



THESIS  
2  
2000



This is to certify that the  
dissertation entitled

**A FICTITIOUS DOMAIN SOLVER FOR COMPUTATION OF THE  
EFFECTIVE PROPERTIES OF MATERIAL DISTRIBUTIONS  
GENERATED BY ITERATED AFFINE**

presented by

*Edward Brennan*

has been accepted towards fulfillment  
of the requirements for

Master's degree in Mechanical Engineering

Dr. Alejandro Diaz

  
Major professor

Date April 26, 2000

**LIBRARY**  
**Michigan State**  
**University**

**PLACE IN RETURN BOX** to remove this checkout from your record.  
**TO AVOID FINES** return on or before date due.  
**MAY BE RECALLED** with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

**A FICTITIOUS DOMAIN SOLVER FOR COMPUTATION OF THE EFFECTIVE  
PROPERTIES OF MATERIAL DISTRIBUTIONS GENERATED BY ITERATED AFFINE  
MAPS**

By

**Edward M. Brennan**

**A THESIS**

Submitted to  
**Michigan State University**  
In partial fulfillment of the requirements  
For the degree of

**MASTER OF SCIENCE**

Department of Mechanical Engineering

2000

## ABSTRACT

# A FICTITIOUS DOMAIN SOLVER FOR COMPUTATION OF THE EFFECTIVE PROPERTIES OF MATERIAL DISTRIBUTIONS GENERATED BY ITERATED AFFINE MAPS

BY

*EDWARD M. BRENNAN*

The effective constitutive properties of a homogeneous material with a fractal shaped microstructure were solved in two dimensions. The change in material stiffness versus change in fractal mapping depth was studied.

Example problems were solved and compared to an optimal rank-2 geometry.

An image-based analysis was performed to obtain the effective material properties. The fictitious domain was used to embed the base cell with periodic boundary conditions. Then, the domain was discretized using finite elements and a Preconditioned Conjugate Gradient (PCG) solver minimized the potential energy. The preconditioner matrix was inverted using a second PCG algorithm. This PCG inside PCG approach makes the convergence rate relatively insensitive to problem scale. Example problems are included that study sensitivity to problem size.

## ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Alejandro Diaz. Your guidance and support during my time here at Michigan State University has been invaluable. I would also like to thank my committee members Dr. Feeny and Dr. Benard. I appreciate your time and effort in reviewing this work as well as answering my questions.

# Table Of Contents

## Chapter 1

Introduction .....	1
1.1 Research Objectives.....	4
1.2 Approach .....	6
1.3 Organization of Thesis .....	7

## Chapter 2

Iterative Solution Of Elasticity Problems .....	9
2.1 Introduction.....	9
2.2 Fictitious Domain Method .....	10
2.3 Solution Using Finite Elements .....	14
Governing Equations .....	14
Boundary Conditions .....	16
Evaluation of Element Matrices.....	18
2.4 PCG Iterative Solver.....	19
Need For An Iterative Solver.....	20
2.4.1 Matrix Preconditioning.....	21
The homogeneous stiffness matrix.....	21
2.4.2 PCG in PCG Method .....	23
First Preconditioner .....	25
Second Preconditioner.....	25
2.5 Performance Of The Two Level PCG Algorithm .....	27
2.5.1 Sensitivity To Problem Size .....	27
2.5.2 Affect Of Changing Domain Resolution.....	28
2.5.3 PCG in PCG Development.....	32
2.6 Accuracy .....	35
2.6.1 Stressed Plate With Hole.....	35
2.6.2 'El' Bracket Example .....	39
2.6.3 Annular Circle Example .....	41
2.7 Summary .....	43

## Chapter 3

Computation of Constitutive Properties .....	44
3.1 Background .....	44
3.2 Homogenization Method.....	46
Problem At Small Scale In Cell Y.....	53
Problem At The Large Scale, The Homogenized Problem On $\Omega$ .....	53
Boundary Conditions .....	56
3.3 Verification of Accuracy .....	57

3.4 Summary .....	59
<b>Chapter 4</b>	
<b>Analysis Of Structures Created By Iterated Affine Maps.....</b>	<b>61</b>
4.1 Introduction.....	61
4.2 Background .....	62
4.2.1 Key Concepts.....	62
4.2.2 Iterative Affine Mapping .....	65
4.3 Performance of Iterative Affine Maps.....	67
4.3.1 Problem Statement .....	67
4.3.2 Baseline Material Distribution .....	69
4.3.3 Strain Energy Density Formulation For Rank-2 Materials .....	71
4.4 Examples .....	72
Sponge' Geometry Example .....	73
'Plus' Geometry Example .....	76
'EI' Geometry Example.....	77
'Inv4' Geometry Example.....	82
'Inv8' Geometry Example.....	84
'Fd4' Geometry Example .....	86
Relative Performance of Fractal Shapes .....	88
4.5 Summary .....	91
<b>Chapter 5</b>	
<b>Conclusions .....</b>	<b>92</b>
5.1 Summary .....	92
PCG Performance .....	92
Performance Of Fractal Patterns.....	93
5.2 Areas Of Future Work .....	93
Bibliography .....	95
<b>Appendix A .....</b>	
<b>A.1 Coefficients for Fractal Maps.....</b>	<b>97</b>
<b>A.2 Stress In An Axially Loaded Block.....</b>	<b>100</b>
<b>A.2 Analytical Solution Of The Stressed Plate With Hole.....</b>	<b>101</b>
<b>Appendix B: Description Of Subroutines .....</b>	<b>102</b>
<b>Appendix C: Program Input.....</b>	<b>106</b>

## List of Figures

1.1	Example of geometric pattern applied twice	3
2.1	Example design domain embedded in fictitious domain	10
2.2	Illustration of periodicity of the fictitious domain	11
2.3	Design domain with constraints and loading	16
2.4	PCG in PCG algorithm	24
2.5	Design domain $\omega$ and boundary conditions for performance trials	28
2.6	Adjustment of design domain size and resolution	29
2.7	PCG1 Iterations	30
2.8	PCG2 iterations	31
2.9	PCG Iterations for test preconditioner	33
2.10	PCG Iterations for test preconditioner	34
2.11	PCG1 Iterations for PCG in PCG method	34
2.12	Stressed plate with hole	36
2.13	Problem domain with symmetry	36
2.14	Stress contour plot of stressed plate with hole	38
2.15	Stress contour plot generated by ANSYS 5.6	38
2.16	Loading and geometry of 'el' bracket	39
2.17	Stress contour plot of Bracket	40
2.18	ANSYS 5.6 analysis of 'el' bracket	40
2.19	Annular circle with boundary conditions	41
2.20	Stress contour plot of disk with hole	42
2.21	Stress contour plot from ANSYS 5.6	42
3.1	Periodic microstructure in a homogenized material	46
3.2	Periodic boundary conditions applied to fictitious domain	57
3.3	Rank-1 material used for algorithm verification	58
4.1	Geometry generated by parameters in table 1	64
4.2	Rank-2 material distribution at two levels	69
4.3	Parameters $\gamma$ and $\delta$ characterize the rank-2 pattern	70
4.4	'Sponge' fractal geometry	73
4.5	Fractals generated by successive mappings from table 4.1	74
4.6	Stiffness analysis for sponge shaped geometry of varying density	75
4.7	'Plus' fractal geometry	76
4.8	'El' fractal geometry	77
4.9	'Plus' shaped geometry	78
4.10	Stiffness of 'Plus' shape	79
4.11	'El' pattern generated by successive mappings	80
4.12	Stiffness of 'el' shape	81
4.13	'Inv4' fractal geometry	82
4.14	Stiffness of the 'inv4' shape	83
4.15	'Inv8' fractal geometry	84
4.16	Stiffness of the 'inv8' shape	85
4.17	'f4d' fractal geometry	86
4.18	Stiffness of the 'fd4' shape	87

4.19	Stiffness comparison of fractal patterns for $\varepsilon_{II}/\varepsilon_1 = 1.0$ .....	89
4.20	Stiffness comparison of fractal patterns for $\varepsilon_{II}/\varepsilon_1 = 0.5$ .....	90
4.21	Stiffness comparison of fractal patterns for $\varepsilon_{II}/\varepsilon_1 = -0.75$ .....	90
A.1	'Stress analysis of fractal geometry depth 2.....	100
A.3	Illustration of coordinate axis for plate with hole .....	101

# Chapter 1

## Introduction

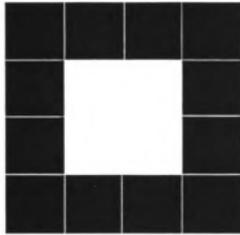
Material can be removed from a homogeneous structure by punching holes in the structure. Along with the resulting reduction in weight, the material stiffness is also reduced. The stiffness of the structure is a function of its weight as well as the special distribution of the holes that are punched. Naturally, some structural shapes created by hole punching are stiffer than others for a given loading, even if the structures have the same weight. Now consider that a geometrical pattern is employed to determine where material is to be removed from a homogeneous domain (or where to punch holes in a material). As expected, for a given weight, some geometrical patterns will be better at keeping a structure stiff than others.

The goal of this project is to study the stiffness of a structure as weight is systematically removed using geometric patterns. By means of mathematical

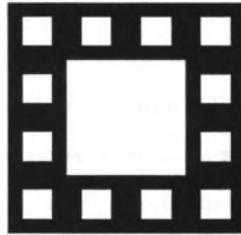
formulas, highly structured geometric patterns can be generated in an iterative fashion to choose where to remove material (where to the punch holes). Then, these patterns can be applied to a domain of uniform material. As more iterations of the geometrical pattern are applied, more material is removed. As the number of iterations approaches infinity, the material weight will approach zero.

When a material with a microstructure is viewed from a macroscopic vantage point, it appears to be homogeneous. But, when one zooms in for closer inspection, one can see the fine geometrical pattern of the microstructure. The smallest repetitive unit of holes in a microstructure is called a base cell. Here it is assumed that, the micro-pattern of the entire structure can be represented by just the base pattern that is repeated periodically. It will be shown that the periodicity of the base cell makes the analysis of a material microstructure possible.

The type of geometric pattern employed in this project has some interesting properties. The pattern can be applied once to slice up the homogeneous base cell into pieces, removing a portion of material. Figure 1.1a shows a solid square broken up into 12 solid pieces and one large void in the center. Then, the same pattern can be re-applied, or mapped, to each remaining solid piece. Figure 1.1b shows this same pattern applied to each of the 12 remaining solid pieces in figure 1.1a to produce 144 solid pieces and 13 holes. The result is a geometric pattern that is systematically refined into pieces containing the same pattern. This patterning technique generates what is known as fractal geometry.



(a) One iteration



(b) Two iterations

Figure 1.1: Example of geometric pattern applied twice

Fractal patterns are often encountered in nature. One such example is the branch configuration of a tree. A branch has the same pattern as the tree itself. Furthermore, a twig that extends from the branch has the same ‘tree’ pattern repeated again and so on.

This project presents the first time that the mechanical properties of materials created by these unique geometrical patterns have been analyzed. Does a certain geometrical pattern contain any inherent advantage as a method of weight reduction? How does its stiffness / weight ratio behave as material is removed? What happens to the material properties of a fixed domain as the geometry of the patterns is changed? These are some of the questions that this project attempts to answer.

Using an iterative pattern to reduce material weight requires a very fine mesh when performing analysis. As expected, the associated system of equations used to solve the elasticity problem is very large. For example, the problem domain

presented in figure 1.1b is twelve times larger than in figure 1.1a after just one iteration. That increase in complexity corresponds to the number of degrees of freedom (using finite element quadrilaterals) increasing by a factor of 16 after just one iteration. Thus, a new analysis tool had to be created to efficiently solve large systems of equations. The state of the art consists of analysis using the fictitious domain and wavelet compression to iteratively solve problems with a large number of degrees of freedom. DeRose [8] recently used a wavelet-Galerkin method to solve state equations for his optimization program. The wavelet approach was developed to solve these problems because it was thought that convergence rates using finite elements would not be acceptable. This project proposes that finite elements can result in suitable convergence rates for 2-D elasticity analysis.

## 1.1 Research Objectives

There are a few reasons why fractal microstructures are of interest in mechanics. First, as already mentioned, fractal patterns are often encountered in nature. Secondly, an iterative-mapping algorithm can systematically remove material from a homogeneous unit cell as shown in figure 1.1. More importantly, certain mappings can remove material away from cell boundaries. This is a favorable feature for keeping a cell stiff while removing material. Finally, fractal patterns have not been used in this context and therefore finding the strengths and weaknesses of such a composite can provide new insights into the problem.

Obtaining a measurement of material stiffness requires that one first solve a plane elasticity problem. The finite element method is a common means to numerically approximate the solution to this problem. However, the analysis of the fractal problem required a new, more efficient, finite element solution, due to the large number of elements required to define a refined fractal shape. This memory requirement forced the use of an iterative solution, which leads to the second objective of the project: to create a design tool to perform finite element analysis and calculate the effective properties of a base cell. The analysis package was to have several exceptional components. It was preferred to employ a meshless fictitious domain. The reasons for this will become evident by a discussion in chapter 2. Also, the iterative solution developed was required to be insensitive to problem size. While an iterative solver requires more computation time, it uses much less in-core computer memory. In addition, using preconditioning schemes can reduce the number of iterations and computation time. The ‘fractal analyzer’ package was written in FORTRAN 77 programming language to be a stand alone executable. To meet the stated objectives, the development of this rather complex program was broken down into the following phases:

1. *Develop global stiffness matrix assembler.* Because of the uniform elements encountered when using a fictitious domain, the element assembly is a critical first development of the solver.

2. *Develop a stand-alone finite element analysis (FEA) program.* By creating a stand-alone FEA program, the skeletal subroutines were tested and verified. Well know problems were solved and verified with closed form solutions.
3. *Develop an iterative solver.* Memory storage requirements needed to be verified and tested. This required the development of a new iterative solver.
4. *Develop code to solve for effective material properties.* Effective cell properties were calculated using a solution method first presented by Bensousson and Lions [5] and described in chapter 3. The user only needs to input the base cell image and a few material properties to obtain the effective constitutive properties. This type of procedure is called an image-based analysis.

## 1.2 Approach

This project consisted of two main thrusts. First, to develop a two dimensional elasticity analysis code employing use of the fictitious domain, finite elements, and an iterative Preconditioned Conjugate Gradient (PCG) solver. Second, to use the aforementioned program to analyze effective properties of material with microstructure generated from iterated affine maps.

An image based analysis code was developed to solve the large-scale elasticity problems with minimal memory requirements. During this development, new approaches were used to solve the system of linear equations. A conjugate gradient algorithm was chosen to solve the system of equations. To increase performance,

system preconditioning in the conjugate gradient algorithm was studied. An efficient preconditioning scheme was achieved using a 2-level PCG algorithm, inverting the preconditioner using again a PCG algorithm. The technique made the number of iterations of the outer level PCG insensitive to the scale of the problem number of elements. This insensitivity to scale, along with efficient memory management, makes the solution of large-scale problems possible.

The effective properties of homogeneous materials were computed in two dimensions using a solution presented by Sigmund [17]. This method was used to solve for the effective properties of base cells exhibiting different geometrical patterns. Lastly, the material stiffness of these base cells was compared to the optimal rank-2 material solution, e.g. as reported by Bendsoe [3]. The rank-2 material is a layered, multi-scale material that exhibits optimal stiffness under specific loading conditions.

### 1.3 Organization of Thesis

The remainder of the thesis is presented as follows. Chapter 2 describes the methods developed and used in the ‘fractal analyzer’ program. The preprocessor and solver are described and examples that investigate accuracy and sensitivity to problem size are included. Chapter 3 discusses the application of ‘fractal analyzer’ to the calculation of effective material properties using homogenization techniques. Simple rank-1 materials are analyzed for software verification. Chapter 4 presents

the analysis of material microstructure created by successive mappings of a fractal pattern. The results are compared to the benchmark rank-2 material. Appendix A gives supporting data for fractal shapes. Appendix B gives a detailed description of each subroutine used in the program. Appendix C details the input to the program.

# Chapter 2

## Iterative Solution Of Elasticity Problems

### 2.1 Introduction

The analysis of an elasticity problem where material microstructures are characterized by refined fractal geometry requires a large number of finite elements to accurately approximate the exact solution. Thus, the opportunity arose to develop and use a new method to solve large problems. This section discusses the algorithm for the preprocessor and processor used in this analysis. Problem discretization using a fictitious domain and solution using a new iterative minimization scheme are presented. Examples are included to study

accuracy and sensitivity to problem size. The examples in this paper are for 2-D problems but these methods can be extended to 3-D.

## 2.2 Fictitious Domain Method

The preprocessor discretizes the domain using the so-called fictitious domain method. The design domain is embedded into a regular periodic domain consisting of comparatively weak material that is dubbed the fictitious domain. The mesh is uniform over the entire domain. This grid is similar to the pixelized image of a computer screen. Figure 2.1 shows the design domain represented by solid material  $\omega$ . The fictitious domain containing void, or no material, surrounds the design domain. The total domain is labeled  $\Omega$  and is made up of the fictitious domain plus the design domain.

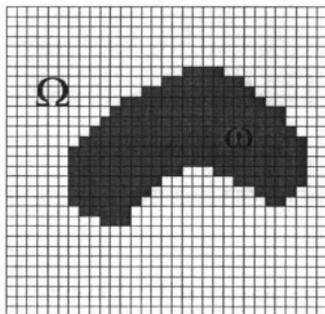


Figure 2.1: Example design domain embedded in fictitious domain

The boundary conditions on the fictitious domain are periodic. The degrees of freedom at the far right of the mesh are the same as those at the left edge of the mesh. Figure 2.2 shows the location of point 'p' in two places on the fictitious domain. Point 'a' is located at each of the domain's four corners. In a sense, the mesh 'wraps' around to the other side. The periodicity can also be thought of as an infinitely repeating mesh in each direction (up, down, left, and right). As will be shown in chapter 3, this setting is advantageous in computations of effective properties of material mixtures.

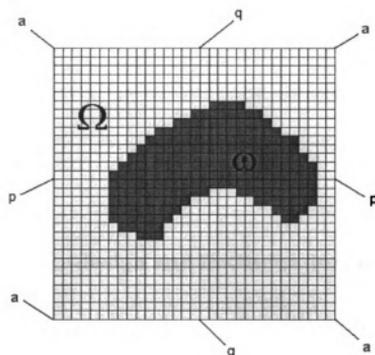


Figure 2.2: Illustration of periodicity of the fictitious domain

For an appropriate choice of fictitious material, the solution to an elasticity problem defined on the periodic domain closely approximates the solution in the original design domain. Thus, the following two problem statements yield approximately the same solution:

**Problem 1 Solution over design domain:**

Find  $u(x_1, x_2) \in R^2$  that

$$\text{Minimizes } \Pi_{\omega}(u) = \frac{1}{2} \int_{\omega} D_{\omega} \varepsilon(u) \varepsilon(u) d\omega - \int_{\Gamma} t u d\Gamma \quad (2.1)$$

Subject to  $u = 0$  on  $\Gamma^u \subset \partial\omega$

Where,

$D_{\omega}$  = material elasticity matrix

$\Pi_{\omega}$  = Potential energy

$t$  = surface traction

$\varepsilon$  = strain

**Problem 2 Solution over design and fictitious domain  $\Omega$ :**

Find  $u_{\Omega} \in R_2$  that

$$\text{Minimizes } \Pi_{\Omega}(u) = \frac{1}{2} \int_{\Omega} D_{\Omega} \varepsilon(u) \varepsilon(u) d\Omega - \int_{\Gamma} t u d\Gamma \quad (2.2)$$

Subject to  $u = 0$  where  $\Gamma^u \subset \partial\omega$  and  $\Gamma'$

These two statements yield approximately the same solution in part because the material in the fictitious domain is defined as very weak compared to the material in the design domain.

An indicator function  $\chi$  can be used to describe the two distinct material constituents that make up the domain, solid and void, i.e.,

$$\forall x \in R, \quad \chi(x) = \begin{cases} 1 & \text{If } x \in \omega \\ 0 & \text{If } x \notin \omega \end{cases} \quad (2.3)$$

The element material stiffness matrix is multiplied by the indicator function before it contributes to the global stiffness matrix. Therefore, the fictitious domain has no effect on the solution in the design domain (as long as the degrees of freedom in the design domain are constrained).

The effective material density given by the indicator function as either 0 or 1 controls the stiffness of each element in the domain. For plane stress, using an isotropic material, the material matrix is expressed as,

$$D = \chi(x) \times \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1+\nu)}{2} \end{bmatrix} \quad (2.4)$$

where,

**E** = Young's Modulus

**$\nu$**  = Poisson's Ratio

**$\chi$**  = Material indicator function (0 = void; 1 = solid)

The fictitious domain mesh has several advantages over the standard mesh found in a common commercial finite element (FE) package. First of all, its geometry is very straightforward. This translates into fast meshing that requires little user experience. Secondly, since the elements are of the same size and shape, numerical integration of each element is not required. Thirdly, as will be shown later, this setting facilitates the construction of an effective preconditioner. One disadvantage of using the fictitious domain is the need to use a fine resolution for objects with a large amount of curvature.

## 2.3 Solution Using Finite Elements

Until recently, linear elastic analysis using the fictitious domain had been done only using Wavelet-Galerkin methods. This next section presents the problem formulation and solution using finite elements.

### Governing Equations

The finite element method is used to solve the plane stress governing equations:

$$\begin{aligned}\frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + f_x &= 0 \\ \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + f_y &= 0\end{aligned}\tag{2.5}$$

where,

$\sigma_x$  = axial stress in the x-direction

$\sigma_y$  = axial stress in the y-direction

$\sigma_{xy}$  = shear stress in the x-y plane

$f_x$  and  $f_y$  = body forces in the x and y directions respectively,

along with the constitutive equation,

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} = [D] \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_{xy} \end{Bmatrix} \quad (2.6)$$

Using the minimum potential energy theorem, and upon discretization using finite elements, problem (2.5) becomes

Find  $u \in \Omega$  that

$$\text{Minimizes } \Pi = \frac{1}{2} u^T K u - u^T F \quad (2.7)$$

Subject to  $u = 0$  on  $\Gamma^u \subset \partial\omega$

where,

$K$  = Global stiffness matrix of size  $n$  by  $n$  ( $n$  = number of degrees of freedom)

$u$  = Nodal displacement vector of size  $n$

$F$  = Nodal source vector of size  $n$

$\Pi$  = Potential Energy

### Boundary Conditions

This section discusses the application of natural boundary conditions and essential boundary conditions. Figure 2.3 illustrates a typical problem domain with kinematic boundary conditions on the bottom face and traction on the right face. The design domain  $\omega$  is embedded in the fictitious domain. The essential boundary conditions are applied on  $\Gamma^u \subset \partial\omega$ . The traction, is applied along the boundary  $\Gamma^t$ .

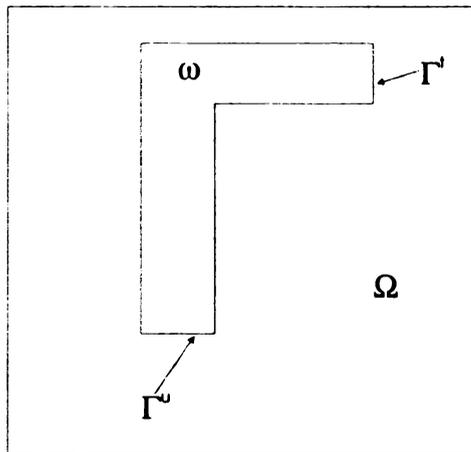


Figure 2.3: Design domain with constraints and loading

Two methods of imposing the constraint boundary conditions,

$u = 0$  where  $\Gamma^u \subset \partial\omega$ , have been studied.

1. A penalty method.
2. Row / column elimination.

Below is a brief discussion of each method.

The penalty method involves reformulating the problem into one of finding the minimum of a modified function,

$$\min \Pi_{\epsilon}(x, y) = \Pi(x, y) + \frac{1}{2} \left( \frac{1}{\epsilon} \right) [H(x, y)]^2 \quad (2.8)$$

Here the objective function is  $\Pi$  subject to the constraint equation  $H(x,y)=0$ .

Then partial differentiation is performed with respect to the modified function  $\Pi_{\epsilon}$  (the  $\frac{1}{2}$  is included in equation 2.8 to simplify differentiation). As the penalty parameter  $(1/\epsilon)$  becomes larger, the constraint equation is increasingly satisfied. In the limit of  $(1/\epsilon)$  approaching infinity, the solution to the modified problem (2.8) approaches the exact solution.

$$\begin{aligned} \frac{1}{\epsilon} &\rightarrow \infty \\ u_{\epsilon} &\rightarrow u \end{aligned} \quad (2.9)$$

The second method involves setting to zero the off-diagonal rows and columns associated with degrees of freedom on  $\Gamma^u$ . Equation 2.10 shows the imposition of boundary conditions to matrix  $[K]$  of equation 2.7.

$$K_0 = [M] + ([I] - [M]) \times [K] \times ([I] - [M]) \quad (2.10)$$

where,

$M$  = diagonal matrix with  $M(i,i) = 1$  where  $\Gamma^{u_i} \subset \partial\omega$ , otherwise  $M(i,i) = 0$ .

$I$  = Identity Matrix

One should note that in problems associated with the computation of effective properties, the only boundary conditions present are the periodic boundary conditions on  $\Omega$ , i.e.,  $\omega$  and  $\Omega$  are identical and  $\Gamma^u = \Gamma^v = \emptyset$ .

### **Evaluation of Element Matrices**

Each pixel in the mesh is treated as a four node quadrilateral finite element.

The element has one node at each corner with 2 degrees of freedom per node.

Because all elements have the same square shape, numerical integration of the

element stiffness matrix is not necessary. When the Jacobian matrix is

calculated, the off diagonal terms are zero. Hence, the stiffness matrix will scale

exactly as is shown by the Jacobian matrix below where  $h$  is the length of an

element side.

$$J = h \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.11)$$

The only difference between the master element used in the normal finite element method and an element used in this method will be a constant scale factor.

## 2.4 PCG Iterative Solver

The global stiffness matrix is not banded due to the periodic nature of the fictitious domain. This makes LU decomposition or Gaussian elimination a poor choice for solving the matrix equations. However, the stiffness matrix is very sparse and clever methods exist for an efficient solution. Also, the domain's periodicity presents some advantageous properties that will be discussed in this section.

One of the goals of this project was to solve large degree of freedom systems with an iterative scheme relatively insensitive to size. This section describes a PCG solver and how it was used to accomplish this task. Several preconditioning schemes were formulated and tested. Finally, this section contains some examples that explore iteration sensitivity to problem size.

### **Need For An Iterative Solver**

Owing to the large problem size, an iterative solver was implemented that would allow for efficient memory use. First, this method would need to solve a system of equations without storing a matrix in its entirety. Second, the solver would

need to store as few vectors as possible such that computer memory limits wouldn't be challenged. Finally, the solver would need to converge upon the solution rapidly and within a number of iterations relatively insensitive to problem size. The PCG method achieved these goals.

The use of the PCG algorithm involves finding conjugate direction vectors to solve equation 2.7. This formulation is more advantageous than a Gaussian elimination approach that requires storing an entire matrix and performing operations on that matrix before solving a system of equations. Equation 2.12 shows how the conjugate gradient method forms the solution vector from the linear combination of scalars and conjugate directions.

$$x^* = \alpha_0 p_0 + \dots + \alpha_{n-1} p_{n-1} \quad (2.12)$$

where,

$x^*$  = solution vector

$\alpha_i$  = scalar

$p_i$  = conjugate direction vector

Equation 2.12 represents one step (step 4) in the complete PCG algorithm, which will be presented in section 2.4.2. It allows the solution to be produced by only using less memory-intensive dot products.

### 2.4.1 Matrix Preconditioning

A preconditioner is used to increase convergence rates of the conjugate gradient method. The preconditioner changes the matrix eigenvalues and makes the system easier to solve. This is called ‘conditioning’ the stiffness matrix because modifying the eigenvalues changes the matrix’s condition number.

#### **The homogeneous stiffness matrix**

When the properties in  $\Omega$  are homogeneous, the resulting homogeneous stiffness matrix has some interesting properties that can be exploited to construct an effective preconditioner. A stiffness matrix arising from a discretization of  $\Omega$  consists of four sub-matrices:

$$K_h = \begin{bmatrix} K_h^{11} & K_h^{12} \\ K_h^{21} & K_h^{22} \end{bmatrix} \quad \text{Dimension } (n \text{ by } n) \quad (2.13)$$

where the notation  $[ ]_h$  denotes that the material properties are homogeneous, and

$$K_h^{12} = K_h^{21} \quad (2.14)$$

Each sub-matrix is a block circulant matrix that contains  $N \times N$  circulant matrices. Here  $N$  is the number of elements per side of  $\Omega$ .

The circulant property of matrices means that each row of the matrix can be created by cyclically shifting each element from the row above by one place to the right. A block circulant matrix is obtained by moving each block sub-matrix from the above row of blocks by one to the right. A symmetric circulant matrix is shown in (2.15),

$$A = \begin{bmatrix} a & b & c & d & c & b \\ b & a & b & c & d & c \\ c & b & a & b & c & d \\ d & c & b & a & b & c \\ c & d & c & b & a & b \\ b & c & d & c & b & a \end{bmatrix} \quad (2.15)$$

The effect is to be able to have information about the entire matrix by only storing one row of that matrix. We shall use  $K_h$  to construct a preconditioner. However,  $K_h$  is a positive semi-definite matrix. Thus, it is singular and it has two zero eigenvalues. Because the preconditioner must be inverted, the homogeneous stiffness matrix cannot be used without some modification.

## 2.4.2 PCG in PCG Method

Below is the definition of the standard PCG algorithm to solve the system of equations,

$$P_1^{-1}Ku = P_1^{-1}F \quad (2.16)$$

where  $P_1$  is the preconditioner matrix and  $K(n$  by  $n)$  is symmetric and positive definite.

**Algorithm: PCG1**

Given an initial guess  $u_0$ ,

1.  $r_o = F - Ku_o$

$$z_o = P_1^{-1}r_o$$

$$p_0 = z_0$$

2. For  $j = 0, 1, \dots, n$  while  $\|r_j\| > a$  given tolerance

3.  $\alpha_j = (r_j \cdot z_j) / (Kp_j \cdot p_j)$

4.  $u_{j+1} = u_j + \alpha_j p_j$

5.  $r_{j+1} = r_j - \alpha_j Kp_j$

6.  $z_{j+1} = P_1^{-1}r_{j+1}$

7.  $\beta_j = (r_{j+1}^T \cdot z_{j+1}) / (r_j^T \cdot z_j)$

8.  $p_{j+1} = z_{j+1} + \beta_j p_j$

9. end

In the PCG method, the preconditioner,  $P$ , must be inverted in step 6. This presents a problem. A good preconditioner matrix can be difficult to invert, yet the inversion process must be fast enough that the solution time is not compromised.

The solution proposed here is to use a second PCG algorithm to invert  $P_1$  in step 6. This allows for the best possible preconditioner choice to be used for the first (outer) PCG algorithm. Figure 2.4 shows the main functions of the two PCG solvers. The inner algorithm is called PCG2 and is used to invert  $P_1$ .

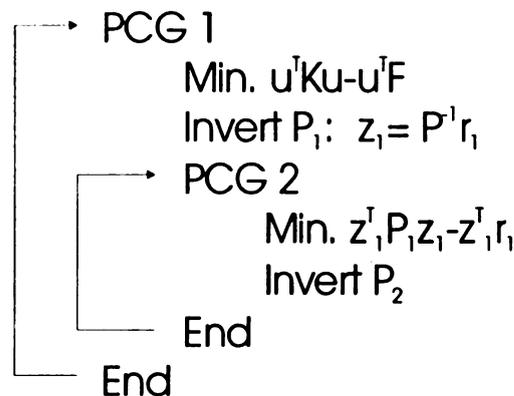


Figure 2.4: PCG in PCG algorithm

As shown in figure 2.4, each iteration of PCG1 may contain many PCG2 iterations. However, using PCG2 allows for a good choice of preconditioner  $P_1$ . It will be shown that devoting time to ensure that a good preconditioner is used pays off. The preconditioner for the second (inner) PCG also needs to be efficient, but more importantly it needs to be very easy to invert. The performance of the PCG algorithm is discussed later in this chapter.

### **First Preconditioner**

The preconditioner used for the first (outer) PCG is:

$$P_1 = [M] + ([I] - [M]) \times [K_h] \times ([I] - [M]) \quad (2.17)$$

where,

$[K_h]$  = the homogeneous stiffness matrix of equation 2.13 with boundary conditions imposed by the row and column elimination method as mentioned earlier.

$[I]$  = the identity matrix.

$[M]$  = a diagonal matrix with  $M(i,i) = 1$  for the  $i^{\text{th}}$  constrained degree of freedom and zero otherwise.

Note: the inclusion of boundary conditions makes  $P_1$  non-singular even though  $K_h$  is singular.

### **Second Preconditioner**

The preconditioner for PCG2 is,

$$P_2 = (K_h + Q) \quad (2.18)$$

where,

$$Q = \begin{bmatrix} c[1] & [0] \\ [0] & c[1] \end{bmatrix} \quad (2.19)$$

$$c \approx \left(\frac{1}{N^2}\right) \quad (2.20)$$

The strategy to invert the second preconditioner is to make the homogeneous stiffness matrix positive definite by adding a matrix  $Q$ . The matrix  $Q$  is of size  $n$  by  $n$  and has only 2 non-zero eigenvalues whose corresponding eigenvectors are the rigid body modes of  $K_h$ . As stated earlier, the homogeneous matrix  $K_h$  has some memory saving properties (it can be stored in only  $n$  memory locations). In addition, some clever techniques can be employed to invert the matrix. For one, the eigenvalues of  $K_h$  can be computed by using the Fast Fourier Transform (FFT). This makes inversion possible by performing the FFT on each individual block twice to generate a tri-diagonal matrix of eigenvalues. Then,  $Q$  from equation 2.19, is added simply by adding  $C$  to the zero eigenvalue of  $K_h$ . Then the matrix is inverted. Finally, the inverse FFT is taken and the blocks of the inverted matrix are reconstructed.

## 2.5 Performance Of The Two Level PCG Algorithm

This section takes a look at the performance of the iterative solver and the fictitious domain methodology. Also, the benefits of using the PCG inside PCG method are shown. Additionally, the advantages and disadvantages of different preconditioners are studied in the development of the PCG in PCG method. Finally, the effects of increasing problem size are considered.

### 2.5.1 Sensitivity To Problem Size

The insensitivity of the PCG solver to problem size is illustrated by solving a plane stress elasticity problem. Different total mesh and resolution combinations were varied and compared.

The fractal shaped structure shown in Figure 2.5 was used to test convergence properties. The design domain material values where  $\nu = 0.3$  and  $E_1 = 1.0$ . The fictitious domain was made of weak material,  $E_2 = 0.0$ , and the boundary conditions and loading shown in Figure 2.5 were applied.

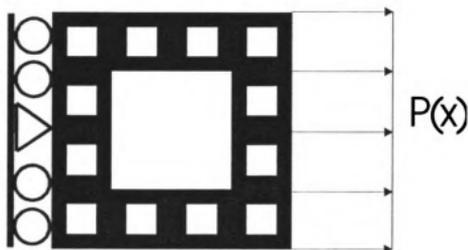


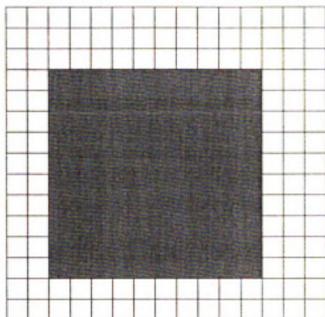
Figure 2.5: Design domain  $\omega$  and boundary conditions for performance trials

## 2.5.2 Affect Of Changing Domain Resolution

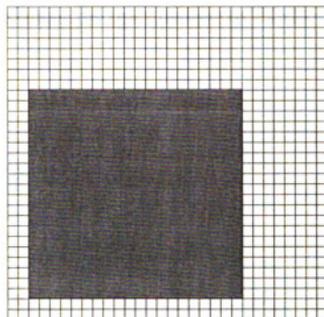
In this study three domains of three different sizes are tested for convergence. Figure 2.6 illustrates the different meshes with differing design and total domain sizes increasing. Table 2.1 shows the size of the design and total domain doubling for D1, D2, and D3. The design domain geometry is the same fractal shown in figure 2.5. Also, it is evident that the proportion of design size / fictitious domain size remains constant.

Design	$\omega$ Discretization	$\omega$ DOF	$\Omega$ Discretization	Total DOF
1	54 x 54	5832	64 x 64	8192
2	108 x 108	23328	128 x 128	32768
3	216 x 216	93312	256 x 256	131072

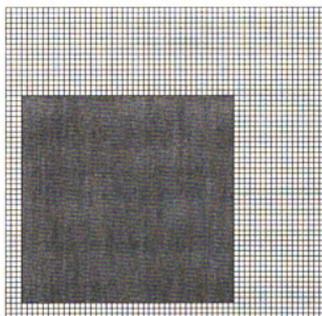
Table 2.1: Discretization levels for analysis of PCG performance



(a)



(b)



(c)

**Figure 2.6: Adjustment of design domain size and resolution**

Figure 2.7 illustrates that the total number of degrees of freedom driven by the total domain size,  $\Omega$ , has little or no effect on the number of PCG1 iterations.

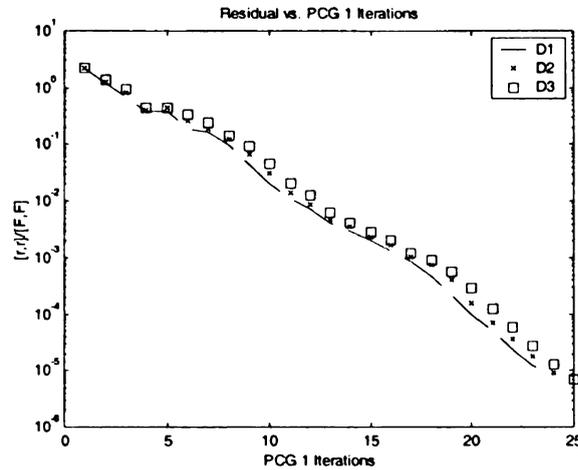


Figure 2.7: PCG1 Iterations

What then, is the cost of adding additional degrees of freedom? Additional time is spent inverting  $P_1$ , the preconditioner of PCG1. The price to pay for a good preconditioner is the time spent inverting it. Figure 2.8 shows the growth in iterations of the second PCG loop, PCG2, as the size of the design domain is increased. These results are from the trial domains described in table 2.2 and the test geometry shown in figure 2.5.

Resolution Level	$\omega$ Discretization	$\omega$ DOF	$\Omega$ Discretization	Total DOF
5	32 x 32	2048	512 x 512	524288
6	64 x 64	8192	512 x 512	524288
7	128 x 128	32768	512 x 512	524288
8	256 x 256	131072	512 x 512	524288

Table 2.2: Domain characteristics to test PCG2

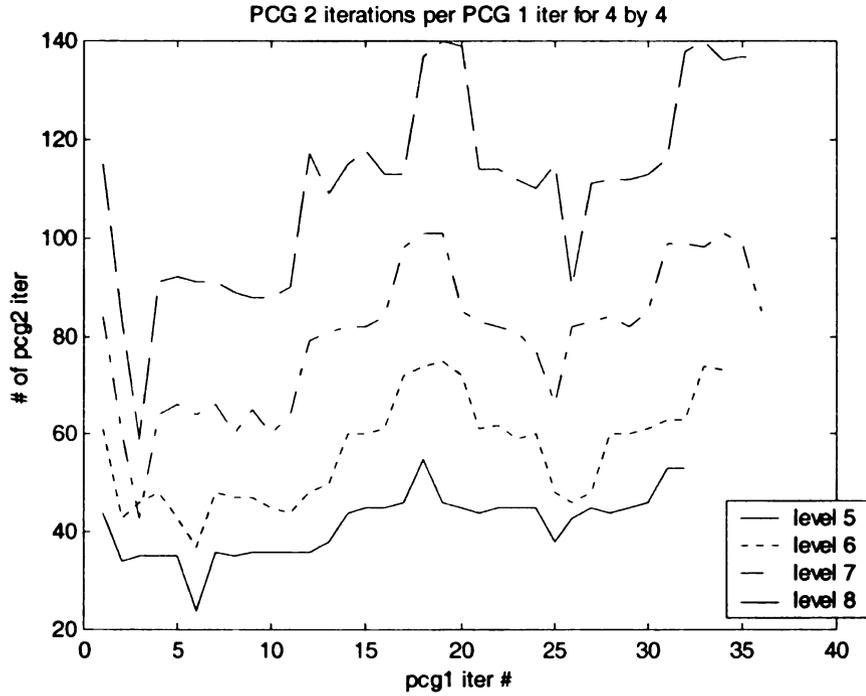


Figure 2.8: PCG2 iterations

**Remark 1:** It was observed that the size of the fictitious domain needed to be greater than approximately 10% of the design domain. If the design domain took up more than 90% of the total domain, the number of iterations rose considerably and in some circumstances the problem could not be solved.

**Remark 2:** Solution time varied as a function of several factors, the most noticeable being problem difficulty. As the problem became more

complicated, i.e. more boundary conditions were imposed or the geometry complexity increased, the solution time increased.

### 2.5.3 PCG in PCG Development

Considering the computations associated with PCG2, what real benefit is gained from using a preconditioner that is difficult to invert? Can we do better? Different preconditioners were studied in order to find and implement a more efficient solution strategy. Below is a brief discussion of the preconditioners that lead to the development of the PCG in PCG scheme.

The design domain in each test consisted of the well-known elasticity problem of a stressed plate with a center hole shown in figure 2.12. The model details are listed in Table 2.3 below.

$\omega$ Discretization	$\omega$ DOF	$\Omega$ Discretization	Total DOF
28 x 48	2688	64 x 64	8192

Table 2.3

Two single PCG schemes were initially considered. The first preconditioner tried was the matrix of the diagonal of  $[K_x]$ . The performance shown in Figure 2.9 is no better than without a preconditioner.

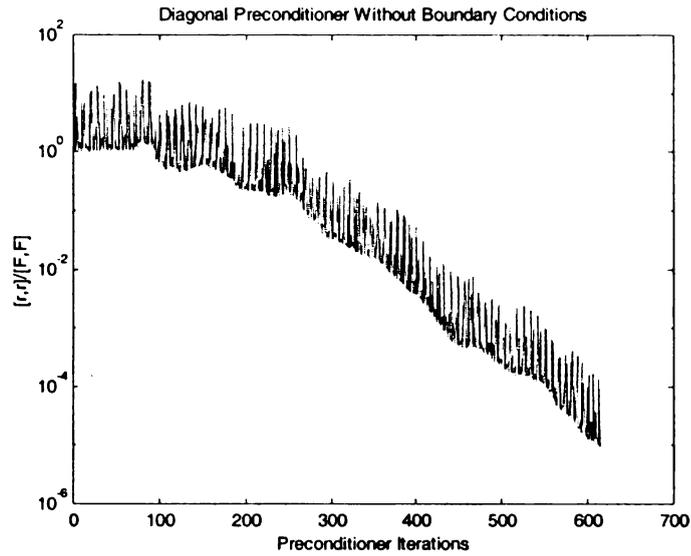


Figure 2.9: PCG Iterations for test preconditioner

In the second attempt, the same diagonal preconditioner was used with boundary conditions applied. A large value was added to  $P_1(i,i)$ , where  $i$  represents a constrained degree of freedom. A decrease in half the number of iterations resulted. Figure 2.10 shows this drastic reduction in iterations.

Finally, the PCG in PCG method was created in order to invert the most desirable of the preconditioners (shown in equation 2.17). A second PCG was introduced to invert the preconditioner. The result was a reduction in the number of iterations by ten-fold as seen in Figure 2.11.

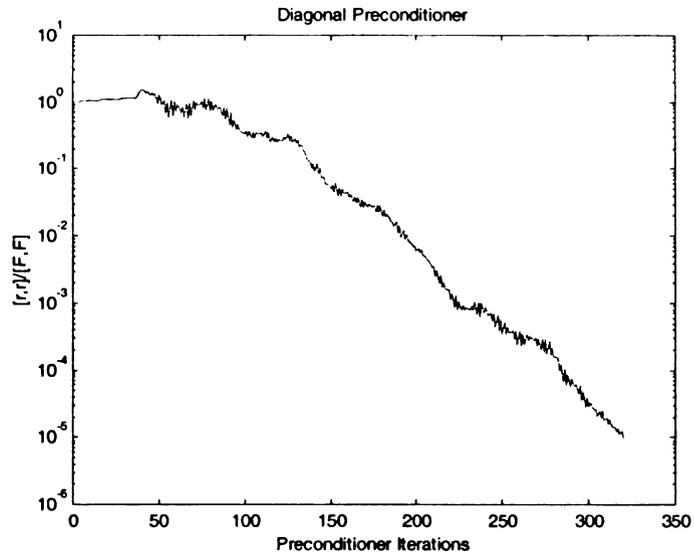


Figure 2.10: PCG Iterations for test preconditioner

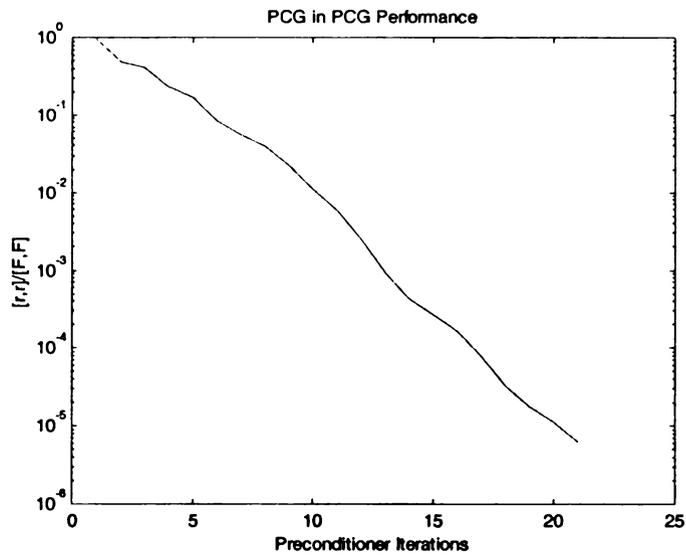


Figure 2.11: PCG1 Iterations for PCG in PCG method

## 2.6 Accuracy

This section explores the accuracy of the PCG in PCG method. Three example problems are solved and the results are compared with the commercial finite element package ANSYS 5.6. The closed form solution of one problem is presented in appendix A.

### 2.6.1 Stressed Plate With Hole

The well-known plane stress elasticity problem of a stressed flat plate with a center hole is examined for accuracy. Figure 2.12 shows the loading and boundary conditions. Opposing traction forces are applied at the left and right sides of the plate.

The problem size was greatly reduced using symmetry. Figure 2.13 shows the problem that was solved by applying symmetry along the vertical and horizontal axis of the hole. The boundary conditions now include two sets of rollers. The first set restricts movement in the x direction only while the other restricts motion in the y direction only.

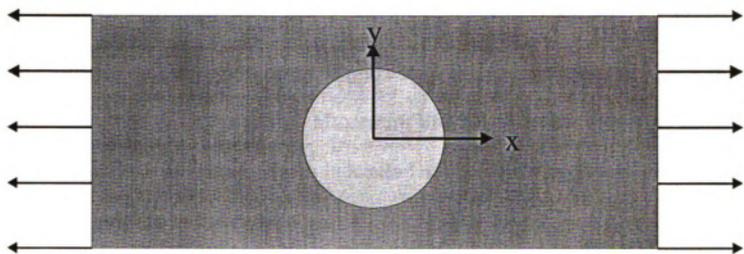


Figure 2.12: Stressed plate with hole

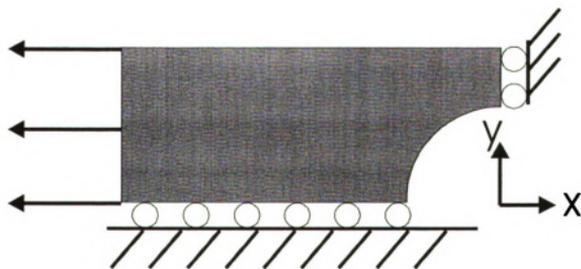


Figure 2.13: Problem domain with symmetry

Using a domain size of 64 by 64, the design domain was meshed as seen in figure 2.14. The design domain is of dimension 48 by 28 in the x and y directions respectively. The hole has a dimension of 12 units radius. The results presented are for an isotropic material with  $E = 2e5$  units and  $\nu = 0.39$ . The load is a constant traction of 10,000 units. Figure 2.14 shows the resulting Von Mises stress distribution in the plate. Maximum Von Mises stress occurs at the bottom of the hole. The minimum stress is located at the hole's horizontal axis about one of its lengths to the right.

Next, a standard finite element implementation (Ansys 5.6) was used to verify the Von Mises stress results. The stresses are similar in magnitude and in distribution. The contour bands in figure 2.14 have the same scale as figure 2.15 to make comparison easier.

The results were also verified with the analytical result that was first calculated by G. Kirsch in 1898. These resulting calculations are provided in appendix section A.3.

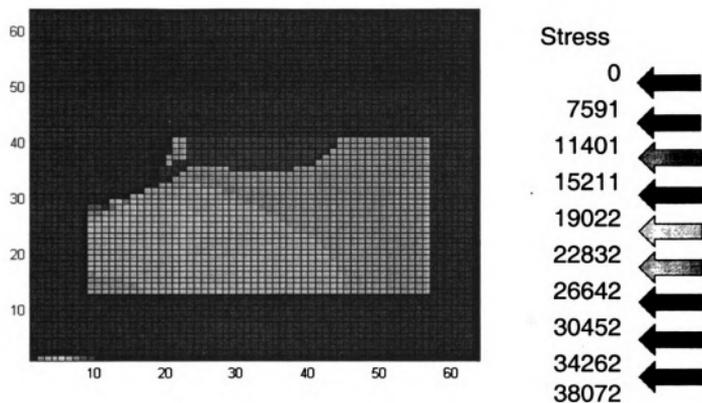


Figure 2.14: Stress contour plot of stressed plate with hole

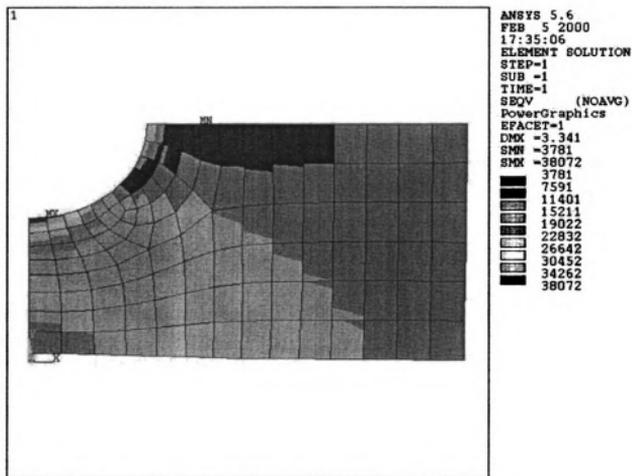


Figure 2.15: Stress contour plot generated by ANSYS 5.6

The next two examples use the following data:

Young's modulus: 100.0

Poisson's ratio: 0.30

Fictitious domain mesh size: 128 x 128 elements

## 2.6.2 'El' Bracket Example

The 'el' bracket geometry and loading is pictured in figure 2.16. The specimen is cantilevered from a wall with a fixed end boundary condition. A pressure of 10 units is applied to the free end as illustrated below.

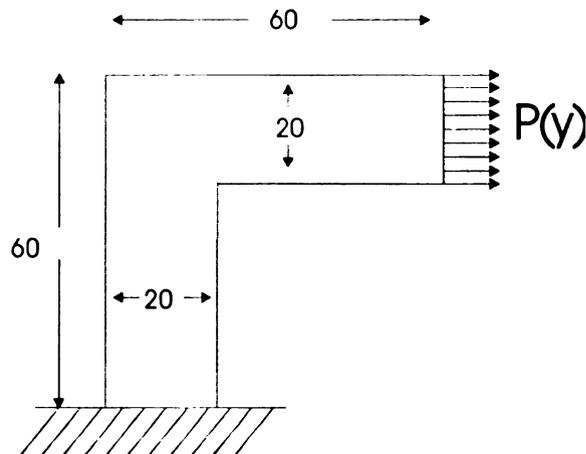


Figure 2.16: Loading and geometry of 'el' bracket

The Von Mises stress contour plot of figure 2.17 shows a larger stress gradient near the fixed end. This solution is similar to the solution generated by ANSYS 5.6 in figure 2.18. The contour bands of each graph show a similar stress pattern. Note that the stress calculated by ANSYS is averaged over each element.

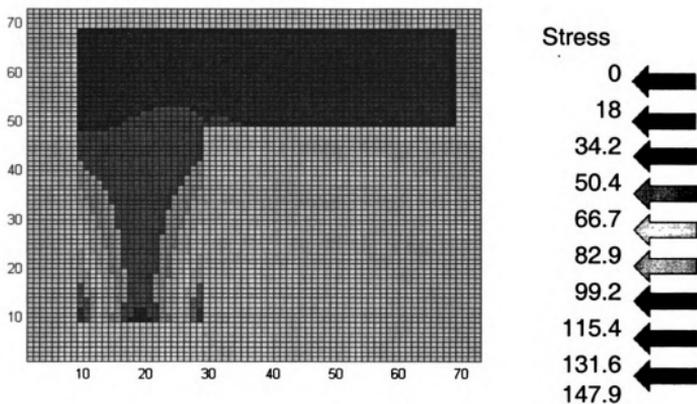


Figure 2.17: Stress contour plot of Bracket

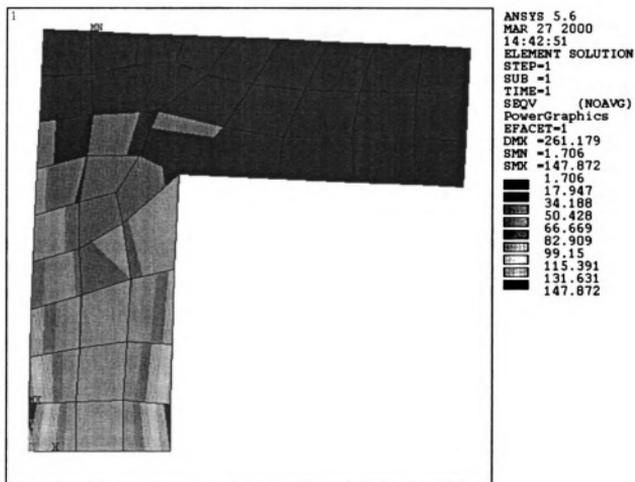


Figure 2.18: ANSYS 5.6 analysis of 'el' bracket

### 2.6.3 Annular Circle Example

The annular circle shown in figure 2.19 is a disk with a hole in the center.

The outer disk radius measures 49 units and the inner hole radius measures 24 units. A pressure of 100 units is applied to a small region on the right side of the outer circle. At the left side, the circle is cantilevered to a wall with a fixed end condition.

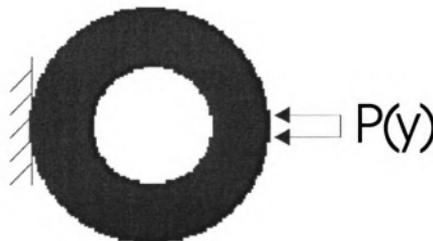


Figure 2.19: Annular circle with boundary conditions

Figure 2.20 shows the stress distribution in the annular circle. The stress is largest near the wall and the load. Regions of high stress also exist near the edge of the hole in the center of the disk.

Comparison to the ANSYS solution yields similar results. The stress distribution patterns are alike as are the relative magnitudes of stress.

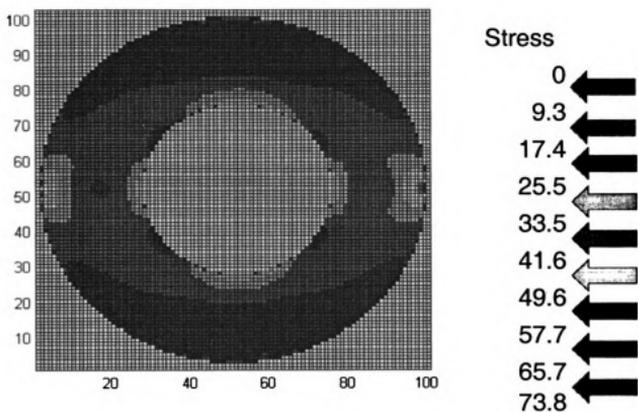


Figure 2.20: Stress contour plot of disk with hole

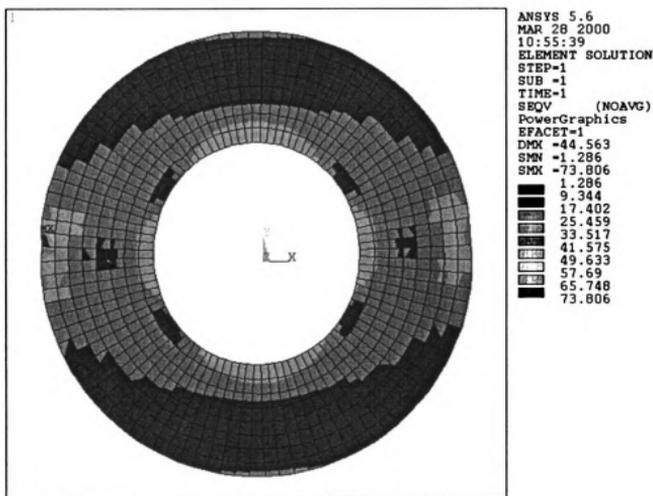


Figure 2.21: Stress contour plot from ANSYS 5.6

## 2.7 Summary

The proposed PCG in PCG solver accurately and efficiently meets the required project specifications. The examples show that the convergence of the PCG solver is independent of the problem size. The fictitious domain approach successfully approximates the solution to linear elasticity problems.

# Chapter 3

## Computation of Constitutive Properties

This section will discuss how effective properties are calculated. Also, a detailed explanation of homogenization techniques is offered. Then, accuracy verification of the computational approach is presented.

### 3.1 Background

A material microstructure can be thought of as a material composed of two distinct constituents that are mixed at a small scale in a pattern that repeats

periodically throughout the domain. When a material with a microstructure is viewed from a macroscopic vantage point, it appears to be homogeneous. But, when one zooms in for closer inspection, one can see the fine geometrical pattern of the microstructure. The smallest repetitive unit in a microstructure is called a base cell. It is assumed that the refined pattern can be represented by this base pattern that is repeated periodically throughout the material. Because of the cell's periodicity, the analysis of only one base cell is sufficient to calculate the effective properties for the entire material.

In this project, the micro-geometry is constructed by puncturing holes into a homogeneous substance. The result is a complex pattern. A different material, as in the case of a composite material, could occupy the refined holes. In this project it will be assumed that one could also leave the hole to be void of material.

The periodic homogenization problem is one of determining the effective material property tensor of structures that have a periodic microstructure. Considering figure 3.1, the effective properties of the macroscopic material are a function of:

1. The properties of the constituent materials labeled, 1 and 2, with elastic tensors  $E_1$  and  $E_2$ , respectively.
2. The special distribution of the two materials in the cell.
3. The orientation angle of the base cell with respect to a global coordinate system  $(x_1, x_2)$ .

The microstructure shown below characterizes the mixture of the two materials, 1 and 2. The base cell geometry in the  $(y_1, y_2)$  domain is repeated periodically in the macroscopic domain  $\Omega$ .

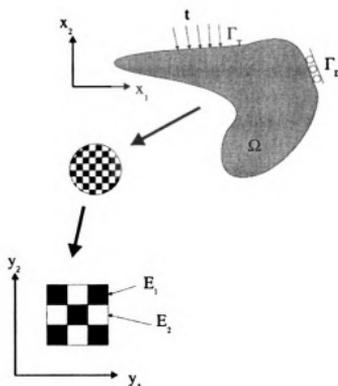


Figure 3.1: Periodic microstructure in a homogenized material

## 3.2 Homogenization Method

The homogenization method allows one to solve elasticity problems in which materials with a microstructure are involved. This method allows for a solution without having to individually discretize each small cell and solve with finite elements. Instead, using homogenization, the behavior of the

microstructure can be approximated by a ‘post processing’ of the macroscopic stress analysis (Guedes [20]).

The methods of periodic homogenization were pioneered in 1978 by Bensoussan, Lions, and Papanicolaou [6]. For the sake of completeness, the following derivation of the homogenization techniques is presented. This derivation follows the methods presented by Bendsoe and Kikuchi [4] and Guedes [20].

Let the domain occupied by the structure be  $\Omega$ , and let the body force  $\mathbf{f}$  be applied to  $\Omega$ . Let the traction  $\mathbf{t}$  be applied along a part of the boundary  $\Gamma_t$  as shown in figure 3.1. Let  $\Gamma_D$  be the part of the boundary where prescribed displacements are specified.

First, assume that the tensor  $E_{ijkl}^\epsilon$  of material constants satisfies the symmetry condition,

$$E_{ijkl}^\epsilon = E_{jikl}^\epsilon = E_{ijlk}^\epsilon = E_{klij}^\epsilon \quad (3.1)$$

Also consider the stress-strain relations,

$$\begin{aligned} \sigma_{ij}^\epsilon &= E_{ijkl}^\epsilon \epsilon_{kl}^\epsilon \\ \epsilon_{kl}^\epsilon &= \frac{1}{2} \left( \frac{\partial u_k^\epsilon}{\partial x_l} + \frac{\partial u_l^\epsilon}{\partial x_k} \right) \end{aligned} \quad (3.2)$$

Next, suppose that a periodic microstructure is in the neighborhood of an arbitrary point  $\mathbf{x}$  in a given linearly elastic structure (see figure 3.1).

Because the body forces and tractions vary within the large scale as well as small scale they are functions of  $x$  and  $x/\varepsilon$ . Here  $\varepsilon$  is a magnification factor that is a measure of the microscopic/ macroscopic dimension ratio (the level of magnification is large enough to scale the microstructure to unit length).

Then, the elasticity tensor is  $E_{ijkl}^\varepsilon(x)$ , where

$$E_{ijkl}^\varepsilon(x) = E_{ijkl}(x, y), \quad y = x/\varepsilon, \text{ for } i, j, k, l = 1, 2, \quad (3.3)$$

Also, for fixed  $x$ ,  $E_{ijkl}(x, y)$  is  $Y$ -periodic where,  $Y = (y_{1R}, y_{1L}) \times (y_{2R}, y_{2L})$ .

With these assumptions, equilibrium of the structure can be characterized by the minimum potential energy principle. Thus, the displacements  $u^\varepsilon$  for equilibrium are the solution to the minimization problem

$$\min_{v^\varepsilon \in U} \Pi^\varepsilon(v^\varepsilon), \quad (3.4)$$

where  $\Pi^\varepsilon$  is the total potential energy defined by

$$\Pi^\varepsilon(v^\varepsilon) = \frac{1}{2} a^\varepsilon(v^\varepsilon, v^\varepsilon) - L(v^\varepsilon), \quad (3.5)$$

$$a^\varepsilon(u, v) = \int_{\Omega} E_{ijkl}^\varepsilon \varepsilon_{kl}(u) \varepsilon_{ij}(v) dx, \quad (3.6)$$

$$L(v) = \int_{\Omega} f^* v \, dx + \int_{\Gamma_i} t^* v \, ds, \quad (3.7)$$

and  $U$  is the admissible linear manifold such that  $v$  is  $Y$ -periodic on the domain  $\Omega$ ,

$$U = \{v : v_i \in H^1(\Omega), v_i = g_i \text{ on } \Gamma_D\}, \quad (3.8)$$

where  $g$  is the specified displacement along the boundary  $\Gamma_D$ , and  $\varepsilon(v)$  is the linearized strain tensor

The solution to problem (3.4),  $u^\varepsilon$ , depends on the parameter  $\varepsilon$  characterizing the microstructure. This dependence of  $u^\varepsilon$  on the large scale and small scale means that an asymptotic expansion can be used with respect to  $\varepsilon$ ,

$$u^\varepsilon(x) = u_0(x) + \varepsilon u_1(x, y) + \dots, \quad y = x / \varepsilon \quad (3.9)$$

Now, assume that an arbitrary admissible displacement  $v^\varepsilon$  is expanded as

$$v^\varepsilon(x) = v_0(x) + \varepsilon v_1(x, y), \quad y = x / \varepsilon \quad (3.10)$$

where  $v_0 \in U$  and  $v_1(x, y)$  is defined in  $\Omega \times Y$ ,  $v_1(x, \cdot)$  is  $Y$ -periodic, and  $v_1 = 0$  on  $\Gamma_D = Y$ . Noting that for any  $Y$ -periodic function  $\varphi$ ,

$$\frac{\partial}{\partial x_i} (\varphi(x, y)|_{y=x/\varepsilon}) = \frac{\partial \varphi}{\partial x_i} + \frac{1}{\varepsilon} \frac{\partial \varphi}{\partial y_i} \quad (3.11)$$

and

$$\lim_{\varepsilon \rightarrow 0} \int_{\Omega} \Phi(x, x/\varepsilon) dx = \frac{1}{|Y|} \int_{\Omega} \int_Y \Phi(x, y) dy dx, \quad (3.12)$$

where  $|Y|$  is the area of one cell,

we see that,

$$\lim_{\varepsilon \rightarrow 0} \Pi^\varepsilon(v^\varepsilon) = \Pi(v_0, v_1), \quad (3.13)$$

where,

$$\Pi(v_0, v_1) = \frac{1}{|Y|} \frac{1}{2} \int_{\Omega} \int_Y E_{ij\mu} (x, y) \left( \frac{\partial v_{0k}}{\partial x_i} + \frac{\partial v_{1k}}{\partial y_l} \right) \left( \frac{\partial v_{0i}}{\partial x_j} + \frac{\partial v_{1i}}{\partial y_j} \right) dy dx - \int_{\Omega} f * v_0 dx - \int_{\Gamma_r} t * v_0 ds \quad (3.14)$$

If the pair  $\{u_0, u_1\}$  is the minimizer of the functional  $\Pi$ , it satisfies the following two equations:

$$\begin{aligned} & \frac{1}{|Y|} \int_{\Omega} \int_Y E_{ijkl}(x, y) \left( \frac{\partial u_{0k}}{\partial x_l} + \frac{\partial u_{1k}}{\partial y_l} \right) \frac{\partial v_{0i}}{\partial x_j} dy dx \\ & = \int_{\Omega} f * v_0 dx + \int_{\Gamma_r} t * v_0 ds \quad \text{for every } v_0, \end{aligned} \quad (3.15)$$

and

$$\frac{1}{|Y|} \int_{\Omega} \int_Y E_{ijkl}(x, y) \left( \frac{\partial u_{0k}}{\partial x_l} + \frac{\partial u_{1k}}{\partial y_l} \right) \frac{\partial v_{1i}}{\partial x_j} dy dx = 0 \quad \text{for every } v_1 \quad (3.16)$$

If  $u_1$  is assumed to be decomposed into

$$u_{1k}(x, y) = -\chi_k^{pq}(y) \frac{\partial u_{0p}}{\partial x_q}(x), \quad (3.17)$$

and if  $\chi^{pq}$  satisfies

$$\int_Y \left( E_{ijkl} - E_{ijpq} \frac{\partial \chi_p^k}{\partial y_q} \right) \frac{\partial v_{1i}}{\partial y_j} dy = 0 \quad \text{for } k, l = 1 \text{ and } 2, \quad (3.18)$$

the second equation (3.16) is automatically satisfied. Substitution of equation 3.17 into the first equation (3.15) yields the homogenized equation

$$\int_{\Omega} E_{ijkl}^H(x) \frac{\partial u_{0k}}{\partial x_i} \frac{\partial v_{0i}}{\partial x_j} dx = \int_{\Omega} f^* v_0 dx + \int_{\Gamma_T} t^* v_0 ds \quad \text{for every } v_0, \quad (3.19)$$

where,

$$E_{ijkl}^H(x) = \frac{1}{|Y|} \int_Y (E_{ijkl}(x, y) - E_{ijpq}(x, y) \frac{\partial \chi_p^k}{\partial y_q}) dy \quad (3.20)$$

Next, define

$$a_H(u, v) = \int_{\Omega} E_{ijkl}^H(x) \frac{\partial u_k}{\partial x_i} \frac{\partial v_i}{\partial x_j} dx, \quad (3.21)$$

$$a_Y(\chi^k, v) = \int_Y E_{ijpq}(x, y) \frac{\partial \chi_p^k}{\partial y_q} \frac{\partial v_i}{\partial y_j} dy, \quad (3.22)$$

and

$$L_Y^k(v) = \int_Y E_{ijkl} \frac{\partial v_i}{\partial y_j} dy \quad (3.23)$$

Thus, the following problem at the microscopic level and associated homogenized problem at the macroscopic level constitute necessary (and sufficient) conditions that are obtained by passing to the limit  $\varepsilon \rightarrow 0$  in the minimum principle.

**Problem At Small Scale In Cell Y:**

$$\chi^k \in U_Y : a_Y(\chi^k, v) = L_Y^k(v) \quad \text{for every } v \in U_Y, \quad (3.24)$$

where  $U_Y$  is the admissible space defined in the cell Y:

$$U_Y = \{v : v_i \text{ is Y- Periodic}\} \quad (3.25)$$

**Problem At The Large Scale, The Homogenized Problem On  $\Omega$ :**

$$u \in U : a_H(u, v) = L(v) \quad \text{for every } v \in U_0, \quad (3.26)$$

where  $U_0$  is the homogeneous case of  $U$ , i.e.,  $g = 0$ .

The microscopic problem can be solved using the finite element method. In doing so, the cell domain Y is discretized using finite elements and the admissible space  $U_Y$  is approximated by  $U_{Yh}$ . Thus, the finite element

approximation on  $\chi_h^{kl} \in U_{\gamma_h}$  of the  $\chi^{kl}$  is obtained as the solution of the discrete problem

$$\chi_h^{kl} \in U_{\gamma_h} : a_{\gamma}(\chi_h^{kl}, v_h) = L_{\gamma}^{kl}(v_h) \quad \text{for every } v_h \in U_{\gamma_h}, \quad (3.27)$$

Using the above approximation, the homogenized elasticity tensor is defined by

$${}_h E_{ijkl}^H(x) = \frac{1}{|Y|} \int_Y (E_{ijkl}(x, y) - E_{ijpq}(x, y) \frac{\partial \chi_{hp}^{kl}}{\partial y_q}) dy \quad (3.28)$$

Following Sigmund [17], we can define

$$P^{ij} = y_j \delta_{ik} e_k \quad (3.29)$$

where,

$\delta =$  kronecker delta

$e_k =$  Cartesian base vector

And note that equation 3.28 can be stated as,

$$\int_Y E_{ijpq} \left( \frac{\partial \chi_p^{(kl)}}{\partial y_q} - \frac{\partial P_p}{\partial y_q} \right) \frac{\partial v_i}{\partial y_j} dY = 0 \quad (3.30)$$

where,

$$\frac{\partial \mathcal{P}_p^{(kl)}}{\partial y_q} = \varepsilon_{pq}^{0(kl)} = \text{three independent cases of initial strain}$$

$$\frac{\partial \mathcal{X}_p^{(kl)}}{\partial y_q} = \varepsilon_{pq}^{*(kl)} = \text{solution to the problem}$$

Given the substitutions above, equation 3.30 can be written as simply,

$$E_{ijkl}^H = \frac{1}{|Y|} \int_Y E_{ijpq} (\varepsilon_{pq}^{0(kl)} - \varepsilon_{pq}^{*(kl)}) dY \quad (3.31)$$

Solving 3.31 involves creating a finite element model and solving for  $\varepsilon^*$  given three independent cases of initial strain,  $\varepsilon^0$ . The initial strains are shown in equation 3.32 below. The initial strains  $\varepsilon_{pq}^{0(11)}$  and  $\varepsilon_{pq}^{0(22)}$  are normal strains.

The third initial strain,  $\varepsilon_{pq}^{0(12)}$ , is a shear strain, i.e.,

$$\varepsilon_{pq}^{0(kl)} =$$

$\varepsilon_{11}^{0(11)} = 1.0$	$\varepsilon_{11}^{0(22)} = 0.0$	$\varepsilon_{11}^{0(12)} = 0.0$
$\varepsilon_{22}^{0(11)} = 0.0$	$\varepsilon_{22}^{0(22)} = 1.0$	$\varepsilon_{22}^{0(12)} = 0.0$
$\varepsilon_{12}^{0(11)} = 0.0$	$\varepsilon_{12}^{0(22)} = 0.0$	$\varepsilon_{12}^{0(12)} = 1.0$

(3.32)

The initial strains are applied to the finite element model as equivalent nodal loads. Finally, equation 3.33 can be written in terms of mutual energies. The expansion of this equation allows for direct calculation of the six unique entries of the constitutive properties tensor.

$$E_{ijkl}^H = \frac{1}{|Y|} \int_Y (E_{pqrs} (\varepsilon_{pq}^{0(kl)} - \varepsilon_{pq}^{*(kl)}) (\varepsilon_{rs}^{0(ij)} - \varepsilon_{rs}^{*(ij)})) \quad (3.33)$$

The constitutive tensor can be written in the following matrix form,

$$[E] = \begin{bmatrix} E_{1111} & E_{1122} & E_{1112} \\ E_{1122} & E_{2222} & E_{2212} \\ E_{1112} & E_{2212} & E_{1212} \end{bmatrix} \quad (3.34)$$

## Boundary Conditions

The boundary conditions for the homogenization problem are periodic. As such, the fictitious domain approach is well suited to enforcing these boundary conditions. Thus, filling the entire domain with a square base cell automatically enforces equation 3.35 and additional constraints aren't required. In other words, there is no weak material surrounding the design domain. The periodic boundary conditions are shown in figure 3.2 and can be stated as,

$$\begin{aligned}
 v_1|_{y_1=0} &= v_1|_{y_1=y_1^0}, v_2|_{y_1=0} = v_2|_{y_1=y_1^0} \\
 v_1|_{y_2=0} &= v_1|_{y_2=y_2^0}, v_2|_{y_2=0} = v_2|_{y_2=y_2^0}
 \end{aligned}
 \tag{3.35}$$

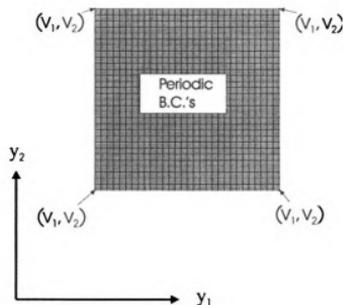


Figure 3.2: Periodic boundary conditions applied to fictitious domain.

### 3.3 Verification of Accuracy

This section verifies the accuracy of the calculations of effective base cell properties. The results are validated by a comparison to a closed form solution for layered materials.

The rank-1 material is a sandwich of strong material held together by a weak material. Figure 3.3 shows a rank-1 microstructure that repeats periodically throughout a domain. The material labeled  $E'$  is stiffer than the material labeled  $E$ . Note that this geometry would be the stiffest possible microstructure per weight for a loading parallel to the  $y_2$  axis. However, if

material  $E^-$  were void it would collapse under any loading with components in the  $y_1$  direction.

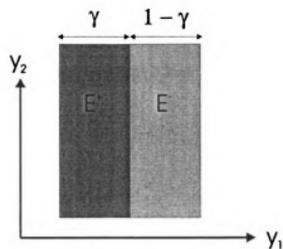


Figure 3.3: Rank-1 material used for algorithm verification

Bendsoe [3] presented an analytical solution for the effective properties of the rank-1 material shown in figure 3.3. In that solution, two material layers were isotropic, had the same Poisson's ratio  $\nu$ , the Young's moduli were  $E^-$  and  $E^+$  respectively, and the corresponding layer thicknesses  $\gamma$  and  $(1-\gamma)$ . Using the matrix notation of equation 3.34, the five non-zero stiffnesses are given by equation 3.36 below.

$$E_{1111}^H = I_1, \quad E_{2222}^H = (1-\nu^2)I_2 + \nu^2 I_1 \quad (3.36)$$

$$E_{1212}^H = \frac{1-\nu}{2} I_1, \quad E_{1122}^H = \nu I_1$$

$$I_1 = \frac{E^- E^+}{[\gamma E^- + (1-\gamma) E^+]}, \quad I_2 = \gamma E^+ (1-\gamma) E^-$$

The accuracy of the finite element implementation will be tested using a base cell with a 16 by 16 finite element mesh constructed to model figure 3.3. The material characteristics of the layers are:  $E^* = 100$ ;  $E = 20$ ,  $\nu = 0.3$ ;  $\gamma = 0.5$ . Below are the constitutive property matrices calculated analytically and computationally.

$$[E] = \begin{bmatrix} 36.63 & 10.99 & 0 \\ 10.99 & 63.30 & 0 \\ 0 & 0 & 12.82 \end{bmatrix}$$

(a) Analytical solution

$$[E] = \begin{bmatrix} 36.63 & 10.99 & 0 \\ 10.99 & 63.30 & 0 \\ 0 & 0 & 12.82 \end{bmatrix}$$

(b) Computational solution

Note that the computational results are accurate to four significant digits.

### 3.4 Summary

The homogenized properties were calculated using finite element analysis of the base cell. Boundary conditions were automatically imposed by the periodicity of the mesh. The fictitious domain approach was well suited to the calculation of effective material properties because of its square domain and periodic properties. Results were successfully verified by comparison to a known closed form solution for rank-1 materials.

# Chapter 4

## Analysis Of Structures Created By Iterated Affine Maps

### 4.1 Introduction

This section looks at the performance of a base cell with geometry created by iterated maps. The effective properties will be compared to the benchmark rank-2 material. Then, the change in cell stiffness with respect to changing fractal coefficients will be studied. Also, this section explains some important properties of fractal geometries and how they are generated using iterated affine maps. In addition, the iterative affine transformations will be explained as well as how they are applied to homogenization.

## 4.2 Background

### 4.2.1 Key Concepts

The first concept of fractal geometry is fractal dimension. Man made geometry can be described almost exclusively by Euclidian geometry. The building blocks of Euclidian geometry are smooth objects such as lines, planes, cylinders, and rectangular volumes. These objects have integer dimensions. For example, to define a point on a line requires one uniquely defined point. Hence, a line is exactly one-dimensional. To define a point on a plane, one requires two numbers to define it, usually from an orthogonal coordinate system. This concept can be extended to three-dimensions.

We now consider a more mathematical definition of dimension. The given definition considers how the size of an object changes as a linear dimension increases. For example, as the linear dimension of a line segment doubles, its length doubles by a factor of 1. As the linear dimensions of a rectangle are doubled, its area increases by a factor of 4. The relationship between the dimension  $D$ , linear scaling  $L$ , and the increase in size can be written as,

$$S=L^D$$

or,

$$D = \frac{\log(S)}{\log(L)}$$

For Euclidian shapes, the dimension given by the above definition is always an integer of 1, 2, or 3. Fractal shapes behave differently. If a planar fractal shape is linearly scaled by a factor  $L$ , then its area does not scale by an integer power of  $L$ . Using the concepts from above, one can find the fractal dimension by dividing the log of the number of self-similar pieces by the log of the magnification factor. This will result non-integer dimension.

Another important property of fractal geometry is self-similarity. A figure is termed self-similar if it can be decomposed into parts that are exact replicas of the whole. The twig on a tree is an example of self-similarity in nature. Another example would be to stand between two mirrors facing each other. One would see one's self-looking into a mirror that would again be looking at one's self in a self-similar manner.

A third important property is box self-similarity. This is simply a definition that a self-similar figure can be finitely 'boxed in' (discretized) for so long, at which point refining the mesh no longer yields differences in the figure. The box self-similarity property can be used to test for self-similarity. Obviously, if the additional mesh refinement yields the same figure then it is self-similar.

Finally, the depth of an iterated map and level of resolution of a mesh are two additional significant characteristics. The depth is defined as how many

times the mapping has been applied. The level of resolution is the size of the base cell mesh and is a power of 2 such as  $2^j$ , where  $j$  is the level of resolution of the mesh.

For this project, self-similarity is desired in the base cell images. For that reason, adequate refinement is required. Because we are meshing the image with an equally spaced grid, we are ensuring the definition of self-similarity by the 'box self-similarity' definition. However, problems arise when base cells of dimension other than  $2^j$  are generated. As mentioned in the last chapter, the base cell dimension of  $2^j$  is required so the FFT can be implemented in the PCG computations. Thus, if a fractal only produces a mesh that can be box self-similar with an odd integer number of elements, then the FFT can't be employed. This limited the selection of possible base cells to shapes that could be self-similar and meshed into an even multiple of elements.

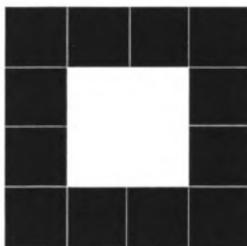


Figure 4.1: Geometry generated by parameters in table 1

## 4.2.2 Iterated Affine Mapping

In this study, a deterministic algorithm uses affine maps to create the true fractal geometry (before discretization). An affine transformation is a combination of linear transformation (given by a linear operator) and then a translation applied to an initial shape. The linear operator can be very complex (such as a scaling and rotation), or it can be a simple scaling of an object. Starting with an initial set of points, an affine transformation is mapped  $n$  times. After each succeeding mapping, the cell density decreases toward zero.

The mesh size of the fractal geometry becomes large as the number of mappings increases. As an example, the discretization size of the geometry in figure 4.1 increases at the rate  $4^n$  where  $n$  is the number of mappings. Thus, the memory efficient PCG in PCG solver presented earlier becomes a useful tool to study these geometries.

Below is the notation used to describe the affine maps:

$$\xi_i(x) = \xi_i \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} = A_i x + t_i \quad (4.1)$$

This transformation maps  $i$  quadrilaterals with corners defined by  $a, b, c,$  and  $d$ . The values  $e$  and  $f$  define the location of the quadrilateral. Table 4.1 shows the coefficients used to define the ‘sponge’ pattern of figure 4.1. This

first mapping is made up of 12 squares of size  $\frac{1}{4}$  by  $\frac{1}{4}$ , whose  $x$  and  $y$  positions are defined by  $e$  and  $f$  respectively.

$\xi$	a	b	c	d	e	F
1	0.25	0	0	0.25	0	0
2	0.25	0	0	0.25	0	0.25
3	0.25	0	0	0.25	0	0.50
4	0.25	0	0	0.25	0	0.75
5	0.25	0	0	0.25	0.25	0
6	0.25	0	0	0.25	0.25	0.75
7	0.25	0	0	0.25	0.50	0
8	0.25	0	0	0.25	0.50	0.75
9	0.25	0	0	0.25	0.75	0
10	0.25	0	0	0.25	0.75	0.25
11	0.25	0	0	0.25	0.75	0.50
12	0.25	0	0	0.25	0.75	0.75

Table 4.1: Coefficients for sponge shape

As the 4 by 4 pattern of table 4.1 is mapped, the amount of material in the cell exhibits exponential decay. Table 4.2 shows the fraction of the cell occupied by material and the number of elements required to define the domain as fractal maps are applied.

Depth	Weight	Number of Elements In $\omega$
0	1.000	1
1	0.750	4
2	0.563	16
3	0.422	64
4	0.316	256
5	0.237	1024
6	0.178	4096

Table 4.2: Weight reduction due to mapped material removal

## 4.3 Performance of Iterative Affine Maps

The goal is to test the stiffness of material created by iteratively mapped geometries. The previous chapter discussed the computation of effective properties associated with specific material arrangements. Using those techniques the base cells are now compared by strain energy density, a measure of overall stiffness. For a fixed density, the material is put side by side to a rank-2 material exhibiting optimal strain energy per unit weight.

The following assumptions are imposed:

1. The deformation is linear elastic
2. All constituents are isotropic
3. Plane stress conditions

### 4.3.1 Problem Statement

The goal here is to study the effective stiffness of materials with a fractal, periodic microstructure. The strain energy density present in a cell subjected to a fixed strain field provides a measure of the stiffness. A larger amount of strain energy density corresponds to a stiffer structure. Thus, the problem becomes to study the strain energy density for a fixed amount of material and a fixed strain field  $\epsilon(w)$ . For a fixed strain field the strain energy density is:

$$W(\boldsymbol{\varepsilon}) = \frac{1}{2} \boldsymbol{\varepsilon}^T [\mathbf{E}] \boldsymbol{\varepsilon} \quad (4.2)$$

where,

$\mathbf{E}$  = effective material property tensor

$\boldsymbol{\varepsilon}$  = fixed strain field

For an orthotropic material, the effective property matrix,  $\mathbf{E}$ , can be written as,

$$[\mathbf{E}] = \begin{bmatrix} E_{11} & E_{12} & 0 \\ E_{21} & E_{22} & 0 \\ 0 & 0 & E_{66} \end{bmatrix} \quad (4.3)$$

The strain energy density in equation 4.2 can be expanded to,

$$W = \frac{1}{2} E_{11} \varepsilon_{11}^2 + \frac{1}{2} E_{22} \varepsilon_{22}^2 + E_{12} \varepsilon_{11} \varepsilon_{22} \quad (4.4)$$

For a given effective property matrix  $[\mathbf{E}]$ , the strain energy can be expressed as a function of varying principal strains  $\varepsilon_I$  and  $\varepsilon_{II}$ . We choose  $|\varepsilon_I| > |\varepsilon_{II}|$  such that the strain ratio  $\eta = \frac{\varepsilon_{II}}{\varepsilon_I}$  is in the range  $-1 < \eta < 1$ . The strain energy

density can be re-written as,

$$W = \frac{1}{2} E_{11} \eta^2 + E_{12} \eta + \frac{1}{2} E_{22} \quad (4.5)$$

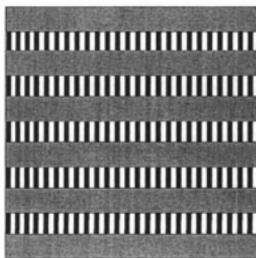
### 4.3.2 Baseline Material Distribution

An analytical calculation of effective properties exists for a material distribution exhibiting the greatest stiffness for a given amount of material when oriented along the axis of principal strain. The material distribution is called rank-2, and will serve as a benchmark with which iteratively mapped geometry can be compared.

The rank-2 microstructure is created by mixing material at two scales. At the first scale, the geometry appears layered like a rank-1 material (figure 4.2 a). At the second scale, perpendicular and between the first scale, is a material mixture that gives the impression of a rank-1 material (figure 4.2 b).



a.) First scale



b.) Second scale

Figure 4.2: Rank-2 material distribution at two levels

The rank-2 material is unique because it provides no shear stiffness ( $E_{66}=0.0$ ). Hence, if oriented correctly, it provides optimum stiffness in the directions of principle strain. On the other hand, if loads are applied that differ from the design load (which provides for optimal stiffness), the material is unstable and will perform mechanism-type motion (Sigmund [17]).

The layered pattern of the rank-2 material is characterized by the parameters  $\delta$  and  $\gamma$  as shown in figure 4.3.

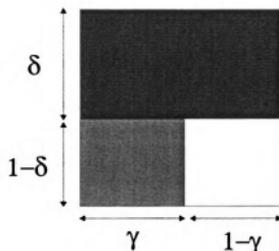


Figure 4.3: Parameters  $\gamma$  and  $\delta$  characterize the rank-2 pattern.

It can be thought of as alternating layers of stiff and rank-1 material with average densities of  $\delta$  and  $1-\delta$  respectively. The average densities of stiff and flexible layers within the rank-1 material are  $\gamma$  and  $1-\gamma$ . Then, the bulk density of the material can be written as,

$$\rho = \delta + \gamma - \gamma\delta \quad (4.6)$$

Note that as  $\gamma$  approaches 1.0, the structure becomes a rank-1 material. As  $\delta$  approaches 1.0, the structure becomes uniformly solid. Naturally, the stiffest material distribution would be for  $\delta = 1.0$ .

### 4.3.3 Strain Energy Density Formulation For Rank-2 Materials

The effective properties for a rank-2 material can be derived using the homogenization formulas presented in chapter 3 as described by Jog, Haber, and Bendsoe [11].

The rank-2 material that exhibits an optimal strain energy curve for a fixed  $\rho$  has continuously adjusted layer densities. This means for differing  $\eta$  there are different combinations of optimal  $\delta$  and  $\gamma$ . In other words, as the material axes are rotated to remain aligned with the instantaneous principal axis, a new geometric optimality exists. For differing values of  $\rho$  there are four regions or modes, each with its own optimal values of  $\delta$  and  $\gamma$ . Modes I and II are defined by values of  $\delta$  and  $\gamma$  between zero and one. Modes III and IV correspond to lower-bound and upper-bound constraints on  $\delta$  (i.e. values of 0.0 and 1.0 respectively).

As presented by Jog, Haber, and Bendsoe [11], the strain energy density for the four modes is defined as:

Mode-I

$$W = \frac{E(1 + 2\eta(1 - \rho + \nu\rho) + \eta^2)}{2(1 - \nu)(2 - \rho + \nu\rho)}$$

Mode-III

$$W = \frac{1}{2}\rho E \quad (4.7)$$

Mode-II

$$W = \frac{E(1 - 2\eta(1 - \rho - \nu\rho) + \eta^2)}{2(1 + \nu)(2 - \rho - \nu\rho)}$$

Mode-IV

$$W = \frac{E(1 + 2\nu\eta + \eta^2)}{2(1 - \nu^2)}$$

The ranges of validity for the four modes are functions of  $\rho$  and  $\eta$ ,

$$\text{mode-I: } \frac{1 + \eta}{1 - \nu} < \rho < 1$$

$$\text{mode-II: } \frac{1 - \eta}{1 + \nu} < \rho < 1 \quad (4.8)$$

$$\text{mode-III: } 0 \leq \rho \leq \frac{1 + \eta}{1 - \nu} \text{ and } 0 \leq \rho \leq \frac{1 - \eta}{1 + \nu}$$

$$\text{mode-IV: } \rho = 1$$

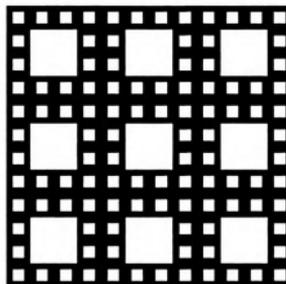
## 4.4 Examples

This section presents analysis of fractal patterns created by the iterative method described earlier. Fair comparison of two different fractal geometries would require that the cell density be the same. This is seldom possible when discretely iterated shapes are involved. Instead, the materials are compared to the analytical calculation of rank-2 material for a given density. The example patterns are solved with the finite element solver described in chapter 2 and the homogenization techniques of chapter 3. The isotropic

material tensor was created using  $E^0 = 100.0$  and  $\nu^0 = 0.30$ . The convergence tolerance for the residual in each iterative solver was  $1 \times 10^{-5}$ .

4	6	8	12
3			11
2			10
1	5	7	9

a) 'Sponge' blueprint



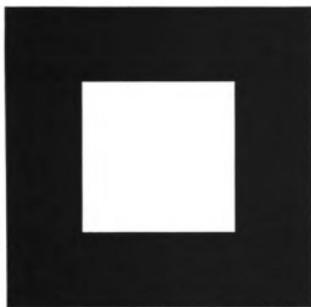
b) Chain of 9, Depth 2 cells

Figure 4.4: 'Sponge' fractal geometry

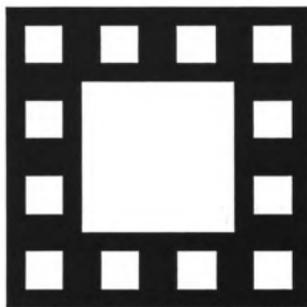
### 'Sponge' Geometry Example

Figure 4.5 shows four sequential fractal maps that were created using the 'sponge' map described earlier in section 4.2.3. First, the pattern removes a square of material away from the center of the cell. Then, at each consecutive iteration, a square is removed from the center of each sub-cell. As a result, material is removed furthest away from the edges of each cell. However, because of sharp corners, stresses are quite large at a hole's edge. Appendix A contains a plot illustrating the Von Mises stress of the cell under axial loading.

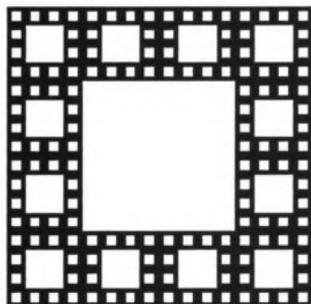
a.) Depth 1



b.) Depth 2



c.) Depth 3



d.) Depth 4

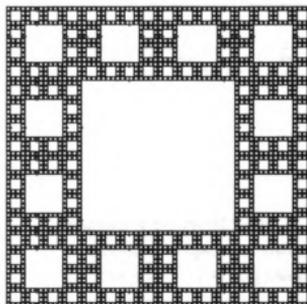
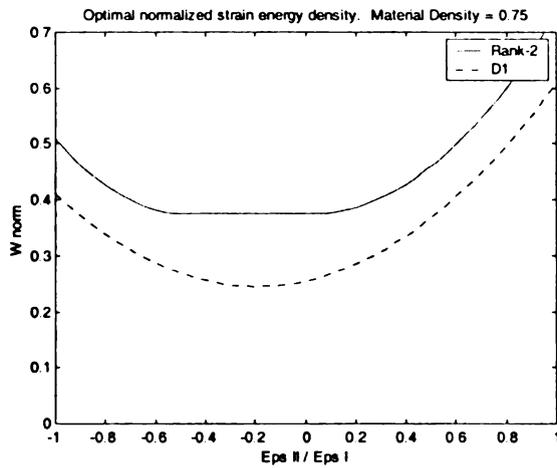
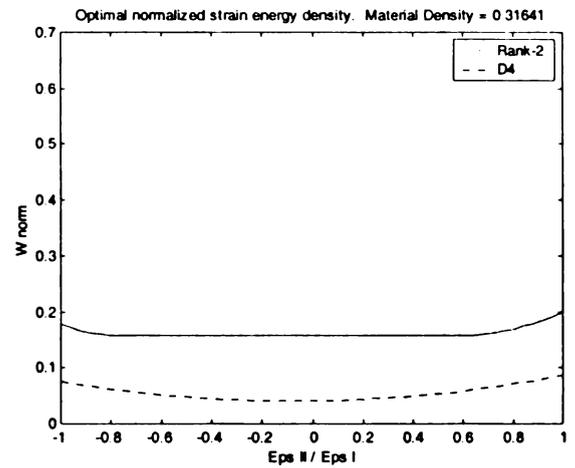


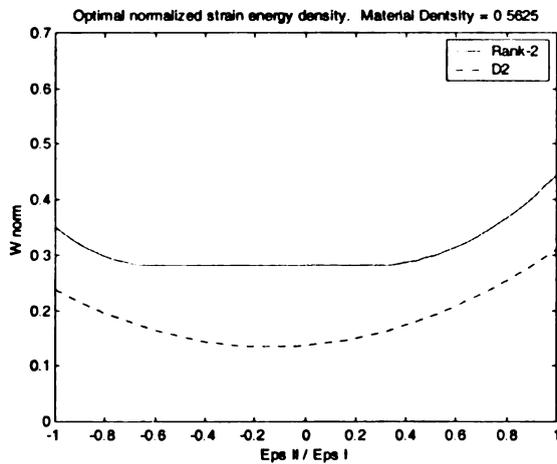
Figure 4.5: Fractals generated by successive mappings from table 4.1



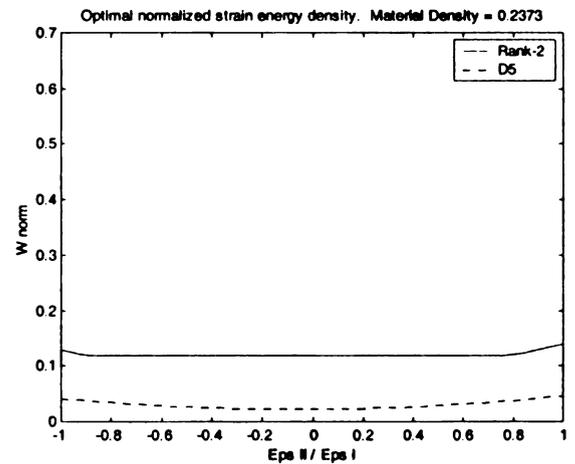
a.)



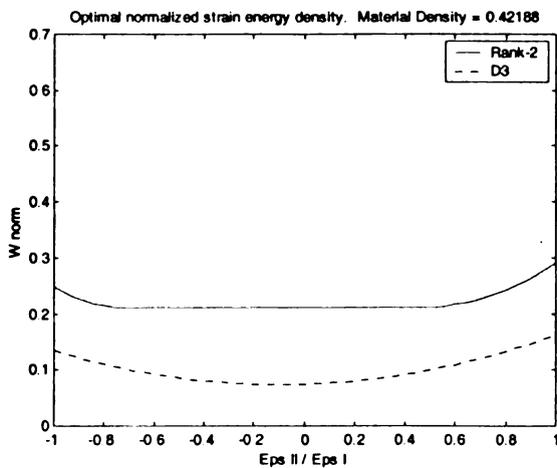
d.)



b.)



e.)



c.)

**Figure 4.6: Stiffness analysis for sponge shaped geometry of varying density.**

Figure 4.6 shows the strain energy density plots of the ‘sponge’ shaped base cell for various cell densities. The performance of the rank-2 material is also plotted. Note that the strain energy density for the rank-2 material is greater than that of the ‘sponge’ geometry. Furthermore, the curves are not symmetric about the axis  $\epsilon_{11} / \epsilon_1 = 0.0$ . The fractal dimension of this shape is 1.79.

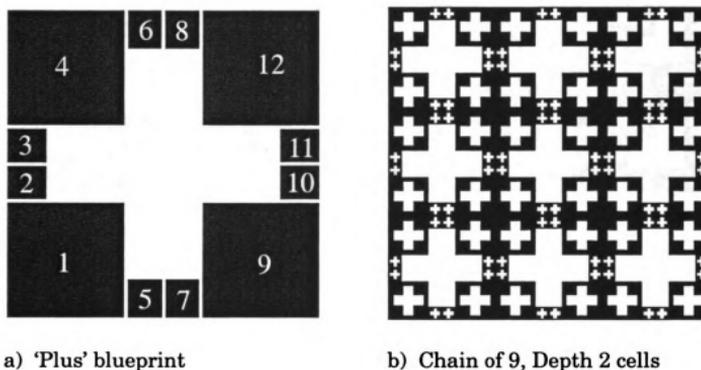
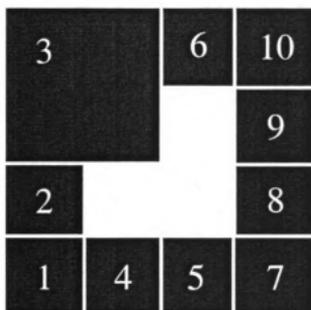


Figure 4.7: ‘Plus’ fractal geometry

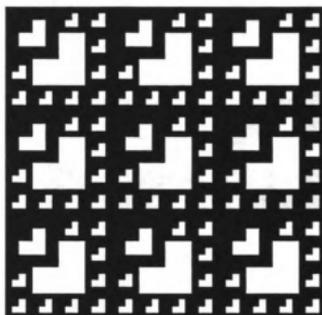
### ‘Plus’ Geometry Example

The next example showcases the ‘plus’ geometry of figure 4.7. The fractal coefficients of the ‘plus’ geometry are documented in table A.1 of appendix A. Material is removed from each cell in the pattern of a ‘plus’ sign at two levels. The first level is four large squares and the second level is eight smaller

squares. As the pattern is iteratively mapped, the cell remains symmetric about the horizontal and vertical axes. This pattern will reduce more cell weight per iteration than the 'sponge' shape of figure 4.5. However, the 'plus' pattern is less stiff than the 'sponge' pattern.



a) 'El' blueprint



b) Chain of 9, Depth 2 cells

Figure 4.8: 'El' fractal geometry

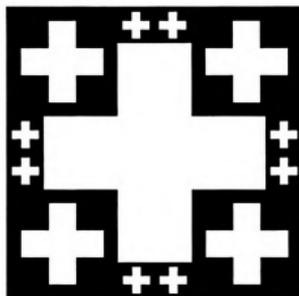
### 'El' Geometry Example

Shown in figure 4.8, the 'el' pattern is not symmetric about the horizontal or vertical axes. The coefficients to construct this shape are presented in table A.2 appendix A. This geometry also has two different scale levels in the affine maps. In figure 4.8 it is shown by affine map 3 having a larger

a.) Depth 1



b.) Depth 2



c.) Depth 3

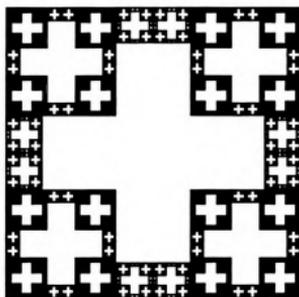
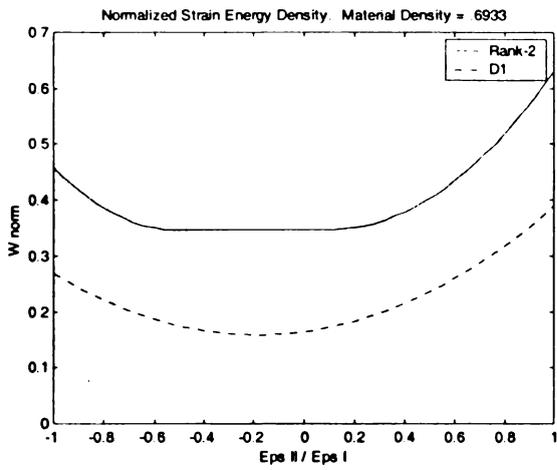
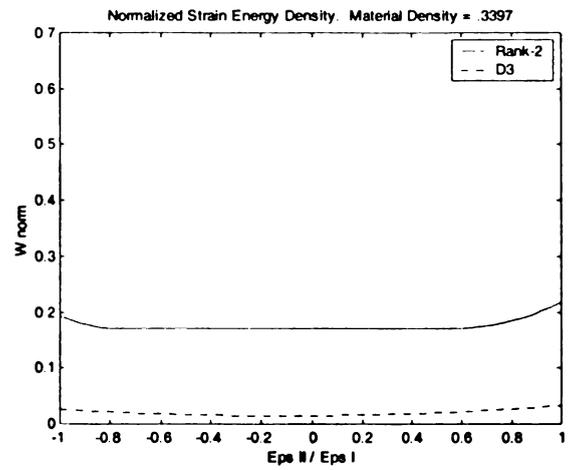


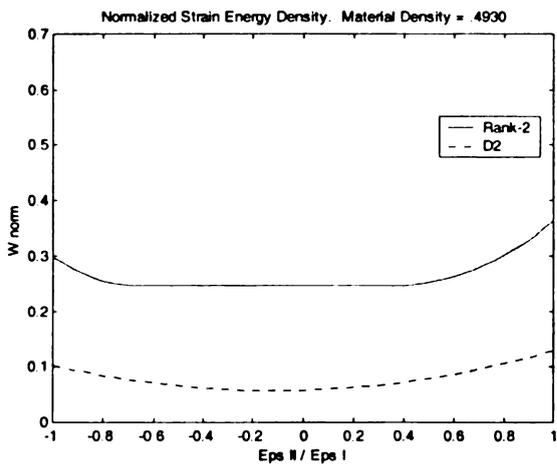
Figure 4.9: 'Plus' shaped geometry.



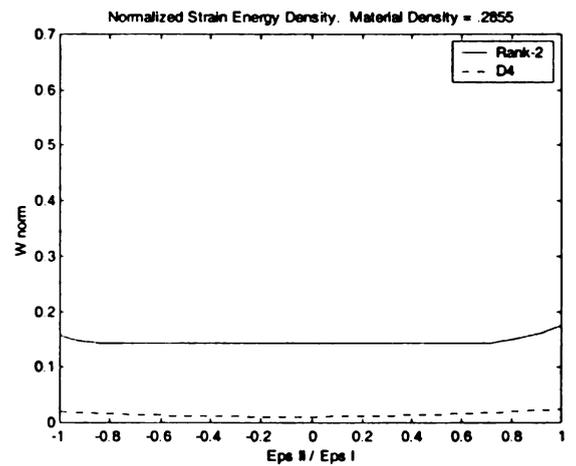
a.)



c.)



b.)



d.)

**Figure 4.10: Stiffness of 'Plus' shape.**

a.) Depth 1



b.) Depth 2



c.) Depth 3

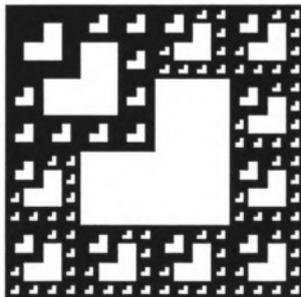
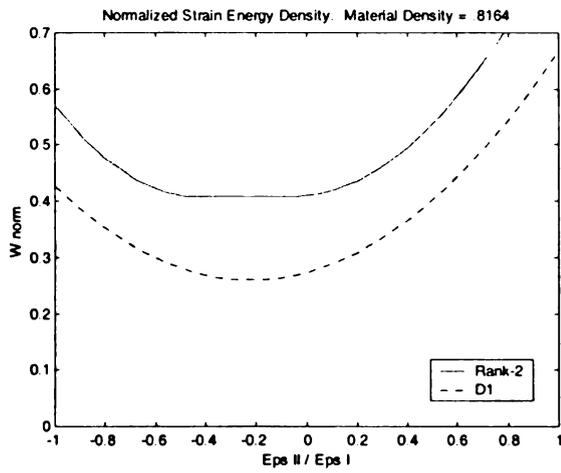
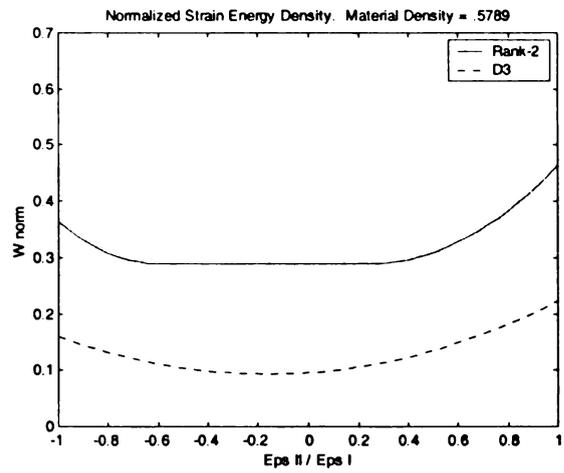


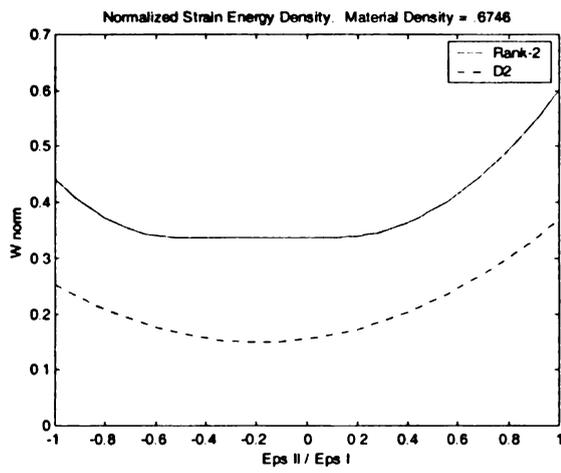
Figure 4.11: 'El' pattern generated by successive mappings



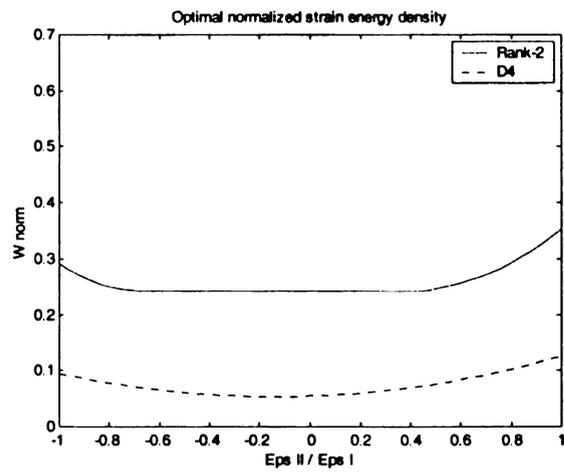
a.)



c.)



b.)



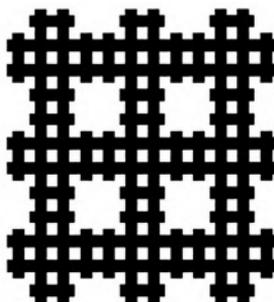
d.)

Figure 4.12: Stiffness of 'el' shape.

transformation scaling than the other affine maps. Figure 4.12 shows the strain energy density comparison to the rank-2 material. The stiffness per unit weight of this cell is also less than the 'sponge' shape as shown in figures 4.19-4.21.



a) Depth 1

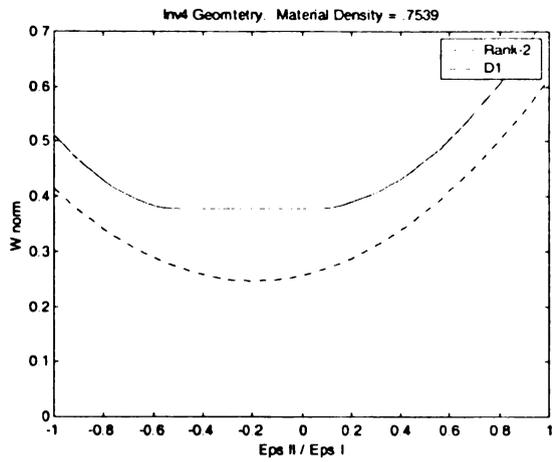


b) Chain of 9, depth 2 cells

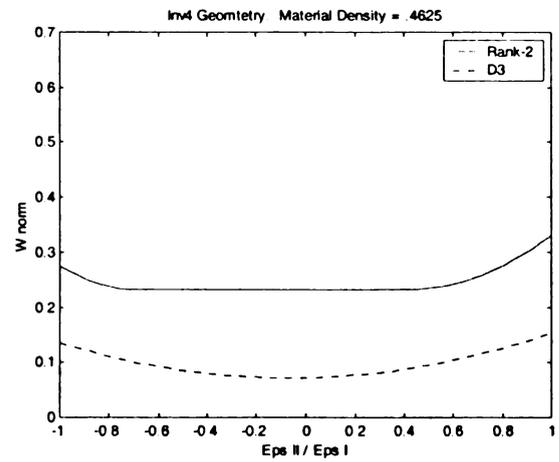
Figure 4.13: 'Inv4' fractal geometry

### 'Inv4' Geometry Example

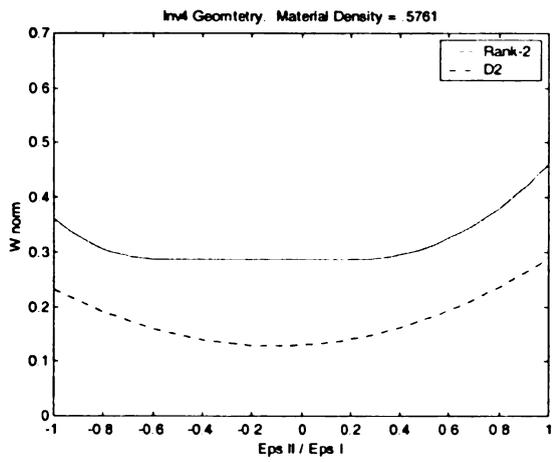
The 'Inv4' geometry shown above is a symmetric pattern whose performance falls between the 'sponge' and 'plus' shapes. It doesn't reduce weight as severely as the 'plus' per iteration, but it is a stiffer pattern per unit weight (see figure 4.19-4.21). Figure 4.13b shows the result of connecting 9 cells of



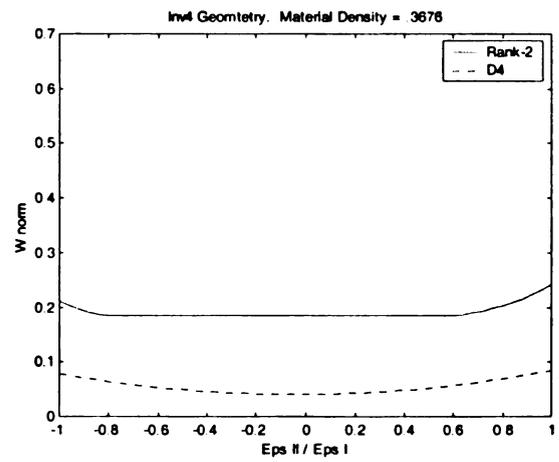
a.)



c.)



b.)



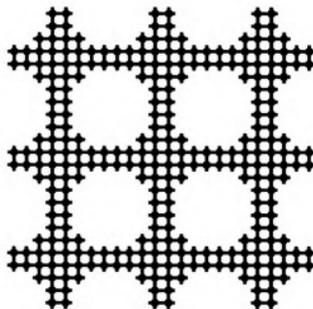
d.)

Figure 4.14: Stiffness of the 'inv4' shape.

the depth two pattern of 'inv4'. The affine maps for the 'inv4' pattern are listed in table A.4 of appendix A. The 'inv4' pattern has a fractal dimension of 1.79.



a) Depth 1

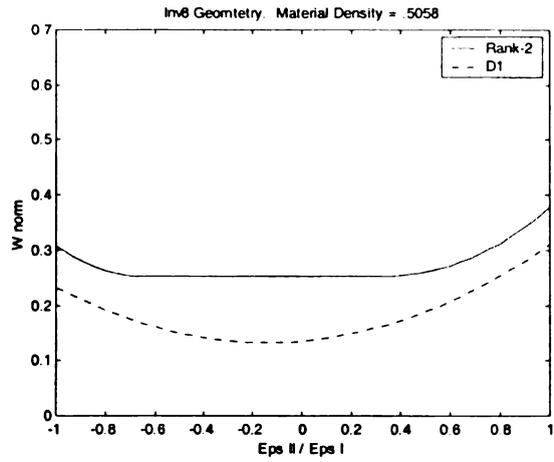


b) Chain of 9, Depth 2 cells

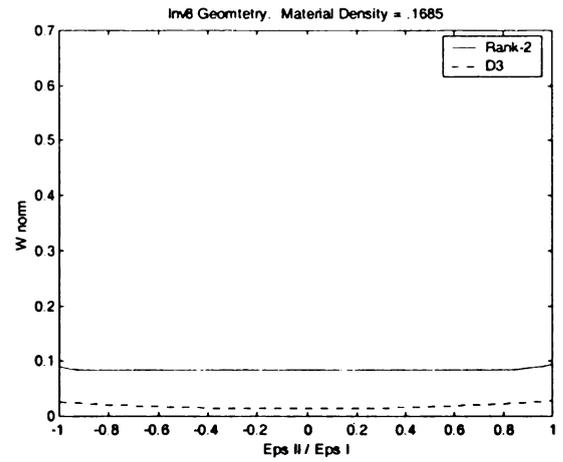
Figure 4.15: 'Inv8' fractal geometry

### 'Inv8' Geometry Example

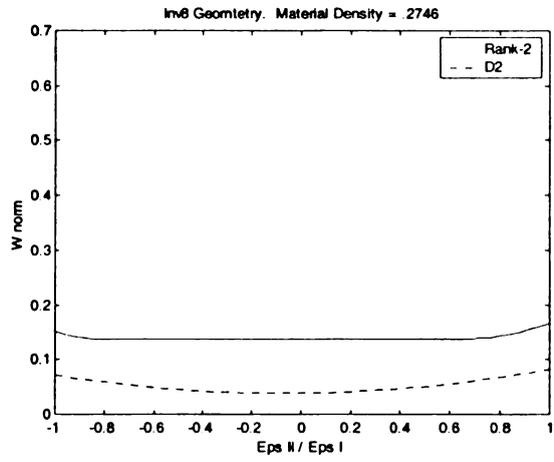
The 'Inv8' pattern was the stiffest fractal pattern per unit weight that was tested. As shown in figure 4.15, the 'Inv8' pattern has a greater strain energy density per density than the 'sponge' shape. Figure 4.15b shows the small-scale structure resulting from the depth 2, 'Inv8' base pattern. The 'inv8' pattern has a fractal dimension of 1.67.



a.)



c.)

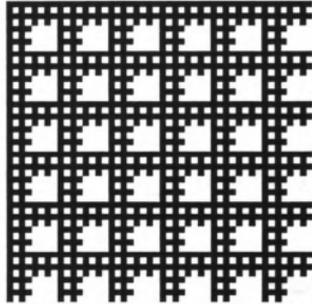


b.)

**Figure 4.16: Stiffness of the 'inv8' shape compared to the rank-2 material.**



a) Depth 1



b) Chain of 9, Depth 2 cells

Figure 4.17: 'fd4' fractal geometry

### 'Fd4' Geometry Example

The 'fd4' fractal pattern has very similar performance to the 'inv4' pattern.

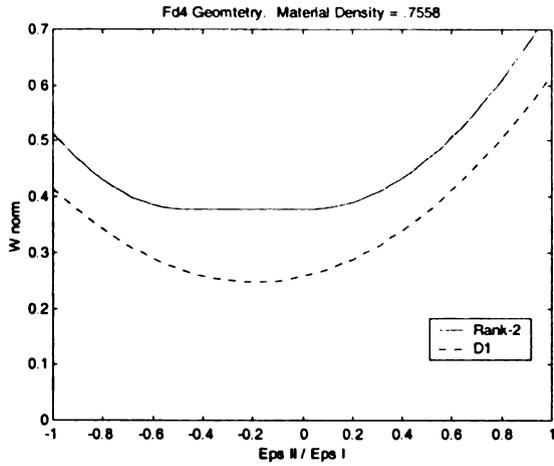
The differences are in the pattern symmetry. One cell of the 'fd4' pattern is not vertically or horizontally symmetric. Nevertheless, the effective

properties calculation considers the material to be periodic as shown in 4.17b.

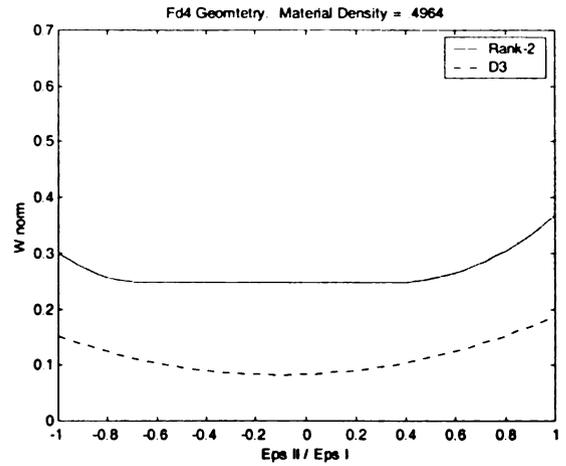
Thus, the structure is similar in the vertical and horizontal directions, which is why its stiffness in these directions is similar. It should be noted that the

'inv4', 'sponge', and 'fd4' shapes all reduce the same fraction of weight per

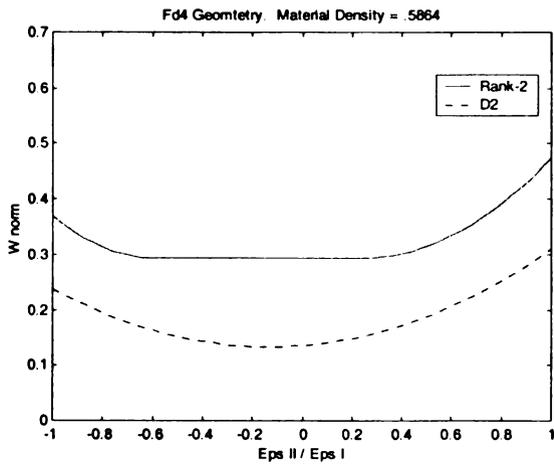
iteration. Also, the 'fd4' pattern has a fractal dimension of 1.79.



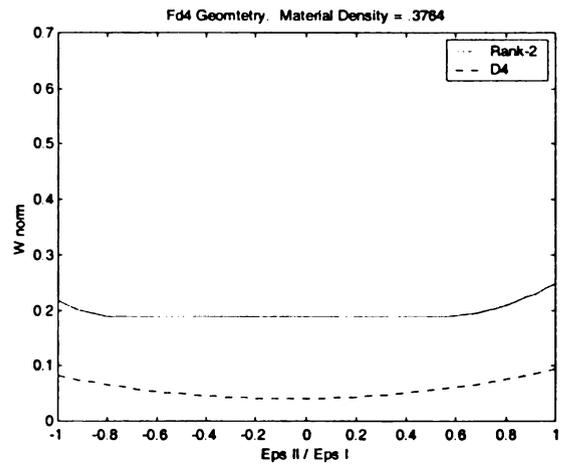
a.)



c.)



b.)



d.)

**Figure 4.18: Stiffness of the 'fd4' shape.**

## Relative Performance of Fractal Shapes

Figure 4.19-4.21 show the relative stiffness of the six fractal patterns when  $\epsilon_{II}/\epsilon_I = 1.0, 0.5,$  and  $-0.75$  respectively. The stiffness is directly proportional to normalized strain energy density. Each series of points represents a sequence of a mapped fractal shape. The black curve represents the benchmark rank-2 material. This material was calculated at each density corresponding to the discretely mapped fractal patterns.

All six fractal shapes that were analyzed had a lower strain energy density than the optimal solution for all ratios of principal strain. Thus, the rank-2 microstructure presents a stiffer solution per unit weight. It is evident that the 'Inv8' shape is the stiffest of the six tested fractal shapes. Conversely, the 'plus' and 'el' shapes are the most compliant patterns for a given density.

It should be noted that the 'sponge', 'fd4', and 'inv4' fractal patterns all have the same fractal dimension. As the number of mapping iterations increases, these three shapes all reduce the effective density at the same rate. The discrepancy in effective density on the following graphs is due to discretization error of the image based input generation software.

*Remark:* if the rank-2 material was allowed to have its upper bound value of  $\delta = 1.0$ , and thus a uniformly solid material, the blue curve would be a

straight line. Instead the blue curve is a mode-II distribution described in equations 4.7 and 4.8.

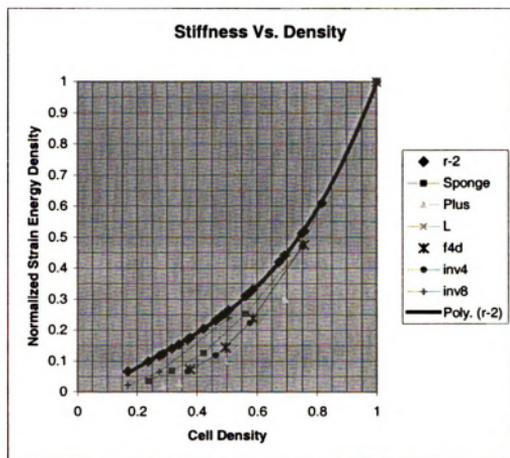


Figure 4.19: Stiffness comparison of fractal patterns for  $\epsilon_{II}/\epsilon_I = 1.0$

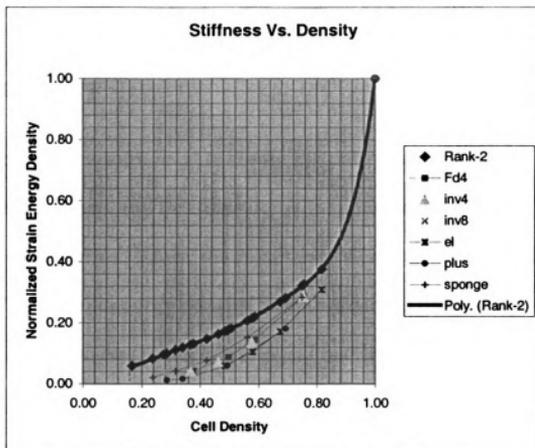


Figure 4.20: Stiffness comparison of fractal patterns for  $\epsilon_{II}/\epsilon_I = 0.5$

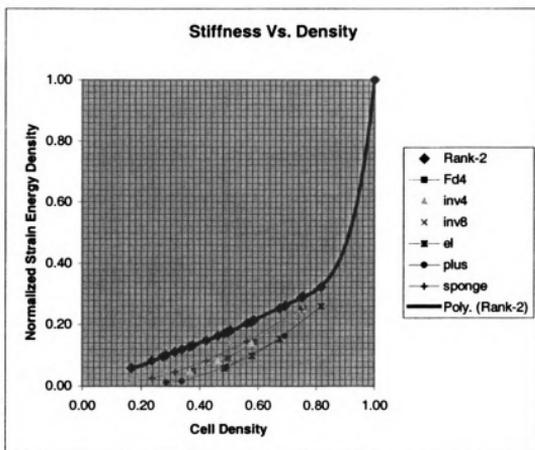


Figure 4.21: Stiffness comparison of fractal patterns for  $\epsilon_{II}/\epsilon_I = -0.75$

## 4.5 Summary

Iterated affine maps were used to create the patterned periodic microstructure. Then, the fictitious domain approach was applied to solve for the effective properties. Then, the results were compared to known optimal solutions. The strain energy density formulation was used as a metric for comparison between two materials of equal material density.

# Chapter 5

## Conclusions

### 5.1 Summary

The fictitious domain method was used with a new iterative scheme to solve for the effective properties of two-dimensional iteratively mapped microstructures. Below is a list of conclusions concerning PCG performance, effective property calculations, and performance of fractal patterns.

#### **PCG Performance**

1. The fictitious domain and PCG in PCG methods produced size insensitive convergence rates.
2. The proposed method met the memory storage requirements for efficient computer use.

3. The implementation of the image-based method of periodic microstructure analysis was successful. Simple image based input was functional and practical.

## **Performance Of Fractal Patterns**

1. By comparing strain energy density, the fractal shaped microstructure did not emit greater stiffness than the optimal rank-2 material.
2. The 'inv8' was the most efficient pattern of material removal. Per unit weight, the 'inv8' pattern was the stiffness.
3. The iterative mapping technique proved to be a convenient way of creating material microstructures.
4. As material was removed, stiffness decreased from the base cell in a regular fashion.

## **5.2 Areas Of Future Work**

Based on the conclusions, the following areas should be explored:

1. Analyze all mesh sizes: A new FFT method could be implemented for the preconditioning system such that mesh sizes other than powers of two can be explored. This would increase the possible fractal geometries that can be studied.

2. Create an 'optimal' pattern: Optimize the stiffness of the base cell by finding fractal mapping coefficients  $a, b, c, d, e,$  and  $f$  such that the compliance is minimized. This could be achieved by modifying the existing program.
3. Add graphical user interface (GUI): Although the current method uses an image for input, a truly integrated system is more desirable. A complete package could have an integrated preprocessor and processor with graphical implementation of boundary conditions.

# Bibliography

- [1] Barnsley, Michael, "Fractals Everywhere," Academic Press/Morgan Kaufmann, 2nd edition (June 1993), pp. 531.
- [2] Bendsoe, Philip Martin, "Optimization of Structural Topology, Shape, & Material", Springer, 1995.
- [3] Bendsoe, Philip Martin, "Optimal shape design as a material distribution problem", Springer, 1989.
- [4] Bendsoe, P.M., Kikuchi, N., "Generating Optimal Topologies in Structural Design Using a Homogenization Method," Computer Methods in Applied Mechanics and Engineering, 1988. pp. 197-224.
- [5] Bensoussan, A., Lions, J.L., Papanicolaou, G., "Asymptotic analysis for periodic structures," Amsterdam, North-Holland. 1978.
- [6] Bjorck, Ake, "Numerical Methods," Prentice-Hall, New Jersey, 1974. pp. 161.
- [7] Chandrupatla, Tirupathi R. and Belegundu, Ashok, "Introduction To Finite Elements in Engineering," Prentice Hall, New Jersey. 1991.
- [8] DeRose, G., "Solving Topology Optimization Problems Using Wavelet-Galerkin Techniques," 1999.
- [9] Diaz, Alejandro R., "A Wavelet-Galerkin Scheme for Analysis of Large-Scale Problems on Simple Domains," International Journal For Numerical Methods in Engineering, 1999. pp. 1610-1612.
- [10] Diaz, A., Sigmund, O., "Checkerboard Patterns in Layout Optimization," 1994.

- [11] Jog, C.S. and R.B. Haber, "A Displacement-Based Topology Design Method with Self Adaptive Layered Materials," Kluwer Academic Publishers, 1993. pp. 219-230.
- [12] Luenberger, David G. , "Introduction to Linear and Nonlinear Programming," Addison-Wesley Publishing Company, Inc. 1965. pp. 168-176.
- [13] The Math Works Inc., "The Student Edition of MATLAB," Prentice Hall, NJ, 1995.
- [14] Pedersen, P., "On Optimal Orientation of Orthotropic Materials," Structural Optimization. 1989. pp. 101-106.
- [15] Reddy, J.N., "An Introduction to the Finite Element Method," McGraw-Hill Inc. 1993. pp. 455-476.
- [16] Saad, Yousef, "Iterative Methods for Sparse Systems," PSW Publishing Company, 1996. pp. 245-251.
- [17] Sigmund, Ole, "Materials with Prescribed Constitutive Parameters: An Inverse Homogenization Problem," pp. 1-10.
- [18] Volterra, Enrico, "Advanced Strength of Materials," Prentice Hall Inc., New Jersey, 1971, pp. 159-163.
- [19] Zienkiewicz, O.C., "The Finite Element Method, Fourth Edition," McGraw-Hill Inc. 1989, pp. 70-79.
- [20] Guedes, Jose, "Nonlinear Computational Models For Composite Materials Using Homogenization," A Thesis.

# Appendix A

## A.1 Coefficients for Fractal Maps

$\xi$	a	b	c	d	e	f
1	3/8	0	0	3/8	0	0
2	1/8	0	0	1/8	0	3/8
3	1/8	0	0	1/8	0	1/2
4	3/8	0	0	3/8	0	5/8
5	1/8	0	0	1/8	3/8	0
6	1/8	0	0	1/8	3/8	7/8
7	1/8	0	0	1/8	1/2	0
8	1/8	0	0	1/8	1/2	7/8
9	3/8	0	0	3/8	5/8	0
10	1/8	0	0	1/8	7/8	3/8
11	1/8	0	0	1/8	7/8	1/2
12	3/8	0	0	3/8	5/8	5/8

Table A.1: 'Plus' fractal coefficients

$\xi$	a	b	c	d	e	f
1	1/4	0	0	1/4	0	0
2	1/4	0	0	1/4	0	1/4
3	1/2	0	0	1/2	0	1/2
4	1/4	0	0	1/4	1/4	0
5	1/4	0	0	1/4	1/2	0
6	1/4	0	0	1/4	1/2	3/4
7	1/4	0	0	1/4	3/4	0
8	1/4	0	0	1/4	3/4	1/4
9	1/4	0	0	1/4	3/4	1/2
10	1/4	0	0	1/4	3/4	3/4

Table A.2: 'El' fractal coefficients

x	a	b	C	d	e	f
1	1/4	0	0	1/4	0	0
2	1/4	0	0	1/4	0	1/4
3	1/4	0	0	1/4	0	1/2
4	1/4	0	0	1/4	0	3/4
5	1/4	0	0	1/4	1/4	1/4
6	1/4	0	0	1/4	1/4	3/4
7	1/4	0	0	1/4	1/2	0
8	1/4	0	0	1/4	1/2	3/4
9	1/4	0	0	1/4	1/2	1/4
10	1/4	0	0	1/4	3/4	1/4
11	1/4	0	0	1/4	1/2	1/2
12	1/4	0	0	1/4	3/4	3/4

Table A.3: 'f4' fractal coefficients

$\xi$	a	b	c	d	e	f
1	1/4	0	0	1/4	1/4	1/4
2	1/4	0	0	1/4	0	1/4
3	1/4	0	0	1/4	0	1/2
4	1/4	0	0	1/4	1/4	1/2
5	1/4	0	0	1/4	1/4	0
6	1/4	0	0	1/4	1/4	3/4
7	1/4	0	0	1/4	1/2	0
8	1/4	0	0	1/4	1/2	3/4
9	1/4	0	0	1/4	1/2	1/4
10	1/4	0	0	1/4	3/4	1/4
11	1/4	0	0	1/4	1/2	1/2
12	1/4	0	0	1/4	3/4	1/2

Table A.4: 'Inv4' fractal coefficients

$\xi$	a	b	c	d	e	f
1	1/8	0	0	1/8	3/8	0
2	1/8	0	0	1/8	3/8	1/8
3	1/8	0	0	1/8	3/8	1/4
4	1/8	0	0	1/8	3/8	3/8
5	1/8	0	0	1/8	3/8	1/2
6	1/8	0	0	1/8	3/8	5/8
7	1/8	0	0	1/8	3/8	3/4
8	1/8	0	0	1/8	3/8	7/8
9	1/8	0	0	1/8	0	3/8
10	1/8	0	0	1/8	1/8	3/8
11	1/8	0	0	1/8	1/4	3/8
12	1/8	0	0	1/8	1/2	3/8
13	1/8	0	0	1/8	5/8	3/8
14	1/8	0	0	1/8	3/4	3/8
15	1/8	0	0	1/8	7/8	3/8
16	1/8	0	0	1/8	1/2	0
17	1/8	0	0	1/8	1/2	1/8
18	1/8	0	0	1/8	1/2	1/4
19	1/8	0	0	1/8	1/2	3/8
20	1/8	0	0	1/8	1/2	1/2
21	1/8	0	0	1/8	1/2	5/8
22	1/8	0	0	1/8	1/2	3/4
23	1/8	0	0	1/8	1/2	7/8
24	1/8	0	0	1/8	0	1/2
25	1/8	0	0	1/8	1/8	1/2
26	1/8	0	0	1/8	1/4	1/2
27	1/8	0	0	1/8	3/8	1/2
28	1/8	0	0	1/8	5/8	1/2
29	1/8	0	0	1/8	3/4	1/2
30	1/8	0	0	1/8	7/8	1/2
31	1/8	0	0	1/8	1/4	1/4
32	1/8	0	0	1/8	1/4	5/8
33	1/8	0	0	1/8	5/8	1/4
34	1/8	0	0	1/8	5/8	5/8

Table A.5: 'Inv4' fractal coefficients

## A.2 Stress In An Axially Loaded Block

The isotropic material tensor was created using  $E^0 = 100.0$  and  $\nu^0 = 0.30$ . An axial load was applied to the structure as shown below:

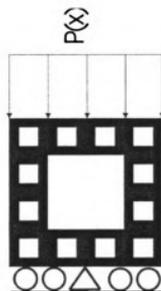


Figure A.2: Loading conditions for stress analysis

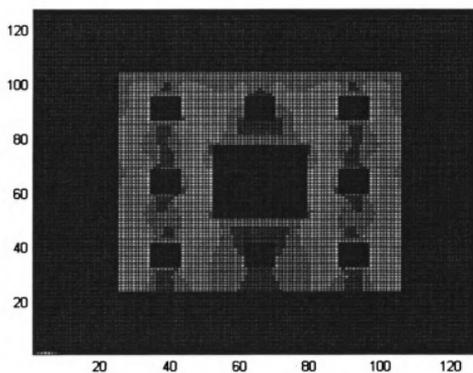


Figure A.1: Stress analysis of fractal geometry depth 2

### A.3 Analytical Solution Of The Stressed Plate With Hole

$r$	$\sigma_r$	$\sigma_\theta$	Von Mises Stress
12.0	0.0	30000.0	30000.0
14.0	2923.8	21770.1	20465.4
16.0	3691.4	17558.6	16034.8
18.0	3703.7	15185.2	13713.7
20.0	3456.0	13744.0	12383.1
22.0	3135.0	12815.4	11570.9
24.0	2812.5	12187.5	11053.0
26.0	2514.6	11745.7	10712.1
28.0	2249.1	11424.4	10482.4
30.0	2016.0	11184.0	10324.7
32.0	1812.7	10999.7	10214.7

Table A.3: Value of stress along the  $r$  axis, see figure A.3

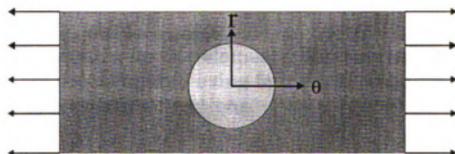


Figure A.3: Illustration of coordinate axis for plate with hole

# Appendix B

## Description Of Subroutines

### Input routines

#### INPUT

This subroutine reads the program input data from two files. 'INPUT.DAT' contains material properties. 'RHO.DAT' contains the size N by N mesh in binary format. Material is denoted by a 1 and void by 0.

#### STIFF

Loads local stiffness matrix for one element. Element equations and finite element model were pre-computed using four noded quadrilateral elements employing linear interpolation functions.

#### OUTSTRAIN

This subroutine outputs strain.

### Math Subroutines

#### DOTP

Calculates the scalar (Dot) product

#### MATMULT

Multiplies two matrices.

#### IMATMULT

Multiplies the global stiffness matrix, with boundary conditions applied, by a vector. This matrix times vector operation is done iteratively such that the global stiffness matrix is never assembled.

$$[M] + ([I] - [M]) \times K \times ([I] - [M]) \times p$$

- I is the identity matrix
- M is a matrix of zeros with  $M(i,i) = 1$  where i is a constrained degree of freedom
- K is the global stiffness matrix
- p is a vector

### HOMOMATMULT

Multiplies a homogenous matrix, with boundary conditions applied, by a vector. This matrix times vector operation is done iteratively such that the global stiffness matrix is never assembled. Identical to IMATMULT except rho is not multiplied by the material matrix.

### DIAGBLKCIRC

Takes fft of vector twice. This creates the diagonal blocks.

### EXPBLKCIRC

This routine ifft's the blocks.

### INVTRID

This subroutine inverts a tri-diagonal matrix.

### FFT

This subroutine takes the fast Fourier transform of a vector of size  $2^j$ .

## Solver Subroutines

### PRE\_P\_INVERT (called once)

Builds, fft's, and inverts the second preconditioner  $P_2 = (A_h + Q)$ .

### PCGSOLVER

Solve:

$$[M] + ([I] - [M]) \times [K] \times ([I] - [M]) \times x = ([I] - [M]) \times F$$

Using a Preconditioned Conjugate Gradient (PCG) algorithm. The Preconditioner for this system is:

$$P_1 = [M] + ([I] - [M]) \times [A_h] \times ([I] - [M])$$

However, this is difficult to invert. Therefore,

$$z = P_1^{-1} \times r$$

is solved using a second preconditioner. This can be found in SECONDARYPCG.

SECONDARYPCG (In PCGSOLVER loop)

$$z = P_1^{-1} \times r$$

Is solved using the same PCG algorithm with the preconditioner:

$$P_2 = (A_h + Q)$$

Notice that  $A_h$  is a singular matrix and that the rigid body mode and thus its singularity is eliminated by the addition of  $Q$ .

P\_INVERT (In SECONDARYPCG loop)

Multiplies:

$$P_1^{-1} \times r$$

This requires the FFT of vector  $r$  and the multiplication and IFFT of the product.

AHASSEMBLER (Called Once)

Assembles  $A_{uu}$ ,  $A_{uv}$ ,  $A_{vu}$ , and  $A_{vv}$ .

MAKEF (Called once)

Creates the three initial strain vectors.

### **Post Processor Subroutines**

STRESSSTRAIN (Called 3 times)

For each prestrain, this subroutine calculates the strain field over the entire domain. Also, it calculates the Von Mises stress over the domain.

GMAT (Called once)

Loads data for the stress calculation.

HOMOGENIZATION

**Calculates base cell effective properties.**

# Appendix C

## Program Input

The following is a listing of the sample input to the fractal analyzer program.

Two input files are required:

1. Rho.dat contains the image. Integer values are required without spaces. Below is a sample

```
111111111
111111111
111111111
111000111
111000111
111000111
111111111
111111111
111111111
```

2. Input.dat contains the required information in FOTRAN77 format.

Below is a sample input,

```
SPONGE d2, 1/16/2000
100.0
.30
1.0
256
```

```
Title line, 60 characters maximum
Young's Modulus, F8.2
Poisson's ratio, F8.2
P ratio, not currently used
Domain, must be an integer power of 2
```

MICHIGAN STATE UNIV. LIBRARIES



31293020611699