

C++とSFMLとQtで数値解析 ソフトウェアを開発した話

23 Hogaraka

有限要素法 って？

構造を解析するために使われた数値
解析手法

メリット 複雑な形状に対応できる

デメリット 大局的な保存則しか満た
していない

2次元ポアソン方程式をC++でQtと
SFMLを用いて可視化したことを記す

数値解析について補足

数値解析とは手で解いて求める解析
的な解を得るのが困難である場合に
代わりに関数列(定数値の配列を
関数とみたもの)を求める手法

求める方程式の性質と導出

$$r(x) = \Delta u(x) + f(x)$$

$$\int_{\Omega} \omega(x) r(x) d\Omega = 0$$

有限要素法を重み付き残差法で解析
重み関数はガラーキン法を用いて
ポアソン方程式を求める。

有限要素法につかう最終的な式

$$A_{ij} = \int_{\Omega} \nabla \phi_i \cdot \nabla \phi_j d\Omega$$

$$\mathbf{b}_i = \int_{\Omega} \phi_i f d\Omega + \int_{\Gamma_2} \phi_i p d\Gamma - \int_{\Omega} \nabla \phi_i \cdot \nabla u_g d\Omega$$

$$b_i = \frac{y_j - y_k}{2\Delta_e} \quad A' = \begin{pmatrix} A_{kk}^e & A_{kl}^e & A_{km}^e \\ A_{jk}^e & A_{jl}^e & A_{jm}^e \\ A_{mk}^e & A_{ml}^e & A_{mm}^e \end{pmatrix}$$

$$c_i = \frac{x_k - x_j}{2\Delta_e} \quad A_{ij}^e = \Delta_e (b_i^e b_j^e + c_i^e c_j^e)$$

ここで i, j, k は A' における添え字で $j = (i + 1) \bmod 3, k = (i + 2) \bmod 3$

$$A = \Sigma A'$$

$$\mathbf{b}_e = \begin{pmatrix} \mathbf{b}_k \\ \mathbf{b}_l \\ \mathbf{b}_m \end{pmatrix} = \frac{\Delta_e}{12} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} f_k \\ f_l \\ f_m \end{pmatrix}$$

尚、基本境界条件では $i = j$ の場合上書きし、直接 $A_{ij} = u_g$

$i \neq j$ で $A_{ij} = 0$ とする

これをメッシュごとに計算
して重ね合わせる



$$AU = \mathbf{b}$$

具体的な実装

メッシュ分割にはpythonを用いてnumpyで計算

残りはC++

poses.csv

x,y

0.000000,0.000000

0.333333,0.000000

0.666667,0.000000

1.000000,0.000000

0.000000,0.333333

0.333333,0.333333

~~~~~

計17行

triangle.csv

v1,v2,v3

11,14,10

14,11,15

8,13,12

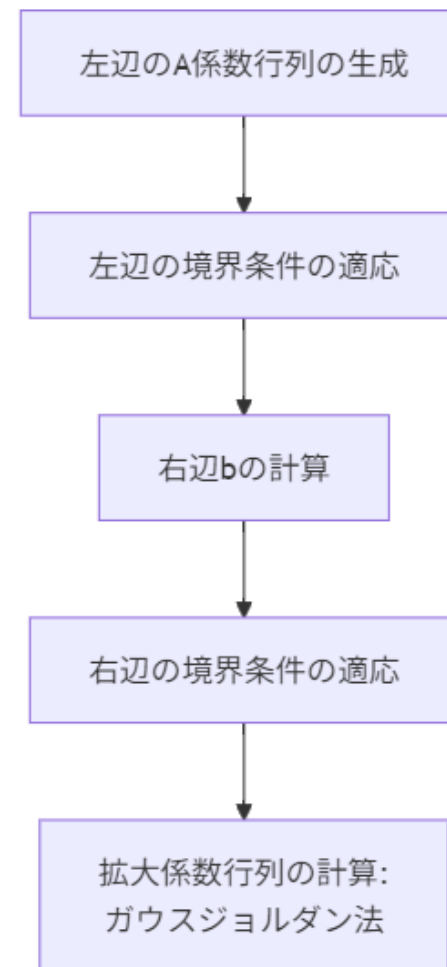
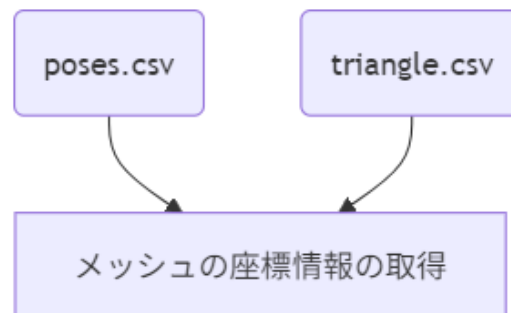
13,8,9

14,13,10

13,9,10

~~~~~

計 19行



具体的な計算～メッシュが4*4行列の場合～

メッシュ:左上から

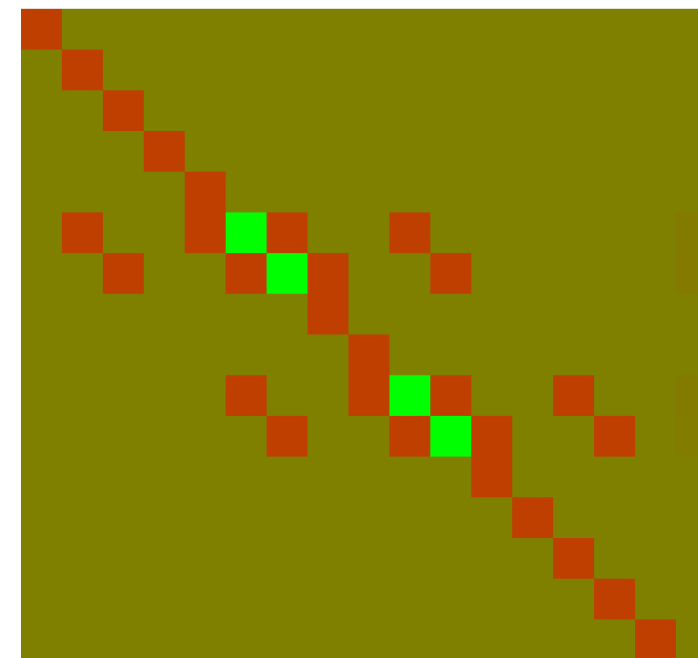
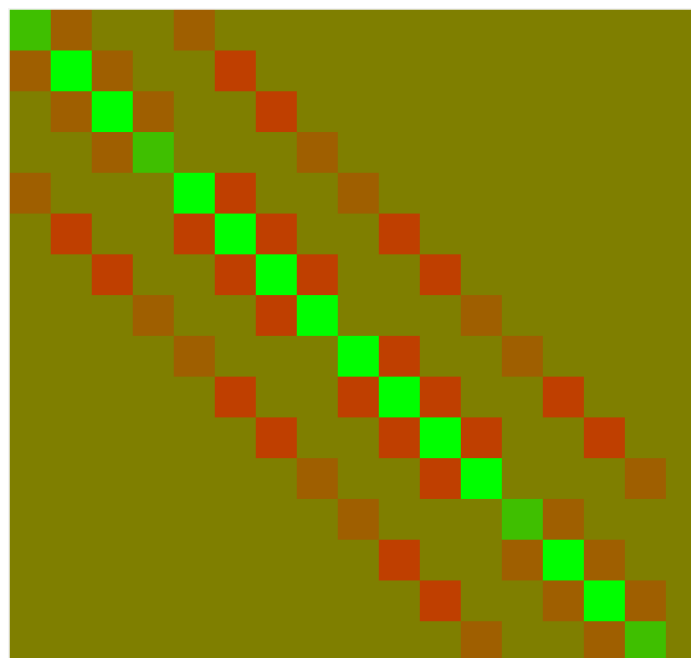
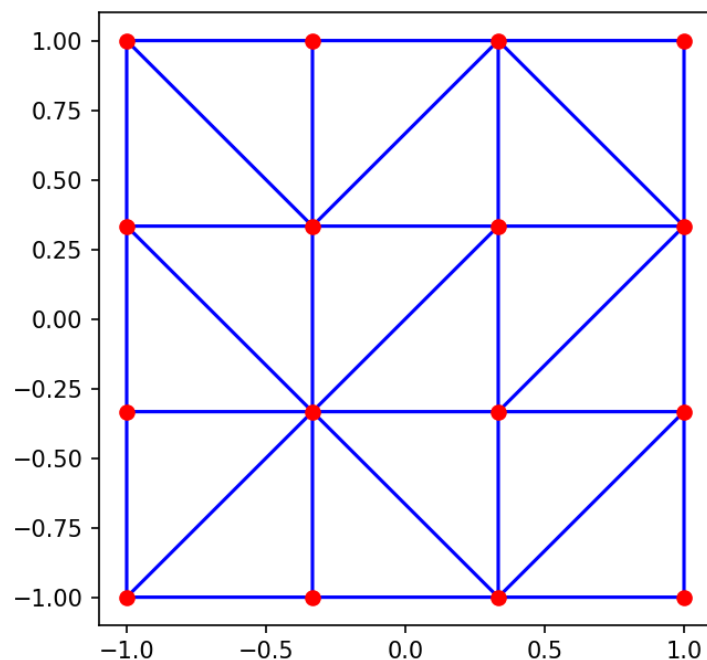
0 1 2 3

4 5 6 7

~~~~~

[A b] 16\*17行列  
[-2,2]

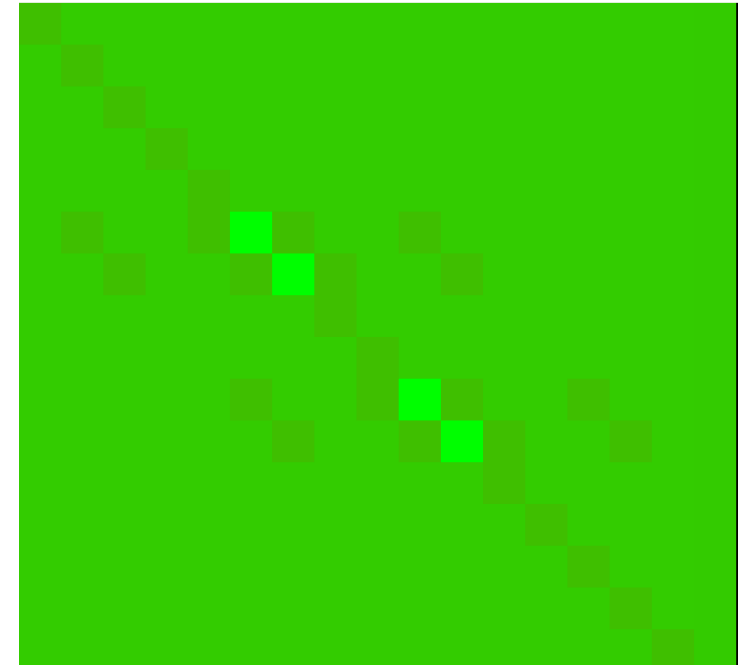
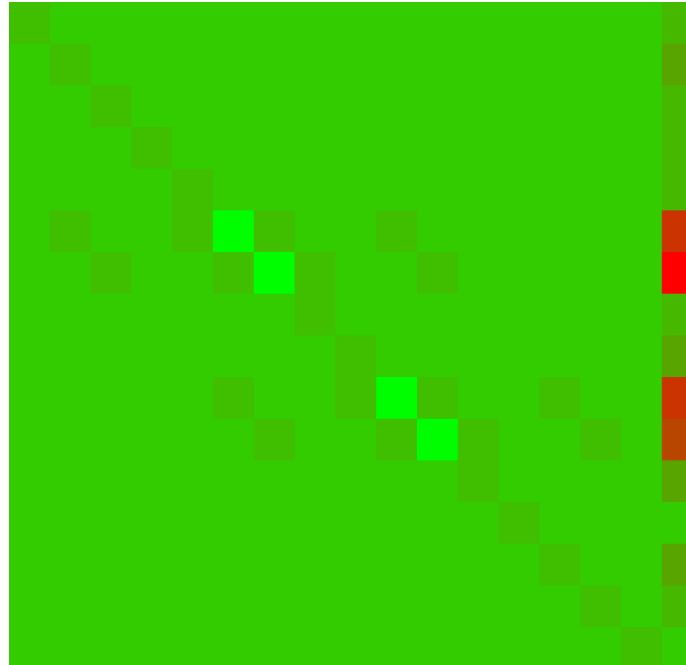
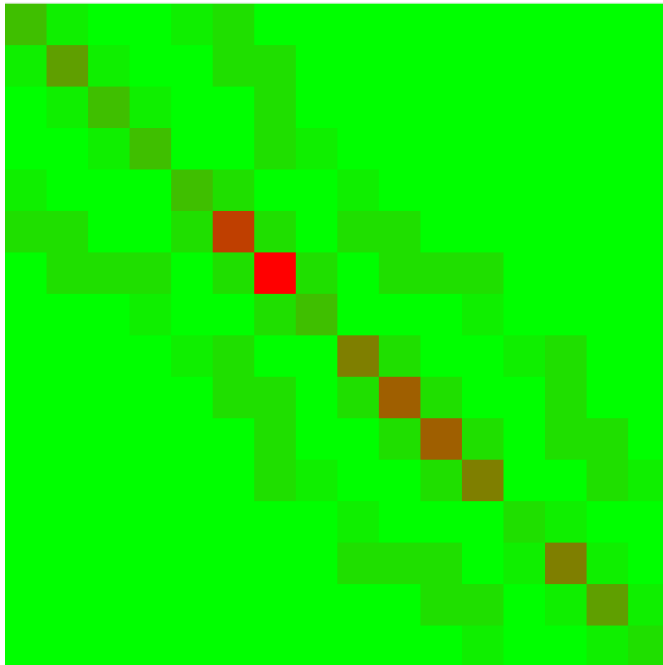
境界条件付き [A b]  
16\*16行列 [-2,2]



# 具体的な計算～メッシュが4\*4行列の場合～

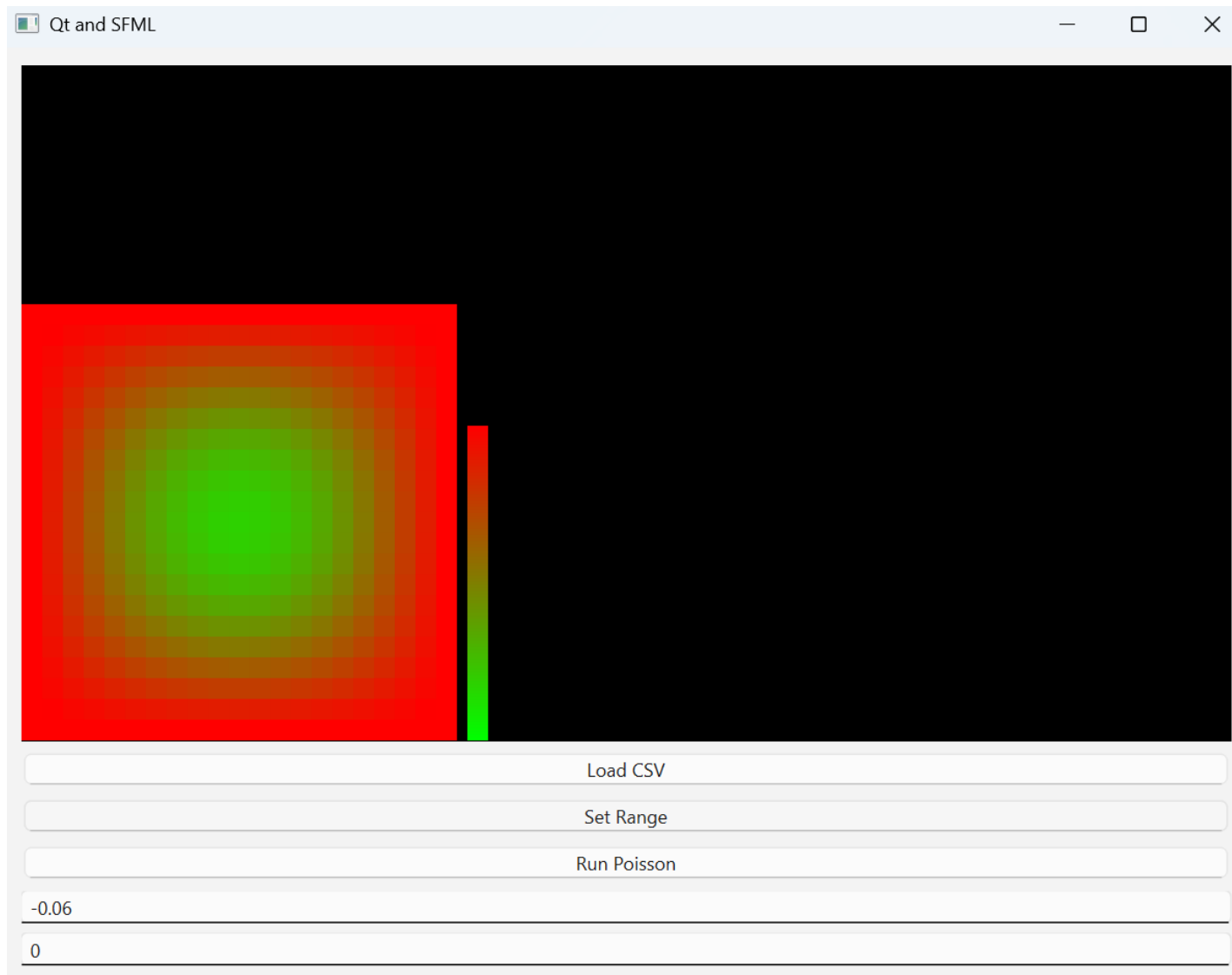
$M/\Delta$  16\*16行列 [0,16]  $[A \ M * f/\Delta]$  16\*17行列 [-4,16]

境界条件付き  
 $[A \ b]$  ( $b = M * f$ )



実際の様子

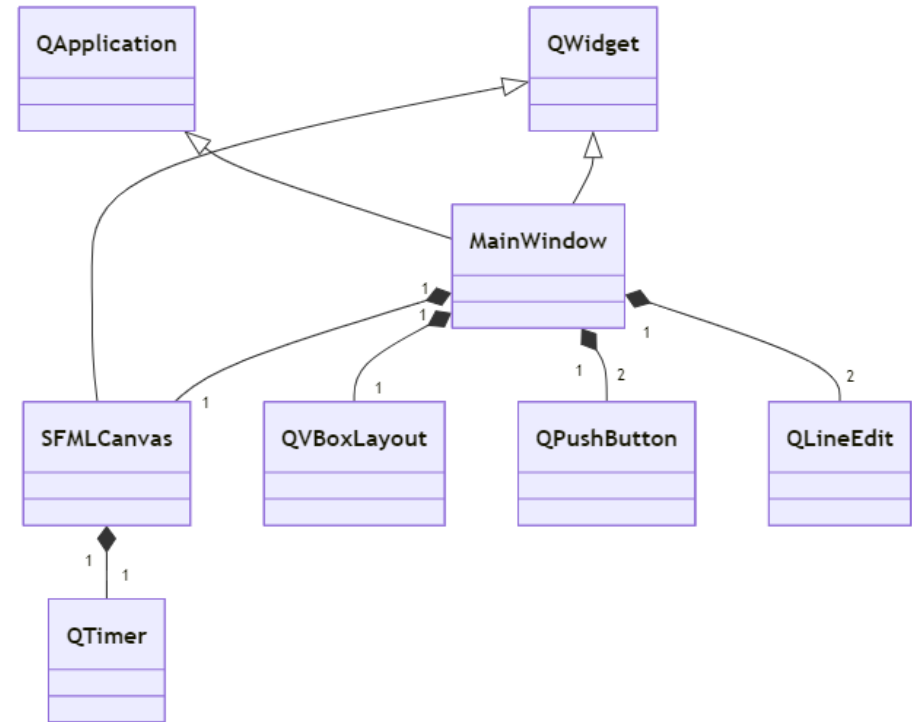
~メッシュが21\*21行列の場合~





# Qt, SFML についてのアーキテクチャ

- QtはC++のクロスプラットフォームGUI開発ライブラリ  
→ 日本語表記やテキスト、ボタン処理などが実装しやすく
- SFMLはOpenGLのラッパーライブラリ。クロスプラットフォームでグラフィック処理が簡便に、かつOpenGLよりもレンダリングコンテキストや入出力の点でOS依存性が低い
- 右は簡単なクラス図

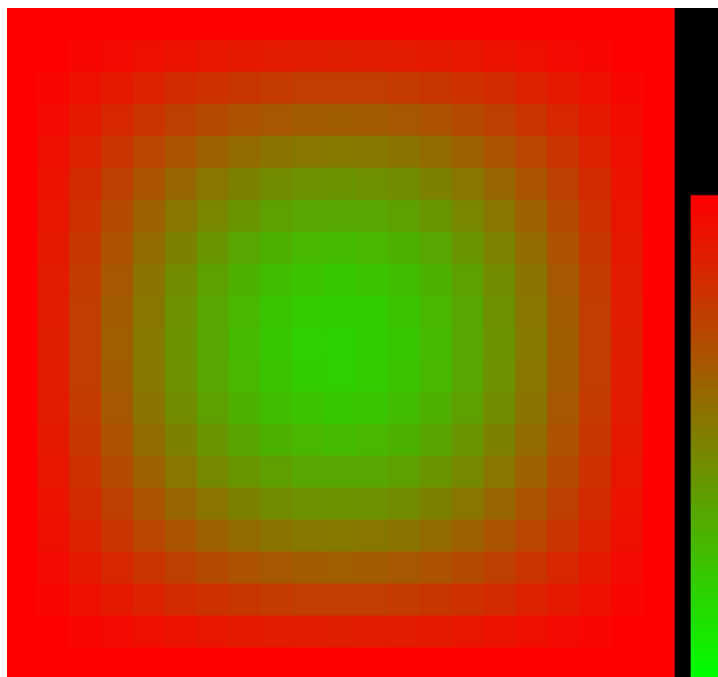


# 具体的な 解析結果

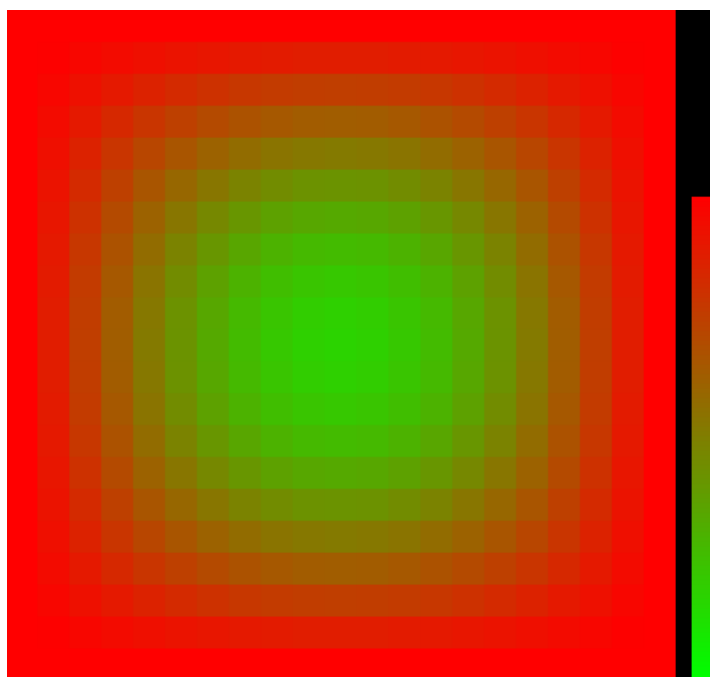
$$\Delta u = \sin(\pi x)\sin(\pi y)$$

$[0, 1]^2$ において 境界条件は  $0(x = \pm 1, y = \pm 1)$

解析解は  $u = -\frac{\sin(\pi x)\sin(\pi y)}{2\pi^2}$  である。



数値解析結果



真の値

両者ともに $[-0.06, 0]$   
で色を表示

誤差はおよそ**0.7%**

↑小さい!!

# Git hub

具体的なコードはこちらから



## 参考文献

有限要素法による流れのシミュレーション 計算力学学会  
2015年