Contemporary Mathematics
Volume **586**, 2013
http://dx.doi.org/10.1090/conm/586/11655

LNG_FEM: Graded Meshes on 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 2D domains with straight edges for different boundary conditions. We shall focus on a detailed exposition on the implementation of the software. In addition, we demonstrate that LNG_FEM is equipped with advanced algorithms and data structures to perform efficiently in numerical tests. We hope that LNG_FEM can broaden the use and understanding of graded mesh in the finite element approximation of singular solutions.

We develop a software package entitled LNG_FEM, which comes from "Linear Graded Finite Element Method". LNG_FEM is a free software package, written in C, for the generation of graded meshes in general 2D domains with polygonal structures [**6, 7, 11**], and for the construction of linear finite element solutions of elliptic boundary value problems. As a by-product of our research on numerical approximations of singular functions, LNG_FEM is a fast, memory-efficient, user-friendly package that can handle different boundary conditions. LNG_FEM is designed for studying and demonstrating grading algorithms, as well as educating students on the finite element method.

This expository article shall mainly present instructions on the use of LNG_FEM (Section 2). In Section 3, we also briefly describe several features on the algorithms and data structures, which make LNG_FEM a reliable and efficient software package feasible for various problems. We keep the package up-to-date. Suggestions to improve the software in any aspect are welcome.

1. Graded meshes

It is well known that elliptic boundary value problems may have singular solutions, even when the given data is smooth. Some typical situations that lead to singular solutions include: non-smooth domains, changes in boundary conditions, interfaces (especially the non-smooth ones) in transmission problems, and differential operators with non-smooth coefficients. Various numerical schemes have been developed to improve the convergence rate of the finite element approximations of

2010 *Mathematics Subject Classification.* Primary 65Y04; Secondary 65N30.

H. Li was partially supported by the NSF Grant DMS-1158839. V. Nistor was partially supported by the NSF Grants OCI-0749202 and DMS-1016556. LNG_FEM is freely available at http://hengguangli.blogspot.com/2009/04/lngfem.html.

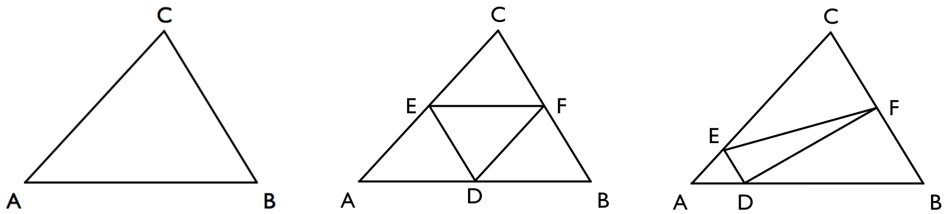


FIGURE 1. The initial triangle ABC (left); the uniform refinement, $\kappa = 0.5$ (center); the κ-refinement with $\kappa < 0.5$ for vertex A (right), $\kappa = \frac{|AD|}{|AB|} = \frac{|AE|}{|AC|} = \frac{|DE|}{|BC|}$.

these singular solutions. Among these numerical schemes, based on *a priori* estimates of the equation in special function spaces [**5, 9, 8, 13**], mesh grading has proved to be a powerful technique [**1, 4, 3, 2, 10, 11, 12, 14**].

Consider the following model problem in a polygonal domain Ω with the mixed boundary condition,

$$(1.1) \qquad -\Delta u = f \quad \text{in} \quad \Omega, \qquad \partial_\nu u = 0 \quad \text{on} \quad \partial\Omega_N, \qquad u = 0 \quad \text{on} \quad \partial\Omega_D.$$

Assuming a smooth f, let $S \subset \bar{\Omega}$ be the set of singular points near which the solution u is not locally in H^2. Then, there is a systematic construction of graded meshes [**3, 11, 14**] to deal with the lack of regularity of the solution at those points.

DEFINITION 1.1. Let $\mathcal{T}$ be a triangulation of Ω. We require that every point in S be a vertex in $\mathcal{T}$ and no two singular points belong to the same triangle. For each point in S, define a grading parameter $\kappa \in (0, 1/2]$. Then *the κ-refinement of $\mathcal{T}$*, denoted by $\kappa(\mathcal{T})$ is obtained by dividing each edge AB of $\mathcal{T}$ in two parts as follows. If neither A nor B is a singular point, then we divide AB into two equal parts. Otherwise, if $A \in S$, we divide AB into AC and CB such that $|AC| = \kappa|AB|$. This procedure will divide each triangle of $\mathcal{T}$ into four triangles.. Let $\mathcal{T}_0$ be an initial triangulation with the above properties. Then, we define by induction $\mathcal{T}_{n+1} = \kappa(\mathcal{T}_n)$.

For each point in the singular set S, there is an optimal range for the grading parameter κ [**3, 11**], determined by the regularity estimate of the solution u in weighted Sobolev spaces. Once κ is chosen within that range, the graded mesh $\mathcal{T}_n$ yields finite element approximations of equation (1.1) with optimal convergence rates. (This result applies to general uniformly strongly elliptic equations mentioned above with various singular solutions.)

2. Instructions

One of the purposes of the software package LNG_FEM is to encourage a broader understanding and use of graded meshes for the finite element approximation of singular solutions of elliptic PDEs. Therefore, in addition to ensuring the reliability and efficiency of the algorithms, we also try to maintain a friendly user interface. 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 two directories (/Sourcefiles and /Results, respectively), to simplify the initialization of the program and the analysis of the results afterwards.

2. Generality. The user is allowed to set up various parameters such as the

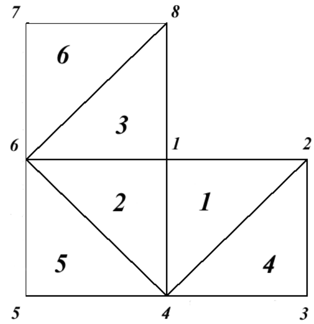


FIGURE 2. An initial triangulation of the L-shape domain with the Dirichlet boundary condition: the numbering of nodes and triangles.

computational domain, boundary conditions, and the grading parameter for each specific singular point, by customizing the input files. The current model problem in the package is equation (1.1) with $f = 1$. It is possible to work on more general equations by updating the corresponding module in the package.

3. ***Efficiency.*** LNG_FEM is equipped with advanced algorithms that perform efficiently in terms of storage and speed. We thus managed to minimize the time for mesh generation and matrix assembling. 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 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.

We now provide a detailed instruction on the implementation of the package.

2.1. Source files. After unzipping the downloaded file, we shall have created the directory /LNG_FEM, including two sub-directories (/Sourcefiles and /Results) and several other files. Note that the the executable file is .out. In particular, /Sourcefiles contains information on the initial triangulation, boundary conditions, and grading parameters. We elaborate on the settings in /Sourcefiles by taking the mesh in Figure 2 as an example.

Below are the rules for the initial triangulation.

1. Any vertex or singular point of the domain is an initial node.

2. An initial triangle cannot contain more than one singular point of the domain.

3. Any node cannot have more than six adjacent triangles.

4. Suppose that there are n initial triangles. The numbering of the triangles can be any one of the $n!$ permutations of the set $\{i,\ 1 \leq i \leq n\}$. Figure 2 shows only one possible numbering. Suppose there are l singular points and m non-singular nodes in the initial nodes. Then, the numbering of the singular nodes can be any one of the $l!$ permutations of the set $\{i,\ 1 \leq i \leq l\}$; the numbering of the non-singular nodes can be any one of the $m!$ permutations of the set $\{i,\ l+1 \leq i \leq l+m\}$. For example, "1" has to be assigned to the node on the re-entrant corner in Figure 2, because it is the only singular point. The numbering of the other seven nodes can be any one of the 7! permutations of the set $\{i,\ 2 \leq i \leq 8\}$.

More precisely, as in Figure 3, for **LNG_Initialnode.txt**, the integer in the first row indicates the number of nodes (eight nodes in Figure 2). Starting from the second row, the ith row, $i \geq 2$, lists the coordinates (x, y) of the $(i-1)$st initial node. (The second node is $(1, 0)$, for example.) **LNG_Initialtriangle.txt** contains

```
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. Files (in the directory /Sourcefiles) for the triangulation in Figure 2 with the grading parameter $\kappa = 0.2$: LNG_Initialnode.txt (left), LNG_Dedge.txt (top), LNG_Initialtriangle.txt (right), LNG_Ratiocontrol.txt (bottom).

information on initial triangles. The first number in the file stands for the number of initial triangles, while the three-tuple in the ith row, $i \geq 2$, identifies the vertices of the $(i-1)$st triangle with the numbering of vertices in the *ascending* order. For example, since the third triangle in Figure 2 has vertices $1, 6$, and 8, the 4th row of LNG_Initialtriangle.txt is 1 6 8 in Figure 3. With these two source files, it suffices to pass all geometric information of the domain to the program.

LNG_Dedge.txt is to specify boundary conditions, namely, the Dirichlet and Neumann edges, respectively. The first integer in the first row represents the number of non-duplicate endpoints (each point counted only once) of Dirichlet edges; the second integer is the number of Dirichlet edges. The integers in the second row are the numberings 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. The Dirichlet edges are $1-2$, $2-3$, $\cdots$, and $8-1$, imposing the pure Dirichlet condition on the L-shape domain. Mixed boundary conditions can be imposed similarly.

LNG_Ratiocontrol.txt controls the grading parameter $\kappa \in (0, 1/2]$ (Figure 1) for specified singular nodes. The first integer of the file is the number of singular nodes, while the decimal in the ith row, $i \geq 2$, is the grading parameter for the $(i-1)$st singular node. As in LNG_Ratiocontrol.txt (Figure 3), there is one singular node on the L-shape domain (the first node) and the grading parameter is $\kappa = 0.2$.

Source files for some typical 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. To implement these files, one can simply replace the original files in /Sourcefiles by the source files given in LNG_Demo.

2.2. Outputs. With all the source files ready, one can open a command terminal and enter the directory /LNG_FEM. To implement, type ./LNG_FEM.out in the terminal and follow the on-screen instructions (Figure 4).

Note that if we choose to compare consecutive solutions, it calculates the H^1-error between the current numerical solution and the solution from last implementation. Therefore, to compare solutions on the third-level mesh and on the 4th-level mesh, we need to compute the solution on the third level first and make another run on the 4th level for the comparison.

All the outputs are automatically placed into the directory /Results, with a MATLAB-recognizable format for the visualization of solutions. We prepared two .m files LNG_MESH.m and LNG_PLOT.m in the package, for graphing graded

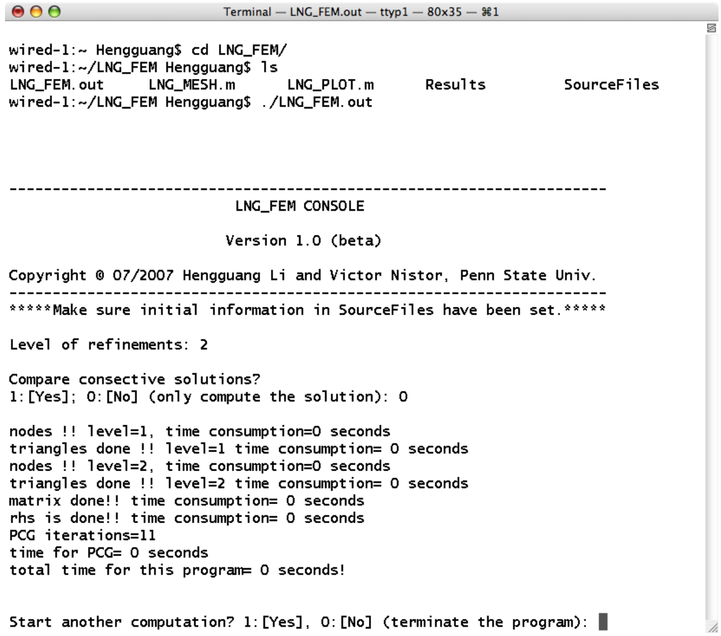


FIGURE 4. The interface of LNG_FEM.

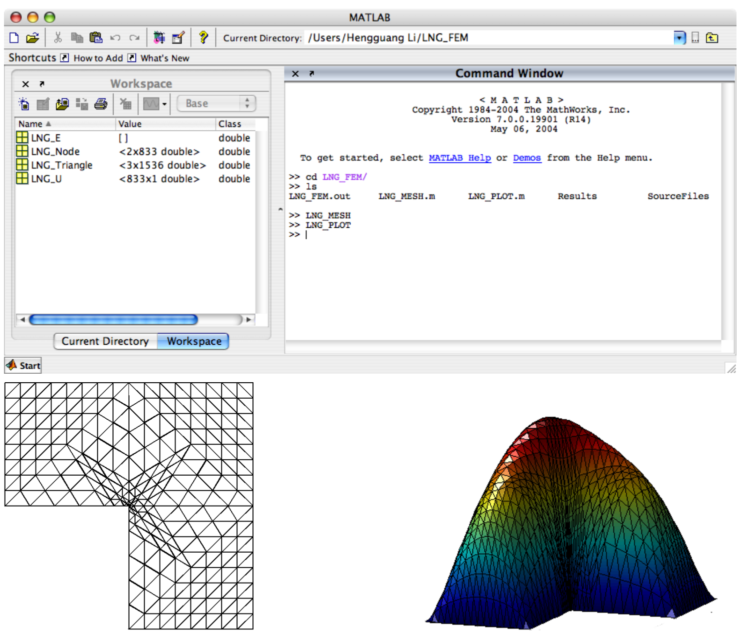


FIGURE 5. MATLAB graphs the 4th-level graded mesh (left) and the corresponding solution (right) on the L-shape domain from the initial triangulation in Figure 2.

meshes and solutions, respectively. After launching MATLAB, set the current directory to be /LNG_FEM. Figure 5 shows a screen shot of MATLAB and the

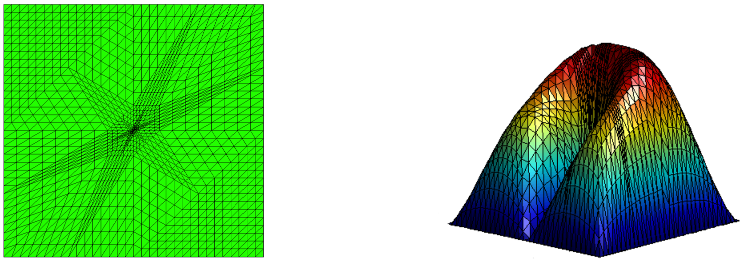
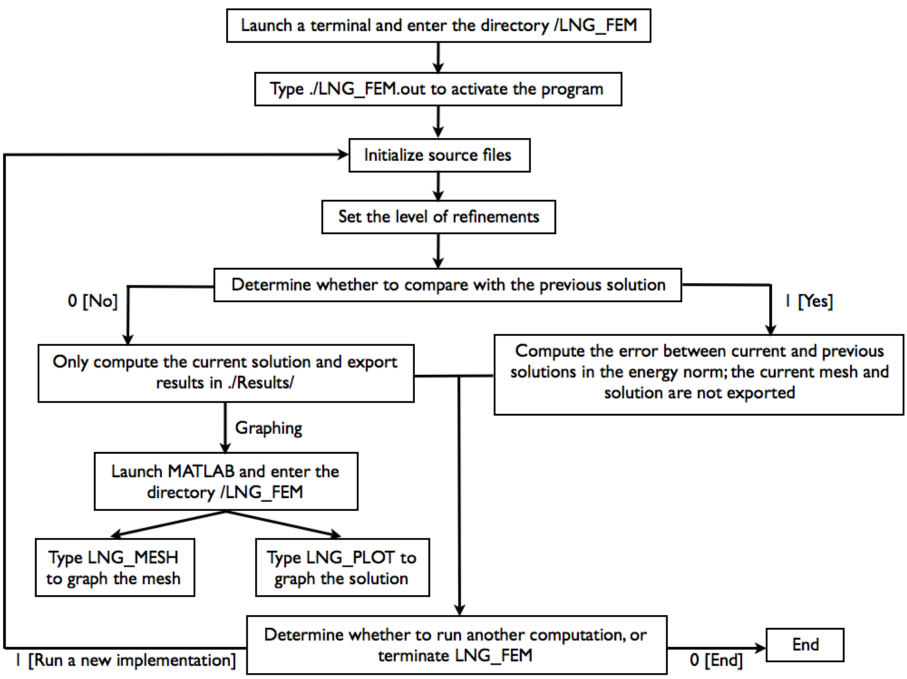


FIGURE 6. The graded mesh toward the tip of crack at the center of the domain after four refinements (left), $\kappa = 0.2$; the corresponding numerical solution of equation (1.1) (right).

resulting pictures of these commands. As a reminder, if we choose to compare solutions, LNG_FEM does not export the current mesh or the current solution.

The files in /Results are: LNG_Node.txt including the coordinates of nodes; LNG_Triangle.txt containing the vertex numbering of triangles; LNG_U.txt being the finite element solution; LNG_Rtp.txt specifying the triangles a node belongs to. LNG_E.txt is for graphing and LNG_Pre.txt is the number of nodes in the mesh.

2.3. The processing flow and examples. We provide a concise diagram below for the working procedure of LNG_FEM. In addition, besides for the L-shape domain, we include other examples (Figures 6 – 8) from the package for illustrations.



3. Algorithms

We used a compact format [**15**] to store sparse matrices and vectors. Pointers were used to assign and release vectors dynamically to minimize the use of the memory. In particular, the memory needed for LNG_FEM to generate meshes,

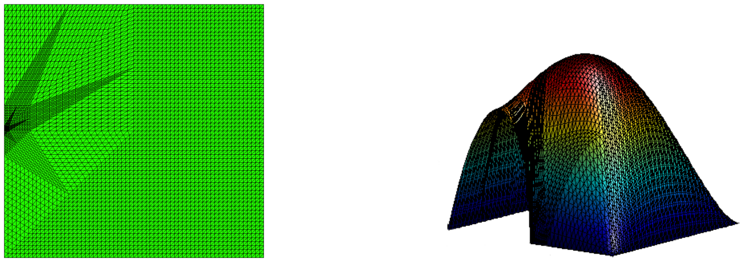


Figure 7. The graded mesh toward the point where the boundary condition changes, $\kappa = 0.2$, level=5 (left); the corresponding numerical solution of equation (1.1) (right).

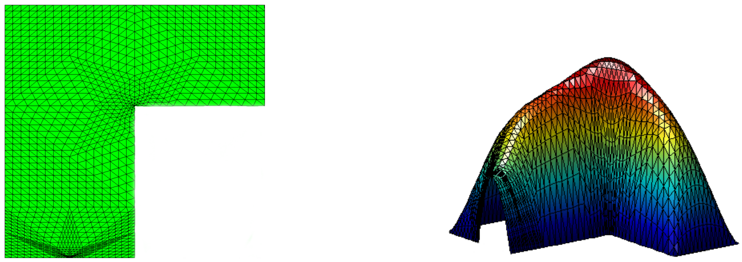
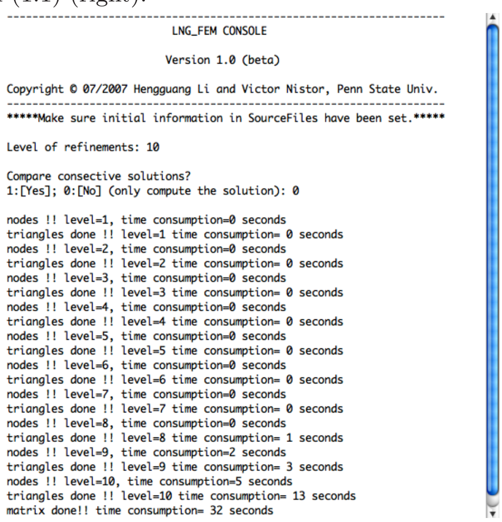


Figure 8. The graded mesh after four refinements for distinct singular points: re-entrant corner, $\kappa = 0.3$, and the point where the boundary condition changes, $\kappa = 0.2$ (left); the corresponding numerical solution for equation (1.1) (right).



```
-------------------------------------------------------------------
                         LNG_FEM CONSOLE

                       Version 1.0 (beta)

Copyright © 07/2007 Hengguang Li and Victor Nistor, Penn State Univ.
-------------------------------------------------------------------
*****Make sure initial information in SourceFiles have been set.*****

Level of refinements: 10

Compare consective solutions?
1:[Yes]; 0:[No] (only compute the solution): 0

nodes !! level=1, time consumption=0 seconds
triangles done !! level=1 time consumption= 0 seconds
nodes !! level=2, time consumption=0 seconds
triangles done !! level=2 time consumption= 0 seconds
nodes !! level=3, time consumption=0 seconds
triangles done !! level=3 time consumption= 0 seconds
nodes !! level=4, time consumption=0 seconds
triangles done !! level=4 time consumption= 0 seconds
nodes !! level=5, time consumption=0 seconds
triangles done !! level=5 time consumption= 0 seconds
nodes !! level=6, time consumption=0 seconds
triangles done !! level=6 time consumption= 0 seconds
nodes !! level=7, time consumption=0 seconds
triangles done !! level=7 time consumption= 0 seconds
nodes !! level=8, time consumption=0 seconds
triangles done !! level=8 time consumption= 1 seconds
nodes !! level=9, time consumption=2 seconds
triangles done !! level=9 time consumption= 3 seconds
nodes !! level=10, time consumption=5 seconds
triangles done !! level=10 time consumption= 13 seconds
matrix done!! time consumption= 32 seconds
```

Figure 9. The processing time for LNG_FEM.

assemble matrices, solve the system of equations, and to compare solutions, is linearly dependent of the problem size. For example, starting with eight initial triangles, as for the domain with a crack (Figure 6), LNG_FEM needs 1.7GB of memory to refine the mesh 10 times, which generates $2^{23} \approx 8.4 \times 10^6$ triangles; and it needs 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 10th level on regular desktops and more on relatively powerful machines.

The algorithms for the mesh generation and matrix assembling were carefully designed, such that the computational cost in the final triangulation almost linearly depends on the number of triangles. We timed the program for 10 consecutive refinements on the original domain with eight triangles for the crack problem (Linux Redhat 9.0 with two 2.8GHz Intel Xeon processors and 2GB of memory, Figure 9). It takes unnoticeable little 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, namely 32 seconds.

In fact, the most time consuming part is solving the system of equations. With the built-in PCG solver, it takes about 20 minutes for the 9th level. In fact, it is the only module that is not optimized in LNG_FEM. We are working on a multigrid solver which will definitely lead to a speed boost for the program.

References

1. T. Apel, A.-M. Sändig, and J. 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** (1996), no. 1, 63–85.
2. I. Babuška, R.B. Kellogg, and J. Pitkäranta, *Direct and inverse error estimates for finite elements with mesh refinements*, Numer. Math. **33** (1979), no. 4, 447–471.
3. C. Băcuţă, V. Nistor, and L. Zikatanov, *Improving the rate of convergence of 'high order finite elements' on polygons and domains with cusps*, Numer. Math. **100** (2005), no. 2, 165–184.
4. S.C. Brenner, J. Cui, T. Gudi, and L.-Y. Sung, *Multigrid algorithms for symmetric discontinuous Galerkin methods on graded meshes*, Numer. Math. **119** (2011), no. 1, 21–47.
5. M. Costabel, M. Dauge, and C. Schwab, *Exponential convergence of hp-FEM for Maxwell equations with weighted regularization in polygonal domains*, Math. Models Methods Appl. Sci. **15** (2005), no. 4, 575–622.
6. M. Dauge, *Elliptic boundary value problems on corner domains*, Lecture Notes in Mathematics, vol. 1341, Springer-Verlag, Berlin, 1988.
7. P. Grisvard, *Singularities in boundary value problems*, Research Notes in Applied Mathematics, vol. 22, Springer-Verlag, New York, 1992.
8. V.A. Kondrat′ev, *Boundary value problems for elliptic equations in domains with conical or angular points*, Transl. Moscow Math. Soc. **16** (1967), 227–313.
9. V.A. Kozlov, V. Mazya, and J. Rossmann, *Spectral problems associated with corner singularities of solutions of elliptic equations*, Mathematical Surveys and Monographs, vol. 5, AMS, Rhode Island, 2001.
10. H. Li, *A-priori analysis and the finite element method for a class of degenerate elliptic equations*, Math. Comp. **78** (2009), 713–737.
11. H. Li, A. Mazzucato, and V. Nistor, *Analysis of the finite element method for transmission/mixed boundary value problems on general polygonal domains*, Electron. Trans. Numer. Anal. **37** (2010), 41–69.
12. H. Li and V. Nistor, *Analysis of a modified Schrödinger operator in 2D: regularity, index, and FEM*, J. Comput. Appl. Math. **224** (2009), no. 1, 320–338.
13. S. Nicaise, *Polygonal interface problems*, Lang, Peter Publishing, Incorporated, 1993.
14. G. Raugel, *Résolution numérique par une méthode d'éléments finis du problème de Dirichlet pour le laplacien dans un polygone*, C. R. Acad. Sci. Paris Sér. A-B **286** (1978), no. 18, A791–A794.
15. L. Zikatanov, *A note on FE assembly*, 2006.

Hengguang Li, Department of Mathematics, Wayne State University, Detroit, MI 48202, USA. E-mail address: hli@math.wayne.edu

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