LNG_FEM: GENERATING GRADED MESHES AND SOLVING ELLIPTIC EQUATIONS ON 2-D DOMAINS OF POLYGONAL STRUCTURES

By

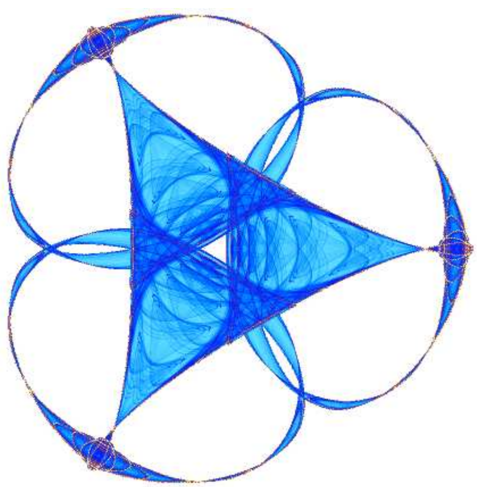
Hengguang Li

and

Victor Nistor

IMA Preprint Series # 2172

(August 2007)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA

400 Lind Hall

207 Church Street S.E.

Minneapolis, Minnesota 55455–0436

Phone: 612-624-6066 Fax: 612-626-7370

URL: http://www.ima.umn.edu

LNG_FEM: GENERATING GRADED MESHES AND SOLVING ELLIPTIC EQUATIONS ON 2-D DOMAINS OF POLYGONAL STRUCTURES

HENGGUANG LI AND VICTOR NISTOR

Abstract. We develop LNG_FEM, a software package for graded mesh generation and for solving elliptic equations. LNG_FEM generates user-specified graded meshes on arbitrary two-dimensional domains with straight edges, for different boundary conditions, once initial information is passed to the program by appropriately filling out some source files. We focus on a detailed instruction on the implementation of the software after a brief literature review of elliptic boundary value problems and graded meshes. Then, we show examples to point out that LNG_FEM is equipped with advanced algorithms and data structures to perform efficiently. LNG_FEM is to popularize the use and understanding of graded mesh in the finite element method. We hope it triggers more interest and discoveries both in academia and in industry.

Contents

1. Introduction

LNG_FEM, short for the linear graded finite element method (FEM), is a free software package written in C, for generation of graded meshes in general two-dimensional domains with polygonal structures [16, 24, 26], and for construction of linear finite element solutions for elliptic boundary value problems (EBVPs). LNG_FEM comes from part of our research on regularity of solutions of EBVPs, particularly from our effort on recovering the *quasi-optimal* rate of convergence of numerical solutions, when the solution possesses corner-like singularities that are caused by non-smoothness of the boundary or changes of boundary conditions [16, 19, 24]. LNG_FEM is a fast, memory-efficient, user-friendly package that is developed for general domains with straight edges in 2-D, and for different boundary conditions. During the development, information on geometry of the domain, mesh generation, linear finite elements, and solvers for algebraic systems are sealed in appropriate data structures. Therefore, with this flexibility of easy updates and

Date: August 8, 2007.
Department of Mathematics, The Pennsylvania State University (UP), *E-mail address*: li_h@math.psu.edu; nistor@math.psu.edu.

modification, LNG_FEM not only provides academia a powerful tool to demonstrate algorithms, educate students, strengthen research results, but also has great potential to effectively handle practical problems in industry.

Instead of details in algorithms and data structures, this article mainly presents instructions on the use of LNG_FEM (Section 2). Concise information is conveyed on preparation of input files before running the program, and on the way the outputs are arranged, to help readers get used to LNG_FEM and be able to work on their own problems in seconds.

Moreover, in Section 3, for those who are concerned with the reliability and efficiency of the software, we briefly describe several features on algorithms and data structures, which make LNG_FEM not only a universal set of the graded FEM for EBVPs on general 2-D domains with straight edges, also an optimal generator of meshes and finite elements in terms of storage and speed.

LNG_FEM is freely available at *http://www.math.psu.edu/li_h/Research.htm*. Extended questions and requests of source files can be sent to Hengguang Li at *li_h@math.psu.edu*. We keep LNG_FEM up-to-date. Suggestions to improve the software in any aspect are appreciated.

It is reasonable to review some related terms for better understanding the motivation and the inside machinery of LNG_FEM, especially for those who are new in this field, while experienced users can jump to Section 2 for instructions directly.

1.1. **Elliptic boundary value problems (EBVPs).** EBVPs are a class of partial differential equations (PDEs) with boundary conditions. Basic properties of solutions of EBVPs, such as existence, uniqueness, regularity, etc. have existed in [17, 18, 31, 33]. Following the equivalence of the bilinear form of the equation and the norm of some Sobolev space, existence and uniqueness can usually be guaranteed by Lax-Milgram Theorem for strictly positive elliptic operators. Regularity, describing smoothness of the solution, however, may be of great importance for numerical methods, since no explicit formula is available for the solution in general.

To fix ideas, we consider the following equation associated to the Laplace operator on a bounded two-dimentional domain Ω throughout our presentation. The results, however, extend to general second-order elliptic PDEs.

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ \partial_\nu u = 0 & \text{on } \partial_N \Omega, \\ u = 0 & \text{on } \partial_D \Omega. \end{cases} \tag{1}$$

A fundamental theorem on regularity of the solution in Equation (1) claims that the solution u is always in the Sobolev space $H^{m+1}(\Omega) \cap H^1_0(\Omega)$, $m \geq 0$, as long as the given datum f belongs to the space $H^{m-1}(\Omega)$, provided that $\partial_N \Omega = \emptyset$, and $\partial\Omega$ is smooth enough.

Theorem 1.1. *Suppose that the boundary $\partial\Omega$ of the domain is smooth and Equation* (1) *has the homogeneous Dirichlet boundary condition $u|_{\partial\Omega} = 0$. Then for $m \geq 0$, there is a unique solution u for Equation* (1)*, satisfying*

$$||u||_{H^{m+1}(\Omega)} \leq C||f||_{H^{m-1}(\Omega)},$$

where C depend only on m and Ω.

The above theorem, however, requires some assumptions on the domain and on the equation, which can be violated easily in practice. For instance, boundary conditions may change types; the boundary of the domain may not be smooth enough (corners, edges, etc.); some derivatives of the coefficients in the equation may have jumps (transmission problems [29, 30]). Any of these situations could generate a certain type of singularities in the solution and cause unexpected difficulties in numerical simulations.

1.2. **Graded meshes.** As a powerful numerical method in practical computation to solve PDEs, the FEM grabs tremendous interest from academic research as well [3, 5, 12, 14, 15, 32]. Starting from a domain decomposition, FEM typically uses piecewise polynomials as basis functions to approximate the solution.

From now on, we concentrate on bounded two-dimensional domains with straight edges. For domains with curved boundaries, other techniques are needed for analysis in the FEM, while they usually do not present the same difficulty as we will handle at this time. Let $\mathbb{P}$ be such a domain with straight edges. The following definition [12] gives a classical decomposition with triangles on $\mathbb{P}$.

Definition 1.2. Let $\{\mathcal{T}^h\}$, $0 < h \leq 1$, be a family (triangulation) of triangles on $\mathbb{P}$ such that

$$\max\{\text{diam } T : T \in \mathcal{T}^h\} \leq h\text{diam } \mathbb{P}.$$

Let B_T be the largest ball contained in T. The family (triangulation) is said to be **quasi-uniform** if there is $\rho > 0$ such that

$$\min\{\text{diam } B_T : T \in \mathcal{T}^h\} \geq \rho h\text{diam } \mathbb{P}.$$

Quasi-uniform triangulations based on Definition 1.2 leads to an optimal convergence rate for finite element solutions of EBVPs in the following sense.

Theorem 1.3. *Suppose that the solution $u \in H^{m+1}(\mathbb{P})$ and the right hand side $f \in H^{m-1}(\mathbb{P})$ in Equation (1). Then, the finite element solution u_h with piecewise polynomials of degree m satisfies*

$$||u - u_h||_{H^1(\mathbb{P})} \leq Ch^m|u|_{H^{m+1}(\mathbb{P})} \leq Ch^m||f||_{H^{m-1}(\mathbb{P})},$$

where C does not depend on the right hand side or the level of refinements.

As mentioned in Subsection 1.1, the solution can be less regular and even not in $H^2(\mathbb{P})$ when the boundary has conical or angular points or the boundary conditions change. In fact, these corner-like singularities that fail to satisfy the assumptions of Theorem 1.3 will slow down the rate of convergence on quasi-uniform meshes [2, 4, 6, 7, 21, 22, 28, 29, 36].

Methods of generating *quasi-optimal* meshes based on different principals have been the interest of many people [1, 6, 7, 27]. The graded mesh [2, 6, 7] is one of them that has proved to be successful to recover the *quasi-optimal* rate of convergence of the numerical solution. The basic idea of graded meshes is to make use of *a priori* estimates to determine the ratio of decay for triangles near singular points on the boundary. The error between u and u_h is evenly distributed on different triangles by controlling the size of the triangle accordingly: if a triangle dose not contain singular vertices, it is triangulated into four smaller triangles of equal size in the next level, by joining the midpoints of each edge; if a triangle has a singular vertex Q_I, we still generate four smaller triangles, but based on a different ratio of decay κ. (See Figure 1.) More examples will be shown in Section 2.

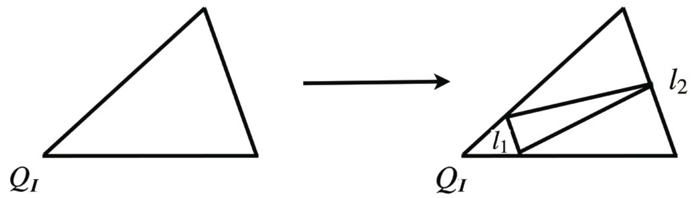


Figure 1. one refinement of triangle T with a singular vertex Q_I, $\kappa = l_1/l_2$

The following estimate [7] regarding the graded meshes claims that the *quasi-optimal* rate of convergence can be obtained for numerical solutions on these meshes in terms of dimensions of subspaces, when an appropriate decay ratio is chosen for every singular point [7, 24].

Theorem 1.4. *Let $0 < \epsilon < \eta$, where $0 < \eta \leq 1$ is related to the smallest positive eigenvalue of the resulting operator from the Mellin transform for the singular point. Since η can be determined explicitly or approximately, we let $\kappa = 2^{-m/\epsilon}$ be the ratio of decay of triangles. Denote by u_n the numerical solution of Equation (1) on the mesh after n successive refinements, and by V_n the finite subspace on the mesh. Then*

$$||u - u_n||_{H^1(\mathbb{P})} \leq C\,dim(V_n)^{-m/2}||f||_{H^{m-1}(\mathbb{P})},$$

for $f \in H^{m-1}(\mathbb{P})$, where C is independent of n.

As one of our attempts to popularize advanced numerical techniques for public uses, LNG_FEM gives users the freedom to specify singular points, to customize different ratios of decay, and to compute or compare solutions on various domains. With imbedded MATLAB commands in LNG_FEM, one can even get more insights of graded meshes by visualizing the results. See Section 2 for details.

1.3. **Numerical solvers.** Different numerical methods for PDEs preserve a similar routine, to translate continuous problems into discrete systems of algebraic equations. These discretizations usually give rise to millions of or even more unknowns that have to be treated well to achieve fairly good numerical approximations. A special version of the *curse of dimensionality* denies most of exact solvers immediately for such systems, even though they are accurate. Thus, theory and algorithms based on iterative methods dominate this field. In particular, as one of the most efficient techniques for solving these large systems, the multigrid (MG) method [8, 9, 10, 13, 20, 34, 35, 37, 38] has shown its strength on solving EPVBs.

We here quote a recent result on the convergence rate of the MG method on systems from graded meshes to illustrate its efficiency [11]. In fact, it claims that the MG V-cycle converges uniformly on *quasi-optimal* graded meshes for corner-like singularities.

Theorem 1.5. *Let B^v be the iterator for the MG V-cycle, and A be the differential operator in Equation (1). Denote by $||\cdot||^2 = (\cdot,\cdot)_a = (A\cdot,\cdot)$ the norm induced by A. Suppose linear finite elements are used for the equation. Then, for the resulting system of algebraic equations from the discretization on quasi-optimal graded meshes, every iteration of the MG V-cycle contracts $||I - B^vA||_a$ in the following* *rate,*

$$||I - B^v A||_a = \frac{c_0}{1 + c_0} \leq \frac{c_1}{c_1 + c_2 k},$$

where I is the identity operator; c_1 and c_2 are constants depending only on the problem and the smoother, not on the level of refinements; k is the number of iterations on every subspace.

In the first version of LNG_FEM, the preconditioned conjugate gradient (PCG) method was imbedded for solving the resulting system of equations. It is not optimal, but an algorithm of acceptable efficiency. This is the only part that does not hold the optimality in our package. An imbedded multigrid solver will be released soon.

2. Instructions on LNG_FEM

The main purpose of the development of LNG_FEM is to encourage pubic understanding and uses of graded meshes for EPVPs on general polygonal domains. To fulfill this goal, we not only emphasize the reliability and efficiency of algorithms, but also try to make the interface as friendly as possible, so that users can enjoy exciting numerical techniques more than spend much time on learning the software. The main features of LNG_FEM include:

1. Modularization. *Related algorithms and data structures are grouped in different modules for easy updates and modification; input files and outputs of the program are bundled in /Sourcefiles and /Results, respectively, to simplify the process of initialization and analysis of the result afterwards.*
2. Generality. *Users have the freedom to set up various domains, boundary conditions, the ratio of decay for any singular point by typing information in corresponding input files. The current model PDE in LNG_FEM is associated with the Laplace operator. One can even change the PDE by simply updating several lines in one of these modules of the source code.*
3. Efficiency. *LNG_FEM is equipped with advanced algorithms and perform efficiently in terms of storage and speed. We managed to minimize the time for mesh generation and matrix assembly. Consequently, LNG_FEM is comparable with other fast mesh generators nowadays. Details will be discussed in Section 3.*
4. Analysis of the result. *LNG_FEM can either compute the numerical solution on the current mesh, or, in addition, compare the current numerical solution with the solution from the previous mesh to provide the convergence rate. The mesh and the solution can be visualized easily in MATLAB with embedded commands in the package.*

With all the features in mind, we here demonstrate that it is easy to take full advantage of LNG_FEM as well by giving step-by-step instructions. Recall that the model PDE inside is Equation (1) with $f = 1$. Before going to the details, one can have the first experience to run the program with an L-shape domain (Figure 2) by entering the directory /LNG_FEM and typing ./LNG_FEM.out in the command terminal. An expository diagram on the processing flow of the program is given in Figure 6.

2.1. **Source files.** After unzipping the downloaded file, we see two sub-directories (/Sourcefiles and /Results), and some other .out or .m files under the directory /LNG_FEM, where the .out file is the executable file for the program. As its name indicates, /Sourcefiles contains information on the initial triangulation, boundary conditions, parameters of the grading that needs users to set up. In this subsection, we elaborate on these files in /Sourcefiles by taking the triangulation in Figure 2 as an example, which comes with the package.

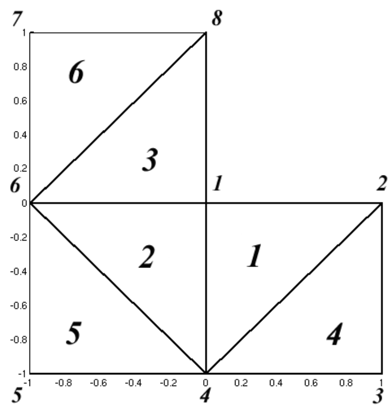


Figure 2. an initial triangulation of the L-shape domain for the homogeneous Dirichlet boundary condition with numbering on nodes and triangles

First of all, we follow several rules to generate initial triangles, nodes, and to number them.

1. Any vertex or singular point not in H^2 of the domain is an initial node.
2. An initial triangle can not contain more than one singular point of the domain.
3. Any node can not have more than six adjacent triangles.
4. There is no restriction on the numbering of triangles. We are free to assign numbers starting from one to triangles. If the domain has no singular point, we have the same situation on the numbering of initial nodes. The only restriction is that when the domain has n singular points, $n \geq 1$, we preserve the first n natural numbers $\{1, 2, \cdots, n\}$ for these points. After numbering singular nodes with these n integers, we are free again to denote every other node by a unique integer $> n$. For example, "1" has to be assigned to the node sitting on the re-entrant corner in Figure 2, since it is the only singular point. Then, we can actually re-arrange the numbering of triangles and other nodes in a different way than in Figure 2. It does not affect the final result.

LNG_Initialnode.txt is for initial nodes, and it always has the same format as the particular LNG_Initialnode.txt corresponding to Figure 2. (See Figure 3.) The integer in the first row indicates the number of nodes (8 nodes in Figure 2). Starting from the second row, the ith row, $i \geq 2$, lists the coordinates (x, y) of the $(i-1)$th initial node. (The second node sits at $(1, 0)$, for example.)

LNG_Initialtriangle.txt contains information on initial triangles, which has a similar structure as LNG_Initialnode.txt. The first number in the file stands for the number of initial triangles, while the three numbers in the ith row, $i \geq 2$, identify the vertices of the $(i-1)$th triangle with the numbering of vertices in the *increasing* order. Thus, since the third triangle in Figure 2 has vertices $1, 6$, and

8, the 4th row of LNG_Initialtriangle.txt is filled with 1 6 8 in Figure 3. One may have already noted that, it suffices to pass all geometric information of the domain to the program, when the above two files are made in this way.

```
8
0 0
1 0
1 -1
0 -1
-1 -1
-1 0
-1 1
0 1
```

```
8 8
1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 1
```

```
6
1 2 4
1 4 6
1 6 8
2 3 4
4 5 6
6 7 8
```

```
1
0.2
```

Figure 3. **source files corresponding to Figure 2 with ratio of decay = 0.2 (clockwise): LNG_Initialnode.txt (left); LNG_Dedge.txt; LNG_Initialtriangle.txt (right); LNG_Ratiocontrol.txt**

LNG_Dedge.txt is to specify boundary conditions. Namely, which edge has the zero Dirichlet boundary condition and which has the zero Neumann boundary condition. The first integer in the first row represents the number of non-duplicate endpoints (every point is counted only once) of Dirichlet edges, and the second integer is the number of Dirichlet edges. Note that we currently do not allow the Neumann-Neumann boundary condition on both adjacent edges of a node. The integers in the second row are the numbering of the endpoints of Dirichlet edges. For example, LNG_Dedge.txt in Figure 3 implies that there are eight Dirichlet nodes, and eight Dirichlet edges. With the numbering of endpoints, these edges are $1-2$, $2-3$, $\cdots$, and $8-1$, which imposes the homogeneous Dirichlet condition on the whole boundary of the L-shape in Figure 2. One will have no difficulty to impose mixed boundary conditions by understanding the format.

LNG_Ratiocontrol.txt controls the ratio of decay $0 < \kappa < 1$ (Figure 1) for specified singular nodes. The first integer of the file simply tells the program the number of singular nodes, while the decimal in the ith row, $i \geq 2$, is the particular ratio of decay of triangles near the $(i-1)$th singular node. As in the file LNG_Ratiocontrol.txt of Figure 3, there is one singular node on the L-shape domain (the first node) with the ratio of decay $\kappa = 0.2$.

More Source files for popular domains and boundary conditions (cracks, mixed boundary conditions, multiple singular nodes, etc.) can be found in LNG_Demo, which is downloadable on our web page. Users can either investigate the files for different situations to get used to the format, or replace the original files in /Sourcefiles by the given source files, run the program and see different solutions.

2.2. **Outputs.** When all source files are filled appropriately, one can open a command terminal and enter the directory /LNG_FEM, so that LNG_FEM.out, /Sourcefiles, and /Results are in the current directory. To activate the program, type ./LNG_FEM.out in the terminal and follow instructions to make decisions (Figure 4).

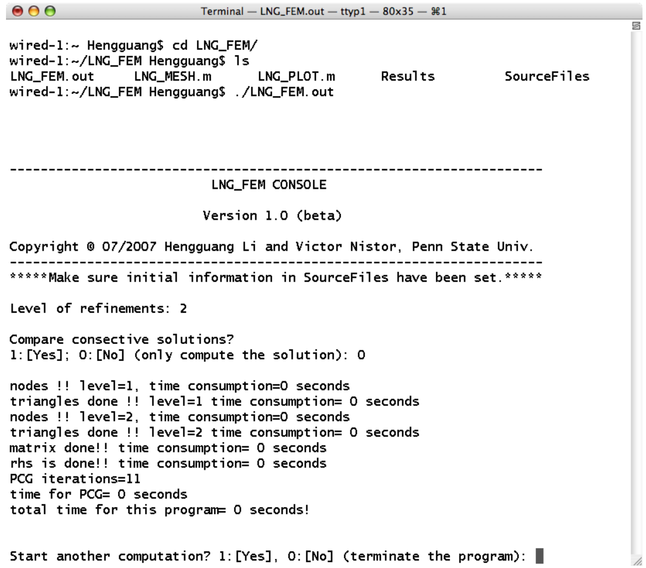


Figure 4. interface of LNG_FEM

We note that if one chooses to compare consecutive solutions, it always calculates the H^1-error between the current numerical solution and the solution from last implementation. Therefore, to compare solutions on the mesh of level 3 and on the mesh of level 4, we need to compute the solution on level 3 first, and make another run on level 4 for the comparison.

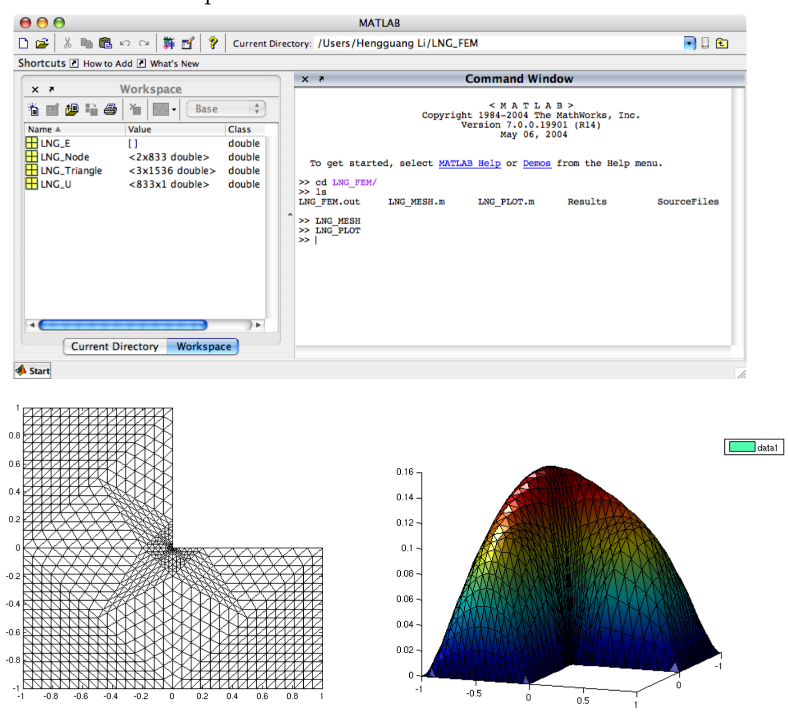


Figure 5. MATLAB graphs the 4th-level graded mesh (left) and the corresponding solution (right) on the L-shape domain from the initial triangulation in Figure 2.

Unlike merely computing the solution, LNG_FEM does not export information on the current mesh and on the solution if we make the choice to compare solutions. Therefore, when we choose to compare solutions in LNG_FEM and use MATLAB to sketch the solution, it actually graphs the solution from the previous level.

All outputs of the program are placed into the directory /Results. We format the files to fulfill the requirements in MATLAB for visualization of solutions. For users' convenience, we prepare two .m files LNG_MESH.m and LNG_PLOT.m in the package, for graphing graded meshes and solutions, respectively. These two commands and other MATLAB functions can be used when LNG_FEM finishes the computation and placing the outputs. After launching MATLAB, to use these two built-in commands, we set the current directory to be /LNG_FEM. Figure 5 shows a screen shot of MATLAB and the resulting pictures of these commands.

We conclude this subsection by briefly reviewing the files in the directory /Results: LNG_Node.txt includes the coordinates of nodes; LNG_Triangle.txt contains the vertex numbering of triangles; LNG_U.txt is the finite element solution; LNG_Rtp.txt specifies which triangles a node is in; LNG_E.txt is needed for graphing and LNG_Pre.txt is the number of nodes in the mesh.

2.3. **The processing flow and examples.** Details on the arrangement of source files and outputs have been done in the two subsections above, while a concise diagram (Figure 6) generates a macroscopic picture for the working procedure of the program.

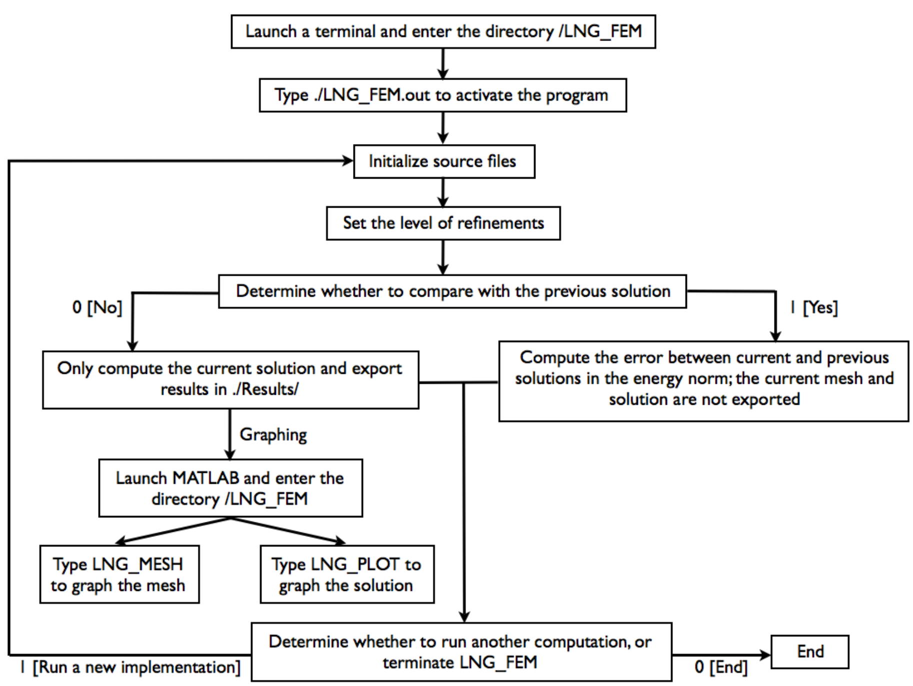


Figure 6. the processing flow of LNG_FEM

In addition, besides the example for the L-shape domain, we here include other pictures from LNG_FEM to help users get more experience on it. The source files of these examples are all available in LNG_Demo online.

1. A domain with a crack + the Dirichlet boundary condition.

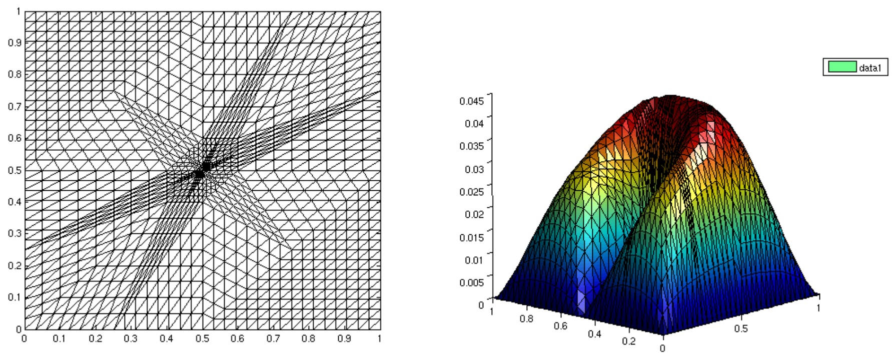


Figure 7. the graded meshes toward the tip of crack at (0, 0) after 4 refinements (left), $\kappa = 0.2$; the numerical solution of Equation (1), $f = 1$, on the mesh (right)

2. Mixed boundary conditions.

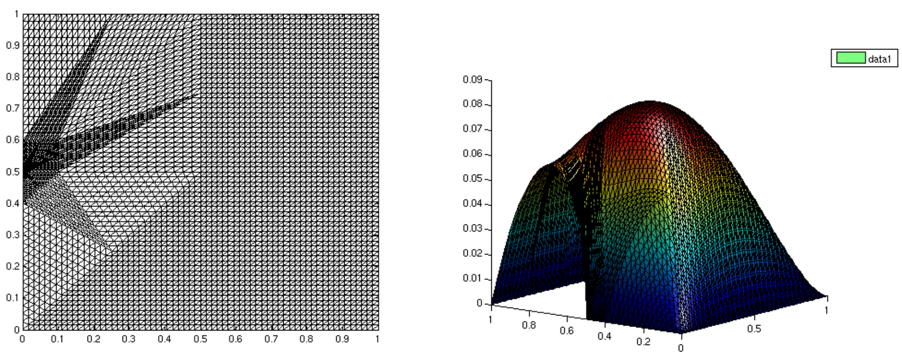


Figure 8. the graded meshes toward the point where the boundary condition changes (left), $\kappa = 0.2$, level=5; the numerical solution of Equation (1), $f = 1$, on the mesh (right)

3. A domain with multiple singular nodes.

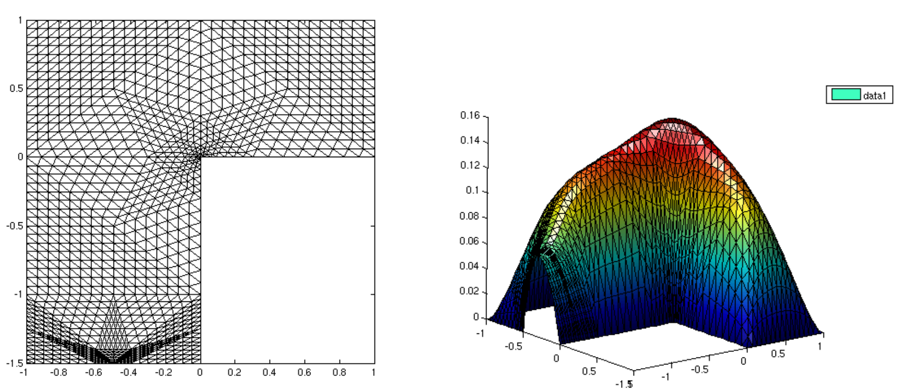


Figure 9. the graded mesh after 4 refinements for two singular points: re-entrant corner $\kappa = 0.3$ and mixed boundary conditions $\kappa = 0.2$ (left); the numerical solution for Equation (1) with $f = 1$ (right)

The optimality of the graded meshes in terms of the convergence rate of numerical solutions based on *a prior* estimates can be found in [7, 24].

3. The performance of LNG_FEM

In this section, we briefly describe the performance of LNG_FEM in terms of storage and speed. More technical details will be discussed in a forthcoming paper.

Equipped with advanced numerical algorithms and data structures, LNG_FEM is reliable, fast, and memory-efficient. It recognizes source files written in the appropriate format with the initial triangulation satisfying several rules in Section 2.

On the outside of the program, it handles general domains and different boundary conditions with a friendly interface to interact with users. Inside the package, we used the compact format [39] to store sparse matrices and vectors, and pointers to assign and release vectors dynamically to minimize the consumption of the memory. To be more precise, the memory needed for LNG_FEM to generate meshes, assemble matrices, solve the system of equations, and to compare solutions, is linearly dependent of the problem size (number of nodes). It is the optimal rate we can achieve. For example, starting with 8 initial triangles, as on the domain with a crack, LNG_FEM needs 1.7GB of memory to refine the mesh 10 times, which gives $2^{23} \approx 8.4 \times 10^6$ triangles, and 430MB of memory for the 9th refinement, with $2^{21} \approx 2.1 \times 10^6$ triangles. Therefore, we can easily go up to the 9th level on regular PCs, and more on relatively powerful machines.

For any numerical software, a usual question is, How fast is it? With the careful design of algorithms, we managed to optimize the procedure for the mesh generation and matrix assembly, such that the computation cost for them in the final triangulation almost linearly depends on the number of triangles. We timed the program for the refinement of the 10th level on the original domain with 8 triangles for the crack problem. The test was in Linux (Redhat 9.0) with two 2.8GHz Intel Xeon processors and 2GB of memory. The result is shown in the figure below.

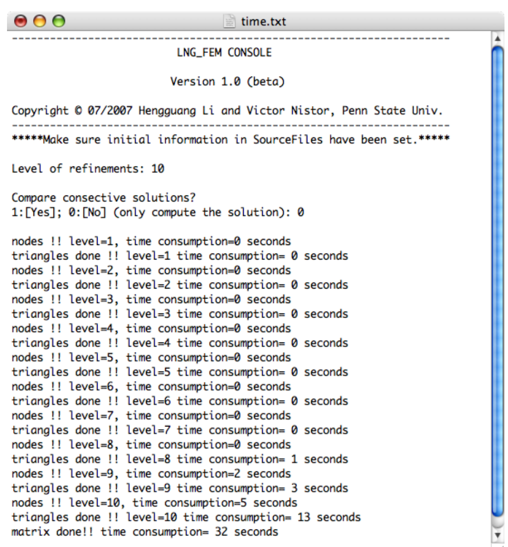


Figure 10. time consumption of LNG_FEM to generate 8.4 million triangles

It takes unmeasurable time for LNG_FEM to generate $2^{17} \approx 1.3 \times 10^5$ triangles, 6 seconds to generate 2.1 million triangles, and 24 seconds to generate 8.4 million triangles. Assembling the matrix takes a little longer, but only 32 seconds. In fact, the most time consuming part in the program is solving the system of equations. With the build-in PCG solver, we need to wait about 20 minutes to see the result on the 9th level, which is acceptable for the normal use. It is the only module that is not optimized in LNG_FEM. We are working on a multigrid solver and hope it will speed up this part soon.

Acknowledgments. We would like to thank Long Chen, Anna Mazzucato, and Ludmil Zikatanov for useful discussions and suggestions during the development of the software.

References

[1] T. Apel, M. Berzins, P. K. Jimack, G. Kunert, A. Plaks, I. Tsukerman, and M. Walkley. Mesh shape and anisotropic elements: theory and practice. In *The mathematics of finite elements and applications, X, MAFELAP 1999 (Uxbridge)*, pages 367–376. Elsevier, Oxford, 2000.

[2] T. Apel, A. Sändig, and J. R. Whiteman. Graded mesh refinement and error estimates for finite element solutions of elliptic boundary value problems in non-smooth domains. *Math. Methods Appl. Sci.*, 19(1):63–85, 1996.

[3] D. N. Arnold. *A Concise Introduction to Numerical Analysis*. 2001.

[4] I. Babuška. Finite element method for domains with corners. *Computing (Arch. Elektron. Rechnen)*, 6:264–273, 1970.

[5] I. Babuška and A. K. Aziz. *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*. Academic Press, New York, 1972.

[6] I. Babuška, R. B. Kellogg, and J. Pitkäranta. Direct and inverse error estimates for finite elements with mesh refinements. *Numer. Math.*, 33(4):447–471, 1979.

[7] C. Băcuţă, V. Nistor, and L. T. Zikatanov. Improving the rate of convergence of 'high order finite elements' on polygons and domains with cusps. *Numer. Math.*, 100(2):165–184, 2005.

[8] J. H. Bramble. *Multigrid Methods*. Chapman Hall CRC, 1993.

[9] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu. Convergence estimates for multigrid algorithms without regularity assumptions. *Math. Comp.*, 57(195):23–45, 1991.

[10] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In *Sparsity and its applications (Loughborough, 1983)*, pages 257–284. Cambridge Univ. Press, Cambridge, 1985.

[11] J. Brannick, H. Li, and L. Zikatanov. Uniform convergence of the multigrid v-cycle on graded meshes for corner singularities. 2007.

[12] S. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1994.

[13] Susanne C. Brenner. Convergence of the multigrid V-cycle algorithm for second-order boundary value problems without full elliptic regularity. *Math. Comp.*, 71(238):507–525 (electronic), 2002.

[14] F. Brezzi and M Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, New York, 1991.

[15] P. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 4 of *Studies in Mathematics and Its Applications*. North-Holland, Amsterdam, 1978.

[16] M. Dauge. *Elliptic Boundary Value Problems on Corner Domains*, volume 1341 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1988.

[17] L. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. AMS, Rhode Island, 1998.

[18] D. Gilbarg and N. Trudinger. *Elliptic Partial Differential Equations of Second Order*. Classics in Mathematics. Springer, Berlin, 1997.

[19] P. Grisvard. *Singularities in Boundary Value Problems*, volume 22 of *Research Notes in Applied Mathematics*. Springer-Verlag, New York, 1992.

[20] W. Hackbusch. *Multi-Grid Methods and Applications*. Computational Mathematics. Springer Verlag, New York, 1995.

[21] V. A. Kondrat′ev. Boundary value problems for elliptic equations in domains with conical or angular points. *Transl. Moscow Math. Soc.*, 16:227–313, 1967.
[22] V. A. Kozlov, V. Mazya, and J. Rossmann. *Elliptic Boundary Value Problems in Domains with Point Singularities.* AMS, Rhode Island, 1997.
[23] H. Li. The finite element method for a class of degenerate elliptic equations. IMA Preprint 2136 2006.
[24] H. Li, A. Mazzucato, and V. Nistor. On the analysis of the finite element method on general polygonal domains I: a priori estimates. 2007.
[25] H. Li and V. Nistor. Analysis of a modified schrödinger operator in 2d: regularity, index, and FEM. 2007.
[26] A. Mazzucato and V. Nistor. Well posedness and regularity for the elasticity equation with mixed boundary conditions on polyhedral domains and domains with cracks. 2007.
[27] P. Morin, R. Nochetto, and K. Siebert. Convergence of adaptive finite element methods. *SIAM Review*, 2002.
[28] S. A. Nazarov and B. A. Plamenevsky. *Elliptic Problems in Domains with Piecewise Smooth Boundaries*, volume 13 of *Expositions in Mathematics*. de Gruyter, New York, 1994.
[29] S Nicaise. *Polygonal Interface Problems.* Lang, Peter Publishing, Incorporated, 1993.
[30] S. Nicaise and A. Sändig. General interface problems. I, II. *Math. Methods Appl. Sci.*, 17(6):395–429, 431–450, 1994.
[31] J. Rauch. *Partial Differential Equations.* Graduate Texts in Mathematics. Springer-Verlag, Amsterdam.
[32] C. Schwab. *P- And Hp- Finite Element Methods: Theory and Applications in Solid and Fluid Mechanics.* Oxford University Press, 1999.
[33] M. Taylor. *Partial Differential Equations 1, Basic Theory*, volume 115 of *Applied Mathematical Sciences.* Springer-verlag, New York, 1995.
[34] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid.* Academic Press Inc., San Diego, CA, 2001. With contributions by A. Brandt, P. Oswald and K. St üben.
[35] P. Vassilevski. *Multilevel Block Factorization Preconditioners.* Springer-Verlag.
[36] L. B. Wahlbin. On the sharpness of certain local estimates for $\overset{\circ}{H}{}^1$ projections into finite element spaces: influence of a re-entrant corner. *Math. Comp.*, 42(165):1–8, 1984.
[37] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34(4):581–613, 1992.
[38] J. Xu and L. Zikatanov. The method of alternating projections and the method of subspace corrections in Hilbert space. *J. Amer. Math. Soc.*, 15(3):573–597 (electronic), 2002.
[39] L. Zikatanov. A note on FE assembly.

Hengguang Li, The Pennsylvania State University, Math. Dept., University Park, PA 16802, USA
E-mail address: li_h@math.psu.edu

Victor Nistor, The Pennsylvania State University, Math. Dept., University Park, PA 16802, USA
E-mail address: nistor@math.psu.edu