A FEM for the Square Root of the Laplace Operator

Enrique Otárola[1]

[1]Department of Mathematics
University of Maryland, College Park.

AMSC689A Final Project Presentation,
May 2011.

Outline of Topics

The Continuos Problem

The problem we shall be concerned with reads as follows: Given a smooth enough function f, find u such that

$$\left\{ \begin{array}{rcll} (-\Delta)^{1/2} u & = & f & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega, \end{array} \right.$$

where $\Omega \subset \mathbb{R}^d$, with $d = 1, 2$ is a bounded domain with a smooth boundary $\partial\Omega$ and $(-\Delta)^{1/2}$ denotes the square root of the Laplace operator $-\Delta$ in Ω with zero boundary values on $\partial\Omega$.

Applications

Concerning applications, nonlocal operators are of importance in a wide range of applications:

- Finance.
- Image Processing.
- Quasi-geostrophic flow models.
- Modeling hydraulic fractures and the evolution of a viscous liquid thin film.

The development of efficient computational solution techniques for this problem is fundamental.

Definition of the Square Root of the Laplacian

Spectral theory of the Laplacian $-\Delta$ in a smooth bounded domain Ω with zero Dirichlet boundary values. There exists a sequence of eigenvalues

$$0 < \lambda_1 < \lambda_2 \leq \cdots \lambda_k \leq \cdots \to \infty$$

and,

Definition of the Square Root of the Laplacian

Spectral theory of the Laplacian $-\Delta$ in a smooth bounded domain Ω with zero Dirichlet boundary values. There exists a sequence of eigenvalues

$$0 < \lambda_1 < \lambda_2 \leq \cdots \lambda_k \leq \cdots \to \infty$$

and, there exists an orthonormal basis $\{\varphi_k\}$ of $L^2(\Omega)$, where $\varphi_k \in H_0^1(\Omega)$ is an eigenfunction corresponding to λ_k:

$$\begin{cases} -\Delta\varphi_k = \lambda_k\varphi_k & \text{in } \Omega \\ \varphi_k = 0 & \text{on } \partial\Omega, \end{cases} \tag{1}$$

for $k = 1, 2, \cdots$. Regularity theory $\implies$ $\varphi_k \in C^\infty(\bar{\Omega})$ for $k = 1, 2, \cdots$.

Definition of the Square Root of the Laplacian

The square root of the Dirichlet Laplacian, for a smooth function u, is given by

$$u = \sum_{k=1}^{\infty} c_k \varphi_k \mapsto (-\Delta)^{1/2} u = \sum_{k=1}^{\infty} c_k \lambda_k^{1/2} \varphi_k .$$

Density results $\implies$ $(-\Delta)^{1/2} : H_0^1(\Omega) \to L^2(\Omega)$.

Definition of the Square Root of the Laplacian

The square root of the Dirichlet Laplacian, for a smooth function u, is given by

$$u = \sum_{k=1}^{\infty} c_k \varphi_k \mapsto (-\Delta)^{1/2} u = \sum_{k=1}^{\infty} c_k \lambda_k^{1/2} \varphi_k .$$

Density results $\implies$ $(-\Delta)^{1/2} : H_0^1(\Omega) \to L^2(\Omega)$. Then if $f \in L^2(\Omega)$, we have

$$f = \sum_{k=1}^{\infty} f_k \varphi_k \implies c_k = f_k \lambda_k^{-1/2}$$

Numerical disadvantages: We need to find a sufficiently large number of eigenfunctions to obtain an accurate approximation.

Definition of the square root of the Laplacian

On the other hand, this operator can be seen as a singular integral

$$(-\Delta)^{1/2} u(x) = C_d \int_\Omega \frac{u(x) - u(y)}{|x-y|^{d+1}} dy,$$

where C_d is a normalization constant.

Definition of the square root of the Laplacian

On the other hand, this operator can be seen as a singular integral

$$(-\Delta)^{1/2}u(x) = C_d \int_\Omega \frac{u(x)-u(y)}{|x-y|^{d+1}}\,dy,$$

where C_d is a normalization constant.

Numerical disadvantages: the integrand is singular and the matrix obtained is dense. These inconveniences complicate the numerical computation.

Harmonic Extension

The approach presented by X. Cabre and J. Tan, (2010): relation between the nonlocal operator $(-\Delta)^{1/2}$ and its harmonic extension.

Given u defined in Ω, we consider its harmonic extension v in the cylinder $\mathcal{C} := \Omega \times (0, \infty)$, with v vanishing on $\partial_L \mathcal{C} := \partial\Omega \times [0, \infty)$.

$$\begin{cases} -\Delta v &= 0 \quad \text{in } \mathcal{C} = \Omega \times (0,\infty), \\ v &= 0 \quad \text{on } \partial_L \mathcal{C} = \partial\Omega \times [0,\infty), \\ \dfrac{\partial v}{\partial \nu} &= f \quad \text{on } \Omega \times \{0\}, \end{cases}$$

where ν is the unit outer normal to $\mathcal{C}$ at $\Omega \times \{0\}$.

Harmonic Extension

The approach presented by X. Cabre and J. Tan, (2010): relation between the nonlocal operator $(-\Delta)^{1/2}$ and its harmonic extension.

Given u defined in Ω, we consider its harmonic extension v in the cylinder $\mathcal{C} := \Omega \times (0,\infty)$, with v vanishing on $\partial_L \mathcal{C} := \partial\Omega \times [0,\infty)$.

$$\begin{cases} -\Delta v & = & 0 & \text{in } \mathcal{C} = \Omega \times (0,\infty), \\ v & = & 0 & \text{on } \partial_L \mathcal{C} = \partial\Omega \times [0,\infty), \\ \dfrac{\partial v}{\partial \nu} & = & f & \text{on } \Omega \times \{0\}, \end{cases}$$

where ν is the unit outer normal to $\mathcal{C}$ at $\Omega \times \{0\}$. Then,

$$u = \mathrm{tr}_\Omega v := v(\cdot, 0)$$

.

Spaces for v and u

Space for v:

$$H_0^1(\mathcal{C}) := \{v \in H^1(\mathcal{C}) | v = 0 \text{ a.e. on } \partial_L \mathcal{C} = \partial\Omega \times [0,\infty)\}.$$

Space for u:

$$\begin{aligned}\mathcal{V}_0(\Omega) = H_{00}^{1/2}(\Omega) &= \left[H_0^1(\Omega), L^2(\Omega)\right]_{1/2,2} \\ &= \left\{u \in H^{1/2}(\Omega) \middle| \int_\Omega \frac{u^2(x)}{d(x)} dx < +\infty \right\}.\end{aligned}$$

Truncated Problem

Numerically, it cannot be solved because $\mathcal{C}$ is an infinite domain $\implies$ We need to consider a suitable truncated problem.

Truncated Problem

Numerically, it cannot be solved because $\mathcal{C}$ is an infinite domain $\implies$ We need to consider a suitable truncated problem.

Why can we truncate the problem? Given $M > 0$, v satisfies

$$\|\nabla v\|^2_{L^2(\Omega\times(M,\infty))} < e^{-2\sqrt{\lambda_1}M}\|f\|^2_{\mathcal{V}_0(\Omega)^*}.$$

Consider M adequately large and define v^M in a bounded domain $\mathcal{C}_M := \Omega \times (0, M)$, imposing a zero Dirichlet condition on $\Omega \times \{M\}$:

$$\begin{cases} -\Delta v^M = 0 & \text{in } \mathcal{C}_M = \Omega \times (0, M), \\ v^M = 0 & \text{on } \partial_L \mathcal{C}_M := \partial\Omega \times [0, M], \\ v^M = 0 & \text{on } \Omega \times \{M\}, \\ \dfrac{\partial v^M}{\partial \nu} = f & \text{on } \Omega \times \{0\}, \end{cases}$$

Weak Formulation of the Truncated Problem

Find $v^M \in H_0^1(\mathcal{C}_M)$ such that

$$\int_{\mathcal{C}_M} \nabla v^M \cdot \nabla \psi = \int_\Omega f \operatorname{tr}_\Omega \psi, \quad \text{for all } \psi \in H_0^1(\mathcal{C}_M).$$

$$H_0^1(\mathcal{C}_M) := \{ v \in H^1(\mathcal{C}_M) | v = 0 \text{ a.e. on } \partial_L \mathcal{C}_M, \\ \text{and } v = 0 \text{ a.e. on } \Omega \times \{M\} \}.$$

Weak Formulation of the Truncated Problem

Find $v^M \in H_0^1(\mathcal{C}_M)$ such that

$$\int_{\mathcal{C}_M} \nabla v^M \cdot \nabla \psi = \int_\Omega f \operatorname{tr}_\Omega \psi, \quad \text{for all } \psi \in H_0^1(\mathcal{C}_M).$$

$$\begin{aligned} H_0^1(\mathcal{C}_M) := \{ v \in H^1(\mathcal{C}_M) | v = 0 \text{ a.e. on } \partial_L \mathcal{C}_M, \\ \text{and } v = 0 \text{ a.e. on } \Omega \times \{M\} \}. \end{aligned}$$

How good is this truncated problem?

$$M > \frac{1}{\sqrt{\lambda_1}} \ln\left(\frac{2}{\epsilon^2}\right) \implies \|v - v^M\|_{H_0^1(\mathcal{C}_M)} \leq \epsilon \|f\|_{\mathcal{V}_0(\Omega)^*}.$$

Galerkin Approximation

Given a family of partitions $\mathcal{T}_k$ of the domain $\mathcal{C}_M$ into quadrilateral elements, we define for $n \geq 1$

$$\mathbb{V}^{n,0} := \{v \in C^0(\overline{\mathcal{C}_M}) : v|_T \in \mathcal{Q}_n(T)\ \forall T \in \mathcal{T}_k\} \cap H_0^1(\mathcal{C}_M),$$

Galerkin approximation for v^M is given by: Find $v_h^M \in \mathbb{V}^{n,0}$ such that

$$\int_{\mathcal{C}_M} \nabla v_h^M \cdot \nabla w_h = \langle f, \mathrm{tr}_\Omega w_h \rangle, \qquad \text{for all } w_h \in \mathbb{V}^{n,0}.$$

Galerkin Approximation

Given a family of partitions $\mathcal{T}_k$ of the domain $\mathcal{C}_M$ into quadrilateral elements, we define for $n \geq 1$

$$\mathbb{V}^{n,0} := \{v \in C^0(\overline{\mathcal{C}_M}) : v|_T \in \mathcal{Q}_n(T) \ \forall T \in \mathcal{T}_k\} \cap H_0^1(\mathcal{C}_M),$$

Galerkin approximation for v^M is given by: Find $v_h^M \in \mathbb{V}^{n,0}$ such that

$$\int_{\mathcal{C}_M} \nabla v_h^M \cdot \nabla w_h = \langle f, \mathrm{tr}_\Omega w_h \rangle, \qquad \text{for all } w_h \in \mathbb{V}^{n,0}.$$

Standard FEM theory + truncated problem property implies

$$\|v - v_h^M\|_{H_0^1(\mathcal{C})} \leq C \left(\epsilon \|f\|_{\mathcal{V}_0(\Omega)^*} + h\|v\|_{H^2(\mathcal{C}_M)} \right),$$

where $h = \max_{T \in \mathcal{T}} h_T$.

Numerical Implementation in deal.ii

The main function is similar to the steps discussed in class. We implement $\mathcal{Q}_k$ adaptive refinement, $\mathcal{Q}_k$ global refinement, and $\mathcal{Q}_k$ exponential refinement, $k \geq 1$, in dimension $d = 2, 3$.

```
int main ()
{
  const unsigned int dim = 3; //We can choose dim = 2 or dim = 3.
  try{
       std::cout << "Solving with Q1 elements, adaptive refinement" << std::endl
                 << "=============================================" << std::endl
                 << std::endl;

      LaplaceProblem<dim>
        laplace_problem ( LaplaceProblem<dim>::adaptive_refinement);
      laplace_problem.run ();
      std::cout << std::endl;
   }
```

Numerical Implementation in deal.ii

Template class LaplaceProblem. The class that does all the work.

```
template <int dim> class LaplaceProblem{
  public:
    enum RefinementMode{global_refinement, adaptive_refinement, exp_refinement};
    LaplaceProblem (const RefinementMode refinement_mode);
    ~LaplaceProblem ();
    void run ();
```

Member functions. They do what their names suggest.

```
      private:
        void setup_system ();
        void assemble_system ();
        void solve ();
        void refine_grid ();
        void process_solution (const unsigned int cycle);
    };
```

Numerical Implementation in deal.ii

Some member variables:

```
    Triangulation<dim>              triangulation;
    DoFHandler<dim>                 dof_handler;
    FE_Q<dim>                       fe;
    ConstraintMatrix                hanging_node_constraints;
    SparsityPattern                 sparsity_pattern;
    SparseMatrix<double>            system_matrix;
    Vector<double>                  solution;
    Vector<double>                  system_rhs;
    const RefinementMode            refinement_mode;
    ConvergenceTable                convergence_table;
};
```

Numerical implementation in deal.ii

LaplaceProblem::run. The code is implemented in cycles. For each kind of mesh, we consider a fixed number of cycles and we solve.

```
template <int dim> void LaplaceProblem<dim>::run (){
  for (unsigned int cycle=0; cycle<7; ++cycle){
    if (cycle == 0){
    // Here we generate the first mesh!
    GridGenerator::subdivided_hyper_rectangle(triangulation,
                                              subdivisions2,p1,p2,false);
    }else{
 refine_grid ();
```

Numerical Example

We consider $\Omega = (0,1)$ and $f(x) = \pi sin(\pi x)$, then $\mathcal{C}_M = (0,1) \times (0,M)$
$u(x) = sin(\pi x)$ and $v(x,y) = sin(\pi x)e^{-\pi y}$.

$$\|v - v_h^M\|_{H_0^1(\mathcal{C})} \leq C\left(\epsilon\|f\|_{\mathcal{V}_0(\Omega)^*} + h\|v\|_{H^2(\mathcal{C}_M)}\right), \quad \epsilon = \epsilon(M)$$

Numerical Example

We consider $\Omega = (0,1)$ and $f(x) = \pi sin(\pi x)$, then $\mathcal{C}_M = (0,1) \times (0,M)$
$u(x) = sin(\pi x)$ and $v(x,y) = sin(\pi x)e^{-\pi y}$.

$$\|v - v_h^M\|_{H_0^1(\mathcal{C})} \leq C\left(\epsilon\|f\|_{\mathcal{V}_0(\Omega)^*} + h\|v\|_{H^2(\mathcal{C}_M)}\right), \quad \epsilon = \epsilon(M)$$

M should change with h to get $\epsilon \approx h$

$$M = -\frac{2}{\pi}\ln\left(\frac{h}{\sqrt{2}}\right)$$

Some Global Meshes

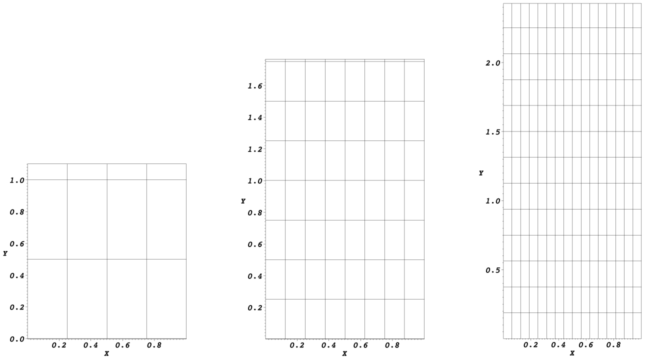
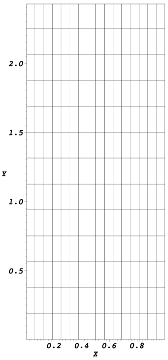


Figure: Degrees of freedom: 20, 81, 238 respectively.

Results with Global Refinement



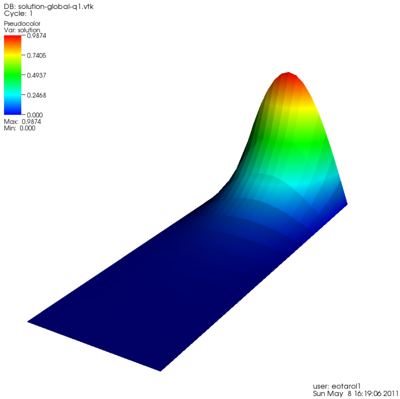


Figure: v_h^M with 238 degrees of freedom.

Results with Global Refinement

Computing L^2 and H^1 error norms.

```
Vector<float> difference_per_cell (triangulation.n_active_cells());

VectorTools::integrate_difference (dof_handler, solution, Solution<dim>(),
                                    difference_per_cell, QGauss<dim>(q_points_norm ),
                                    VectorTools::L2_norm);

const double L2_error = difference_per_cell.l2_norm();
```

Results with Global Refinement

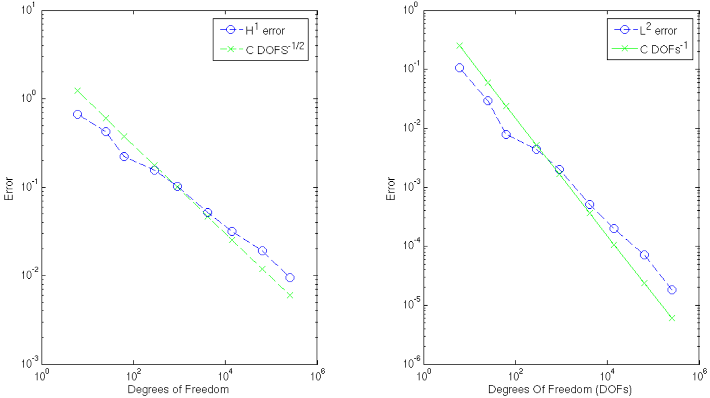


Figure: Decay of the L^2 and H^1 norms of the error.

Estimates for the Function u

What about u?

Estimates for the Function u

What about u? Trace results imply an estimate for u

$$
\begin{aligned}
\|u - u_h^M\|_{H_{00}^{1/2}(\Omega)} &\leq \|v - v_h^M\|_{H_0^1(\mathcal{C})} \\
&\leq C\left(\epsilon\|f\|_{\mathcal{V}_0(\Omega)^*} + h\|v\|_{H^2(\mathcal{C}_M)}\right), \quad \epsilon = \epsilon(M)
\end{aligned}
$$

However, notice that this estimate is not optimal! Optimal estimate

$$
\|u - u_h^M\|_{H_{00}^{1/2}(\Omega)} \leq C\left(\epsilon\|f\|_{\mathcal{V}_0(\Omega)^*} + h^{3/2}\|f\|_{H^1(\Omega)}\right), \quad \epsilon = \epsilon(M)
$$

Results with Global Refinement

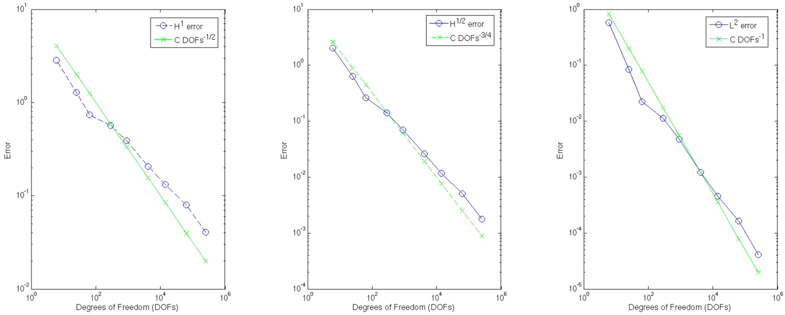


Figure: Decay of the L^2, $H^{1/2}$ and H^1 norms of the error.

Exponential Refinement

We exploit the behavior of the real solution

$$v(x,y) = \sum c_k \varphi_k e^{-\sqrt{\lambda_k} y}, \quad \text{for all } (x,y) \in \mathcal{C},$$

to design an exponential mesh. 2D case: We do global refinement in x and exponential refinement in y. Using interpolation results we get

$$\|v - v_h^M\|^2_{H_0^1(\mathcal{C}_M)} \leq C \sum_{k=1}^{N_y} \left(h_k^y\right)^3 e^{-\sqrt{\lambda_1} y_k} \leq CN^{-1},$$

Exponential Refinement

We exploit the behavior of the real solution

$$v(x,y) = \sum c_k \varphi_k e^{-\sqrt{\lambda_k} y}, \quad \text{for all } (x,y) \in \mathcal{C},$$

to design an exponential mesh. 2D case: We do global refinement in x and exponential refinement in y. Using interpolation results we get

$$\|v - v_h^M\|^2_{H_0^1(\mathcal{C}_M)} \leq C \sum_{k=1}^{N_y} \left(h_k^y\right)^3 e^{-\sqrt{\lambda_1} y_k} \leq CN^{-1},$$

and finally we obtain

$$y_{k+1} = y_k + \frac{1}{k} N^{-2/3} e^{\sqrt{\lambda_1}/3 y_k}.$$

Some Exponential Meshes

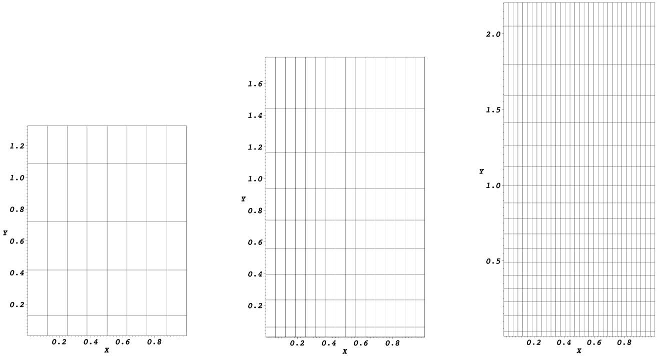


Figure: Degrees of freedom: 54, 170, 627 respectively.

Results with Exponential Refinement

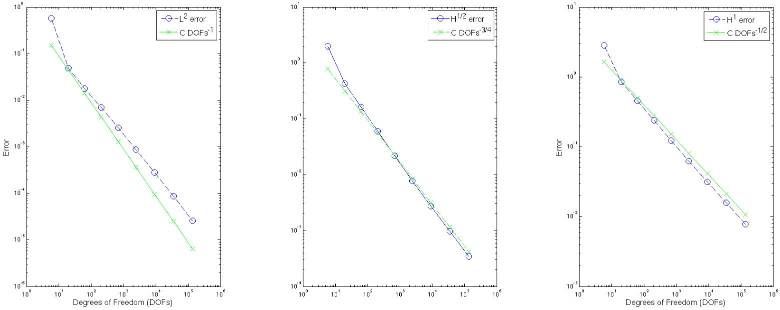


Figure: Decay of the L^2, $H^{1/2}$ and H^1 norms of the error.

Adaptive Refinement

The estimate

$$\|v - v_h^M\|_{H_0^1(\mathcal{C})} \leq C \left(\epsilon \|f\|_{\mathcal{V}_0(\Omega)^*} + h\|v\|_{H^2(\mathcal{C}_M)} \right),$$

is not computable and provides only asysmptotic information. We create a mesh adapted to the function v. Basic ingredient:

$$\|v - v_h^M\|_{H_0^1(\mathcal{C})} \leq C_1 \mathcal{E}_{\mathcal{T}}(v_h^M) \leq C_2 \left(\|v - v_h^M\|_{H_0^1(\mathcal{C})} + \mathrm{osc}_{\mathcal{T}}(v_h^M) \right)$$

Adaptive Refinement

The estimate

$$\|v - v_h^M\|_{H_0^1(\mathcal{C})} \leq C\left(\epsilon\|f\|_{\mathcal{V}_0(\Omega)^*} + h\|v\|_{H^2(\mathcal{C}_M)}\right),$$

is not computable and provides only asysmptotic information. We create a mesh adapted to the function v. Basic ingredient:

$$\|v - v_h^M\|_{H_0^1(\mathcal{C})} \leq C_1\mathcal{E}_{\mathcal{T}}(v_h^M) \leq C_2\left(\|v - v_h^M\|_{H_0^1(\mathcal{C})} + \operatorname{osc}_{\mathcal{T}}(v_h^M)\right)$$

Error Estimator Implemented in deal.ii

$$\mathcal{E}_{\mathcal{T}}^2(v_h^M, T) = \frac{h_T}{24}\int_{\partial T}\left[\frac{\partial v_h^M}{\partial \nu}\right]$$

3D Numerical Example

We consider $\Omega = (0,1) \times (0,1)$ and $f(x) = \sqrt{2}\pi \sin(\pi x) \sin(\pi y)$, then $u(x) = \sin(\pi x) \sin(\pi y)$ and $v(x,y) = sin(\pi x) sin(\pi y) e^{-\sqrt{2}\pi y}$.

We have optimal estimates for every refinement: adaptive, exponential and global. We show the results obtained using Adaptivity.

An Adaptive Mesh. $M = 4$.

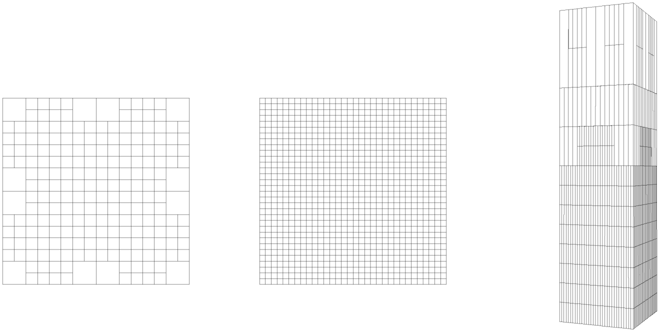


Figure: Degrees of freedom: 13435.

Convergence Table for v

n cells		H^1-error		L^2-error	
0	4	4.016e-01	-	4.790e-02	-
1	32	6.419e-01	-0.68	7.156e-02	-0.58
2	228	6.252e-01	0.04	6.094e-02	0.23
3	1628	4.190e-01	0.58	2.845e-02	1.10
4	11400	2.312e-01	0.86	8.959e-03	1.67
5	79265	1.188e-01	0.96	2.394e-03	1.90
6	549238	5.983e-02	0.99	6.091e-04	1.97

Results with Adaptive Refinement

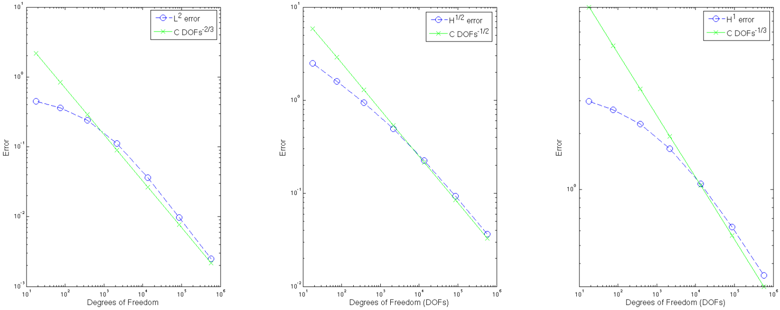


Figure: Decay of the L^2, $H^{1/2}$ and H^1 norms of the error.

Results with Adaptive Refinement

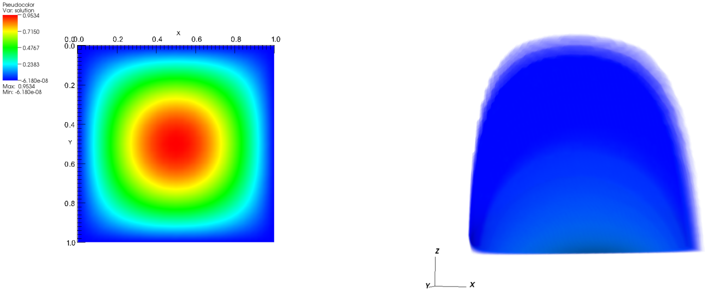


Figure: u_h^M and v_h^M with 13435 degrees of freedom.

3D Numerical Example

We consider the following numerical example. Given a smooth function $f(x,y) = \sqrt{2}\pi \sin(\pi x) \sin(\pi y)$, find u such that

$$\begin{cases} (-\Delta)^{1/2} u & = & f & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega, \end{cases}$$

where $\Omega = (-1,1)^2$ - $\mathrm{disk}_{(0,0)}(0.5)$.

```
GridGenerator::hyper_cube_with_cylindrical_hole (triangulation, inner_radius,
                                   outer_radius, extension, repetition,true);
```

An Adaptive Mesh. M $= 4$.

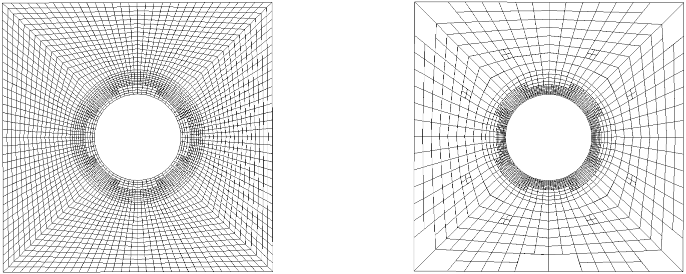


Figure: Meshes for $z = 0$ and $z = 4$.

An Adaptive Mesh. M = 4.

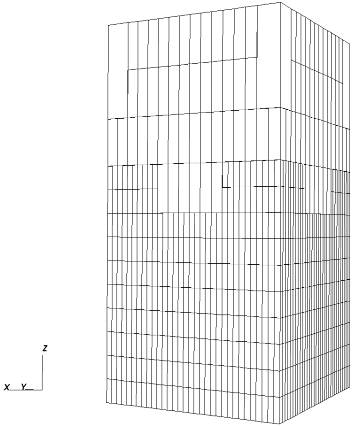


Figure: Degrees of freedom: 22492.

An Adaptive Mesh. M = 4.

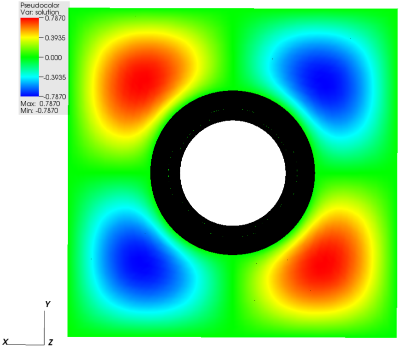


Figure: u_h^M computed with 22492 degrees of freedom.

Questions?