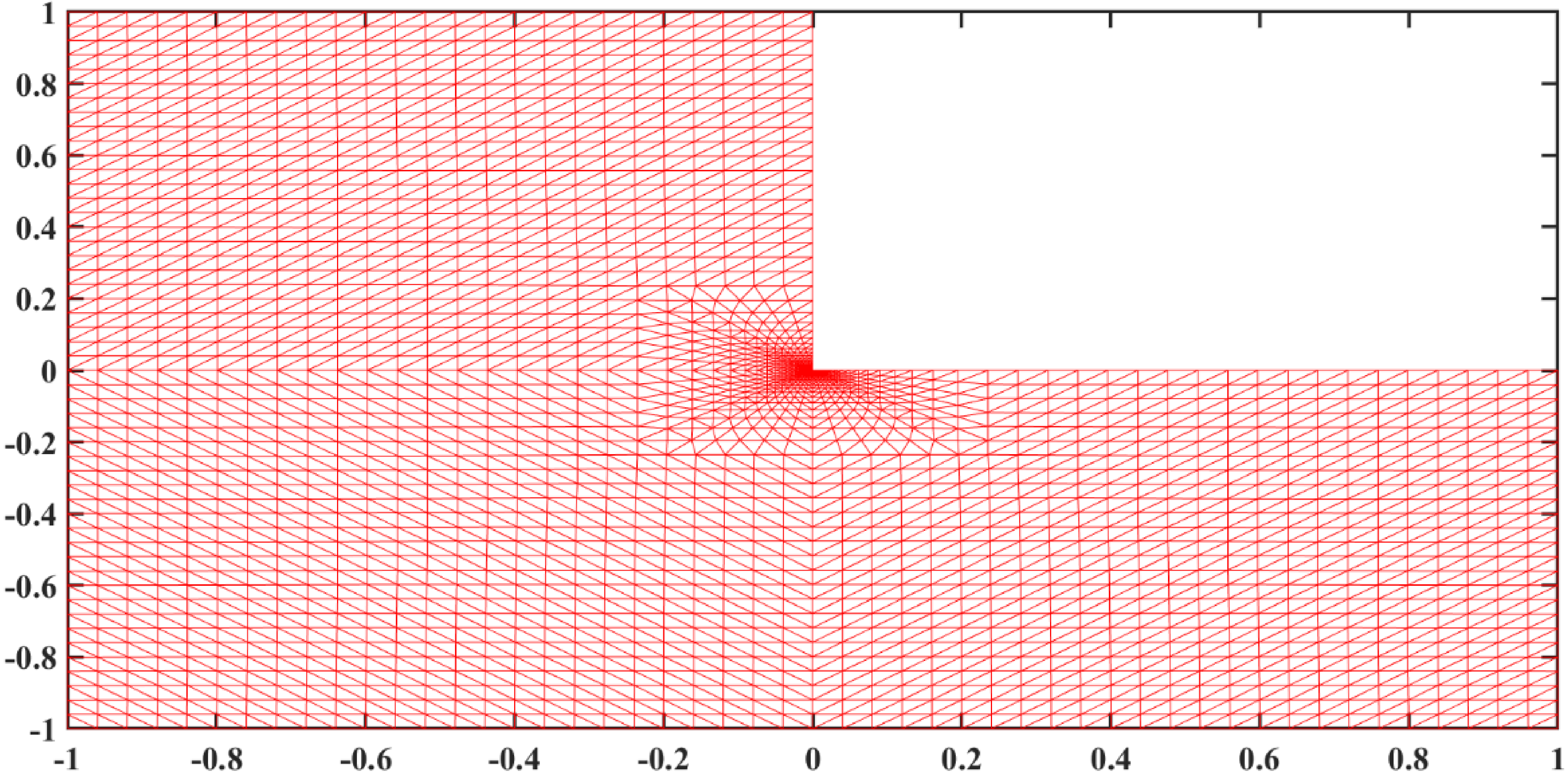
Error analysis for uniform refining (Method - 2)

L shape Problem –Results Discussion – Uniform Grid – Error Order



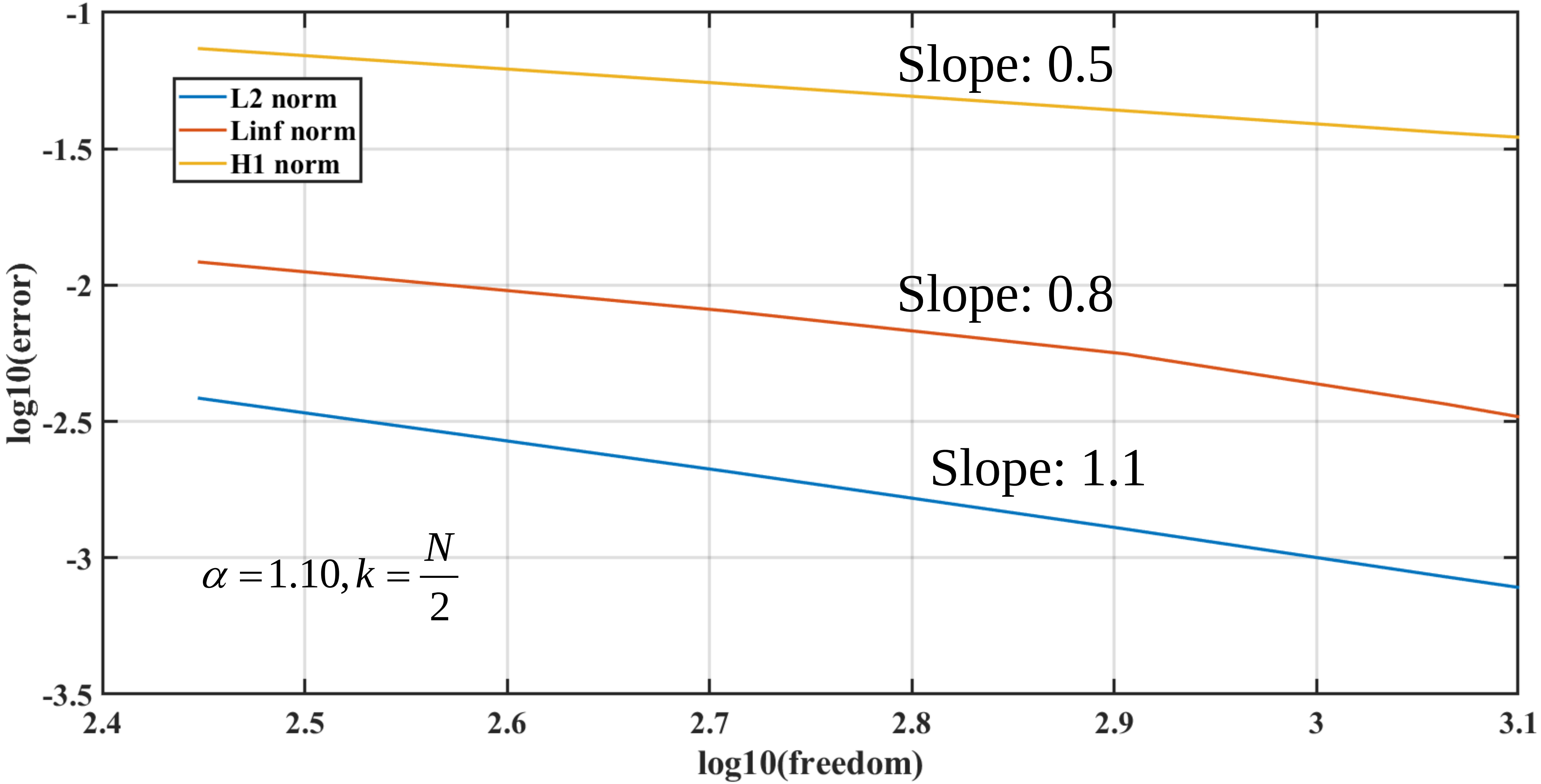
Error analysis for uniform refining (Method - 1)

L shape Problem –Results Discussion –Local Refining



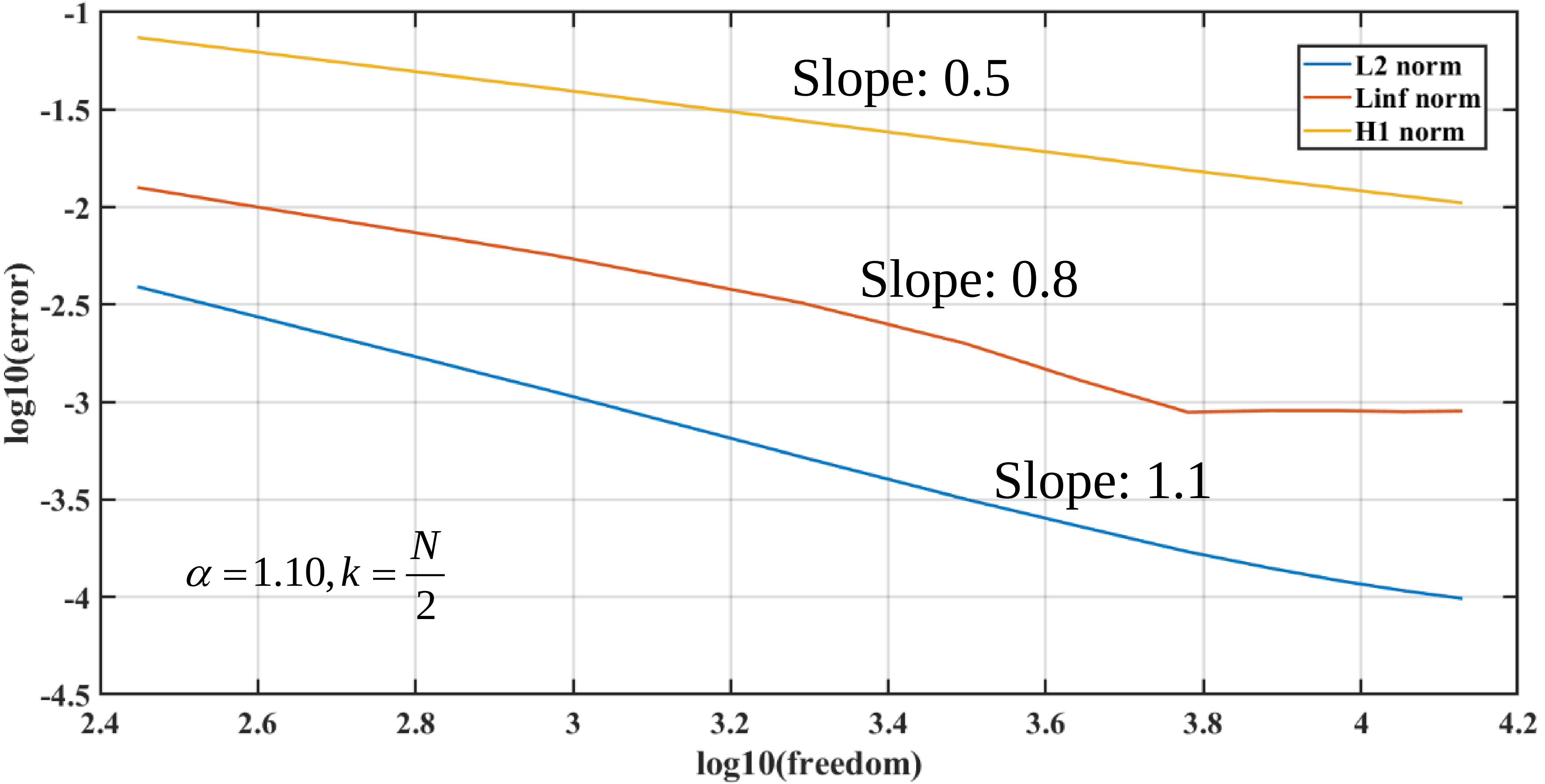
Local refine grid of L shape problem

L shape Problem –Results Discussion – Local Refining



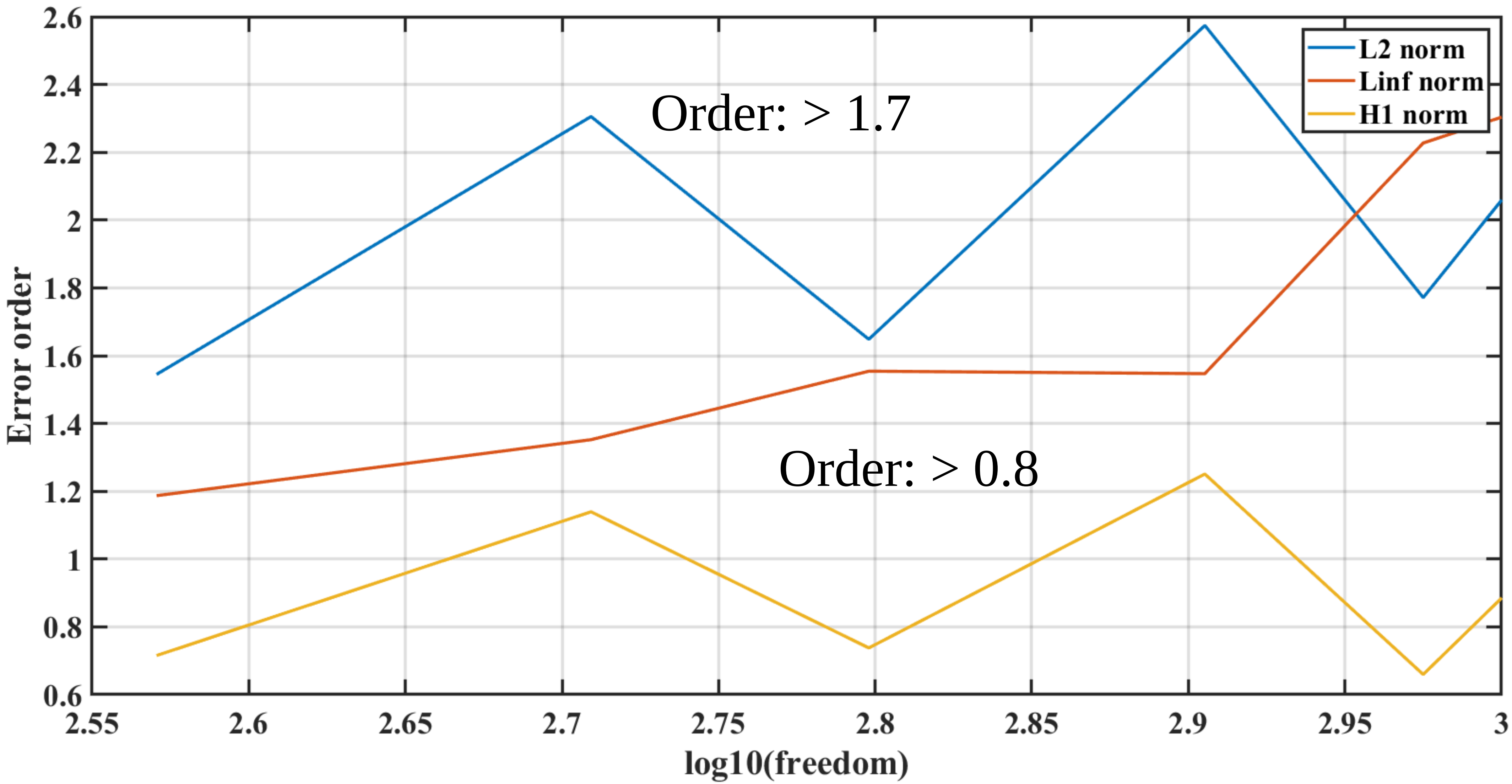
Local refine grid of L shape problem (Method - 1)

L shape Problem –Results Discussion –Local Refining



Local refine grid of L shape problem (Method - 2)

L shape Problem –Results Discussion –Local Refining – Error Order



Local refine grid of L shape problem (Method - 2) Order

Summarize

1. Use 3-node 3-integration point linear triangular element in the FEM analysis

2. For Motz Problem, the error order of L2 norm is 1, the error order of Linf norm and H1 norm is 0.5

3. For L shape Problem, the error order of L2 norm is 1.7, the error order of Linf norm and H1 norm is 0.8

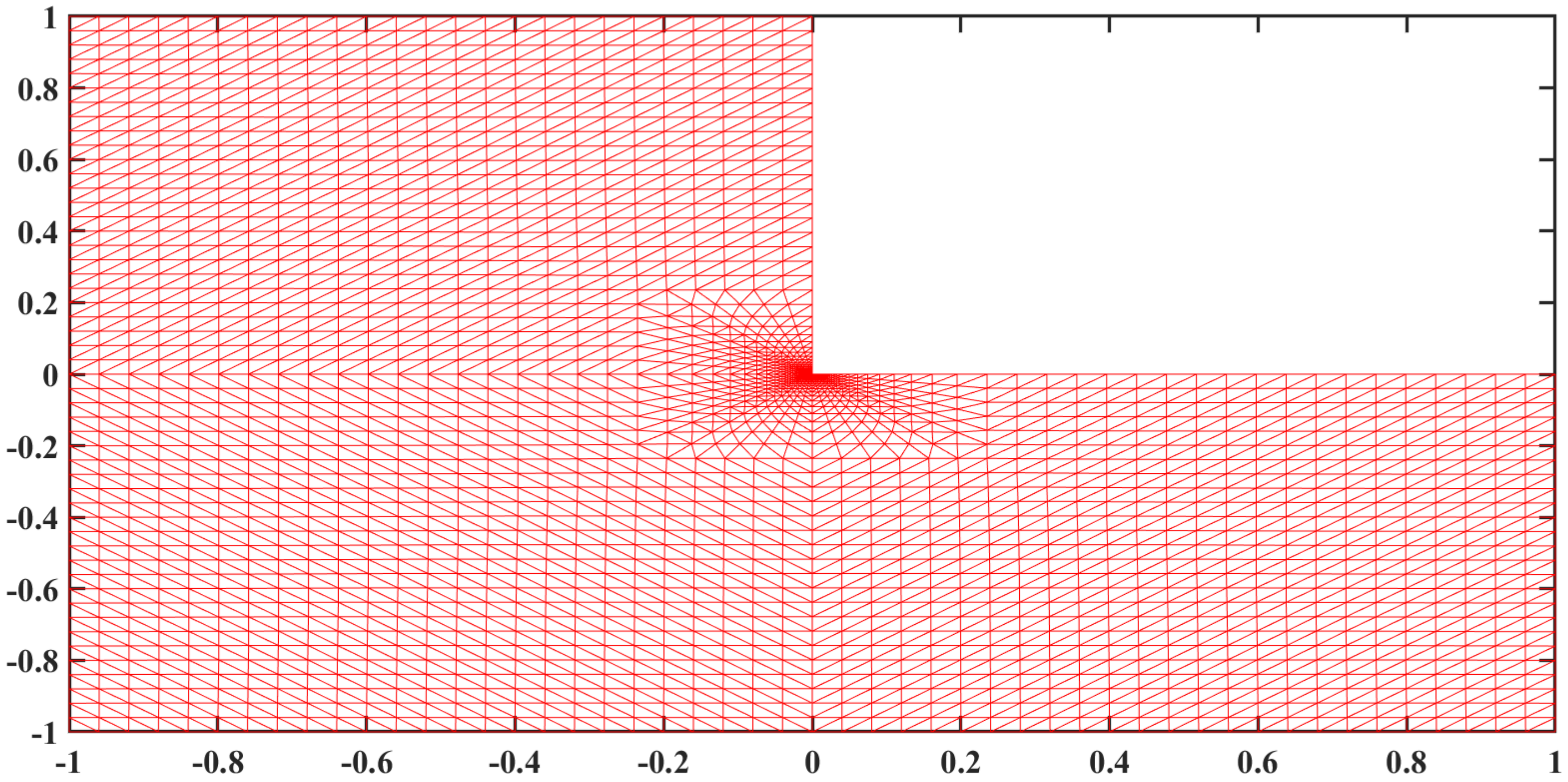
4. Local refining grid can promote the accuracy of the finite element method.

5. Both the refining coefficient and refining layer have effect on the calculation.

Thanks for Watching

Further Study –

Given a point (x,y), which element it belongs to? (nonuniform grid)



Geohash algorithm