The Application of the Multigrid Method to the Finite Element Solution of Solid Mechanics Problems

Thesis by

Ian Dennis Parsons

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1988

(Submitted July 21, 1987)

To Blondie

Acknowledgments

I would like to thank all of the people who made this thesis possible, including: my advisor, Professor John Hall, for providing guidance and motivation; Professor Ares Rosakis, for helpful discussions regarding fracture mechanics; Greg Lyzenga and Moktar Salama at JPL for helping to extend the practical importance of this work; Professor Erik Antonsson for the generous use of his computing resources; Kris Wood for his computer expertise, especially with the PS 300; Tom Heaton at the USGS; Jim Sterling and Mike Moser for the use of their Latex files; Cecilia Lin for providing excellent diagrams; and, finally, Shawn for caring. Financial support was provided in part by the Caltech President's Fund.

Abstract

A multigrid algorithm is described that can be used to obtain the finite element solution of linear elastic solid mechanics problems. The method is applied to some simple two and three dimensional problems to evaluate its strengths and weaknesses. The usefulness of the method is demonstrated by solving some large three dimensional problems of practical interest.

When conditions of near incompressibility are encountered, the multigrid method performs poorly due to a combination of a reduction in the smoothing effect of the Gauss-Seidel relaxation method and coarse mesh locking. These problems can be partially cured by using the Jacobi preconditioned conjugate gradient method to smooth the error, and assembling the coarse mesh stiffness matrices using a reduced integration scheme.

It is also found that the bending behavior of the linear brick and quadrilateral elements used in this thesis slow the convergence of the multigrid method. This effect also causes nonuniform meshes to yield computation times that are not proportional to the problem size; however, the linear dependence can be recovered by increasing the refinement of the finite element meshes. It is demonstrated that reduced integration techniques become less effective in relieving the stiffness of the coarse mesh for nonuniform meshes as the problem size is increased. The solution of a well-conditioned three dimensional test problem shows that the multigrid algorithm requires far less computational effort than a direct method, and that its performance is comparable to that of the Jacobi preconditioned conjugate gradient method.

The usefulness of the multigrid method is demonstrated by applying it to the finite element solution of two solid mechanics problems of engineering interest: the elastostatic state near a three dimensional edge crack, and the relationship between the average offset and the stress drop for two and three dimensional faults in a half-space. The features of the solution to these problems are extensively discussed. It is found that the multigrid method is faster than the Jacobi preconditioned conjugate gradient method when applied to these practical problems.

The investigations described in this thesis reveal some interesting features of the performance of the multigrid method when it is applied to the finite element solution of solid mechanics problems. In particular, the storage requirements of the method are linearly proportional to the problem size. The constant of proportionality depends only on the dimension of the problem. The solution times of the multigrid method are found to be linearly proportional to the problem size if uniform meshes are used. However, this is not true for most of the problems that are solved with nonuniform meshes. The constant of proportionality in the relationship between the problem size and the solution time depends on the particular problem under consideration.

Table of Contents

List of Figures

List of Tables

Part I

Introduction to the Multigrid Method

Chapter 1

Introduction and Objectives

Over the last twenty years the displacement based finite element method has become the most widely used technique for numerically solving engineering solid and structural mechanics problems. The discretization of a particular problem (linear or nonlinear, static or dynamic) invariably results in the generic matrix equation

$$\mathbf{K}\mathbf{x} = \mathbf{f}, \tag{1.1}$$

as the basic solution step, where $\mathbf{K}$ is an $n \times n$ generalized stiffness matrix (often symmetric, positive definite and ill-conditioned), $\mathbf{x}$ is a vector of n unknown displacements, and $\mathbf{f}$ is a vector of n known forces. Equation (1.1) is frequently solved by a direct factorization method based on Gaussian elimination. The high cost of these methods in terms of the computational effort and the storage required makes them impractical for large problems. Iterative methods have lower storage requirements but tend to exhibit slow convergence for ill-conditioned systems.

The objectives of this thesis are to describe the application of a multigrid algorithm (which is essentially a fast iterative method) to the solution of equation (1.1), to evaluate its effectiveness by applying it to some simple two and three

dimensional problems, and to demonstrate its usefulness by solving some large three dimensional solid mechanics problems of practical interest. The history of the multigrid method has been documented elsewhere [27,56]. Although the convergence of the method has been established, applications have concentrated on the field of fluid mechanics and have mostly used a finite difference discretization of the relevant problems [12,25,26,27,56].

Part I of this thesis introduces the multigrid method, and shows how it can be used to solve solid mechanics problems. Chapter 2 outlines the way in which the finite element method is used to obtain approximate solutions to these types of problems. A brief summary of the methods available to solve the linear matrix equation (1.1) is given in chapter 3, together with a discussion of the problem of ill-conditioned systems. In chapter 4, a multigrid algorithm is described that can be used to solve equation (1.1) that arises from the finite element discretization of a solid mechanics problem. The method is applied to some simple two and three dimensional solid mechanics problems in chapter 5 to evaluate its strengths and weaknesses. The computational effort and the storage requirements of the scheme are also compared to those of a direct factorization algorithm and another iterative scheme (the Jacobi preconditioned conjugate gradient method).

In part II, the usefulness of the multigrid method is demonstrated by applying it to the finite element solution of two solid mechanics problems of engineering interest. An investigation of the elastostatic state near a three dimensional edge crack is described in chapter 6. A class of linear elastic problems originating from the field of geophysics is discussed in chapter 7. This involved computing the relationship between the average offset and the stress drop for two and three

dimensional faults in a half-space. The work presented in part II not only shows how the multigrid method can be used to solve real world problems, but also gives extensive discussions of the solutions to these problems.

Part III contains some closing comments and remarks. Chapter 8 presents a summary of the performance of the multigrid method when it was applied to the various problems described in this thesis. The work is also placed in context by discussing other applications of the multigrid method to the field of solid mechanics, as well as some previously published solutions of various large solid mechanics problems that were obtained using different numerical techniques. In the last section of chapter 8, the performance of the multigrid method is compared to that of another iterative scheme that has been used to solve problems in solid mechanics, the element-by-element preconditioned conjugate gradient method. Chapter 9 outlines some ideas for future research. Finally, chapter 10 contains the main conclusions of the work presented in this thesis.

It should be noted that tables and figures are placed immediately after the chapter in which they are first refered to; tables are placed before figures. For example, table 3.1 and figure 3.1 can be found at the end of chapter 3, with table 3.1 preceding figure 3.1

Chapter 2

The Finite Element Method in Solid and Structural Mechanics

2.1 Background

Basically, the finite element method is a general technique by which differential equations (that are usually difficult to solve analytically) can be cast into matrix equations. These matrix equations can then be solved using a computer to yield an approximate numerical solution to the original differential equations. Although the finite element method can be applied to the equations that govern many different physical systems, it is most often used to solve problems in solid and structural mechanics. The most popular variant of the method that is employed in these situations is the displacement based finite element formulation (sometimes known as the direct stiffness method), which has been extensively discussed in the literature [6,33,55,63].

From the point of view of the engineering analyst, solid and structural mechanics problems can be divided into two classes: linear and nonlinear. These can each be further subdivided into static and dynamic problems. This thesis

will concentrate on the linear static case. For all classes of solid and structural mechanics problems, the displacement based finite element method can most easily be formulated by use of the principle of virtual work.

2.2 Formulation of the Finite Element Equilibrium Equations

Although the principle of virtual work can be used to derive the finite element equilibrium equations for nonlinear solid and structural mechanics problems, this section will specifically deal with those that are governed by the equations of linear elasticity. Consider the body shown in figure 2.1 that occupies a region $\mathcal{R}$. The density and the elasticity tensor of the body will be denoted by ρ and $\mathbf{c}$, respectively. The body is subjected to body forces $\mathbf{b}$ that act throughout $\mathcal{R}$, and to surface tractions $\mathbf{s}$ and concentrated forces $\mathbf{f}^i$ that act on the boundary of the body denoted by $\partial\mathcal{R}_s$. Specified displacements can be applied to the body on its boundary denoted by $\partial\mathcal{R}_d$. The specified displacements and the forces acting on the body cause displacements $\mathbf{u}$, strains $\boldsymbol{\epsilon}$, and stresses $\boldsymbol{\sigma}$. The principle of virtual displacements requires that for any compatible, small virtual displacements causing increments $\delta\mathbf{u}$ (which must be zero on $\partial\mathcal{R}_d$) and $\delta\boldsymbol{\epsilon}$, the total internal virtual work is equal to the total external virtual work (i.e. the virtual work done by $\mathbf{b}$, $\mathbf{s}$, $\mathbf{f}^i$, and the inertial forces) [6]. Applying the principle at a particular instant of time, this can be written as

$$\int_{\mathcal{R}} \delta\boldsymbol{\epsilon}^T \boldsymbol{\sigma} dV = \int_{\mathcal{R}} \delta\mathbf{u}^T(\mathbf{b} - \rho\ddot{\mathbf{u}})dV + \int_{\partial\mathcal{R}_s} \delta\mathbf{u}^T \mathbf{s} dA + \sum^{i} \delta\mathbf{u}^{iT}\mathbf{f}^i, \qquad (2.1)$$

where $\delta\boldsymbol{\epsilon}$ and $\delta\mathbf{u}$ are the virtual strains and displacements, respectively, and the dot denotes differentiation with respect to time. The superscript i denotes the displacements or forces at the points where the concentrated nodal loads $\mathbf{f}^i$ are applied.

In finite element analysis, the body is divided into discrete elements that are connected at their nodal points. The displacements within each element are assumed to be a function of the nodal displacements, i.e.

$$\mathbf{u}^e = \mathbf{N}^e\mathbf{a}^e, \tag{2.2}$$

where $\mathbf{u}^e$ are the displacements within a particular element, $\mathbf{a}^e$ are the displacements at the nodes of the element, and $\mathbf{N}^e$ is the element displacement interpolation, or shape function, matrix. The superscript e denotes an element. Using equation (2.2), the element strains are approximated by

$$\boldsymbol{\epsilon}^e = \mathbf{B}^e\mathbf{a}^e, \tag{2.3}$$

where $\mathbf{B}^e$ is the strain-displacement matrix that is obtained using the strain-displacement relations. The stresses within an element are given by

$$\boldsymbol{\sigma}^e = \mathbf{D}^e\mathbf{B}^e\mathbf{a}^e, \tag{2.4}$$

where $\mathbf{D}^e$ is the elasticity matrix of the element. Dividing the body into elements, equation (2.1) can be written as

$$\begin{aligned}\sum^e \int_{\mathcal{R}^e} \delta\,\boldsymbol{\epsilon}^{eT}\boldsymbol{\sigma}^e dV^e = \sum^e \int_{\mathcal{R}^e} \delta\mathbf{u}^{eT}(\mathbf{b}^e - \rho^e\ddot{\mathbf{u}}^e)dV^e \\ + \sum^e \int_{\partial\mathcal{R}_s^e} \delta\mathbf{u}^{eT}\mathbf{s}^e dA^e + \sum^i \delta\mathbf{u}^{iT}\mathbf{f}^i,\end{aligned} \tag{2.5}$$

where $\sum^e$ represents summation over all of the elements in the mesh. Using the finite element approximation given in equations (2.2), (2.3), and (2.4), equation

(2.5) becomes

$$\delta\mathbf{a}^T \left\{ \sum^{e} \int_{\mathcal{R}^e} \mathbf{B}^{eT}\mathbf{D}^e\mathbf{B}^e dV^e \right\} \mathbf{a} = \delta\mathbf{a}^T \left\{ \sum^{e} \int_{\mathcal{R}^e} \mathbf{N}^{eT}(\mathbf{b}^e - \rho^e \mathbf{N}^e \ddot{\mathbf{a}}) dV^e \right\} + \delta\mathbf{a}^T \left\{ \sum^{e} \int_{\partial\mathcal{R}_s^e} \mathbf{N}^{eT}\mathbf{s}^e dA^e \right\} + \delta\mathbf{a}^T \mathbf{f}_a. \quad (2.6)$$

The load vector $\mathbf{f}_a$ represents the concentrated loads. Note that the i^{th} component of $\mathbf{f}_a$ is the concentrated nodal force which corresponds to the i^{th} degree-of-freedom in $\mathbf{a}$.

Equations for the n unknown nodal displacements $\mathbf{a}$ are obtained from equation (2.6) by imposing unit virtual displacements in turn at all displacement components [6]. The n equations (one for each unknown component, or degree-of-freedom, a_i of $\mathbf{a}$) are then assembled into

$$\mathbf{M}\ddot{\mathbf{a}} + \mathbf{K}\mathbf{a} = \mathbf{f}_b + \mathbf{f}_s + \mathbf{f}_a. \quad (2.7)$$

It is more convenient to construct the finite element equilibrium equation (2.7) by assembling the following element matrices and vectors.

$$\mathbf{K}^e = \int_{\mathcal{R}^e} \mathbf{B}^{eT}\mathbf{D}^e\mathbf{B}^e dV^e. \quad (2.8)$$

$$\mathbf{M}^e = \int_{\mathcal{R}^e} \rho^e \mathbf{N}^{eT}\mathbf{N}^e dV^e. \quad (2.9)$$

$$\mathbf{f}_b^e = \int_{\mathcal{R}^e} \mathbf{N}^{eT}\mathbf{b}^e dV^e. \quad (2.10)$$

$$\mathbf{f}_s^e = \int_{\partial\mathcal{R}_s^e} \mathbf{N}^{eT}\mathbf{s}^e dA^e. \quad (2.11)$$

The matrices $\mathbf{M}^e$ and $\mathbf{K}^e$ are the mass and stiffness matrices of the element, respectively, and are assembled into the global mass and stiffness matrices, $\mathbf{M}$ and $\mathbf{K}$. The load vectors $\mathbf{f}_b^e$ and $\mathbf{f}_s^e$ represent the effect of the element body forces and the tractions acting on the surface of the element, respectively.

The above formulation holds for linear dynamic problems. The static case is exactly the same, with the exception that the inertial term is absent.

2.3 Reduced Integration Techniques

Reduced integration is a method that is used in solid and structural mechanics to improve the behavior of various finite elements [6,33,63]. The technique is employed in this thesis when the four node linear quadrilateral element is used to model a nearly incompressible material (i.e. a material whose Poisson's ratio is close to $\frac{1}{2}$) under conditions of plane strain, and when the element is subjected to a bending deformation.

Physical insight can be gained into an element's behavior by examining its modes of deformation. The linear quadrilateral element has three rigid body and five deformation modes, which are shown in figure 2.2. If the element is used to model a nearly incompressible material subjected to plane strain conditions, the bending modes will become very stiff if a 2×2 (exact) Gauss quadrature is used to compute the element stiffness matrix. This phenomenon, which is referred to as locking, is caused by the considerable volumetric strain energy present in the bending modes at the Gauss points. These modes cannot undergo lateral Poisson straining to relieve the volumetric straining.

The element locking can be cured by using a higher order element (e.g. an eight node quadrilateral), or by using a one point Gauss integration scheme to compute the volumetric strain energy. This reduced integration is effective because the volume change in the element's bending modes is zero at this in-

tegration point. One convenient way to implement this technique for isotropic materials is to split the element stiffness matrix into two parts

$$\mathbf{K}^e = \int_{\mathcal{R}^e} \left(\mathbf{B}^{eT} \mathbf{D}^e_\mu \mathbf{B}^e + \mathbf{B}^{eT} \mathbf{D}^e_\lambda \mathbf{B}^e \right) dV^e, \tag{2.12}$$

where the matrices $\mathbf{D}^e_\mu$ and $\mathbf{D}^e_\lambda$ are such that

$$\mathbf{D}^e = \mathbf{D}^e_\mu + \mathbf{D}^e_\lambda, \tag{2.13}$$

and for plane strain are given by

$$\mathbf{D}^e_\mu = \mu \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.14}$$

and

$$\mathbf{D}^e_\lambda = \lambda \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.15}$$

In equations (2.14) and (2.15), λ $(= \frac{E\nu}{(1+\nu)(1-2\nu)})$ and μ $(= \frac{E\nu}{2(1+\nu)})$ are the Lamé and shear moduli of the material, respectively. Reduced integration is used to compute the second term of equation (2.12) since it contains the locking part of the volumetric strain energy.

The four node linear quadrilateral element performs poorly when it is subjected to a bending deformation. Even though compressibility is not a problem, the stiffness of the element's bending modes are too high due to the absence of a lateral normal strain and the presence of shear strains. In practice, better results are often achieved when one point Gauss integration is employed on both the $\mathbf{D}^e_\lambda$ and the $\mathbf{D}^e_\mu$ terms of the element stiffness matrix in equation (2.12), which reduces the bending stiffness to zero. However, this can lead to zero energy modes

of deformation of the mesh, which result in a singular stiffness matrix [33]. An alternative technique, which will be used in chapter 5 to improve the bending behavior of the quadrilateral element, is to use reduced integration on only one of the $\mathbf{D}_\lambda^e$ and $\mathbf{D}_\mu^e$ terms in equation (2.12).

2.4 Imposition of the Boundary Conditions

As indicated in section 2.2, there are two types of boundary conditions that are used in solid and structural mechanics: specified displacements and specified loads. The specified loads may result from applied nodal forces, applied tractions along the element boundaries, or body forces acting on the elements. These loads are represented by the terms on the right-hand side of equation (2.7); i.e. they all contribute to the forces at the nodal points of the finite element mesh.

Specified displacements are usually dealt with by substituting them into equation (2.7), which results in a contribution to the nodal forces and the elimination of the corresponding degrees-of-freedom. However, it will be seen in chapter 4 that this is cumbersome when using the multigrid method. A more convenient approach is to add into equation (2.7) constraint equations that express the prescribed displacement conditions [6]. If the displacement at degree-of-freedom i is specified as b, i.e. $a_i = b$, the constraint equation

$$ka_i = kb \tag{2.16}$$

is added into equation (2.7) where $k \gg k_{ii}$ (k_{ii} is the i^{th} diagonal term of $\mathbf{K}$). Physically, a stiff spring is added at the specified degree-of-freedom, and a large force is applied so that the degree-of-freedom attains the prescribed value. In the

work presented in this thesis, k was chosen to be $k_{ii} \times 10^6$.

This chapter has briefly summarized the way in which the finite element method is used to obtain the matrix equation (2.7) that approximates the partial differential equations that govern linear solid mechanics problems. The reader is referred to standard texts such as [6,33,63] for a more detailed discussion of the technique. The essence of the method is that the problem is reduced to solving equation (2.7) for the n unknown nodal displacements, or degrees-of-freedom, $\mathbf{a}$.

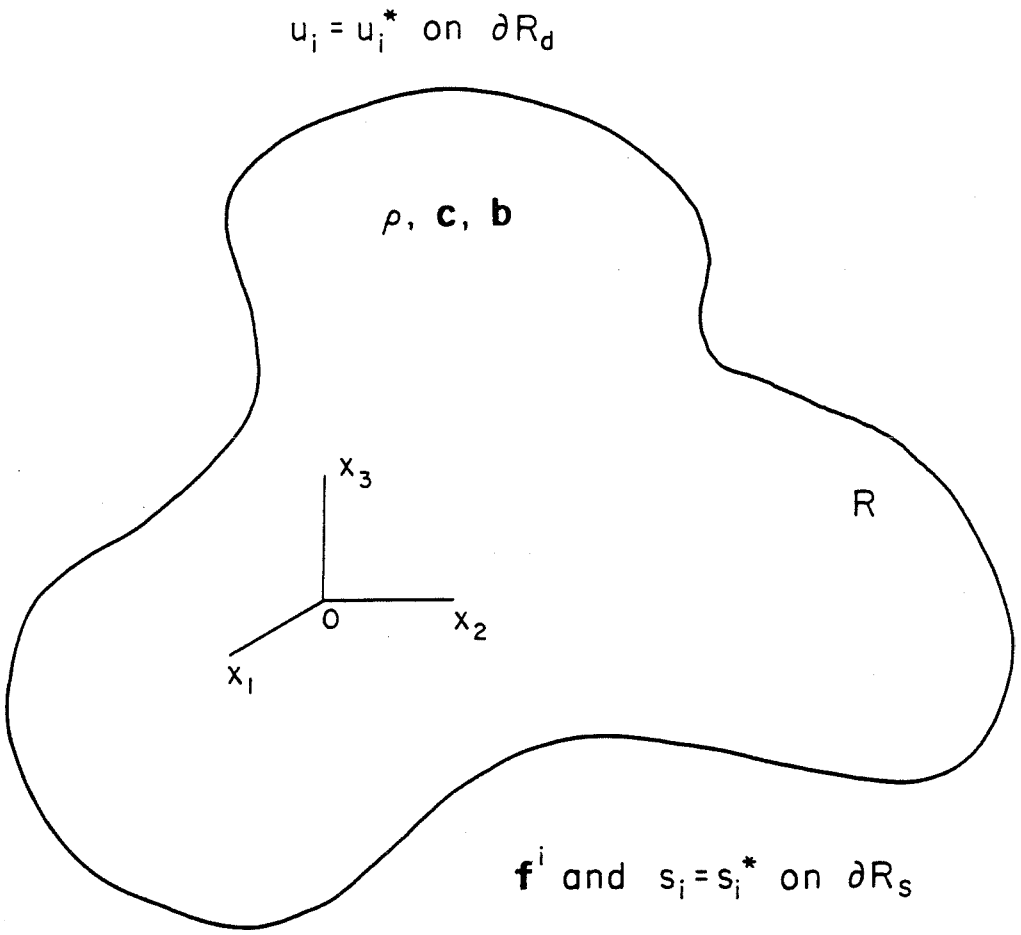


Figure 2.1: General Three Dimensional Body.

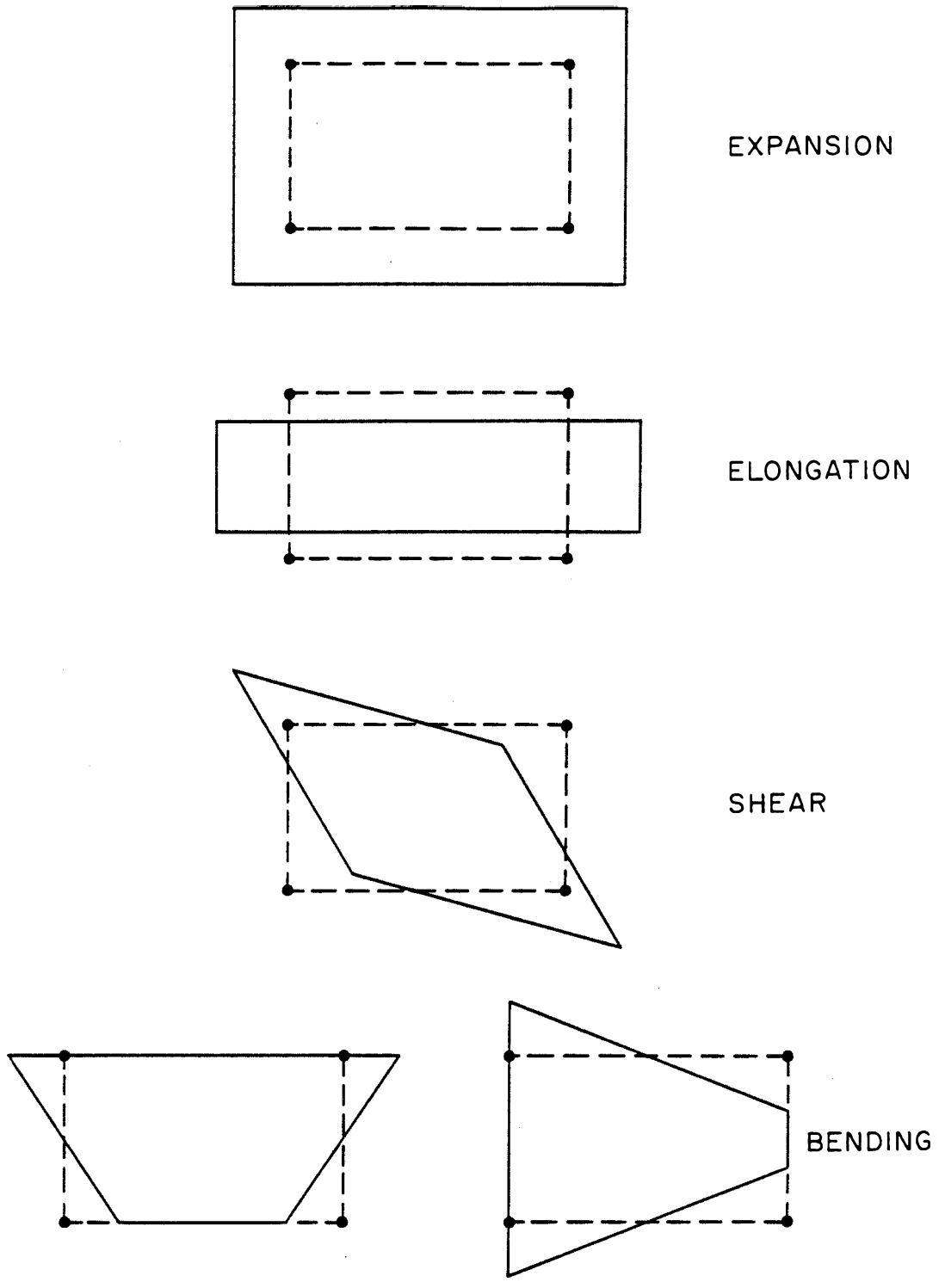


Figure 2.2: The Modes of Deformation of a Four Node Linear Quadrilateral Element.

Chapter 3

Solution of the Finite Element Equilibrium Equations

3.1 General Considerations

The way in which the finite element equilibrium equations are derived for linear solid and structural mechanics problems was outlined in chapter 2. For the static case, the equations can be written as

$$\mathbf{Ka} = \mathbf{f}. \tag{3.1}$$

In this equation, $\mathbf{K}$ is the stiffness matrix of the system, $\mathbf{a}$ is a vector of n unknown displacements, and $\mathbf{f}$ is a vector of n known forces. The original problem has been reduced to numerically solving equation (3.1) for $\mathbf{a}$.

An equation similar to equation (3.1) arises when the linear, dynamic finite element equilibrium equations given in equation (2.7) are solved. If these equations are integrated directly using Newmark's method, the matrix equation that has to be solved at each time step is given by [6]

$$\left\{\frac{1}{\Delta t^2}\mathbf{M} + \frac{\gamma}{\Delta t}\mathbf{C} + \beta\mathbf{K}\right\}\mathbf{a}(t+\Delta t) = \beta\mathbf{f}(t+\Delta t) + \left\{\frac{1}{\Delta t^2}\mathbf{M} + \frac{\gamma}{\Delta t}\mathbf{C}\right\}\mathbf{a}(t)$$

$$+ \left\{\frac{1}{\Delta t}\mathbf{M} + (\gamma - \beta)\mathbf{C}\right\}\dot{\mathbf{a}}(t) + \left\{(\frac{1}{2} - \beta)\mathbf{M} + \frac{\Delta t}{2}(\gamma - 2\beta)\mathbf{C}\right\}\ddot{\mathbf{a}}(t), \quad (3.2)$$

where $\mathbf{C}$ is the damping matrix, Δt is the time step, and β and γ are the parameters of the method. Linear dynamic problems can also be solved by means of the modal superposition technique. This involves obtaining the natural frequencies and normal modes of the body (or structure) under consideration from the equation

$$\mathbf{K}\,\boldsymbol{\phi} = \lambda \mathbf{M}\,\boldsymbol{\phi}, \quad (3.3)$$

where λ and $\boldsymbol{\phi}$ represent the eigenvalues (the square of the natural frequencies) and the eigenvectors (the normal modes) of the system, respectively. Inverse iteration is a common technique that is used to determine λ and $\boldsymbol{\phi}$ [6]. For example, if this algorithm is used to obtain the first frequency and corresponding mode of the system, a matrix equation of the form

$$\mathbf{K}\mathbf{v}^{(k+1)} = \mathbf{M}\mathbf{v}^{(k)} \quad (3.4)$$

has to be solved at each iteration step, where $\mathbf{v}^{(k)}$ and $\mathbf{v}^{(k+1)}$ are the old and new approximations to the first mode. It can be seen that both equations (3.2) and (3.4) have the same form as equation (3.1).

Although the formulation of nonlinear solid mechanics problems was not discussed in chapter 2, it is interesting to briefly discuss the solution of the resulting nonlinear matrix equations. The most frequently used solution technique is some form of Newton-Raphson iteration [6]. For example, for a nonlinear static problem, the equation that has to be solved at each Newton-Raphson iteration step is

$$\mathbf{K}_T^{(k)}\Delta\mathbf{a}^{(k)} = \mathbf{f} - \mathbf{K}(\mathbf{a}^{(k)})\mathbf{a}^{(k)}, \quad (3.5)$$

where $\mathbf{K}(\mathbf{a}^{(k)})$ is the nonlinear stiffness matrix, $\mathbf{K}_T^{(k)}$ is the tangent stiffness matrix, $\mathbf{a}^{(k)}$ is the current approximate solution, and $\Delta\mathbf{a}^{(k)}$ is the correction to $\mathbf{a}^{(k)}$. The matrix equation (3.5), which is similar to (3.1), has to be solved for $\Delta\mathbf{a}^{(k)}$.

Thus the fundamental problem that faces the engineering numerical analyst when a linear, nonlinear, static, or dynamic problem has to be solved is the efficient solution of the generic matrix equation

$$\mathbf{K}\mathbf{x} = \mathbf{f}, \tag{3.6}$$

where $\mathbf{K}$ is an $n \times n$ generalized stiffness matrix, $\mathbf{x}$ is a vector of n unknown displacements, and $\mathbf{f}$ is a vector of n known forces. Future references to the size of a problem will mean the number of unknowns, n, in equation (3.6). This chapter will present the various approaches that have been used to solve equation (3.6), and will highlight the advantages and disadvantages of each method.

There are basically two ways by which equation (3.6) can be solved: directly and iteratively. Direct methods are typically based on Gauss elimination; iterative methods, such as the Gauss-Seidel or conjugate gradient methods, methodically improve an initial guess to the solution of equation (3.6) until an acceptable solution has been obtained. It should be noted that the stiffness matrix $\mathbf{K}$ is symmetric and usually positive definite. The symmetry property allows only one half of the matrix to be stored. The matrix $\mathbf{K}$ has two other properties that result from the local nature of the finite element discretization. Since each degree-of-freedom in a finite element mesh is connected only to its nearest neighbors, $\mathbf{K}$ is sparse and, if the degrees-of-freedom are ordered appropriately, banded. Banded means that all of the nonzero terms in the matrix are confined to lie within a certain distance from the diagonal (i.e. they all lie inside the band-

width of the matrix). The term sparse means that most of the numbers within the bandwidth of $\mathbf{K}$ are zeroes; there are relatively few nonzero terms. The fact that $\mathbf{K}$ is sparse and banded will have important implications for the solution algorithms discussed in this chapter.

3.2 Direct Methods

This section gives a brief discussion of the more common class of solution algorithms: the direct methods. Variants of the method discussed here are employed in the large commercial finite element codes such as ADINA, NASTRAN, and ANSYS.

The most widely used direct methods are based on Gaussian elimination. These methods have the desirable property that the solution $\mathbf{x}$ to equation (3.6) can be obtained in an amount of time that can be predicted in advance. However, the storage requirements can be excessive. In the factorization method, $\mathbf{K}$ is split into its factors,

$$\mathbf{K} = \mathbf{L}_\pi \mathbf{D}_\pi \mathbf{U}_\pi, \tag{3.7}$$

where $\mathbf{D}_\pi$ is a diagonal matrix, and $\mathbf{L}_\pi$ and $\mathbf{U}_\pi$ are lower and upper triangular matrices, respectively, with 1's on their diagonals. Since $\mathbf{K}$ is symmetric,

$$\mathbf{L}_\pi = \mathbf{U}_\pi^T, \tag{3.8}$$

and so

$$\mathbf{K} = \mathbf{U}_\pi^T \mathbf{D}_\pi \mathbf{U}_\pi. \tag{3.9}$$

Once $\mathbf{K}$ has been factored, $\mathbf{x}$ can be obtained in the following way.

- Solve $\mathbf{U}_\pi^T \mathbf{z} = \mathbf{f}$ for $\mathbf{z}$ by a forward reduction.
- Solve $\mathbf{D}_\pi \mathbf{y} = \mathbf{z}$ for $\mathbf{y}$ by simple division.
- Solve $\mathbf{U}_\pi \mathbf{x} = \mathbf{y}$ for $\mathbf{x}$ by a back substitution.

Since $\mathbf{K}$ is symmetric, only the upper half of the matrix is required. This is usually stored in banded or skyline form. Figure 3.1 shows the differences between these two storage methods. Generally, a large number of zero terms within the bandwidth or skyline of $\mathbf{K}$ will be stored initially. This is necessary because these zeroes will be replaced by nonzero terms of the factors of $\mathbf{K}$. The storage required for the direct methods is thus nm real numbers, where m is the bandwidth of the stiffness matrix stored in banded form, or an average bandwidth of the matrix stored in skyline form.

It can be shown that the number of operations required to factor $\mathbf{K}$ is approximately $\frac{1}{2}nm^2$, and the number of operations required to solve for $\mathbf{x}$ using the three steps described above is approximately $2nm$ [6]. (One operation is defined as multiplying or dividing and adding or substracting two real numbers.) It should be noted that m is proportional to $n^{\frac{1}{2}}$ and $n^{\frac{2}{3}}$ for two and three dimensional problems with uniform meshes, respectively. Thus, for three dimensional problems, the solution time and the storage will be proportional to $n^{\frac{7}{3}}$ and $n^{\frac{5}{3}}$, respectively. It will be demonstrated in chapter 5 that these relations severely limit the size of three dimensional problems that can be solved using a direct method.

Many variations of the direct method described in this section have been developed. These include blocked methods, frontal techniques, and substructur-

ing. The reader is referred to review articles such as [42] for a more detailed discussion of these schemes. However, they still cannot escape the high computational cost of the direct method in terms of time and storage. For example, blocked methods reduce the amount of direct access memory that is required by storing the stiffness matrix on disk. This results in an increase in the amount of I/O time required to read and write the matrix to and from the disk.

3.3 Iterative Methods

The other class of algorithms that can be used to solve the linear matrix equation (3.6) are the iterative methods. The main advantage these techniques have over the direct methods is that only the nonzero terms of the stiffness matrix need to be stored. Since $\mathbf{K}$ is a sparse matrix, this represents a considerable saving in storage. However, iterative methods have a disadvantage in that it is not possible to estimate how many cycles of iteration will be required to solve a given matrix equation ahead of time. Many schemes require a large number of cycles for ill-conditioned problems, which typically arise in solid and structural mechanics. (Section 3.4 contains a discussion of the problem of ill-conditioning in these applications.) Therefore iterative methods tend to be slow, and so have been bypassed in favor of direct methods in practical applications.

Another advantage that direct methods have over iterative methods is their ability to deal with multiple right-hand sides. The most expensive part of the direct methods discussed in section 3.2 is the factorization of the stiffness matrix; the back substitution and forward reduction are the only steps that involve the force vector. Therefore only one factorization is required to analyze the behavior

of a structure that is subjected to different loads. On the other hand, if an iterative method is used, each loading would have to be treated as a separate problem; no advantage would be gained from having already solved a different problem on the same structure. This feature of direct methods makes them particularly appealing when dynamic or nonlinear problems are considered, since they essentially consist of a series of different load vectors applied to a generalized stiffness matrix.

The low storage requirements of iterative algorithms in general has fueled a great deal of research aimed at developing a fast iterative method (i.e. one that is not sensitive to the ill-conditioning of problems). An example in the area of solid and structural mechanics is the work in [30,31,32,61], which has attempted to develop an effective preconditioned conjugate gradient method. The work presented in this thesis demonstrates that the multigrid method is also a fast iterative algorithm that is not greatly affected by certain types of ill-conditioning encountered in solid mechanics.

The aim of this section is to briefly describe some of the more popular iterative methods that are used. Some of the algorithms will be used with, or compared to, the multigrid method in the following chapters. The reader is referred to [2,28,49] for more detailed discussions of the schemes discussed here.

The objective of an iterative method is: given an approximation $\mathbf{x}^{(k)}$ to the exact solution $\mathbf{x}$ of equation (3.6), reduce the residual

$$\mathbf{r}^{(k)} = \mathbf{f} - \mathbf{K}\mathbf{x}^{(k)} \tag{3.10}$$

in some way until

$$R = \frac{\| \mathbf{r}^{(k)} \|}{\| \mathbf{f} \|} \leq \varepsilon, \tag{3.11}$$

where $\| \cdot \|$ denotes the Euclidean norm of a vector and ε is the convergence tolerance. It can be shown that the error,

$$\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}, \tag{3.12}$$

and the residual are related by

$$\mathbf{K}\mathbf{e}^{(k)} = \mathbf{r}^{(k)}. \tag{3.13}$$

It is helpful to recognize that the error is a set of displacements, and that the residual is a set of forces; the residual represents the forces that produce the displacements given by the error.

3.3.1 Basic Iterative Methods

An important class of iterative algorithms are the linear stationary methods of the first degree [28], and may be expressed in the form:

$$\left.\begin{array}{ll} \text{Given:} & \mathbf{x}^{(0)},\ k = 0 \\ \text{Iterate over:} & \mathbf{x}^{(k+1)} = \mathbf{G}\mathbf{x}^{(k)} + \mathbf{k} \\ & k = k + 1 \\ \text{until convergence is achieved.} & \end{array}\right\} \tag{3.14}$$

This can also be written as

$$\left.\begin{array}{ll} \text{Given:} & \mathbf{x}^{(0)},\ k = 0 \\ \text{Iterate over:} & \mathbf{r}^{(k)} = \mathbf{f} - \mathbf{K}\mathbf{x}^{(k)} \\ & \mathbf{B}\Delta\mathbf{x}^{(k)} = \mathbf{r}^{(k)} \\ & \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)} \\ & k = k + 1 \\ \text{until convergence is achieved.} & \end{array}\right\} \tag{3.15}$$

Note that

$$\left.\begin{aligned} \mathbf{G} &= \mathbf{I} - \mathbf{B}^{-1}\mathbf{K} \\ \text{and } \mathbf{k} &= \mathbf{B}^{-1}\mathbf{f}. \end{aligned}\right\} \tag{3.16}$$

A necessary and sufficient condition for the convergence of this class of iterative methods is that

$$\rho(\mathbf{G}) < 1, \tag{3.17}$$

where $\rho(\ .\ )$ denotes the spectral radius of a matrix.

It can be seen from equations (3.14), (3.15), and (3.16) that different choices of $\mathbf{B}$ (which is called the splitting matrix) will yield different iteration algorithms. If $\mathbf{B}$ is chosen as the stiffness matrix $\mathbf{K}$, the method will converge in one iteration. However, $\mathbf{B}$ is chosen so that the matrix equation

$$\mathbf{B}\Delta\mathbf{x}^{(k)} = \mathbf{r}^{(k)} \tag{3.18}$$

is easy to solve. Various methods can be generated by considering the sum decomposition of the stiffness matrix,

$$\mathbf{K} = \mathbf{L}_\sigma + \mathbf{D}_\sigma + \mathbf{U}_\sigma \tag{3.19}$$

where $\mathbf{D}_\sigma$ is a diagonal matrix, and $\mathbf{L}_\sigma$ and $\mathbf{U}_\sigma$ are lower and upper triangular matrices, respectively, with zeroes on their diagonals. Since $\mathbf{K}$ is symmetric,

$$\mathbf{L}_\sigma = \mathbf{U}_\sigma^T. \tag{3.20}$$

Table 3.1 lists four of the basic linear stationary methods of the first degree: the Richardson, Jacobi, Gauss-Seidel, and succesive overrelaxation methods. The matrices $\mathbf{G}$ and $\mathbf{B}$, together with the convergence criteria for each method, are given in the table.

3.3.2 Gradient Methods

Another important class of iterative algorithms are those based on gradient methods. The problem of solving the linear matrix equation (3.6) is converted into one of minimizing an associated quadratic functional. It can be shown [49] that solving equation (3.6) is equivalent to minimizing the functional $\mathcal{F}(\mathbf{x}^{(k)})$ with respect to $\mathbf{x}^{(k)}$, where

$$\mathcal{F}(\mathbf{x}^{(k)}) = (\mathbf{x}^{(k)}, \mathbf{K}\mathbf{x}^{(k)}) - (\mathbf{f}, \mathbf{x}^{(k)}), \tag{3.21}$$

and (. , .) denotes the Euclidean scalar product of two vectors. Note that the gradient of $\mathcal{F}(\mathbf{x}^{(k)})$ is given by

$$\nabla\mathcal{F}(\mathbf{x}^{(k)}) = \mathbf{K}\mathbf{x}^{(k)} - \mathbf{f} = -\mathbf{r}^{(k)}. \tag{3.22}$$

The direction of $\nabla\mathcal{F}(\mathbf{x}^{(k)})$ is the direction for which $\mathcal{F}(\mathbf{x}^{(k)})$ at the point $\mathbf{x}^{(k)}$ has the greatest rate of change. This observation leads to the method of steepest descent for minimizing $\mathcal{F}(\mathbf{x}^{(k)})$, and thereby solving equation (3.6) [2,49]. An improved approximation $\mathbf{x}^{(k+1)}$ is computed by moving in the direction of $\nabla\mathcal{F}(\mathbf{x}^{(k)})$ to a point where $\mathcal{F}(\mathbf{x}^{(k+1)})$ is minimal, i.e.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)}\nabla\mathcal{F}(\mathbf{x}^{(k)}), \tag{3.23}$$

where the search parameter $\lambda^{(k)}$ is chosen to minimize $\mathcal{F}(\mathbf{x}^{(k+1)})$. Thus the method

of steepest descent is given by

$$\left.\begin{array}{ll} \text{Given:} & \mathbf{x}^{(0)},\ k=0 \\ \text{Iterate over:} & \mathbf{r}^{(k)} = \mathbf{f} - \mathbf{K}\mathbf{x}^{(k)} \\ & \lambda^{(k)} = \frac{(\mathbf{r}^{(k)},\mathbf{r}^{(k)})}{(\mathbf{r}^{(k)},\mathbf{K}\mathbf{r}^{(k)})} \\ & \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)}\mathbf{r}^{(k)} \\ & k = k+1 \\ \text{until convergence is achieved.} & \end{array}\right\} \tag{3.24}$$

The conjugate gradient method [2,49] is a variation of the method of steepest descent. The new approximate solution to equation (3.6) is given by

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)}\mathbf{p}^{(k)}, \tag{3.25}$$

where $\mathbf{p}^{(k)}$ is a direction vector. For the conjugate gradient method, $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$, and

$$\mathbf{p}^{(k)} = \mathbf{r}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k-1)},\ k \geq 1. \tag{3.26}$$

The parameter $\alpha^{(k)}$ is chosen so that $\mathbf{p}^{(k)}$ is $\mathbf{K}-$conjugate to $\mathbf{p}^{(k-1)}$, i.e.

$$(\mathbf{p}^{(k)}, \mathbf{K}\mathbf{p}^{(k-1)}) = 0. \tag{3.27}$$

As in the method of steepest descent, the search parameter $\lambda^{(k)}$ in equation (3.25)

is chosen to minimize $\mathcal{F}(\mathbf{x}^{(k+1)})$. The conjugate gradient method is

$$\left.\begin{array}{ll} \text{Given:} & \mathbf{x}^{(0)},\ k=0 \\ \text{Initialize:} & \mathbf{r}^{(0)} = \mathbf{f} - \mathbf{K}\mathbf{x}^{(0)} \\ & \mathbf{p}^{(0)} = \mathbf{r}^{(0)} \\ \text{Iterate over:} & \lambda^{(k)} = \frac{(\mathbf{r}^{(k)},\mathbf{r}^{(k)})}{(\mathbf{p}^{(k)},\mathbf{K}\mathbf{p}^{(k)})} \\ & \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)}\mathbf{p}^{(k)} \\ & \mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \lambda^{(k)}\mathbf{K}\mathbf{p}^{(k)} \\ & \alpha^{(k+1)} = \frac{(\mathbf{r}^{(k+1)},\mathbf{r}^{(k+1)})}{(\mathbf{r}^{(k)},\mathbf{r}^{(k)})} \\ & \mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} + \alpha^{(k+1)}\mathbf{p}^{(k)} \\ & k = k+1 \\ \text{until convergence is achieved.} & \end{array}\right\} \tag{3.28}$$

It can be shown [49] that the residual vectors in the conjugate gradient method form an orthogonal system. Since they belong to an $n-$dimensional vector space (and therefore can contain no more than n nonzero vectors), the residual vector must vanish after n cycles of iteration at the most. Thus the conjugate gradient method has the remarkable property that, in theory, the solution to the matrix equation (3.6) is given in, at most, n steps. However, in practice, the mutual orthogonality of the residual vectors cannot be maintained exactly. The more ill-conditioned $\mathbf{K}$ is, the greater the deviation [49].

The sensitivity of the conjugate gradient method to the condition of $\mathbf{K}$ has led to the development of various methods of preconditioning the matrix equation (3.6). Preconditioning is a technique that changes the problem of solving the ill-conditioned equation (3.6) into one of solving the well-conditioned equation

$$\hat{\mathbf{K}}\hat{\mathbf{x}} = \hat{\mathbf{f}} \tag{3.29}$$

[2]. This preconditioned system is obtained by means of a positive definite preconditioning matrix $\mathbf{C}$ such that

$$\mathbf{C} = \mathbf{E}\mathbf{E}^T. \tag{3.30}$$

The preconditioned matrix $\hat{\mathbf{K}}$ is given by

$$\hat{\mathbf{K}} = \mathbf{E}^{-1}\mathbf{K}\mathbf{E}^T; \tag{3.31}$$

also

$$\hat{\mathbf{f}} = \mathbf{E}^{-1}\mathbf{f} \tag{3.32}$$

and

$$\hat{\mathbf{x}} = \mathbf{E}^{-T}\mathbf{x}. \tag{3.33}$$

Applying the conjugate gradient method to the solution of the preconditioned equation (3.29) yields the preconditioned conjugate gradient method for the solution of the original equation (3.6).

$$\left.\begin{array}{ll}
\text{Given:} & \mathbf{x}^{(0)},\ k = 0 \\
\text{Initialize:} & \mathbf{r}^{(0)} = \mathbf{f} - \mathbf{K}\mathbf{x}^{(0)} \\
& \mathbf{d}^{(0)} = \mathbf{C}^{-1}\mathbf{r}^{(0)} \\
& \mathbf{p}^{(0)} = \mathbf{d}^{(0)} \\
\text{Iterate over:} & \lambda^{(k)} = \frac{(\mathbf{r}^{(k)},\mathbf{d}^{(k)})}{(\mathbf{p}^{(k)},\mathbf{K}\mathbf{p}^{(k)})} \\
& \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)}\mathbf{p}^{(k)} \\
& \mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \lambda^{(k)}\mathbf{K}\mathbf{p}^{(k)} \\
& \mathbf{d}^{(k+1)} = \mathbf{C}^{-1}\mathbf{r}^{(k+1)} \\
& \alpha^{(k+1)} = \frac{(\mathbf{r}^{(k+1)},\mathbf{d}^{(k+1)})}{(\mathbf{r}^{(k)},\mathbf{d}^{(k)})} \\
& \mathbf{p}^{(k+1)} = \mathbf{d}^{(k+1)} + \alpha^{(k+1)}\mathbf{p}^{(k)} \\
& k = k + 1 \\
\text{until convergence is achieved.} &
\end{array}\right\} \tag{3.34}$$

Two factors influence the choice of the preconditioning matrix $\mathbf{C}$. The preconditioned matrix $\hat{\mathbf{K}}$ should be better conditioned than the original stiffness matrix $\mathbf{K}$, and the system $\mathbf{C}\mathbf{d}^{(k)} = \mathbf{r}^{(k)}$ should be solved more efficiently than the original matrix equation $\mathbf{K}\mathbf{x} = \mathbf{f}$. The linear stationary methods of the first degree discussed in section 3.3.1 are a good source of preconditioning matrices. One of the simplest choices is to take $\mathbf{C}$ equal to the splitting matrix of the Jacobi iteration method, i.e.

$$\mathbf{C} = \mathbf{D}_\sigma. \tag{3.35}$$

This defines the Jacobi preconditioned conjugate gradient method which will be compared to the multigrid method in chapters 5, 6 and 7.

3.4 The Problem of Ill-Conditioning in Solid and Structural Mechanics

The term ill-conditioned has already been used in this thesis to describe the stiffness matrix $\mathbf{K}$, and has been included in many discussions on the solution of matrix equations [49,57,58,59]. The condition of a matrix is basically a measure of how easily the fundamental matrix equation (3.6) can be solved. Quoting Turing [58], "The expression 'ill-conditioned' is sometimes used merely as a term of abuse applicable to matrices or equations." However, it does have a well-defined mathematical meaning. This section will explain the significance of this expression in both mathematical and physical terms.

3.4.1 The Condition of a Matrix

The spectral condition number of any nonsingular matrix $\mathbf{A}$ is defined as

$$\kappa(\mathbf{A}) = \| \mathbf{A} \|^* \| \mathbf{A}^{-1} \|^*, \tag{3.36}$$

where $\| \cdot \|^*$ is the compatible matrix norm induced by the Euclidean vector norm. If $\mathbf{A}$ is symmetric and positive definite, then

$$\kappa(\mathbf{A}) = \frac{\lambda_{max}(\mathbf{A})}{\lambda_{min}(\mathbf{A})}, \tag{3.37}$$

where $\lambda_{max}(\mathbf{A})$ and $\lambda_{min}(\mathbf{A})$ are the maximum and minimum eigenvalues of $\mathbf{A}$, respectively. The matrix $\mathbf{A}$ is said to be ill-conditioned if it has a large condition number.

One of the characteristics of ill-conditioned matrix equations is that they are sensitive to small perturbations in the data: small changes in the elements of $\mathbf{K}$ or $\mathbf{f}$ can lead to large changes in the solution vector $\mathbf{x}$. This can be demonstrated in the following way [57]. Let $\boldsymbol{\phi}$ be a perturbation in $\mathbf{f}$ that causes a pertubation $\boldsymbol{\xi}$ in the solution $\mathbf{x}$, i.e.

$$\mathbf{K}(\mathbf{x} + \boldsymbol{\xi}) = \mathbf{f} + \boldsymbol{\phi}. \tag{3.38}$$

The relative error is $\frac{\|\boldsymbol{\xi}\|}{\|\mathbf{x}\|}$ in the solution and $\frac{\|\boldsymbol{\phi}\|}{\|\mathbf{f}\|}$ in the data. Since

$$\mathbf{K}\mathbf{x} = \mathbf{f}, \tag{3.39}$$

then

$$\| \mathbf{f} \| \leq \| \mathbf{K} \|^* \| \mathbf{x} \| . \tag{3.40}$$

Also, from equations (3.38) and (3.39),

$$\mathbf{K}\boldsymbol{\xi} = \boldsymbol{\phi}, \tag{3.41}$$

and so

$$\| \boldsymbol{\xi} \| \leq \| \mathbf{K}^{-1} \|^* \| \boldsymbol{\phi} \| . \tag{3.42}$$

Hence an estimate of the ratio of the relative errors in the solution and the data is given by

$$\left.\begin{aligned} \frac{\|\boldsymbol{\xi}\|}{\|\mathbf{x}\|} \Big/ \frac{\|\boldsymbol{\phi}\|}{\|\mathbf{f}\|} &= \frac{\|\boldsymbol{\xi}\|}{\|\boldsymbol{\phi}\|} \cdot \frac{\|\mathbf{f}\|}{\|\mathbf{x}\|} \\ &\leq \| \mathbf{K}^{-1} \|^* \| \mathbf{K} \|^* \\ &= \kappa(\mathbf{K}). \end{aligned}\right\} \tag{3.43}$$

Thus a small change in $\mathbf{f}$ can result in a large change in $\mathbf{x}$ if the matrix $\mathbf{K}$ is ill-conditioned. It should be noted that the expression given in equation (3.43) is a worst case estimate, which may be overly conservative in some cases.

3.4.2 Applications to the Solution of Matrix Equations

The spectral condition number has two important implications for the solution of matrix equations. The accuracy of the solution obtained by any numerical algorithm and the speed of iterative methods (i.e. the number of iterations required for convergence to a given tolerance) both depend on the condition of the matrix.

The accuracy of a numerically obtained solution to $\mathbf{Kx} = \mathbf{f}$ can be determined in the following way [49]. The solution of the matrix equation cannot be evaluated more accurately than is permitted in the calculation of $\mathbf{f} - \mathbf{Kx}'$ for a given vector $\mathbf{x}'$ in the vicinity of the exact solution $\mathbf{x}$ (since the equation is considered solved if the residual is zero). Consider a perturbation $\delta\mathbf{x}$ of the exact solution $\mathbf{x}$ such that $\| \delta\mathbf{x} \|$ is one unit in the last significant figure of the largest component (in absolute value) of $\mathbf{x}$. It should be noted that $\mathbf{x} + \delta\mathbf{x}$ is the largest perturbed vector that can be considered an exact solution to within the

computational accuracy of the machine. A vector $\mathbf{x}'$ can be considered a solution to $\mathbf{Kx} = \mathbf{f}$ to within the machine's accuracy if

$$\left.\begin{aligned} \| \mathbf{f} - \mathbf{K}\mathbf{x}' \| &\leq \| \mathbf{f} - \mathbf{K}(\mathbf{x} + \delta\mathbf{x}) \| \\ &= \| \mathbf{K}\delta\mathbf{x} \| \\ &\leq \| \mathbf{K} \|^* \| \delta\mathbf{x} \| . \end{aligned}\right\} \tag{3.44}$$

This can be written as

$$\| \mathbf{K}(\mathbf{x} - \mathbf{x}') \| \leq \| \mathbf{K} \|^* \| \delta\mathbf{x} \| . \tag{3.45}$$

The objective is to estimate $\| \mathbf{x} - \mathbf{x}' \|$. Using equation (3.45)

$$\left.\begin{aligned} \| \mathbf{x} - \mathbf{x}' \| &= \| \mathbf{K}^{-1}\mathbf{K}(\mathbf{x} - \mathbf{x}') \| \\ &\leq \| \mathbf{K}^{-1} \|^* \| \mathbf{K}(\mathbf{x} - \mathbf{x}') \| \\ &\leq \| \mathbf{K}^{-1} \|^* \| \mathbf{K} \|^* \| \delta\mathbf{x} \| \\ &= \kappa(\mathbf{K}) \| \delta\mathbf{x} \| . \end{aligned}\right\} \tag{3.46}$$

Equation (3.46) means that any vector $\mathbf{x}'$ can be considered a solution to the original equation if it differs from the exact solution $\mathbf{x}$ by $\kappa(\mathbf{K})$ units in the last significant figure of the largest absolute component of $\mathbf{x}$. This implies that in solving the equation $\mathbf{Kx} = \mathbf{f}$ by any numerical method, it is possible to have an error of $\log_{10} \kappa(\mathbf{K})$ significant figures of the largest component of the solution vector, and to have this error in all of the components.

The condition of a matrix therefore plays an important role in determining the accuracy of a numerically obtained solution to $\mathbf{Kx} = \mathbf{f}$. The condition number also has important implications on the performance of iterative methods for solving the matrix equation. For example, it can be shown that, for the three gradient methods discussed in section 3.3.2, $p(\varepsilon)$, the smallest number of iterations k required to reduce the initial error by a factor of ε with respect to the

energy norm $\| \mathbf{x} \|_{\mathbf{K}} = (\mathbf{x}, \mathbf{K}\mathbf{x})^{\frac{1}{2}}$, i.e. k such that

$$\| \mathbf{e}^{(k)} \|_{\mathbf{K}} \leq \varepsilon \| \mathbf{e}^{(0)} \|_{\mathbf{K}}, \tag{3.47}$$

is as follows [2].

$$\left.\begin{array}{ll} \text{Steepest Descent:} & p(\varepsilon) \leq \frac{1}{2}\kappa(\mathbf{K}) \ln \frac{1}{\varepsilon} + 1, \\ \text{Conjugate Gradient:} & p(\varepsilon) \leq \frac{1}{2}\sqrt{\kappa(\mathbf{K})} \ln \frac{2}{\varepsilon} + 1, \\ \text{Preconditioned Conjugate Gradient:} & p(\varepsilon) \leq \frac{1}{2}\sqrt{\kappa(\hat{\mathbf{K}})} \ln \frac{2}{\varepsilon} + 1, \end{array}\right\} \tag{3.48}$$

where $\hat{\mathbf{K}}$ is the preconditioned matrix, see equation (3.31). Therefore, an ill-conditioned system may require more iterations, and hence a larger amount of computational effort, than a well-conditioned system.

3.4.3 Physical Identification of Ill-Conditioned Systems

It is apparent from the above discussion that the condition of a matrix is a well defined mathematical quantity that plays an important role in the solution of $\mathbf{K}\mathbf{x} = \mathbf{f}$. However, the computation of the spectral condition number from equation (3.37) is, in general, prohibitively expensive. It is therefore important that the engineering numerical analyst should have an understanding of the physical causes of ill-conditioning. Equation (3.37) states that the condition number is the ratio of the largest eigenvalue of $\mathbf{K}$ to the smallest. Recalling that $\mathbf{K}$ is a generalized stiffness matrix, the smallest eigenvalue corresponds to the stiffness of the softest mode of deformation of the mesh, while the largest eigenvalue is generally well approximated by the stiffness of the stiffest mode of deformation of the stiffest element. This physical interpretation permits some feeling for the condition number of $\mathbf{K}$ to be obtained.

One of the more common sources of ill-conditioning in solid mechanics is the use of small elements. Physically, as the typical element size is reduced (i.e. as the number of elements in the mesh is increased), the stiffness of the individual elements increases. This means that the largest eigenvalue of the mesh will also increase. However, the smallest eigenvalue will remain largely unchanged. Thus the condition number of the stiffness matrix will increase. This dependence on the element size has been treated more mathematically in [2,49].

Conditions of near incompressibility in three dimensional and plane strain linear elasticity also give rise to ill-conditioned solid mechanics problems. As the material becomes more incompressible, the lowest eigenvalue of $\mathbf{K}$ may not change substantially; however, the stiffness of the expansion mode of deformation of the individual elements will increase dramatically (see section 2.3). Another form of ill-conditioning in solid mechanics arises when the problem to be analyzed consists of materials that have widely different stiffnesses. This case includes that of a rigid inclusion.

The problem of ill-conditioning is severe in the area of structural mechanics. Finite element meshes consisting of beams produce ill-conditioned matrix equations due to the large stiffness of the axial mode of deformation of the elements and the relatively lower stiffnesses of, say, the swaying mode of a frame structure. A similar effect is seen with meshes comprised of shell elements.

When the multigrid method is used to solve ill-conditioned problems in solid mechanics in later chapters of this thesis, its performance will be assessed by using the physical interpretation of ill-conditioning described in this section. Physical reasoning will also be used to explain the performance of the method

when it is applied to problems that are not ill-conditioned in the conventional sense.

Method	G	B	Convergence Criteria [28] (necessary and sufficient)
Richardson	$\mathbf{I}-\mathbf{K}$	$\mathbf{I}$	$\rho(\mathbf{K}) < 2$
Jacobi	$\mathbf{I}-\mathbf{D}_\sigma^{-1}\mathbf{K}$	$\mathbf{D}_\sigma$	$\rho(\mathbf{I}-\mathbf{D}_\sigma^{-1}\mathbf{K}) < 1$
Gauss-Seidel	$-(\mathbf{I}+\mathbf{D}_\sigma^{-1}\mathbf{L}_\sigma)^{-1}\mathbf{D}_\sigma^{-1}\mathbf{U}_\sigma$	$\mathbf{L}_\sigma+\mathbf{D}_\sigma$	Always converges since $\mathbf{K}$ is symmetric and positive definite
Successive Overrelaxation (ω = relaxation factor)	$(\mathbf{I}+\omega\mathbf{D}_\sigma^{-1}\mathbf{L}_\sigma)^{-1}\times$ $((1-\omega)\mathbf{I}-\omega\mathbf{D}_\sigma^{-1}\mathbf{U}_\sigma)$	$\mathbf{L}_\sigma+\frac{1}{\omega}\mathbf{D}_\sigma$	$0 < \omega < 2$ for $\mathbf{K}$ symmetric and positive definite

Table 3.1: Some of the Linear Stationary Methods of the First Degree.

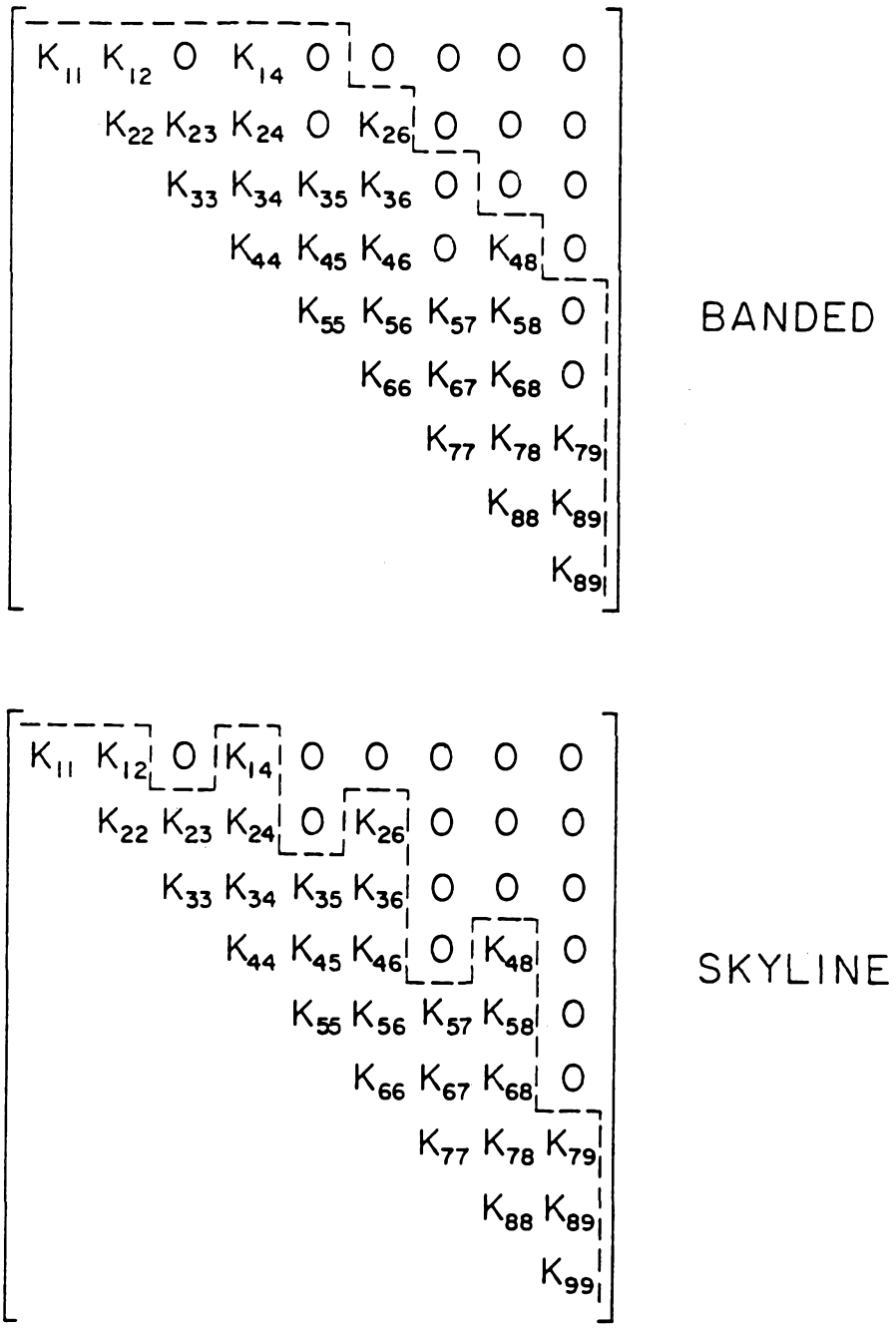


Figure 3.1: Banded and Skyline Storage Schemes for a Symmetric Matrix.

Chapter 4

The Multigrid Method for Solving Linear Matrix Equations

4.1 Introduction and Historical Background

In this chapter, a multigrid method is described that can be used to solve the linear matrix equation (1.1). Before proceeding with a detailed discussion of the algorithm, it is interesting to briefly comment on its development and previous applications. More detailed historical accounts can be found in [27,56]. The development of multigrid methods can be regarded as part of the wider quest to find efficient iterative algorithms. Southwell [51,52] was one of the first researchers to attempt to improve the basic Jacobi and Gauss-Seidel iteration (or relaxation) schemes; special block and group relaxation methods were developed which involved simultaneously relaxing small groups of unknowns. Fedorenko [20] and Bachvalov [3] were the first authors to propose a true multigrid algorithm that explicitly incorporated a coarse grid. However, it was not until Brandt published his extensive paper [12] that the true efficiency of the method was understood. Brandt's important contributions include the introduction of multigrid methods

for nonlinear problems, adaptive techniques, and the theoretical tool of local Fourier analysis. Since then, the numbers of papers, conferences, and books on the general subject has been growing at an increasing rate.

For the purposes of this discussion, the current multigrid literature can be separated into two broad groups: theoretical development and practical applications. Theoretical work, such as [27], has resulted in rather intricate mathematical theories concerning the general convergence of the method. These studies have generally examined the behavior of the algorithm when it is applied to the finite difference solution of elliptic equations in rectangular domains. The applications of these results to the field of solid and structural mechanics is limited. However, the subsequent algorithm development, together with the main result that the computational work required to solve a given problem is only linearly proportional to the number of unknowns, are sufficient to motivate research into the application of the multigrid method in this area.

Most of the published applications of the multigrid method arise from the field of fluid mechanics. In particular, potential and Euler flows, as well as problems governed by the full Navier-Stokes equations, have been solved [13,17,48]. However, the applications dealing explicitly with solid and structural mechanics problems are conspicuous by their absence. The only paper at present known to the author is [5], in which some fairly small (about 2,000 degrees-of-freedom) two dimensional plane stress, plane strain, and axisymmetric linear elastic problems were solved using the finite element method. Other authors have also briefly mentioned the application of the multigrid method to the solution of solid and structural mechanics problems. An application to membrane and plate problems

is given in [11], as well as a discussion of the effect of locking on the performance of the method. The eigenvalues and eigenvectors of a rectangular plate were obtained using the multigrid method in [24].

There is still a great deal of work to be done on the application of the multigrid method to the solution of solid and structural mechanics problems of engineering interest. The remainder of this chapter will describe in detail a multigrid algorithm that can be used to solve the linear matrix equation (1.1) that arises from the finite element discretization of a linear solid mechanics problem.

4.2 The Multigrid Algorithm

The essence of the multigrid method is that iterative (or relaxation) methods are very efficient in reducing the high frequency error component in the approximate solution, but are very slow in reducing the low frequency component. This enables the remaining error (which is smooth after only a few cycles of relaxation) to be approximated by a direct solution of a coarse mesh equivalent of equation (1.1). The low frequency error may be thought of as the component that can be represented on a coarse mesh, which is chosen so that the coarse equivalent of equation (1.1) can be solved cheaply.

The smoothing effect of Gauss-Seidel relaxation on the error can be demonstrated by means of the simple two dimensional point load problem shown in figure 4.1. This problem was solved using the finite element method; the rectangular region was discretized with four node quadrilateral elements, and the resulting matrix equation was solved using Gauss-Seidel relaxation. Figure 4.2 shows the

initial error ($\mathbf{e}^{(0)}$), and the error after 10, 50, and 100 cycles of relaxation ($\mathbf{e}^{(10)}$, $\mathbf{e}^{(50)}$, and $\mathbf{e}^{(100)}$, respectively). It was noted in section 3.3 that the error is a set of displacements, and so the error is plotted as a deformed finite element mesh. The undeformed mesh is shown in broken lines. The initial guess to the solution of the problem was taken as the null vector (i.e. $\mathbf{x}^{(0)} = \mathbf{0}$); therefore, from equation (3.12), the initial error is equal to the exact solution (i.e. $\mathbf{e}^{(0)} = \mathbf{x}$). It is clear from figure 4.2 that Gauss-Seidel relaxation is very effective in reducing the high frequency error component, whereas it is relatively slow in eliminating the remaining low frequency error.

The multigrid algorithm used in this thesis is based on the one given in [56]. To illustrate the method, consider a two mesh (or grid) arrangement. Quantities on the fine and coarse meshes (which will be defined later) are denoted by subscripts f and c, respectively. A single multigrid cycle for the solution of equation (1.1) on the fine mesh is as follows.

- Starting with the current approximation $\mathbf{x}_f^{(k)}$, perform ν_1 relaxation cycles on the fine mesh to obtain $\bar{\mathbf{x}}_f^{(k)}$; ν_1 is chosen so that the high frequency component of the error associated with $\mathbf{x}_f^{(k)}$ is significantly reduced. This results in a smooth error (i.e. an error that can be represented on a coarse mesh).

- Restrict the fine mesh residual onto the coarse mesh to obtain the coarse mesh residual

$$\mathbf{r}_c^{(k)} = \mathcal{I}_f^c \mathbf{r}_f^{(k)}, \tag{4.1}$$

where $\mathcal{I}_f^c$ is the fine to coarse mesh restriction operator.

- Compute the coarse mesh correction, $\Delta \mathbf{x}_c^{(k)}$, by solving

$$\mathbf{K}_c \Delta \mathbf{x}_c^{(k)} = \mathbf{r}_c^{(k)}, \tag{4.2}$$

  where $\mathbf{K}_c$ is the coarse mesh stiffness matrix. Since the error associated with $\bar{\mathbf{x}}_f^{(k)}$ is smooth, $\Delta \mathbf{x}_c^{(k)}$ will be a good approximation to this error.

- Interpolate $\Delta \mathbf{x}_c^{(k)}$ from the coarse mesh to the fine mesh to obtain the fine mesh correction

$$\Delta \mathbf{x}_f^{(k)} = \mathcal{I}_c^f \Delta \mathbf{x}_c^{(k)}, \tag{4.3}$$

  where $\mathcal{I}_c^f$ is the coarse to fine mesh interpolation operator.

- Perform ν_2 cycles of relaxation starting with the new fine mesh approximation

$$\hat{\mathbf{x}}_f^{(k)} = \bar{\mathbf{x}}_f^{(k)} + \Delta \mathbf{x}_f^{(k)}, \tag{4.4}$$

  to obtain $\mathbf{x}_f^{(k+1)}$, the new approximation to the solution of equation (1.1). The parameter ν_2 is chosen so that any high frequency error introduced by the interpolation is significantly reduced.

The two mesh algorithm requires the solution of the matrix equation (4.2) on the coarse mesh. This can be done either by using a direct solution on the coarse mesh, or, if this is too expensive, by introducing a still coarser mesh and using one or more cycles of the two mesh algorithm itself as the solution method. Thus a sequence of coarse meshes can be used to solve equation (1.1) on the finest mesh. The parameter γ will denote the number of multigrid cycles used on each coarse mesh to solve equation (4.2) for the coarse mesh correction. Figure 4.3 shows the structure of a single multigrid cycle with different numbers of meshes

and various values of γ. For obvious reasons, the cycles with $\gamma = 1$ and 2 are called V and W-cycles, respectively.

The effectiveness of the multigrid algorithm can be demonstrated by using it to solve the point load problem shown in figure 4.1. The fine and the coarse meshes shown in figure 4.4 were used in a two mesh method to solve this problem. Gauss-Seidel relaxation was used to smooth the error on the fine mesh; ν_1 and ν_2 were both chosen to be 5. Figure 4.5 shows the behavior of the error on the fine mesh at various stages of one multigrid cycle. The initial error on the fine mesh, $\mathbf{e}_f^{(0)}$, is shown (which is, once again, equal to the exact solution $\mathbf{x}_f$, since $\mathbf{x}_f^{(0)} = \mathbf{0}$), together with the error after 5 (ν_1) cycles of Gauss-Seidel relaxation on the fine mesh ($\bar{\mathbf{e}}_f^{(0)}$), the fine mesh error after the coarse mesh correction had been computed and interpolated to the fine mesh ($\hat{\mathbf{e}}_f^{(0)}$), and the fine mesh error after a further 5 (ν_2) cycles of relaxation ($\mathbf{e}_f^{(1)}$). The effectiveness of the use of relaxation on the fine mesh to reduce the high frequency error, combined with the use of the coarse mesh to capture the low frequency error, is apparent.

The components of the multigrid method (the relaxation method, the coarsening procedure, the restriction operator, the interpolation operator, and the coarse mesh equations) are now discussed.

4.2.1 The Relaxation Method

The function of the relaxation method is to quickly reduce the high frequency error component. Three relaxation schemes are used in this thesis: Gauss-Seidel (GS), successive overrelaxation (SOR), and Jacobi preconditioned conjugate gradient (JCG). These methods were discussed in chapter 3.

4.2.2 The Coarsening Procedure

The three types of finite elements used in this thesis are the eight node linear brick, the four node linear quadrilateral, and the three node linear triangle. One way to define a coarse mesh for these elements is to simply eliminate alternate nodes from the fine mesh by applying constraints to the fine mesh degrees-of-freedom. Figures 4.6, 4.7, and 4.8 show how this is done for the brick, quadrilateral, and triangular elements, respectively. The constraint equations take the following form: for the brick elements

$$\left.\begin{aligned} x_f^a &= x_c^a, \\ x_f^i &= (x_c^a + x_c^b)/2, \\ x_f^j &= (x_c^a + x_c^b + x_c^c + x_c^d)/4, \\ x_f^k &= (x_c^a + x_c^b + x_c^c + x_c^d + x_c^e + x_c^f + x_c^g + x_c^h)/8; \end{aligned}\right\} \tag{4.5}$$

for the quadrilateral elements

$$\left.\begin{aligned} x_f^a &= x_c^a, \\ x_f^e &= (x_c^a + x_c^b)/2, \\ x_f^f &= (x_c^a + x_c^b + x_c^c + x_c^d)/4; \end{aligned}\right\} \tag{4.6}$$

for the triangle elements

$$\left.\begin{aligned} x_f^a &= x_c^a, \\ x_f^d &= (x_c^a + x_c^b)/2. \end{aligned}\right\} \tag{4.7}$$

In the above, x_f^α denotes a fine mesh degree-of-freedom at node α, and x_c^β denotes the corresponding coarse mesh degree-of-freedom at node β.

It should be noted that the constraints given in equations (4.5), (4.6), and (4.7) assume that the fine mesh nodes not in the coarse mesh are approximately midway between the adjacent coarse mesh nodes. For example, in figure 4.6, node

i is roughly midway between nodes a and b, and node j is approximately midway between nodes a, b, c, and d. Otherwise, the coefficients in equations (4.5), (4.6), and (4.7) would have to be different.

4.2.3 The Interpolation Operator

The interpolation operator, $\mathcal{I}_c^f$, transfers displacements from the coarse mesh to the fine mesh. The natural choice for $\mathcal{I}_c^f$ comes from consideration of the coarsening procedure. Equation (4.3) then becomes

$$\Delta \mathbf{x}_f^{(k)} = \mathbf{T} \Delta \mathbf{x}_c^{(k)}, \tag{4.8}$$

where $\mathbf{T}$ is an interpolation matrix that can be conveniently assembled from coarse mesh element expressions of the form of equations (4.5), (4.6), and (4.7).

4.2.4 The Restriction Operator

The restriction operator, $\mathcal{I}_f^c$, transfers forces from the fine mesh to the coarse mesh. Since forces on the fine mesh must do the same work when they are restricted to the coarse mesh, it can be shown using equation (4.8) that

$$\mathbf{r}_c^{(k)} = \mathbf{T}^T \mathbf{r}_f^{(k)}, \tag{4.9}$$

which defines $\mathcal{I}_f^c$.

4.2.5 The Coarse Mesh Equations

Equations (4.8) and (4.9) are sufficient to determine the coarse mesh equations as

$$\mathbf{T}^T\mathbf{K}_f\mathbf{T}\Delta\mathbf{x}_c^{(k)} = \mathbf{r}_c^{(k)}. \tag{4.10}$$

By comparison with equation (4.2), the coarse mesh stiffness matrix can be directly computed from

$$\mathbf{K}_c = \mathbf{T}^T\mathbf{K}_f\mathbf{T}. \tag{4.11}$$

The matrix $\mathbf{K}_c$ is banded and sparse since $\mathbf{T}$ is defined by equations (4.5), (4.6), and (4.7). The coarse mesh stiffness matrix in equation (4.11) is identical to the matrix produced by assembling element matrices from the coarse mesh which are formed from four (figures 4.7 and 4.8) or eight (figure 4.6) fine mesh elements using the constraint equations (4.5), (4.6), and (4.7). Envisaging $\mathbf{K}_c$ as being assembled from coarse mesh elements is convenient since it reveals a deficiency in the way in which the coarse elements are formed. Namely, the advantages of reduced integration are lost on the coarse meshes (see section 2.3). To remedy this situation, an alternative procedure is also employed in this thesis in which the coarse element stiffness matrices are computed by integration in the usual way (with the option of reduced integration), and then assembled to form $\mathbf{K}_c$. A possible disadvantage of this method is the lack of consistent interpolation and restriction operators. However, the previously defined interpolation and restriction operators will be used for both methods of formation of $\mathbf{K}_c$ in this thesis.

4.2.6 Treatment of the Boundary Conditions

The way in which specified displacements and forces are imposed using the displacement based finite element method was discussed in section 2.4. Specified forces present no difficulties when a multigrid method is used to solve for the unknown displacements. Specified displacements are usually dealt with on the element level, resulting in a contribution to the nodal forces. The degrees-of-freedom associated with these displacements are then eliminated from the matrix equation (1.1). As a consequence of the interpolation of displacements between the coarse and the fine meshes, it is more convenient to keep these degrees-of-freedom as unknowns in equation (1.1). This is done by adding a stiff spring and applying a large force to the specified degree-of-freedom so that it attains the specified value (see section 2.4).

4.2.7 Computational Aspects

It is demonstrated in this thesis that, for a particular problem as mesh refinement takes place, the multigrid method can solve the matrix equation (1.1) in a time and with a storage requirement that are both proportional only to the number of unknowns in the finest mesh, n. It is therefore important that the effort required to calculate the coarse mesh stiffness and the interpolation and restriction matrices also be proportional to n. The storage required by all the matrices is kept proportional to n by storing only their nonzero terms (and only the upper half of the symmetric stiffness matrices). The interpolation matrices are assembled on an element level, in a computation time proportional to n. The computation of the coarse mesh stiffness matrices using equation (4.11) in a time that is proportional

to n and with minimal storage requires a rather involved algorithm that is not discussed in detail here.

The computer used in the studies presented in this thesis is a VAX 11/750; all of the computations were performed using the machine's virtual memory. Eight byte double precision storage was used for all of the real numbers; integers were stored as four byte numbers. An iterative method was assumed to have converged when the Euclidean norm of the residual on the finest mesh, $\mathbf{r}^{(k)}$, was 10^{-6} of the norm of the original force vector, $\mathbf{f}$ (i.e. ε in equation (3.11) was 10^{-6}). The initial approximation to the solution on the finest mesh, $\mathbf{x}^{(0)}$, was taken as $\mathbf{0}$.

4.2.8 Multigrid Notation

The following notation is used in order to easily identify the different versions of the multigrid algorithm. A multigrid method that uses coarse mesh stiffness matrices computed using equation (4.11) is denoted by MG_1; a method using assembled coarse mesh stiffness matrices is denoted by MG_2. The relaxation method is identified in the manner given in section 4.2.1. For example, the multigrid method identified as MG_1-GS means that the coarse mesh stiffness matrices were computed using equation (4.11), and that Gauss-Seidel relaxation was used.

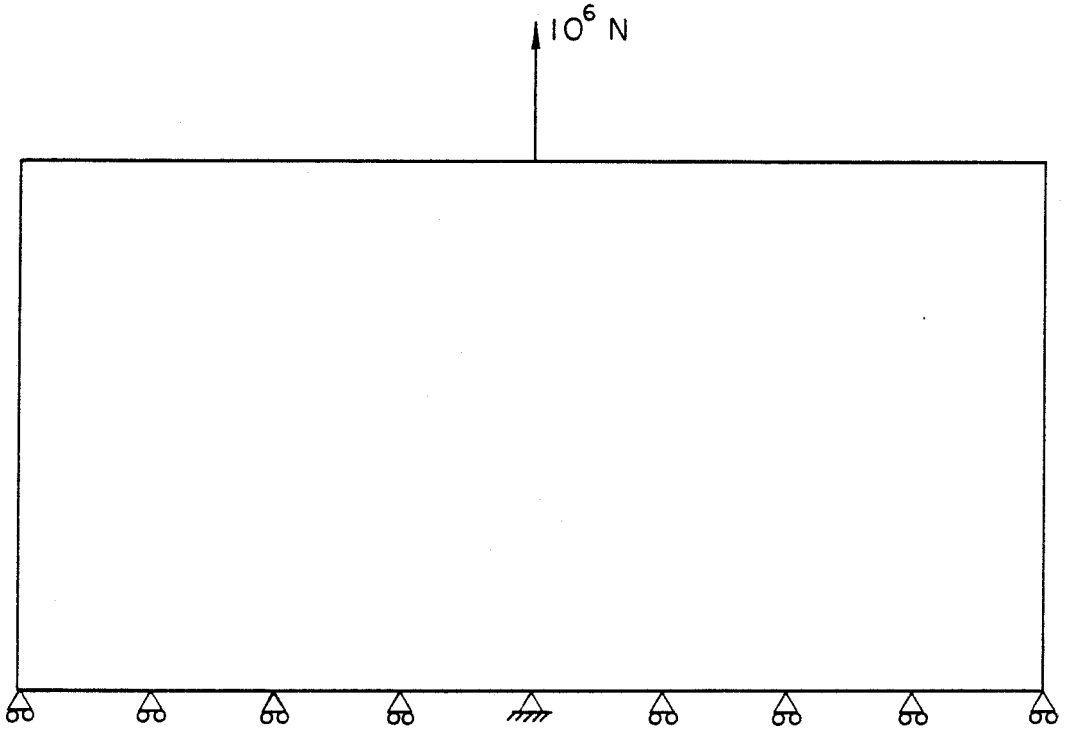


Figure 4.1: Two Dimensional Point Load Problem.

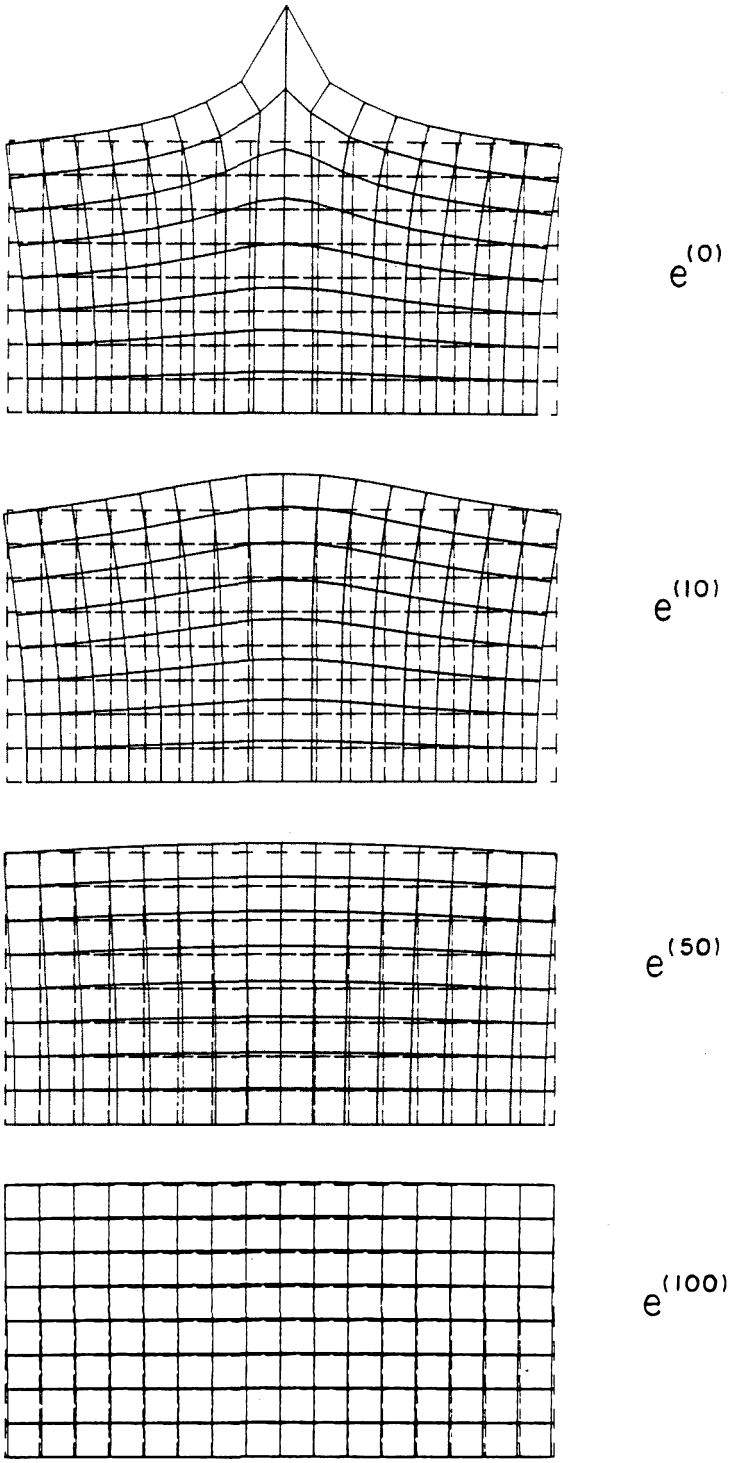


Figure 4.2: Behavior of the Error in the Gauss-Seidel Relaxation Method for the Two Dimensional Point Load Problem.

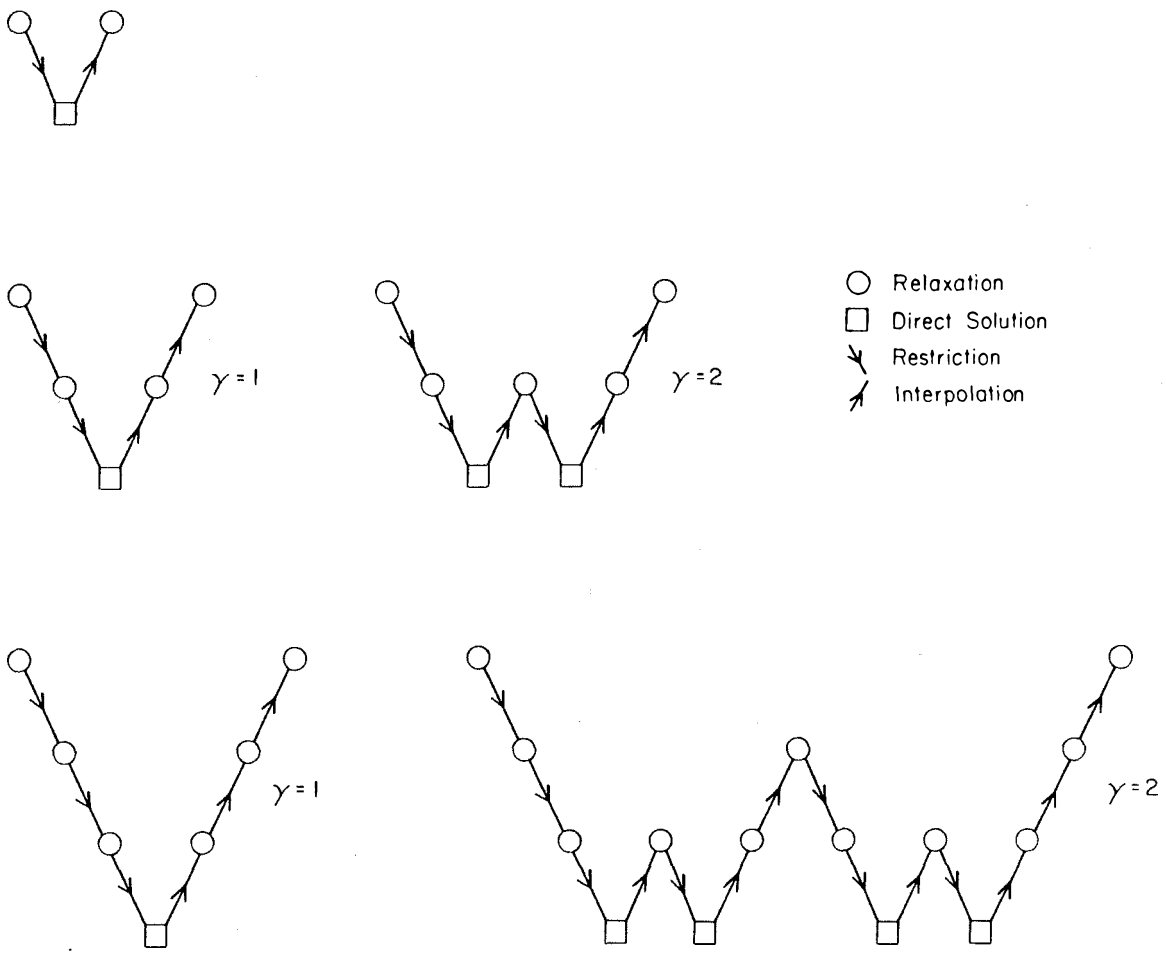
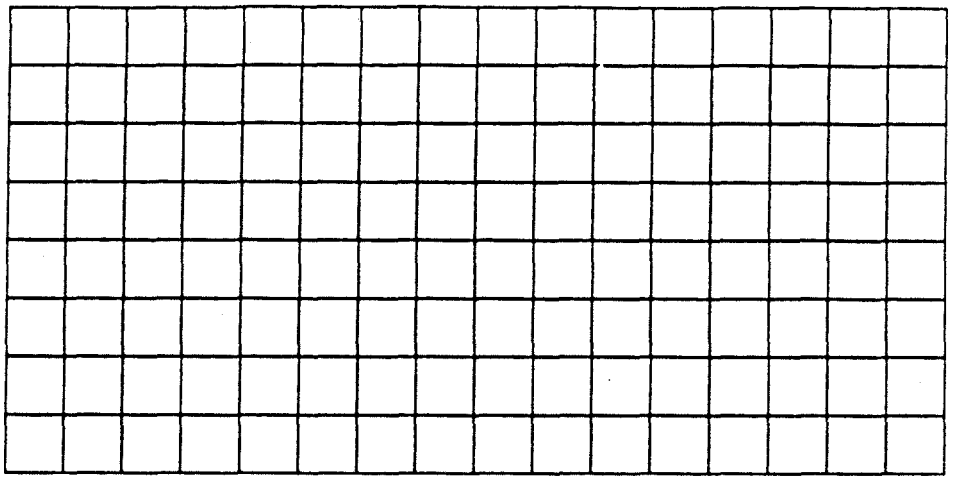
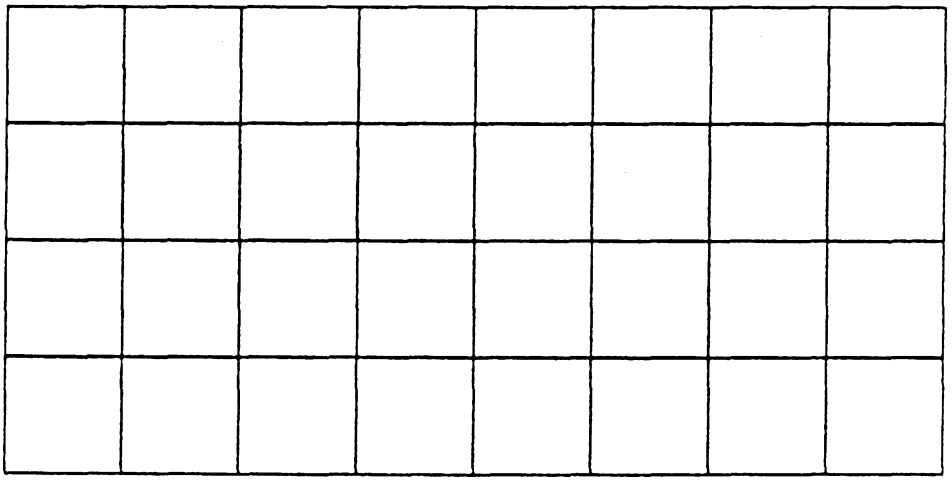


Figure 4.3: Structure of One Multigrid Cycle With Different Numbers of Meshes and Various Values of γ.



(a) Fine Mesh



(b) Coarse Mesh

Figure 4.4: The Fine and Coarse Meshes Used in the Solution of the Two Dimensional Point Load Problem.

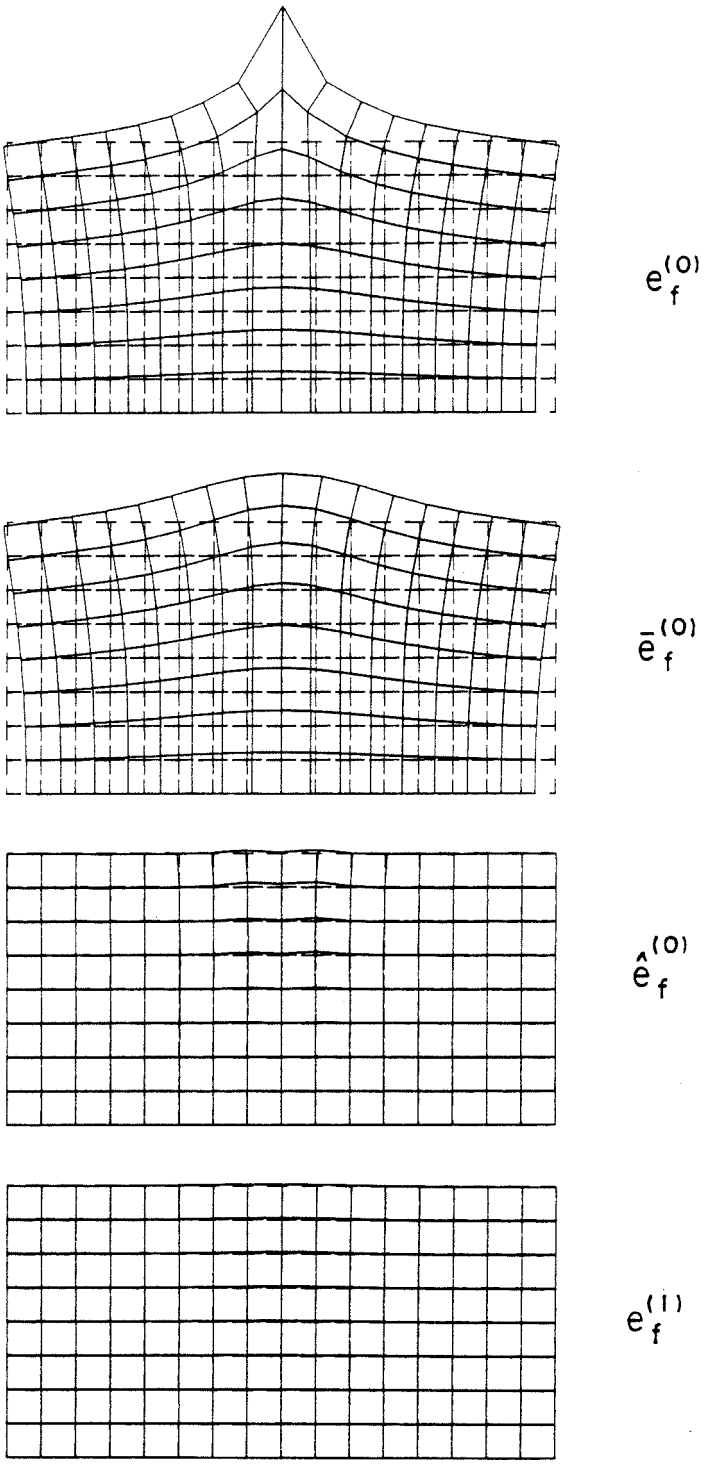


Figure 4.5: Behavior of the Fine Mesh Error in the Multigrid Method for the Two Dimensional Point Load Problem (MG_1-GS Scheme, $\nu_1 = \nu_2 = 5$).

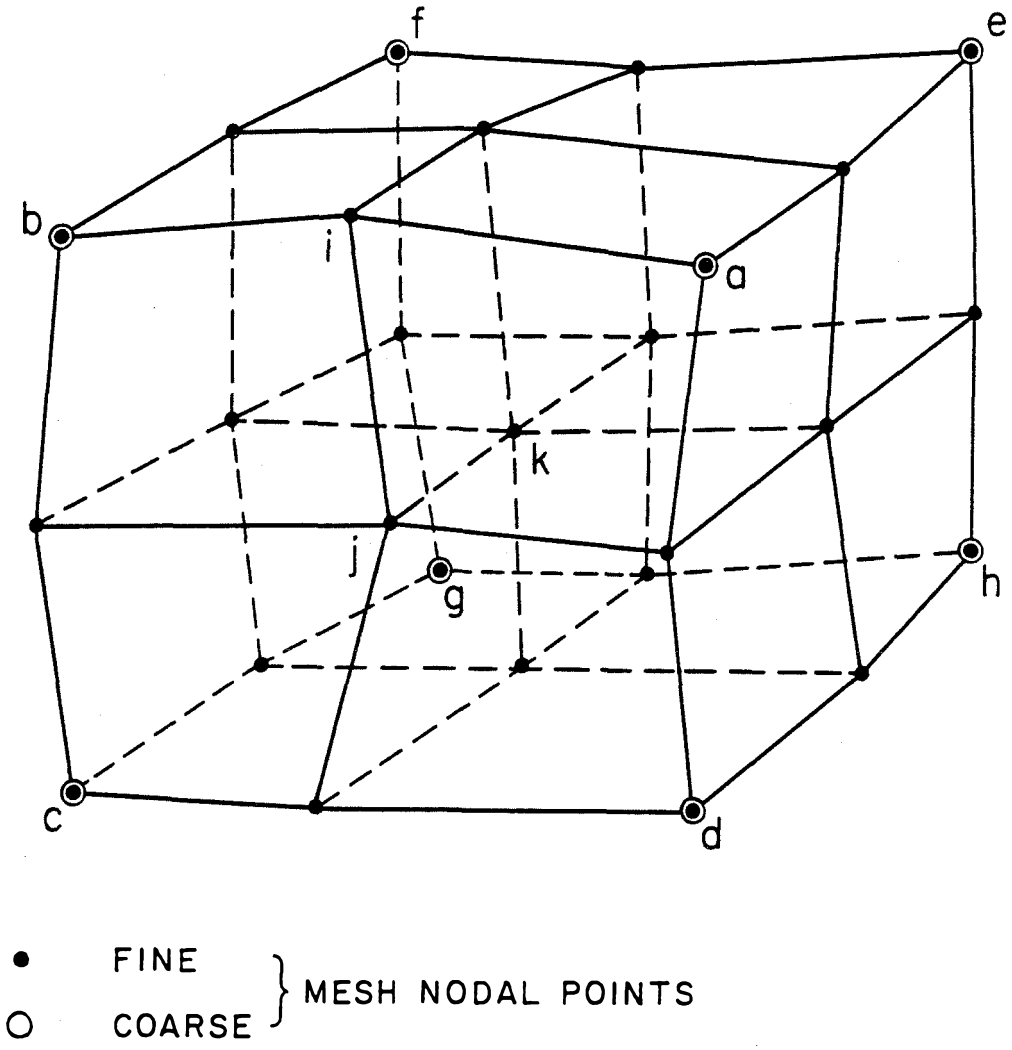


Figure 4.6: Coarsening for the Brick Elements.

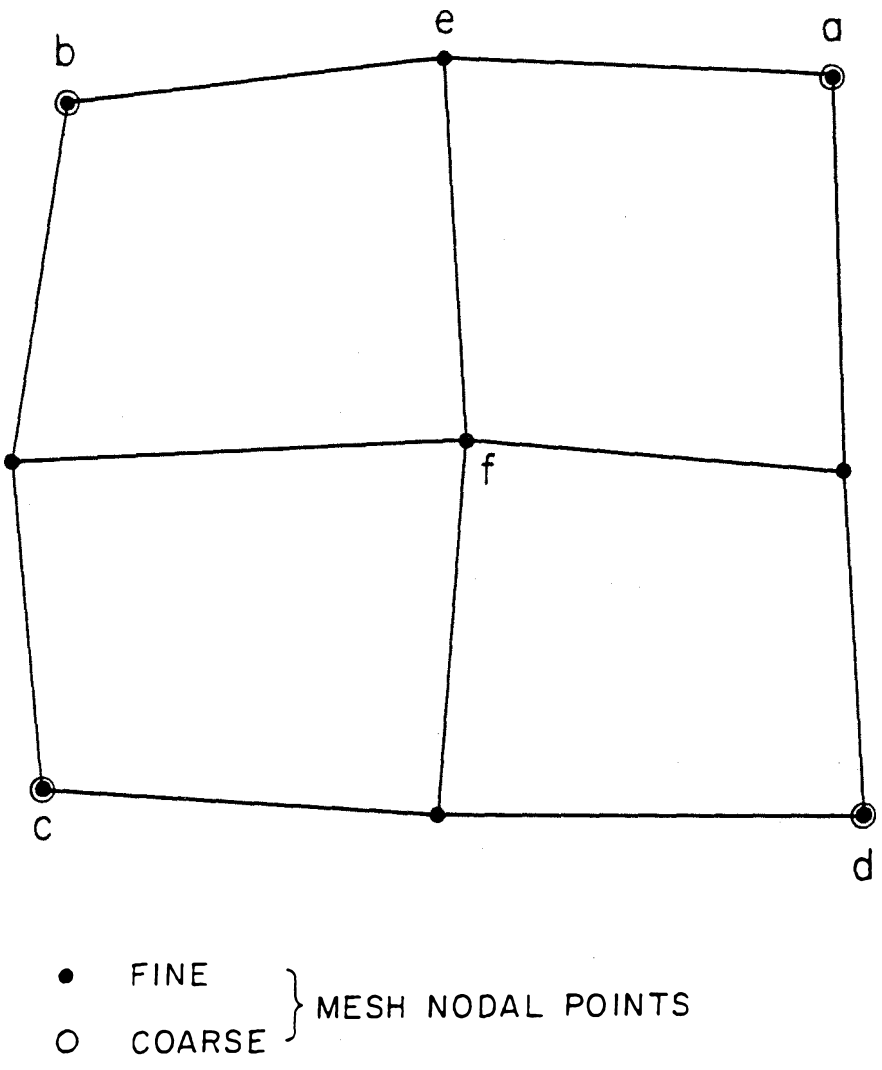


Figure 4.7: Coarsening for the Quadrilateral Elements.

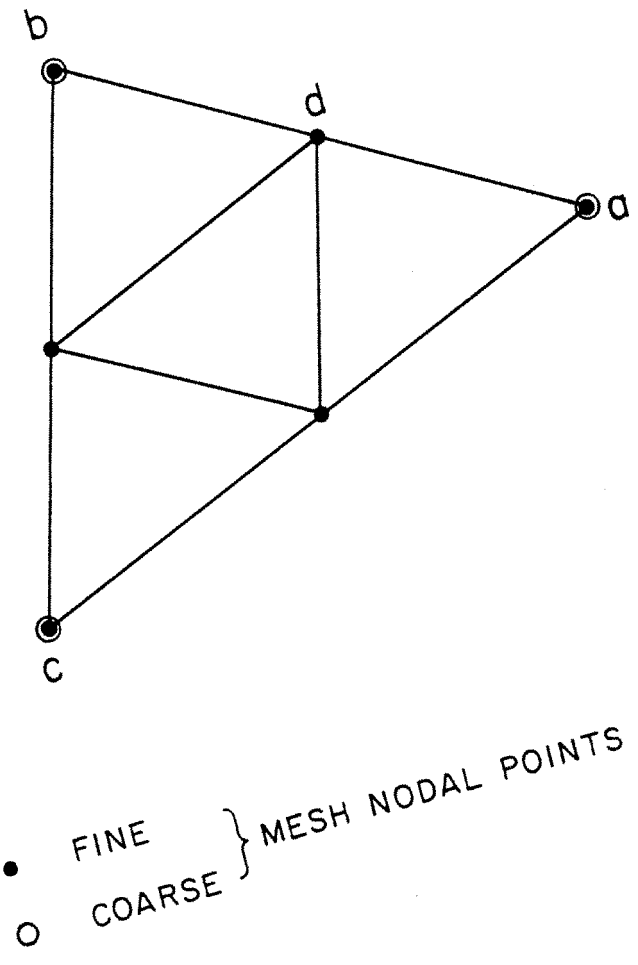


Figure 4.8: Coarsening for the Triangular Elements.

Chapter 5

Evaluation of the Multigrid Method for Solid Mechanics Problems

The objective of this chapter is to evaluate the strengths and weaknesses of the multigrid algorithm described in chapter 4 by applying it to some simple two and three dimensional solid mechanics problems. In section 5.1, different variants of the method are used to solve some two dimensional problems on uniform meshes (a point load, a point load acting on a nearly incompressible material, and a crack). This enables the effect of different relaxation schemes to be examined, and reveals some of the weaknesses of the method. In section 5.2, the influence of nonuniform meshes on the convergence of the multigrid method is determined. Finally, in section 5.3, the performance of the method is compared with two other solution algorithms (the direct factorization and the Jacobi preconditioned conjugate gradient methods) by solving a three dimensional problem on a series of uniform meshes. A discussion of the results presented in this chapter is given in section 5.4.

5.1 Application of the Multigrid Method to Some Two Dimensional Trial Problems

The multigrid algorithm described in chapter 4 was applied to some small two dimensional problems so that its behavior could be examined. A two mesh method was used; the fine and the coarse meshes were shown in figure 4.4. The fine mesh on which the particular problems were posed consisted of 128 four node linear quadrilateral elements and 153 nodal points. Three different problems were solved on this mesh by varying the boundary conditions and the element material properties: a point load, a point load acting on a nearly incompressible material, and a crack subjected to the tensile opening mode of deformation (mode I). Plane strain was assumed in all cases; the mesh was 2 m by 1 m with an out-of-plane dimension of 1 m. Unless stated otherwise, Young's modulus was 207 GN/m^2, Poisson's ratio was 0.3, a 2×2 Gauss (exact) integration scheme was used to compute the fine mesh stiffness matrix, and the relaxation parameters ν_1 and ν_2 were set at 5 for each problem.

5.1.1 A Point Load

The boundary conditions used to specify the point load acting on the fine mesh were shown in figure 4.1. This problem is a useful test of the multigrid method for two reasons: point loads are common boundary conditions in solid mechanics, and the theoretical solution shows singularities in the stresses, the strains, and the displacements at the load application point. Two types of multigrid methods were tried: MG_1-GS and MG_1-JCG. Figure 5.1 shows graphs of $\log_{10}(R)$ versus

the number of multigrid cycles for these schemes, where R is the convergence parameter defined in equation (3.11). During each multigrid cycle, R was computed after each relaxation and immediately after the interpolated coarse mesh correction had been added to the current fine mesh approximation. The discontinuities in $\log_{10}(R)$ shown in figure 5.1 (and in similar plots) occur when the coarse mesh correction has been computed and interpolated to the fine mesh. The MG_1-GS and MG_1-JCG methods converge to a value of $R \leq 10^{-6}$ in five multigrid cycles; however, the GS relaxation scheme requires less computational work.

Figure 5.2 shows the behavior of the error on the fine mesh during the first multigrid cycle for the MG_1-JCG method. A similar diagram was shown in figure 4.5 for the MG_1-GS method. This data shows the error before any relaxation has taken place ($\mathbf{e}_f^{(0)}$), after ν_1 cycles of relaxation ($\bar{\mathbf{e}}_f^{(0)}$), after the interpolation of the coarse mesh correction ($\hat{\mathbf{e}}_f^{(0)}$), and after ν_2 cycles of relaxation ($\mathbf{e}_f^{(1)}$). All of the errors in the two figures are scaled by the same factor. The figures reveal that both relaxation schemes were effective in reducing the high frequency error component and that the coarse mesh enabled the low frequency error component to be obtained from the coarse mesh correction equation. This explains the fast convergence of the MG_1-GS and MG_1-JCG methods.

Figure 5.3 shows the effect of varying the relaxation factor (ω in table 3.1) on the speed of convergence of the MG_1-SOR scheme. The method was assumed to have converged when $R \leq 10^{-6}$. A near optimum convergence rate can be achieved by choosing the relaxation factor as 1, which corresponds to GS relaxation. This agrees with Hackbusch's observation that successive overrelaxation with an optimal relaxation factor cannot be used as a smoothing iteration [27].

5.1.2 A Point Load Acting on a Nearly Incompressible Material

Nearly incompressible elasticity (i.e. a material whose Poisson's ratio is close to $\frac{1}{2}$, such as rubber) is a severe test for any solution algorithm because an ill-conditioned stiffness matrix results in plane strain and three dimensional problems (see section 3.4). Another difficulty that faces the engineering analyst when using four node linear quadrilateral elements is that of element locking: the element's bending modes become very stiff as Poisson's ratio approaches $\frac{1}{2}$ because they involve considerable volumetric strain energy (see section 2.3). As was explained in section 2.3, this can be fixed, and the element's behavior improved, by using a reduced, one point Gauss integration to compute the $\mathbf{D}_\lambda^e$ contribution to the stiffness matrix, since the volume change in the bending modes is zero at this integration point.

The boundary conditions imposed on the fine mesh were the same as those shown in figure 4.1. Several multigrid methods were tested: MG_1-GS, MG_2-GS, MG_1-JCG, and MG_2-JCG. Different values of ν_1 and ν_2 were tried. The MG_2 methods used a coarse mesh stiffness matrix that was assembled from coarse elements computed with reduced integration on the $\mathbf{D}_\lambda^e$ term. The fine mesh stiffness matrix for all of the multigrid methods was calculated using the same reduced integration scheme.

Figure 5.4 shows that the constraints used to compute the coarse mesh stiffness matrix using equation (4.11) will produce considerable local volume changes when the elements associated with the coarse mesh deform in their bending modes. The volume changes are caused by each fine mesh element that makes up the coarse element being constrained to deform in a way that is not volume

preserving; note that these volume changes are inconsistent with the coarse element's true bending mode. Therefore the bending mode of a coarse element will involve considerable volumetric strain energy. This will result in a coarse mesh that locks (i.e. one that is very stiff) for such deformations, even though reduced integration may have been used on the fine mesh. Thus the coarse mesh will not effectively yield the low frequency error component. On the other hand, direct assembly of the coarse mesh stiffness matrix from coarse elements using one point integration to compute the $\mathbf{D}_\lambda^e$ term will avoid the locking problem.

Figure 5.4 also shows that the interpolation procedure used to transfer displacements from the coarse mesh to the fine mesh described in section 4.2.3 will introduce volume changes on the fine mesh. However, these volume changes are associated with a high frequency error component, and should therefore be quickly eliminated by an effective relaxation procedure.

Figure 5.5 shows the effect Poisson's ratio has on the convergence of the various multigrid schemes with different values of ν_1 and ν_2. Although the MG_1 schemes converged in fewer multigrid cycles as ν_1 and ν_2 were increased, the computational work remained approximately constant. However, the choice of ν_1 and ν_2 had a significant effect on the convergence of the MG_2 schemes. The convergence of the MG_2-GS scheme in particular was greatly improved by increasing the values of ν_1 and ν_2. Observation of the error on the fine mesh shows that this was due to the behavior of the high frequency error component. Figure 5.6 shows plots of the error on the fine mesh for the MG_1-GS scheme with $\nu_1 = \nu_2 = 5$ for a Poisson's ratio of 0.498. The initial error ($\mathbf{e}_f^{(0)}$) is shown, together with the error after one ($\mathbf{e}_f^{(1)}$), five ($\mathbf{e}_f^{(5)}$), and ten ($\mathbf{e}_f^{(10)}$) multigrid cycles. Figures 5.7, 5.8, and

5.9 show similar plots of the error when the MG_2-GS scheme was used to solve the same problem, with $\nu_1 = \nu_2 = 5$, 10, and 20. Figure 5.6 shows that the GS relaxation did not reduce the high frequency error as fast as it did for $\nu = 0.3$ (see figure 4.5). It is also apparent that the coarse mesh formed using the constraints was not effective in capturing the low frequency error. However, the MG_1-GS method did converge with $\nu_1 = \nu_2 = 5$.

The plots of the error shown in figures 5.7, 5.8, and 5.9 for the MG_2-GS scheme clearly demonstrate the sensitivity of this method to the choice of ν_1 and ν_2. Observation of the error after one multigrid cycle ($\mathbf{e}_f^{(1)}$) shows how poor the GS relaxation was in smoothing the error for a nearly incompressible material, although its performance was improved by increasing ν_1 and ν_2. However, the coarse mesh assembled using reduced integration was very effective in computing the low frequency error. It is the substantial difference in the performance of the fine mesh relaxation and the coarse mesh that caused the MG_2-GS method to diverge for $\nu_1 = \nu_2 = 5$ (see figure 5.7). The reduction in the high frequency component due to the relaxation was less than the increase due to the interpolation of the correction from the coarse mesh since too few relaxation steps were performed on the fine mesh. On the other hand, for the MG_1-GS method with $\nu_1 = \nu_2 = 5$ (see figure 5.6), the low frequency error component on the fine mesh was not well represented by the direct solution of the coarse mesh correction equation because the coarse mesh locked. This meant that the interpolated high frequency component was also less, in fact small enough to be kept under control by the inefficient GS relaxation. The instability present in the MG_2 scheme for a nearly incompressible material is cured by increasing the values of ν_1 and ν_2, as is evident in figures 5.7, 5.8, and 5.9.

Figure 5.5 also shows that the convergence of the MG-JCG methods were improved by increasing ν_1 and ν_2 as Poisson's ratio approached $\frac{1}{2}$. The smoothing effect of the JCG relaxation was found to be less sensitive to Poisson's ratio than the GS relaxation, which explains why the MG-JCG schemes work better than the MG-GS schemes as Poisson's ratio approaches $\frac{1}{2}$.

For values of Poisson's ratio less than 0.49, the most effective multigrid scheme incorporated GS relaxation (with $\nu_1 = \nu_2 = 5$) and an assembled coarse mesh stiffness matrix with reduced integration. Although the corresponding MG_2-JCG scheme produced convergence in fewer multigrid cycles, the GS relaxation involved less computational work. An assembled coarse mesh with reduced integration is not subject to locking, enabling the low frequency error to be accurately computed. For larger values of Poisson's ratio, the most effective scheme was MG_2-JCG with an assembled coarse mesh stiffness matrix and increased values of ν_1 and ν_2. The JCG relaxation and the increased values of ν_1 and ν_2 are necessary since the smoothing capability of the relaxation methods is reduced for higher values of Poisson's ratio. For a Poisson's ratio of 0.4997, the MG_2-JCG scheme (with ν_1 and ν_2 equal to 20) converged in 18 multigrid cycles. Incidentally, 0.4997 is a typical Poisson's ratio for vulcanized natural rubber [40].

5.1.3 A Crack Subjected to Mode I Deformation

Fracture mechanics has been the subject of many numerical studies. The details of the multigrid solution of a large three dimensional crack problem are given in chapter 6 of this thesis. As a preliminary study, a simple two dimensional example was considered with the same material properties as used in the point

load problem (section 5.1.1). Figure 5.10 shows the boundary conditions used to specify the problem on the fine mesh. The theoretical solution predicts stress and strain singularities at the crack tip [60] (see chapter 6).

Figure 5.11 shows plots of $\log_{10}(R)$ versus the number of multigrid cycles when the crack problem was solved using the MG_1-GS and MG_1-JCG methods with $\nu_1 = \nu_2 = 5$. Both methods converged in 9 multigrid cycles, which is 4 cycles more than was observed for the point load discussed in section 5.1.1.

Figure 5.12 shows the behavior of the error on the fine mesh for the MG_1-GS method before any relaxation has taken place ($\mathbf{e}_f^{(0)}$), after ν_1 cycles of relaxation ($\bar{\mathbf{e}}_f^{(0)}$), after the interpolation of the coarse mesh correction ($\hat{\mathbf{e}}_f^{(0)}$), and after ν_2 cycles of relaxation ($\mathbf{e}_f^{(1)}$). Similar behavior was observed for the MG_1-JCG method. The error after the coarse mesh correction equation had been solved exactly ($\hat{\mathbf{e}}_f^{(0)}$) shows there was a considerable low frequency error that the coarse mesh was unable to capture, consisting mainly of a bending deformation. This can be explained by consideration of the bending behavior of the four node elements associated with the coarse mesh. It was explained in section 2.3 that the stiffness of the bending mode of a four node element is too high due to the absence of a lateral normal strain and the presence of shear strains. Better results can be achieved when one point Gauss integration is employed on both the $\mathbf{D}_\lambda^e$ and the $\mathbf{D}_\mu^e$ terms of equation (2.12), which reduces the bending stiffness to zero. For this crack problem, it was found that a singular coarse mesh stiffness matrix resulted; this was due to the presence of zero energy modes of deformation [33].

As an alternative, reduced integration of only one of the $\mathbf{D}_\lambda^e$ and $\mathbf{D}_\mu^e$ terms was attempted on the coarse mesh stiffness matrices. The MG_2-GS and the MG_2-

JCG methods that resulted from a reduced integration of the $\mathbf{D}_\lambda^e$ term converged in 7 and 8 cycles, respectively. Figure 5.13 shows the behavior of the error on the fine mesh for this MG_2-GS scheme. When the $\mathbf{D}_\mu^e$ term was computed using one point Gauss integration, the MG_2-GS and MG_2-JCG methods both converged in 6 cycles; the error on the fine mesh is shown in figure 5.14 for this MG_2-GS method.

Careful comparison between the error on the fine mesh shown in figures 5.12, 5.13, and 5.14 for the MG_1-GS and MG_2-GS schemes demonstrates that the coarse meshes that were computed using reduced integration captured more of the low frequency error than the coarse meshes formed using constraints. It is also apparent that the high bending stiffness of the coarse mesh is more effectively relieved by using the reduced integration on the $\mathbf{D}_\mu^e$ term than on the $\mathbf{D}_\lambda^e$ term. This was expected since the $\mathbf{D}_\mu^e$ term contains the shear energy that is responsible for the excessive coarse mesh bending stiffness. The reduced integration resulted in a coarse mesh that was able to capture the low frequency error, and therefore yielded an improved coarse mesh correction.

5.2 The Effect of Nonuniform Meshes on the Convergence of the Multigrid Method

An important theoretical property of the multigrid method that has been established for uniform discretizations of elliptic problems, is that the number of operations (and hence solution time) required for convergence is directly proportional to the problem size (i.e. the number of degress-of-freedom in the mesh).

This follows because the number of operations required by the components of each multigrid cycle is proportional to the problem size, and, for uniform discretizations, the total number of cycles required for convergence is independent of the problem size [56]. However, nonuniform meshes are very common in the application of the finite element method; they may be used to model an irregular boundary or to capture high gradients near stress concentrations or singularities. The objective of this section is to examine the effect of nonuniform meshes on the convergence of the multigrid method by using several uniform and nonuniform meshes to solve the point load problem shown in figure 4.1.

Figure 5.15 shows the various meshes that were used in this study. A two mesh MG-GS method with $\nu_1 = \nu_2 = 5$ was employed; the three uniform fine meshes $\Omega_1^{(u)}$, $\Omega_2^{(u)}$, and $\Omega_3^{(u)}$ were each used with the three uniform coarse meshes $\Omega_2^{(u)}$, $\Omega_3^{(u)}$, and $\Omega_4^{(u)}$, respectively. A similar arrangement of the fine and coarse meshes was used with the nonuniform meshes, denoted by superscript (n), shown in figure 5.15. Hence six different pairs of meshes with varying numbers of degrees-of-freedom (indicated in figure 5.15) were used to solve the simple point load problem. It should be noted that the nonuniform mesh, $\Omega_i^{(n)}$, had the same element size around the load application point as was present throughout the corresponding uniform mesh, $\Omega_i^{(u)}$. The material properties and overall dimensions of the meshes were the same as in section 5.1; plane strain conditions were assumed, and 2×2 Gauss integration was used to compute all of the fine mesh stiffness matrices.

Figures 5.16 and 5.17 show the behavior of R (see equation (3.11)) when the two mesh MG_1-GS method was used to solve each of these six problems. It

can be seen that more multigrid cycles were required by the nonuniform meshes for a given R as the problem size was increased. Figures 5.18 and 5.19 show the corresponding behavior of the norm of the error on the fine mesh computed using equation (3.13) for the six problems. These plots show that after five multigrid cycles, the size of the error was smaller for the larger problems. This effect was more pronounced for the uniform meshes.

A general observation that can be made from figures 5.16 and 5.17 is that the multigrid method converges faster (in terms of the number of cycles required) when applied to the uniform meshes. This can be explained by examining the error on the fine mesh during the first multigrid cycle for the fine meshes $\Omega_2^{(u)}$ and $\Omega_2^{(n)}$. Recall that these meshes have the same element size near the load application point, and that the coarse meshes used in the multigrid scheme were $\Omega_3^{(u)}$ and $\Omega_3^{(n)}$. Figures 5.20 and 5.21 show the error on the fine mesh during the first multigrid cycle for the uniform and nonuniform meshes, respectively. The error is shown before any relaxation has taken place ($\mathbf{e}_f^{(0)}$), after ν_1 cycles of relaxation ($\bar{\mathbf{e}}_f^{(0)}$), after the interpolation of the coarse mesh correction ($\hat{\mathbf{e}}_f^{(0)}$), and after ν_2 cycles of relaxation ($\mathbf{e}_f^{(1)}$).

Careful comparison between figures 5.20 and 5.21 show that the low frequency error component is reduced more by the uniform coarse mesh than by the nonuniform coarse mesh. This can be seen by observing the error after interpolation of the coarse mesh correction ($\hat{\mathbf{e}}_f^{(0)}$) on the uniform and nonuniform fine meshes. This is caused by the poor bending behavior of the four node plane elements discussed in section 2.3. The nonuniform coarse mesh has fewer elements than the uniform coarse mesh that can be used to capture the required bending

deformation. (This bending is evident in the plots of $\bar{\mathbf{e}}_f^{(0)}$, the error after ν_1 relaxation cycles, shown in figures 5.20 and 5.21.) Thus the nonuniform coarse mesh is stiffer than the uniform coarse mesh, and so a significant low frequency error component will be present on the nonuniform fine mesh after the interpolation of the coarse mesh correction. It is interesting to note that the high frequency error near the load application point present after the coarse mesh correction has been interpolated ($\hat{\mathbf{e}}_f^{(0)}$) is the same for both the uniform and the nonuniform fine meshes. This demonstrates that the smoothing effect of the GS relaxation is uneffected by the nonuniform mesh.

Another observation that can be made from the plots of R shown in figure 5.17 for the nonuniform meshes is that the value of R after five multigrid cycles is larger for the smaller problems, but that it seems to be reaching a constant value that is independent of the problem size for larger problems. This point is further demonstrated by figures 5.22 and 5.23 which show log-log plots of the value of R after five multigrid cycles versus the problem size for the problems with the uniform and nonuniform meshes. Note that figures 5.22 and 5.23 contain additional data obtained by solving the point load problem on meshes formed by uniformly subdividing meshes $\Omega_1^{(n)}$ and $\Omega_1^{(u)}$ shown in figure 5.15. The figures also show the values of R after five cycles when the MG_2-GS schemes that utilized reduced integration of the $\mathbf{D}_\lambda^e$ and $\mathbf{D}_\mu^e$ terms in equation (2.12) (see section 2.3) were used to solve the point load problem.

It was shown in section 5.1.3 that the bending behavior of the coarse mesh can be improved by using a reduced integration scheme to compute the $\mathbf{D}_\mu^e$ term. Figures 5.22 and 5.23 show that this improvement is present when both the $\mathbf{D}_\lambda^e$

and $\mathbf{D}_{\mu}^{e}$ terms are computed using reduced integration for the uniform and nonuniform meshes, although reduced integration on the $\mathbf{D}_{\mu}^{e}$ term is the most effective technique. However, figure 5.23 shows that the reduced integration becomes less effective as the problem size is increased for the nonuniform meshes. This is not evident for the uniform meshes, see figure 5.22. Both figures suggest that the number of multigrid cycles required by both the uniform and the nonuniform meshes becomes independent of the problem size as the meshes are refined, and that this effect is more pronounced for the nonuniform meshes.

The results presented in this section show that the number of multigrid cycles required by a nonuniform mesh may be greater than that required by a uniform mesh used to solve the same problem, and that this is due to the poor bending behavior of the nonuniform coarse mesh. Consequently, the solution time required by a nonuniform mesh may not be proportional to the problem size. However, this proportionality can be recovered by producing fine meshes by repeatedly subdividing a nonuniform mesh. It is also interesting to note that reduced integration techniques are less effective for nonuniform meshes, especially as the problem size is increased.

5.3 The Performance of the Multigrid Algorithm

This section compares the performance of the multigrid method with some other solution algorithms by solving a three dimensional solid mechanics problem. A cube shown in figure 5.24 and consisting of a material with Young's modulus 207 GN/m^2 Poisson's ratio 0.3 and density 7850 kg/m^3 with one vertical face fixed, subjected to a gravity load and four horizontal point loads acting at the free

corners, was discretized with eight node linear brick elements. Five test problems of different sizes were generated by using regular meshes that had 4, 6, 8, 12, and 16 elements along each side of the cube, and were used to compare the speed and storage requirements of various solution schemes. Six solution algorithms were used to solve the resulting matrix equation: a direct factorization, JCG iteration, MG_1-GS, MG_2-GS, MG_1-JCG, and MG_2-JCG. It should be noted that the MG_2 methods did not use reduced integration on any part of the coarse mesh element stiffness matrix. The multigrid methods consisted of V-cycles (i.e. $\gamma = 1$, see section 4.2) with ν_1 and ν_2 equal to 5. The number of meshes used in the multigrid schemes depended on the size of the problem; generally, the coarsest mesh was the last one that could be produced by repeated coarsening of the finest mesh (i.e. repeatedly grouping eight fine mesh elements together to form a coarse mesh element). The convergence tolerance for each iterative method (ε in equation (3.11)) was 10^{-6}.

Figure 5.25 shows the variation of the total computation time with the problem size (the number of degrees-of-freedom in the finest mesh) for the various solution schemes. Figure 5.26 shows the corresponding solution times (the total time minus the set-up time) and the estimated direct solution time (i.e. the time required to perform $\frac{1}{2}nm^2$ operations on the VAX 11/750, see section 3.2). The set-up time includes the time required to compute and assemble all of the stiffness and interpolation matrices. Due to the excessive computational effort required, the data shown in figures 5.25 and 5.26 for the direct solution of the largest problem (about 14,000 degrees-of-freedom) had to be extrapolated from a partial factorization of the stiffness matrix.

The iterative methods all required approximately the same amount of time. These iterative methods were about two orders of magnitude faster than the direct method for the largest problem considered. The total and solution times measured for the MG_2 schemes (not shown in figures 5.25 and 5.26) were within 1% of the corresponding times of the MG_1 schemes, demonstrating that the difference in the computational effort between assembling and computing the coarse mesh stiffness matrices was negligible. The times required by the multigrid methods depended linearly on the problem size. A least squares straight line fit showed that the JCG solution times varied as $n^{1.5}$, which is similar to the predictions made in [2] for conjugate gradient relaxation.

The estimate of the direct solution time based on the number of operations was within 30% of the measured time for problems smaller than 10,000 degrees-of-freedom. Although this error is large, the estimate does give an order of magnitude prediction of the solution time. The large amount of I/O time required to produce a direct solution causes the estimate to fail for larger problems.

Figure 5.27 shows the amount of in-core storage required for the solution of the test problems. The storage needed by the iterative solutions grew linearly with the problem size, and was about one order of magnitude less than that required by the direct solution for the problem with 14,000 degrees-of-freedom. The large in-core storage requirement of the direct solution algorithm contributed to its slowness since the computer had to spend an increasing amount of time transferring data from disk to high speed memory by means of its virtual memory system. The different storage requirements of the MG_1 and the MG_2 methods were due to the additional information needed by the program to ensure that the

computational effort required in the calculation of the coarse mesh stiffness matrices using equation (4.11) was proportional to the problem size. The additional storage required by the MG-JCG methods (not shown in figure 5.27) was found to be less than 3% of that required by the corresponding MG-GS methods.

The time required to form the coarse mesh stiffness and interpolation matrices in the multigrid methods was less than 9% of the total computation time; the data associated with the coarse meshes amounted to 45% of the total storage requirements. The multigrid method was able to produce fast solutions of the test problems despite these overheads associated with the coarse meshes.

The three dimensional test problem shown in figure 5.24 can be used to demonstrate the effect of the bending behavior of the coarse meshes on the performance of the multigrid algorithm. Table 5.1 shows the number of cycles required by the MG_1-GS method when it was used to solve the five test problems. The table also shows the number of cycles required when the direction of the point loads acting on the cube shown in figure 5.24 were changed from horizontal to vertical. This change of loading subjected the cube to a substantial bending deformation. Table 5.1 demonstrates that this bending resulted in an increase in the number of multigrid cycles required for convergence. This was due to the stiffness of the coarse meshes when they are subjected to bending deformations, which does not allow the low frequency error to be computed accurately.

The results presented in this section demonstrate the inherent inefficiency of the direct method when used to solve three dimensional problems. It is interesting to note that the largest problem considered (which was too big to be solved directly on the VAX 11/750 used in this study) was generated by the dis-

cretization of a cube with 16 eight node linear brick elements along each side. The numerical analysis of many three dimensional solid mechanics problems of engineering interest would require a more refined mesh, producing an even larger problem.

5.4 Discussion of the Results

The relatively simple problems discussed in this chapter have revealed several important features of the multigrid method when it is applied to solid mechanics problems. The poor performance of the method when conditions of near incompressibility are encountered (see section 5.1.2) is due to a combination of a reduction in the smoothing effect of the GS relaxation method and coarse mesh locking. These problems can be partially cured by using the JCG iteration method to smooth the error and assembling the coarse mesh stiffness matrices using a reduced integration scheme.

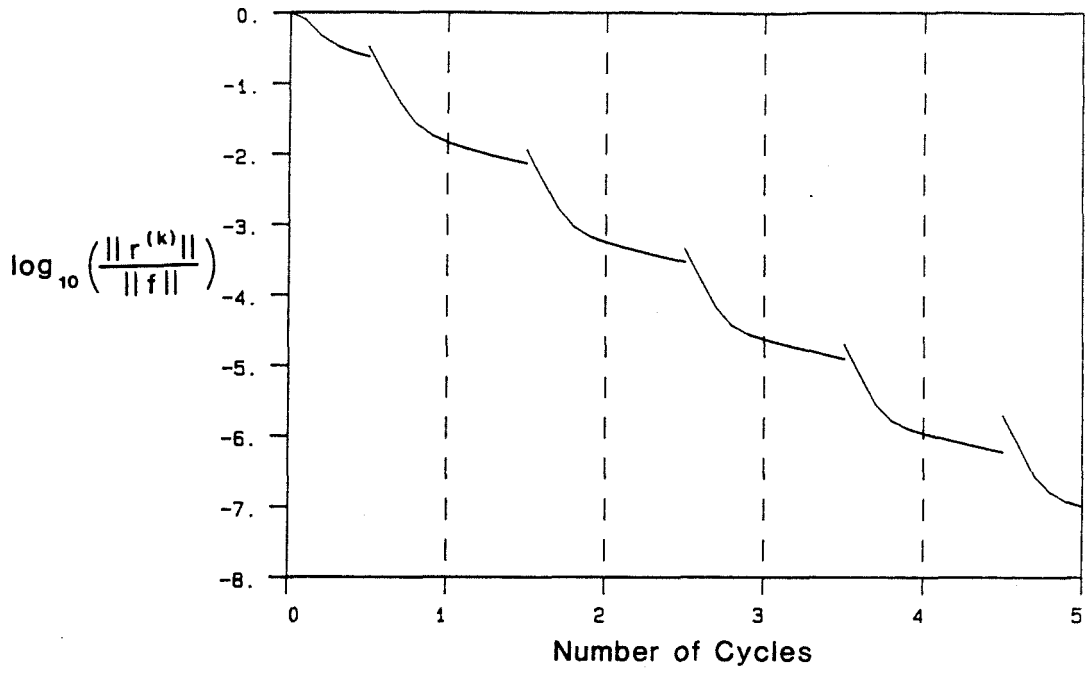
The bending behavior of the linear brick and quadrilateral elements used in this chapter was found to slow the convergence of the multigrid algorithm. The coarse mesh is unable to capture a low frequency error that consists of a bending deformation accurately. This effect also caused nonuniform meshes to yield solution times that were not proportional to the problem size n; however, the linear dependence on n was recovered as it became larger. It was demonstrated in section 5.2 that reduced integration techniques become less effective in relieving the stiffness of the coarse mesh for nonuniform meshes as the problem size is increased.

The studies presented in section 5.3 demonstrate that the multigrid algorithm requires far less computational effort (in terms of CPU time and storage requirement) than a direct method. Furthermore, the performance of the multigrid method is comparable to that of the Jacobi preconditioned conjugate gradient method (JCG) for the well-conditioned problem that was considered. It is interesting to note that the solution times required by the multigrid and the JCG methods were proportional to n and $n^{1.5}$, respectively. This suggests that the multigrid method will perform better than the JCG method for large problems.

The features of the multigrid algorithm that were examined in this chapter will be used to explain the performance of the method when some practical problems in solid mechanics are solved in part II of this thesis. The problems in part II are a good test of the method not only because of their engineering significance, but also because they require the solution of large, ill-conditioned, matrix equations.

n	Number of Cycles for the Loading Shown in Figure 5.24	Number of Cycles for the Bending Deformation
375	4	6
1,029	4	5
2,187	4	7
6,591	5	6
14,739	5	7

Table 5.1: Effect of Bending on the Number of MG_1-GS Cycles Required for Convergence for the Three Dimensional Test Problem.



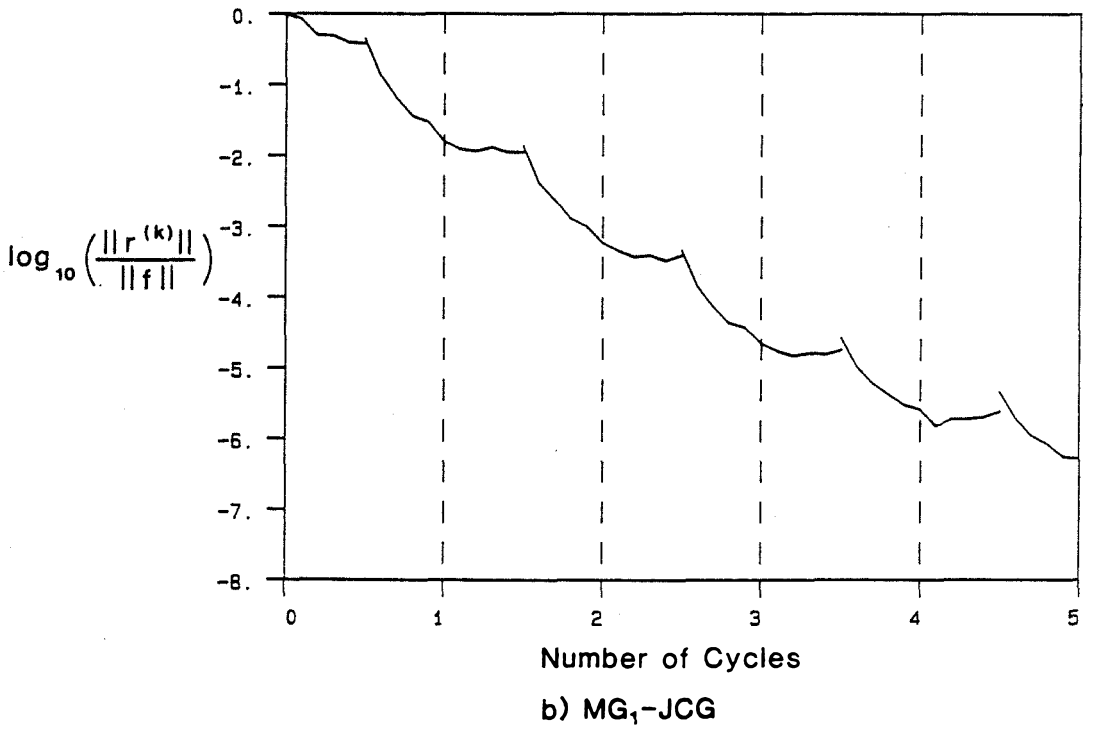


Figure 5.1: Convergence of the MG_1-GS and the MG_1-JCG Methods ($\nu_1 = \nu_2 = 5$) for the Point Load Problem.

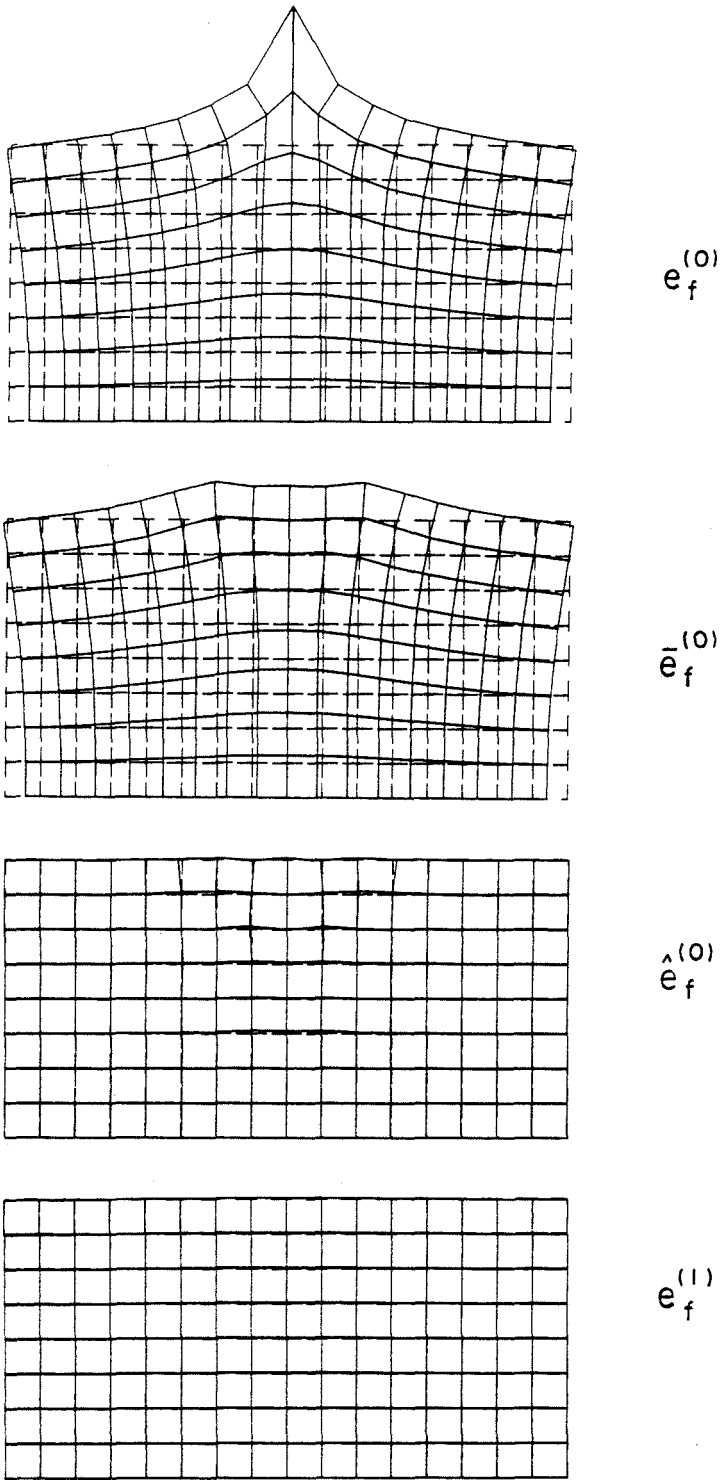


Figure 5.2: Behavior of the Fine Mesh Error in the MG_1-JCG ($\nu_1 = \nu_2 = 5$) Method for the Point Load Problem.

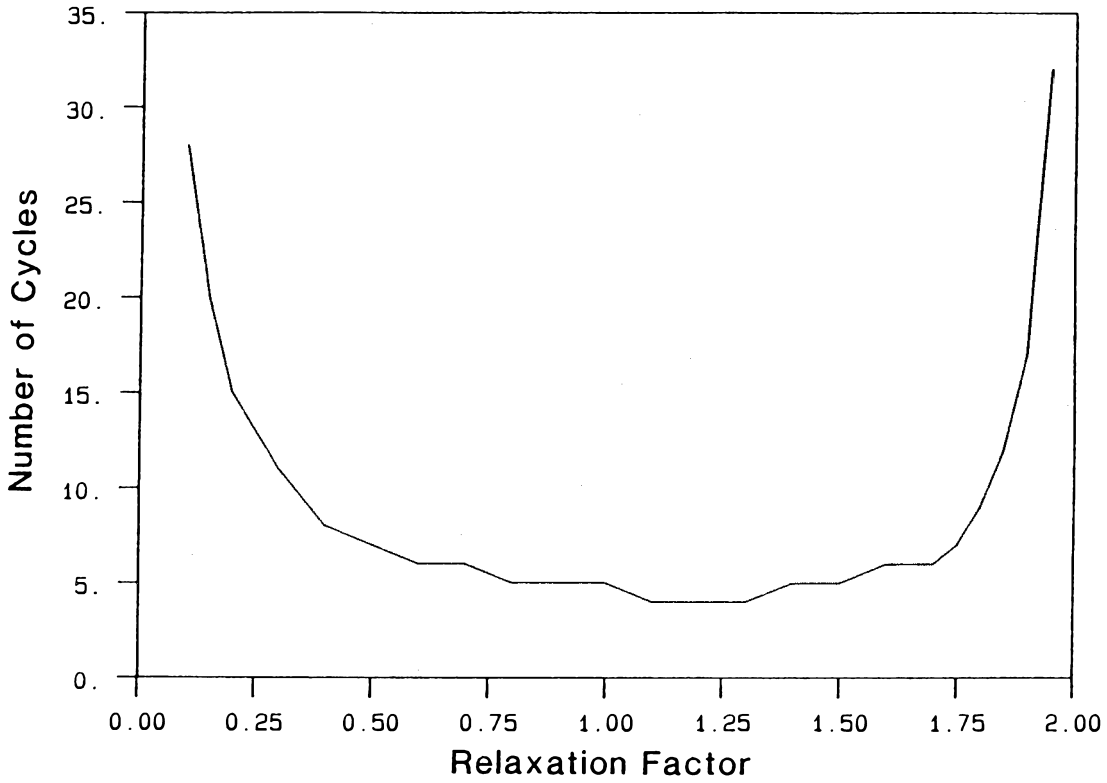


Figure 5.3: Effect of the Relaxation Factor on the Convergence of the MG_1-SOR ($\nu_1 = \nu_2 = 5$) Method for the Point Load Problem.

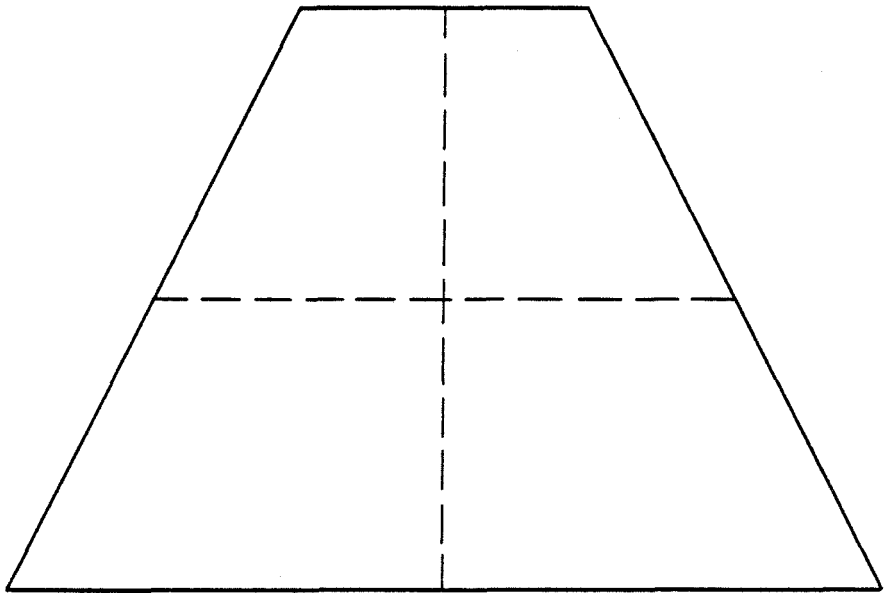


Figure 5.4: Bending Behavior of a Coarse Mesh Element Formed Using Constraints.

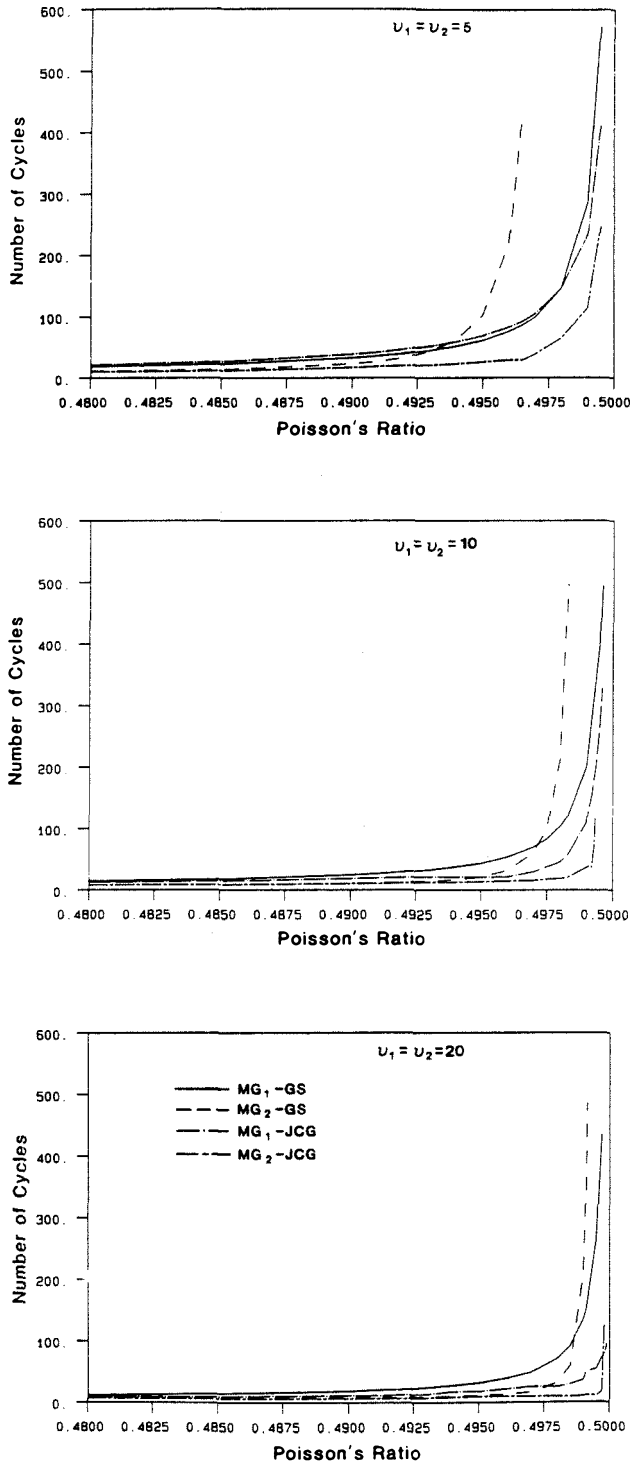


Figure 5.5: The Effect of Poisson's Ratio on the Convergence of Various Multigrid Schemes for the Point Load Problem.

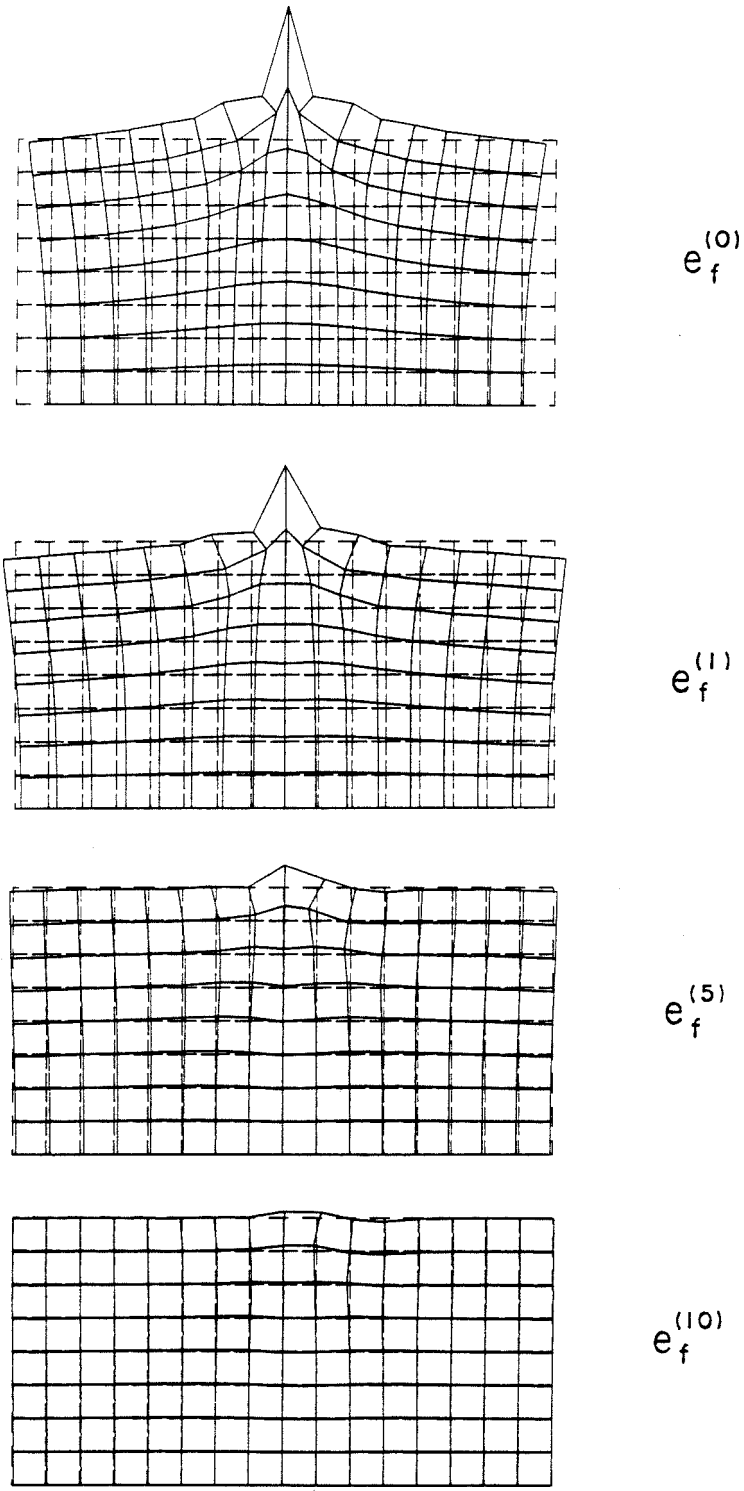


Figure 5.6: Behavior of the Fine Mesh Error in the MG_1-GS ($\nu_1 = \nu_2 = 5$) Method for the Point Load Problem; $\nu = 0.498$.

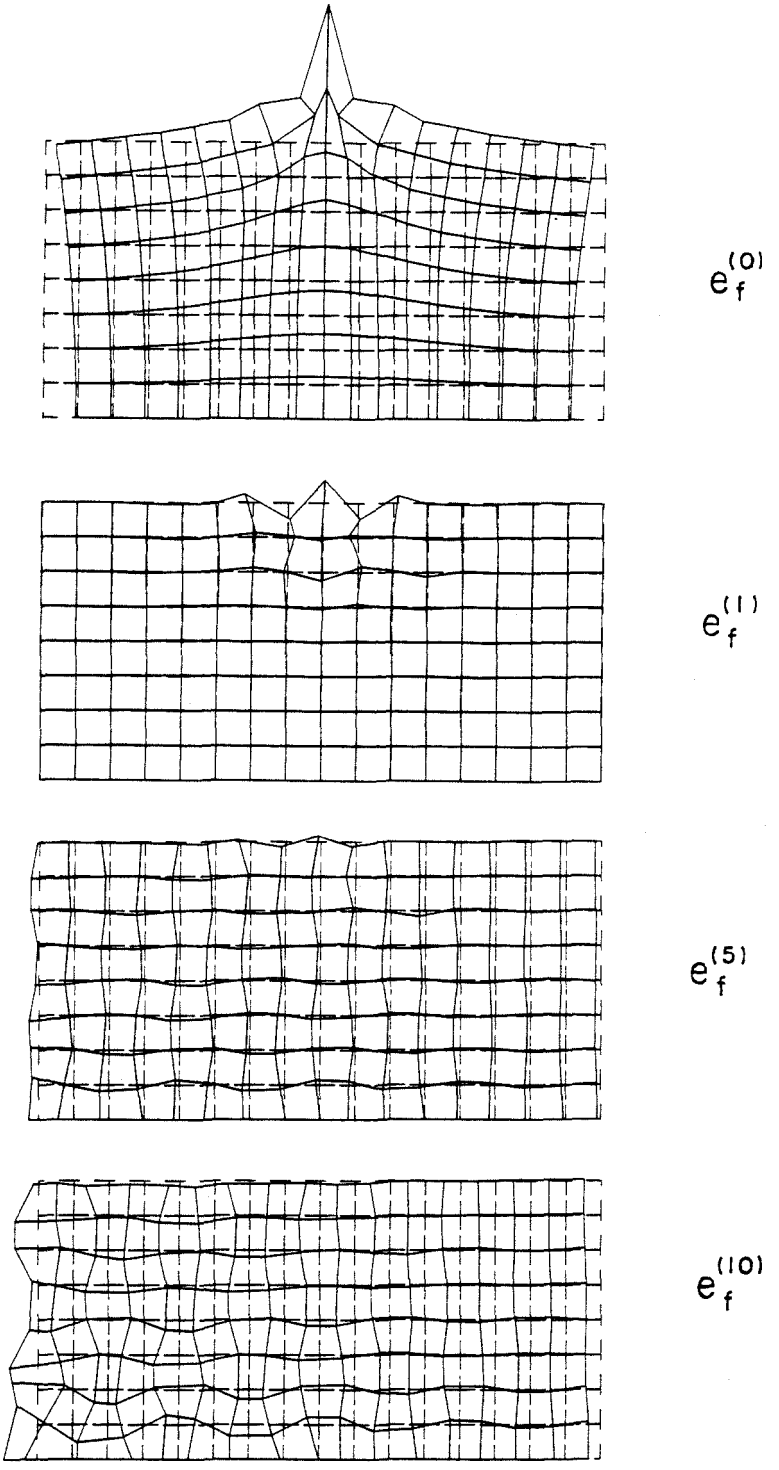


Figure 5.7: Behavior of the Fine Mesh Error in the MG_2-GS ($\nu_1 = \nu_2 = 5$) Method for the Point Load Problem; $\nu = 0.498$.

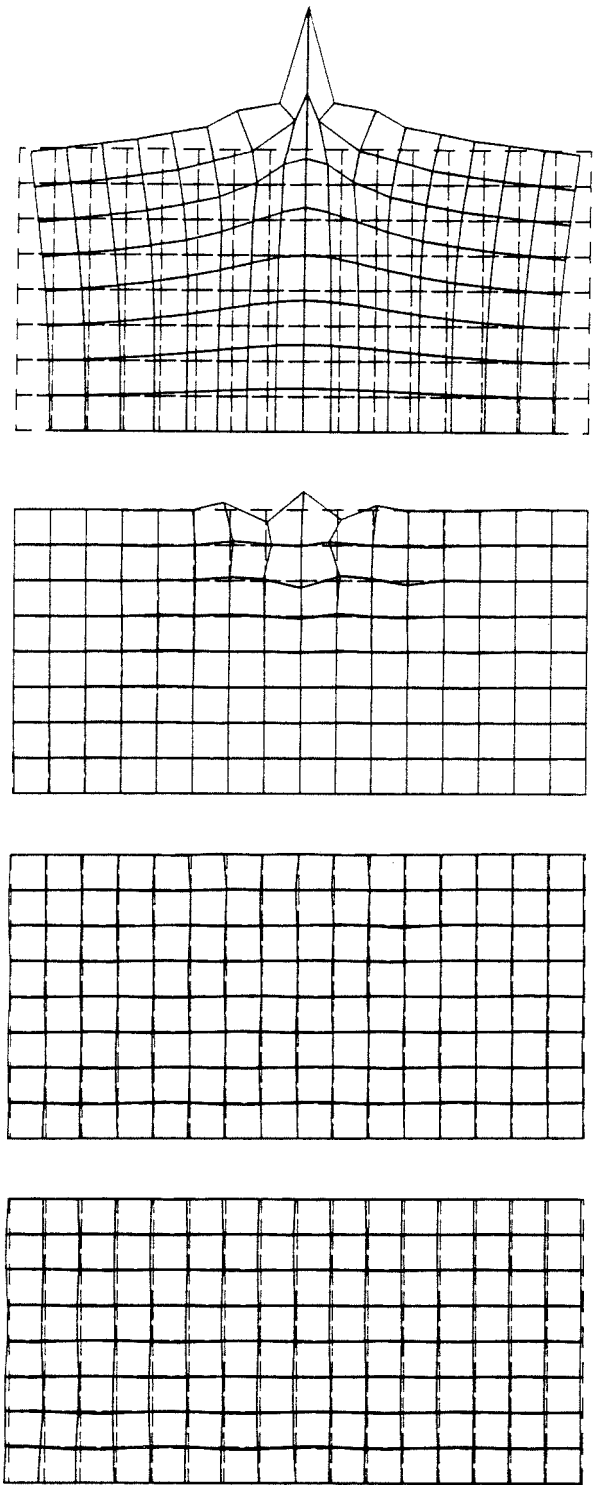


Figure 5.8: Behavior of the Fine Mesh Error in the MG_2-GS ($\nu_1 = \nu_2 = 10$) Method for the Point Load Problem; $\nu = 0.498$.

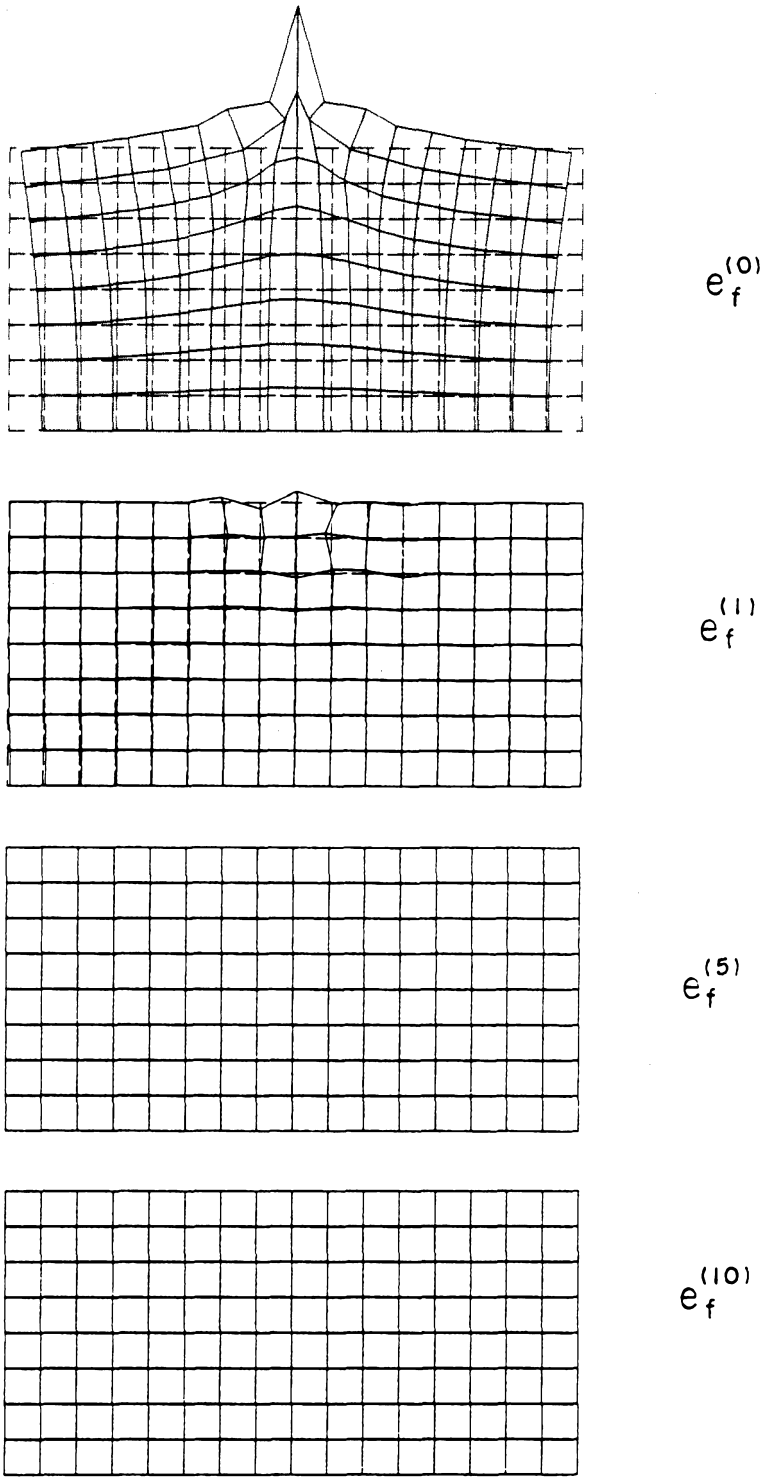


Figure 5.9: Behavior of the Fine Mesh Error in the MG_2-GS ($\nu_1 = \nu_2 = 20$) Method for the Point Load Problem; $\nu = 0.498$.

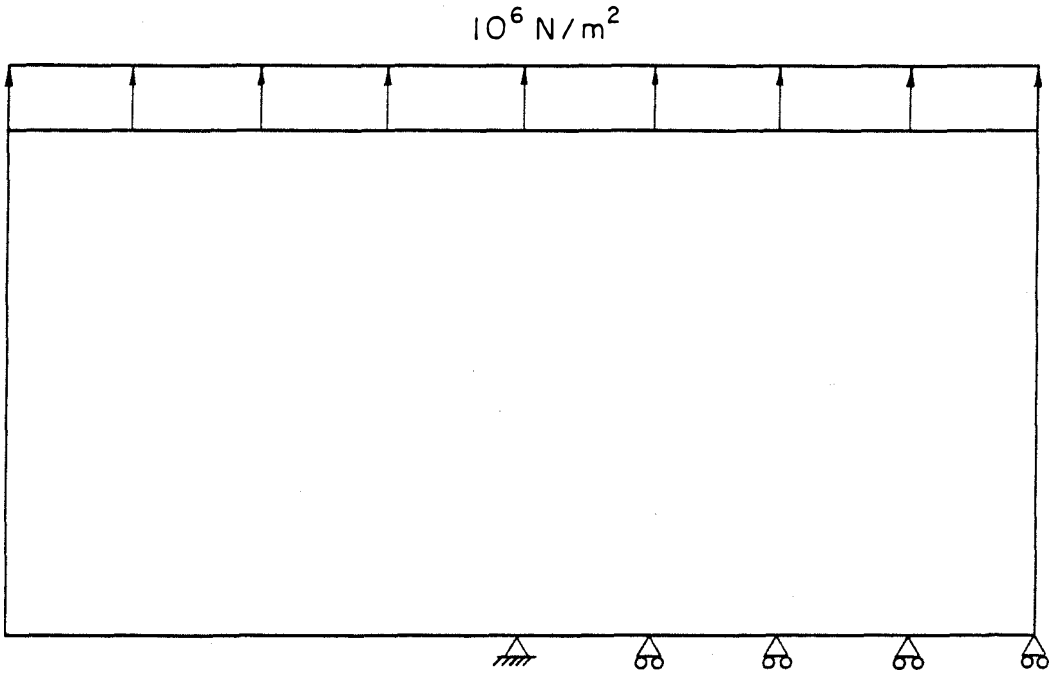
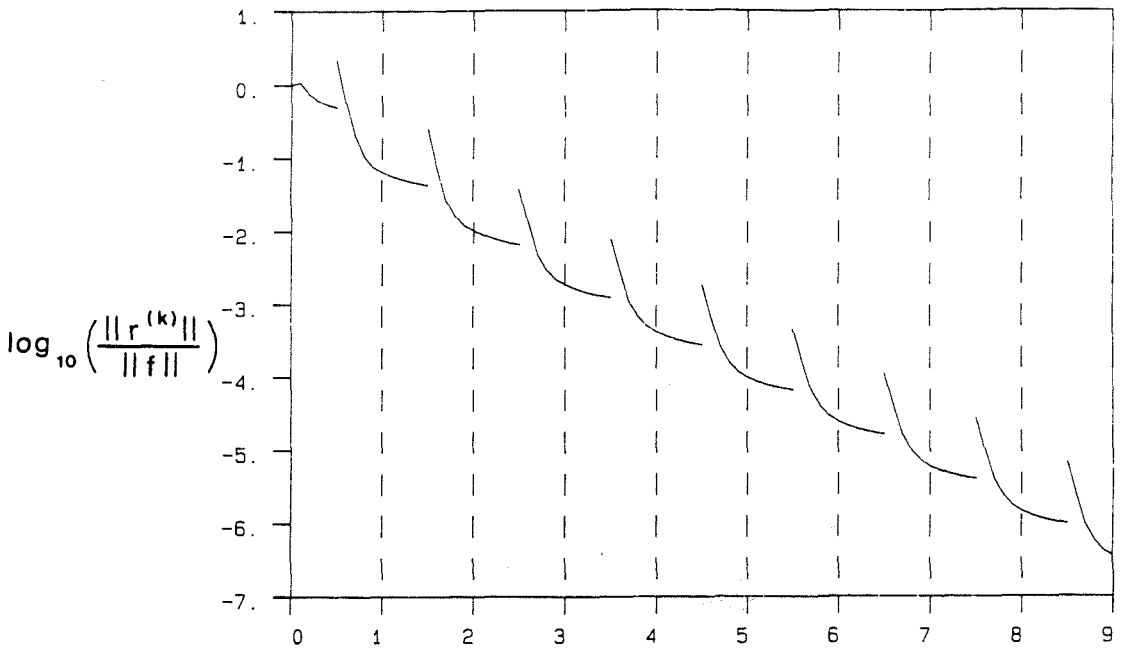


Figure 5.10: Two Dimensional Crack Problem.



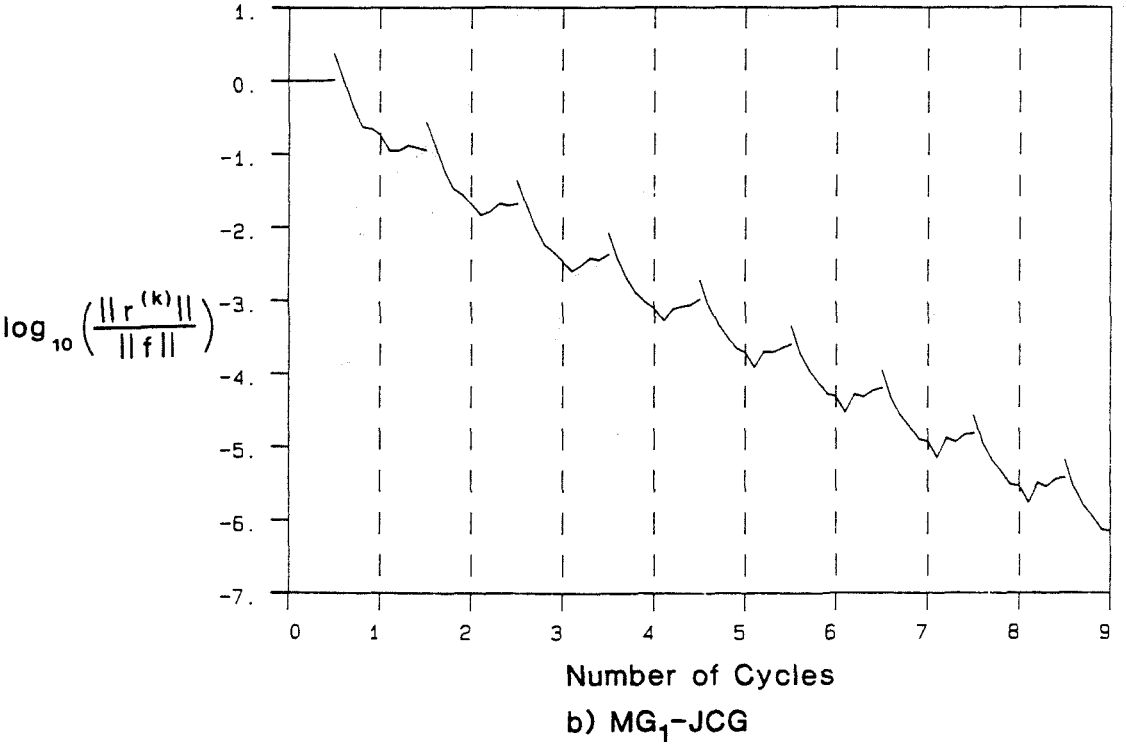


Figure 5.11: Convergence of the MG_1 ($\nu_1 = \nu_2 = 5$) Method With Different Relaxation Schemes for the Two Dimensional Crack Problem.

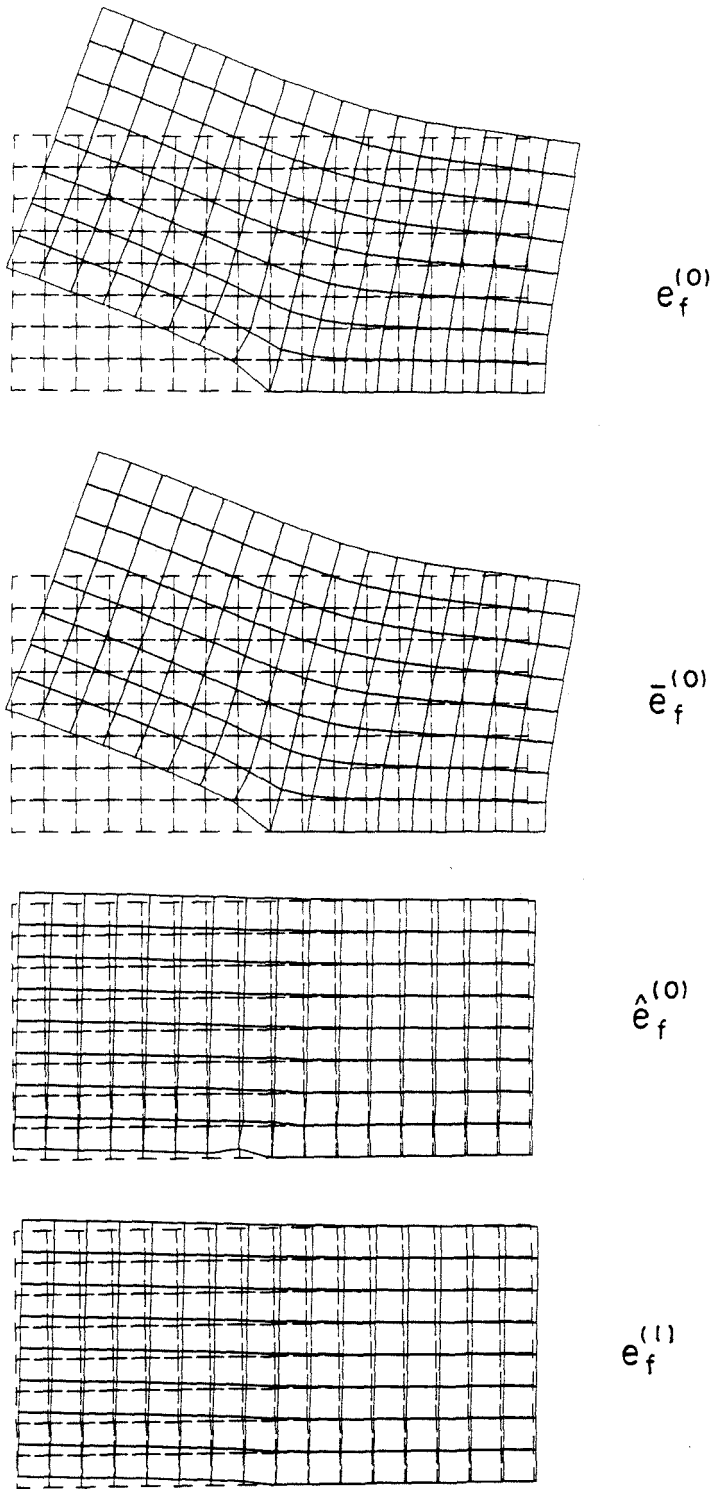


Figure 5.12: Behavior of the Fine Mesh Error in the MG_1-GS ($\nu_1 = \nu_2 = 5$) Method for the Two Dimensional Crack Problem.

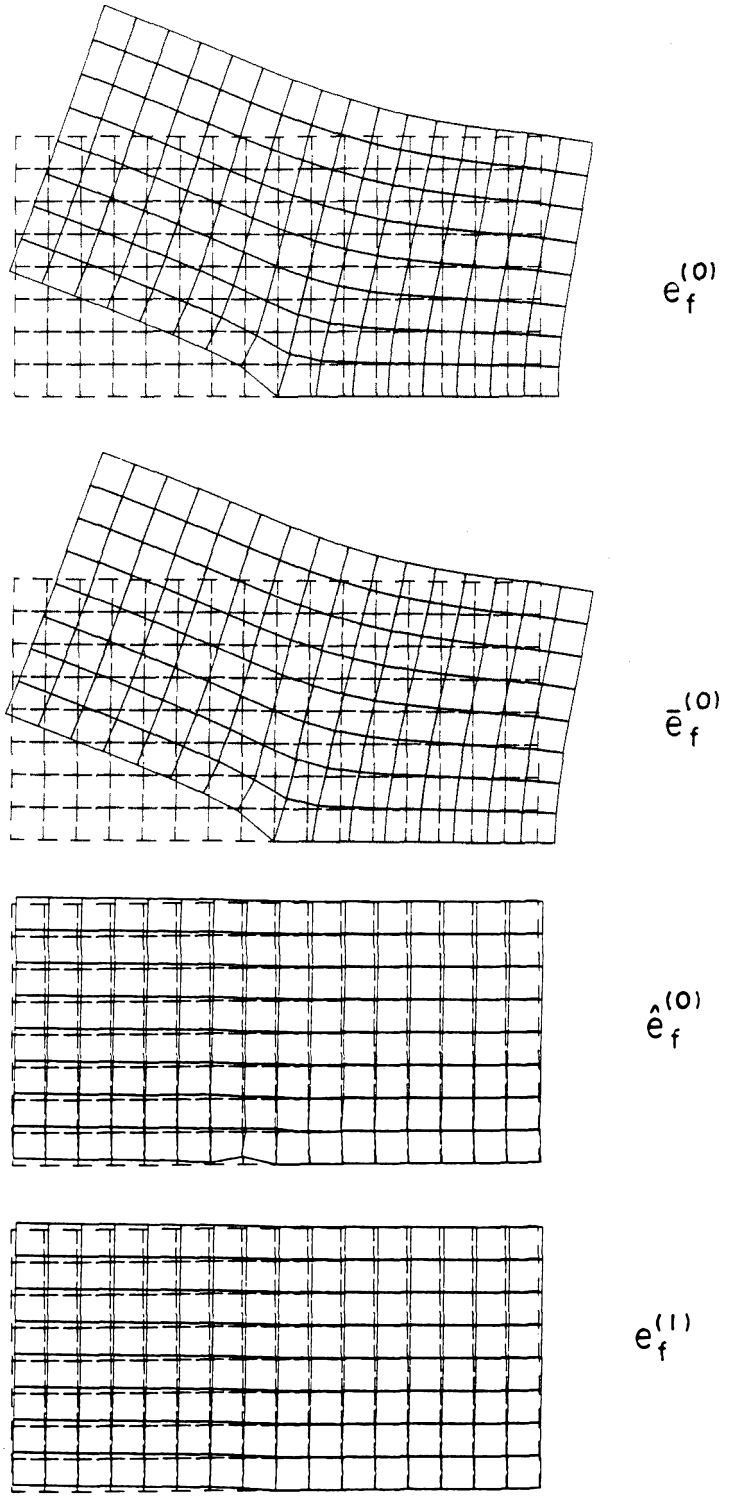


Figure 5.13: Behavior of the Fine Mesh Error in the MG_2-GS ($\nu_1 = \nu_2 = 5$) Method With Reduced Integration on the $\mathbf{D}_\lambda^e$ Term for the Two Dimensional Crack Problem.

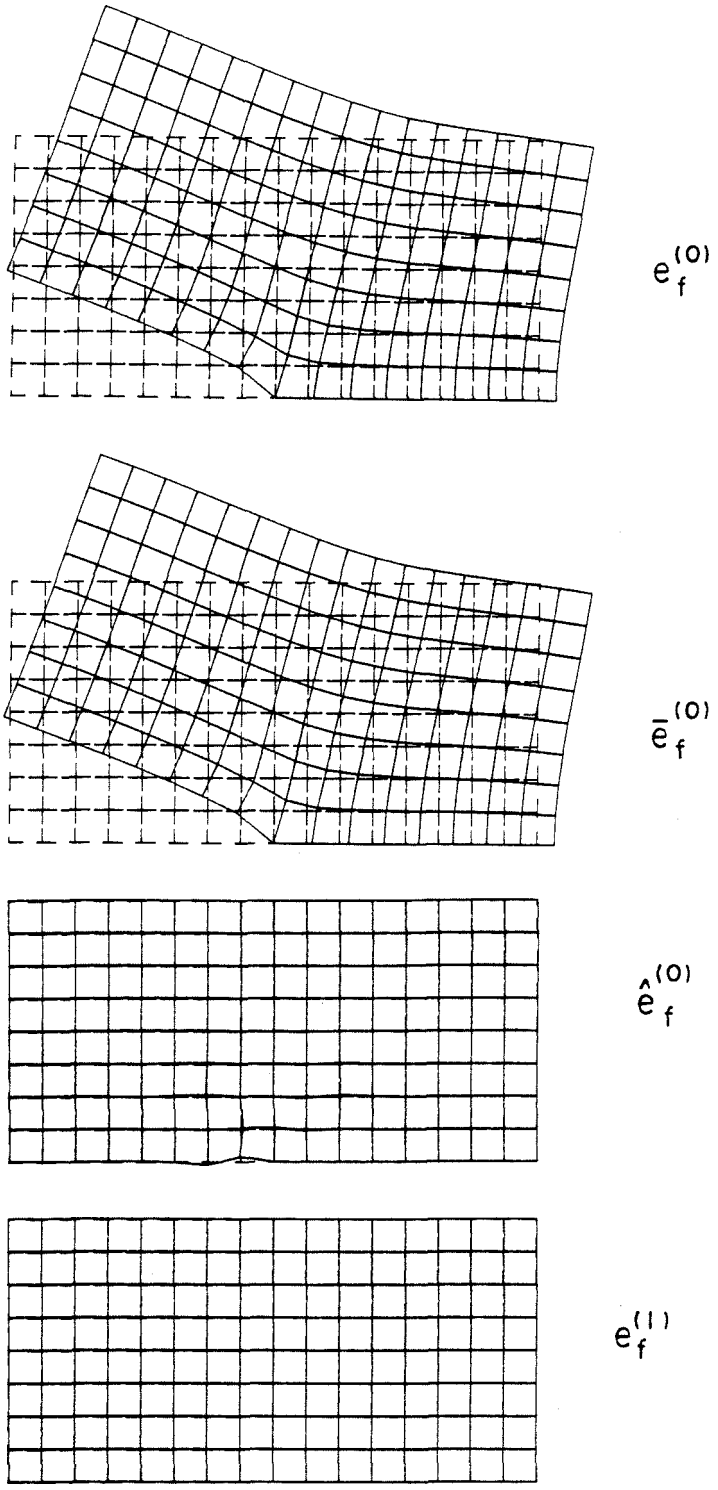


Figure 5.14: Behavior of the Fine Mesh Error in the MG_2-GS ($\nu_1 = \nu_2 = 5$) Method With Reduced Integration on the $\mathbf{D}_\mu^e$ Term for the Two Dimensional Crack Problem.

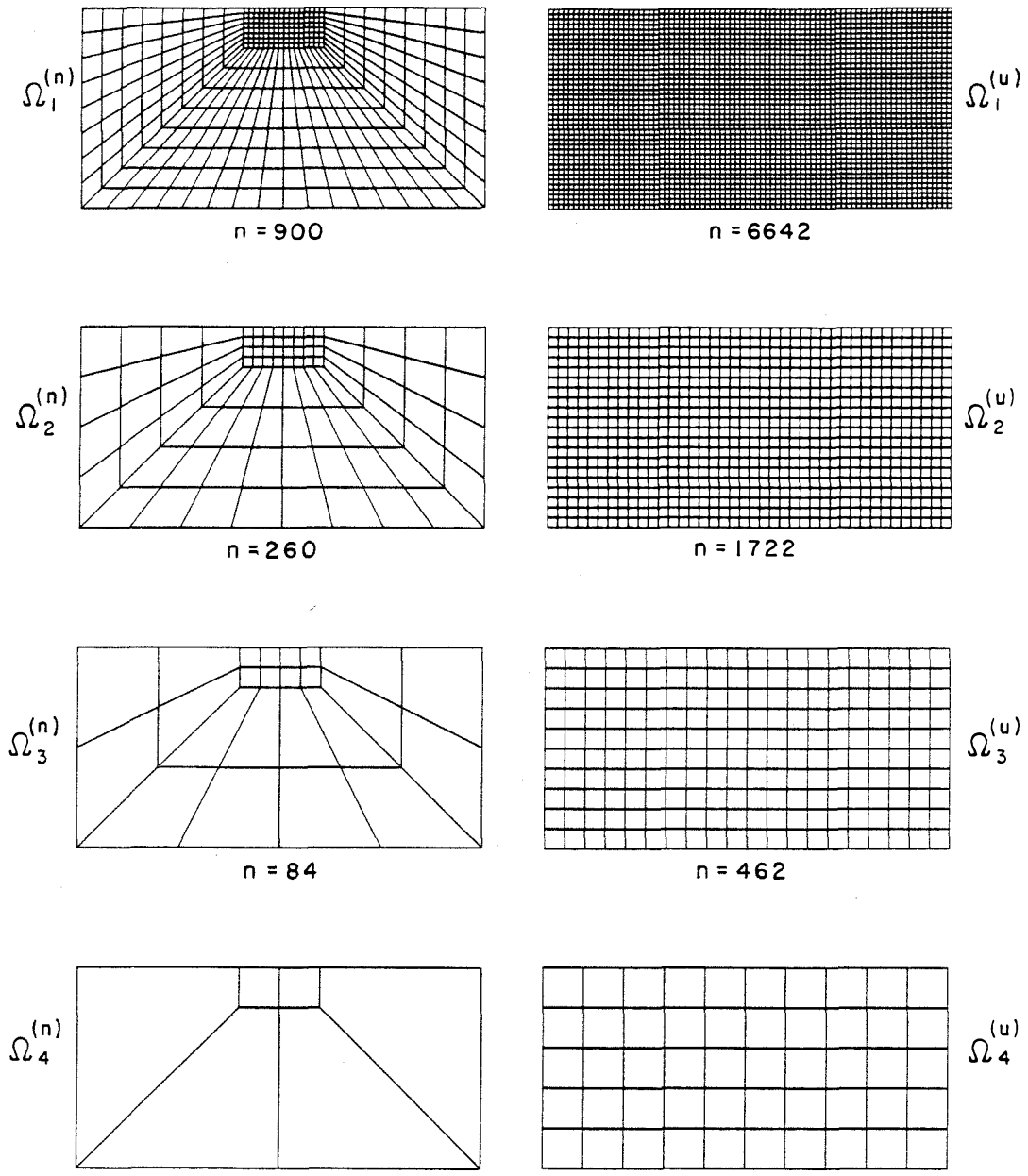


Figure 5.15: The Uniform ($\Omega_i^{(u)}$) and the Nonuniform ($\Omega_i^{(n)}$) Meshes Used to Solve the Point Load Problem.

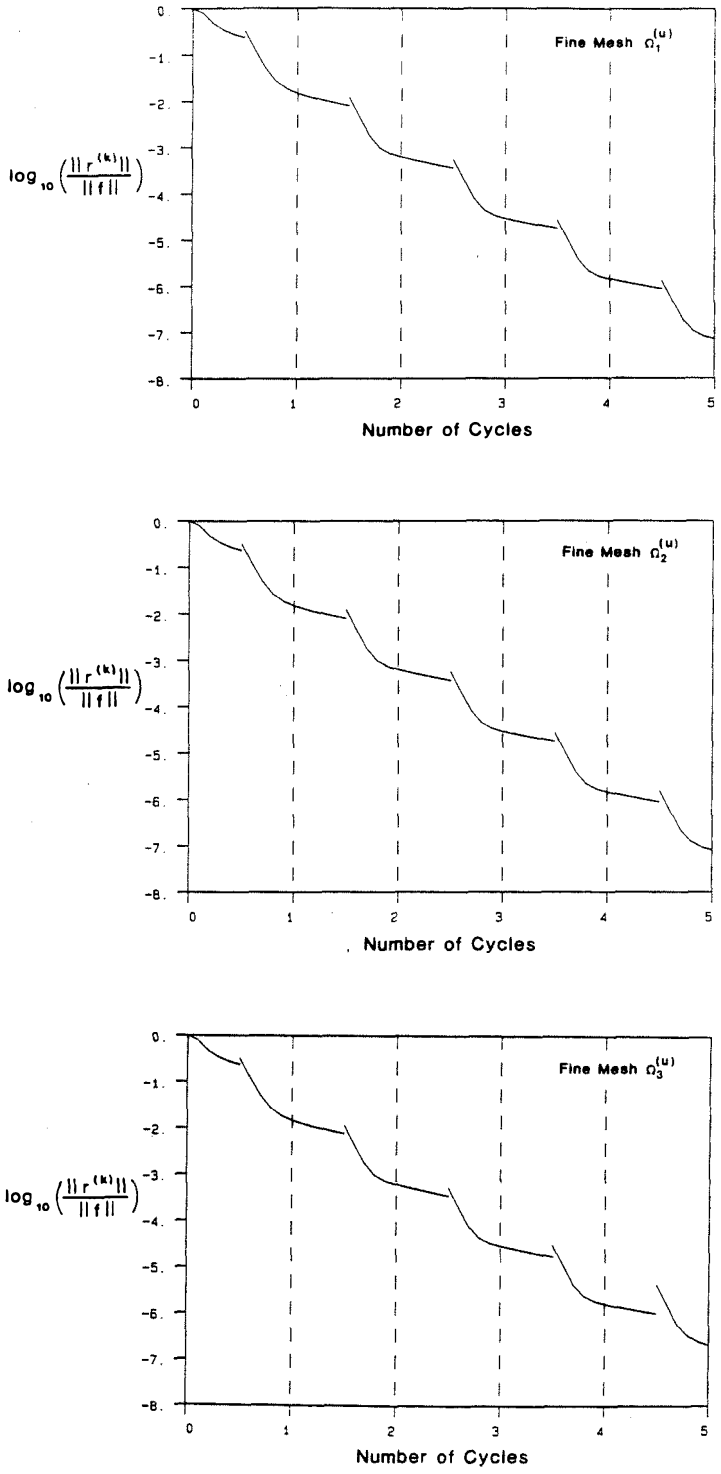


Figure 5.16: Convergence in Terms of the Residual for the Uniform Meshes ($\Omega_i^{(u)}$) Used to Solve the Point Load Problem Using the MG_1-GS ($\nu_1 = \nu_2 = 5$) Method.

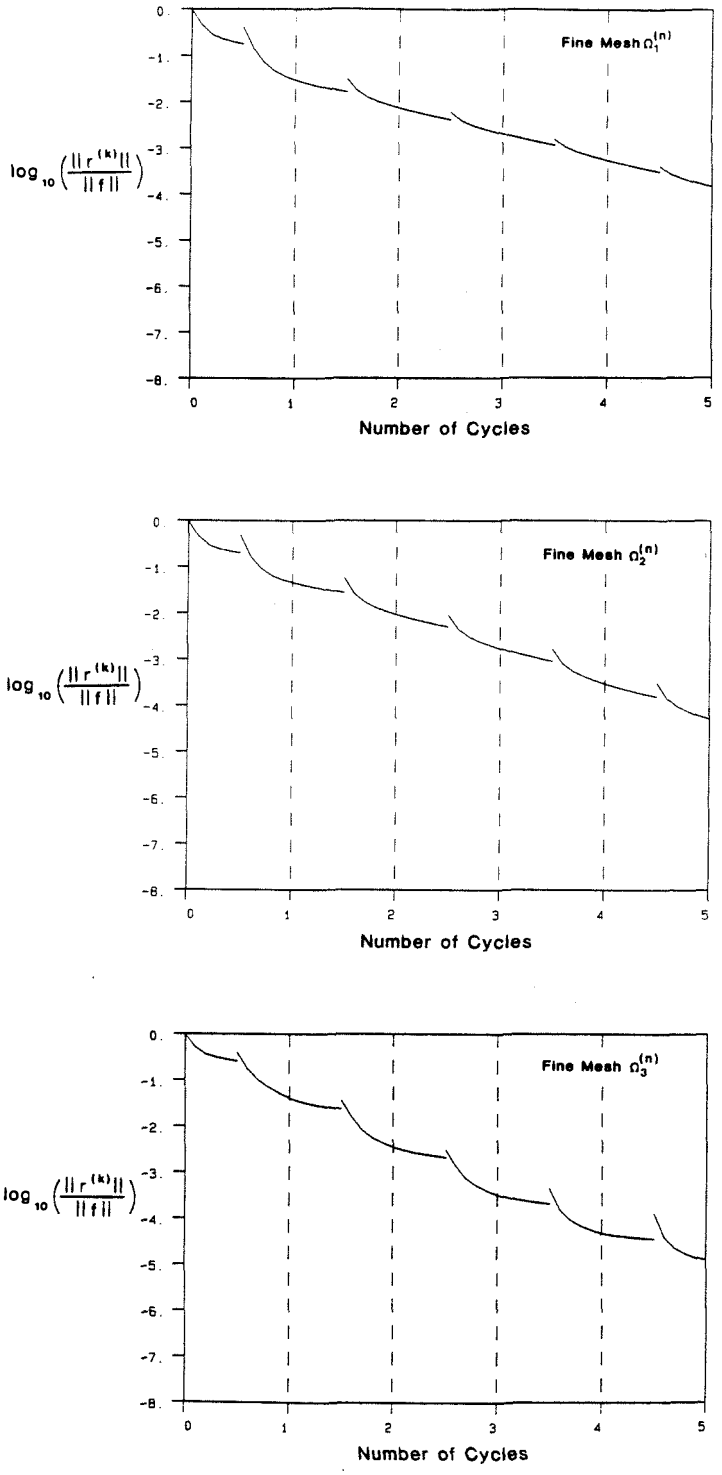


Figure 5.17: Convergence in Terms of the Residual for the Nonuniform Meshes ($\Omega_i^{(n)}$) Used to Solve the Point Load Problem Using the MG_1-GS ($\nu_1 = \nu_2 = 5$) Method.

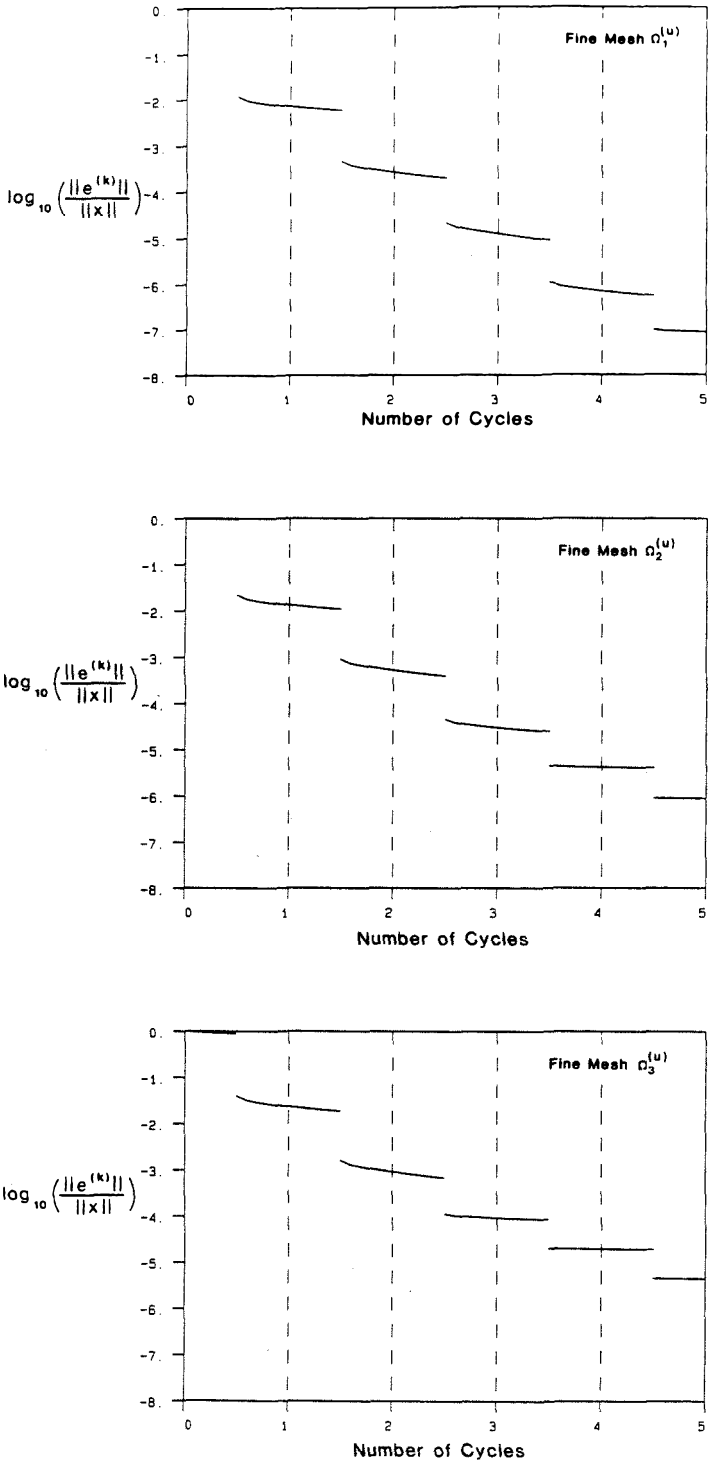


Figure 5.18: Convergence in Terms of the Error for the Uniform Meshes ($\Omega_i^{(u)}$) Used to Solve the Point Load Problem Using the MG_1-GS ($\nu_1 = \nu_2 = 5$) Method.

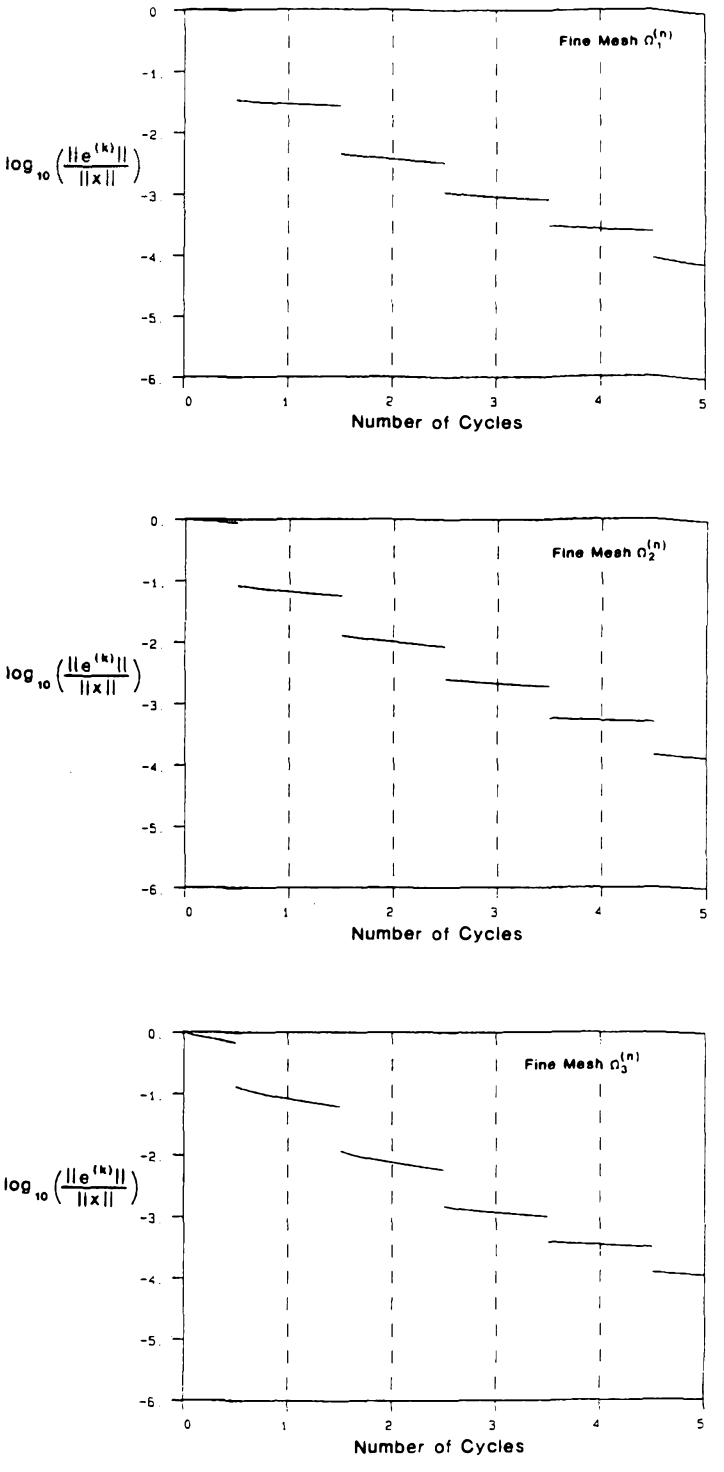


Figure 5.19: Convergence in Terms of the Error for the Nonuniform Meshes ($\Omega_i^{(n)}$) Used to Solve the Point Load Problem Using the MG_1-GS ($\nu_1 = \nu_2 = 5$) Method.

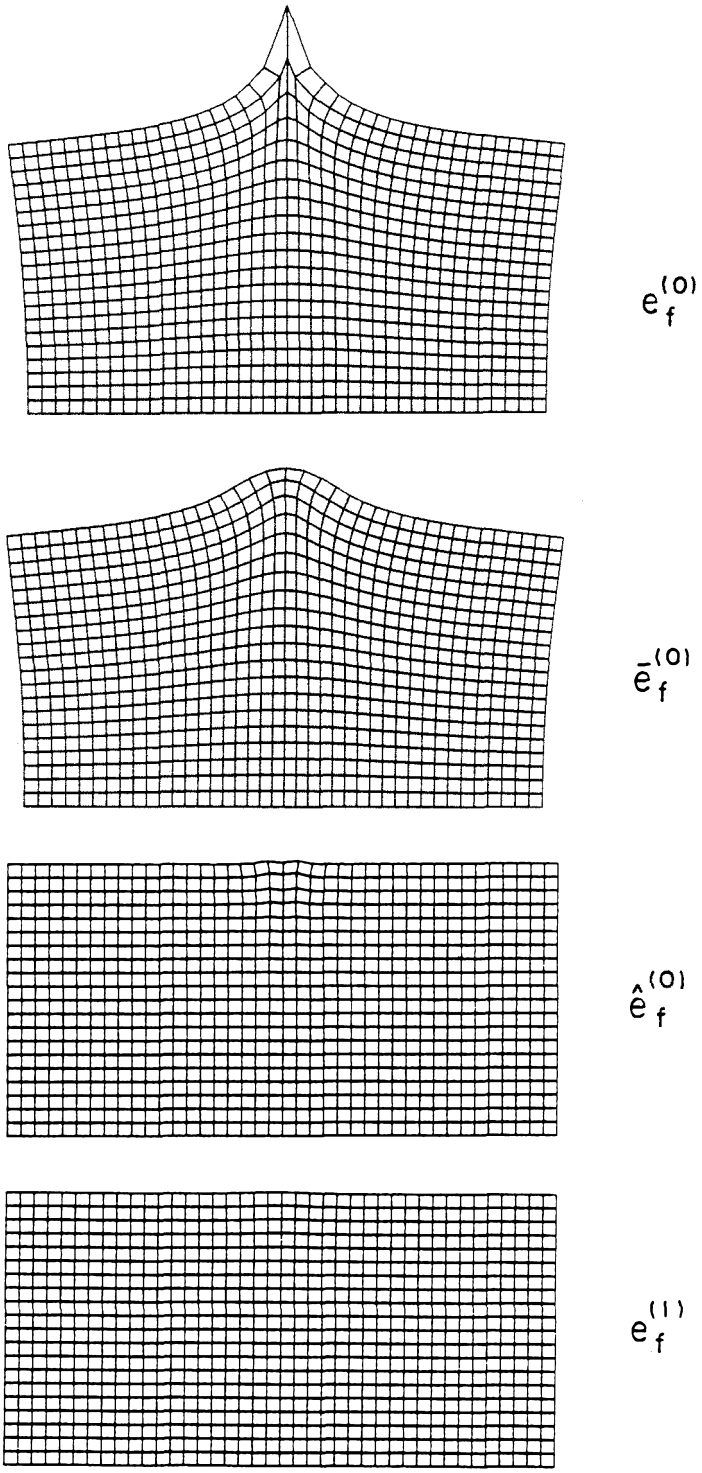


Figure 5.20: Behavior of the Fine Mesh Error on the Uniform Mesh $\Omega_2^{(u)}$ for the Point Load Problem Using the MG_1-GS ($\nu_1 = \nu_2 = 5$) Method.

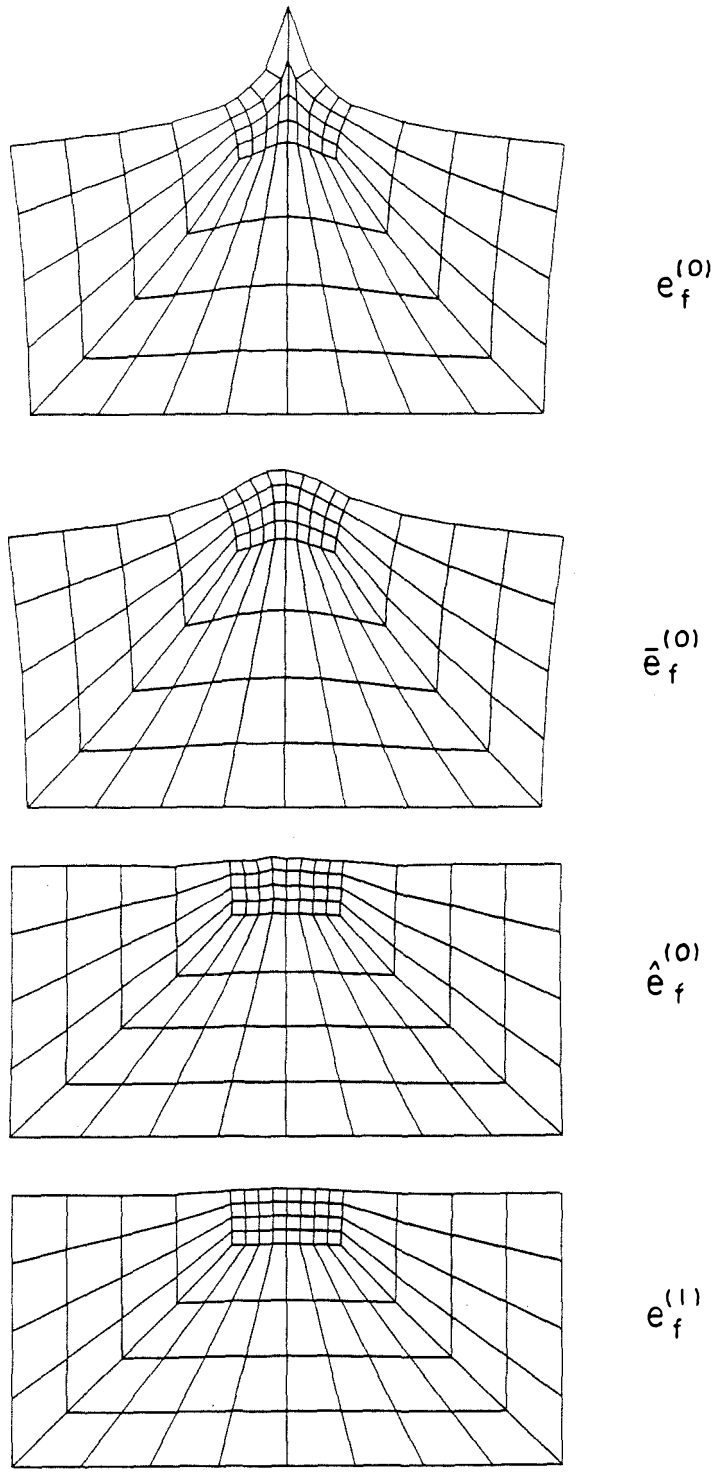


Figure 5.21: Behavior of the Fine Mesh Error on the Nonuniform Mesh $\Omega_2^{(n)}$ for the Point Load Problem Using the MG_1-GS ($\nu_1 = \nu_2 = 5$) Method.

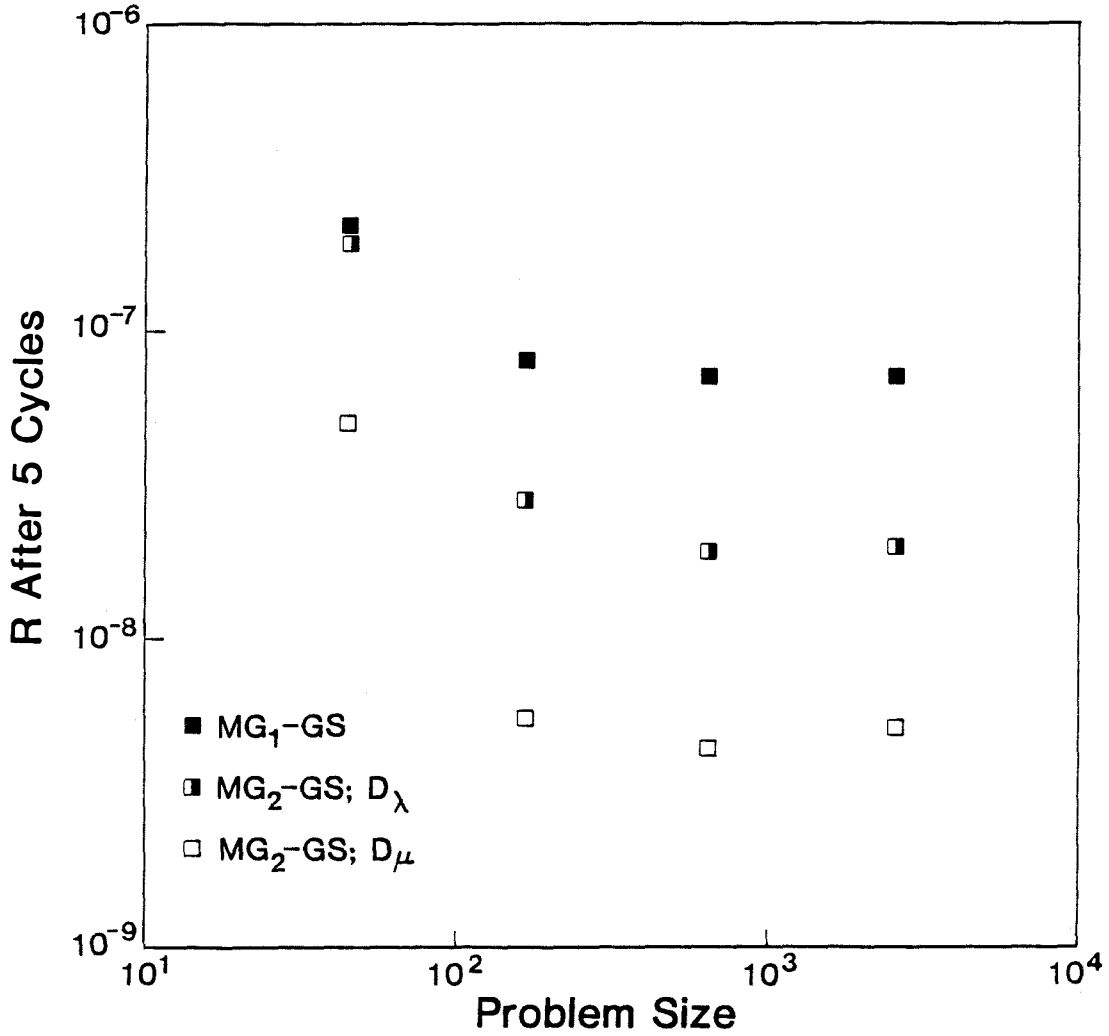


Figure 5.22: Convergence of the MG-GS ($\nu_1 = \nu_2 = 5$) Methods on the Uniform Meshes for Different Problem Sizes.

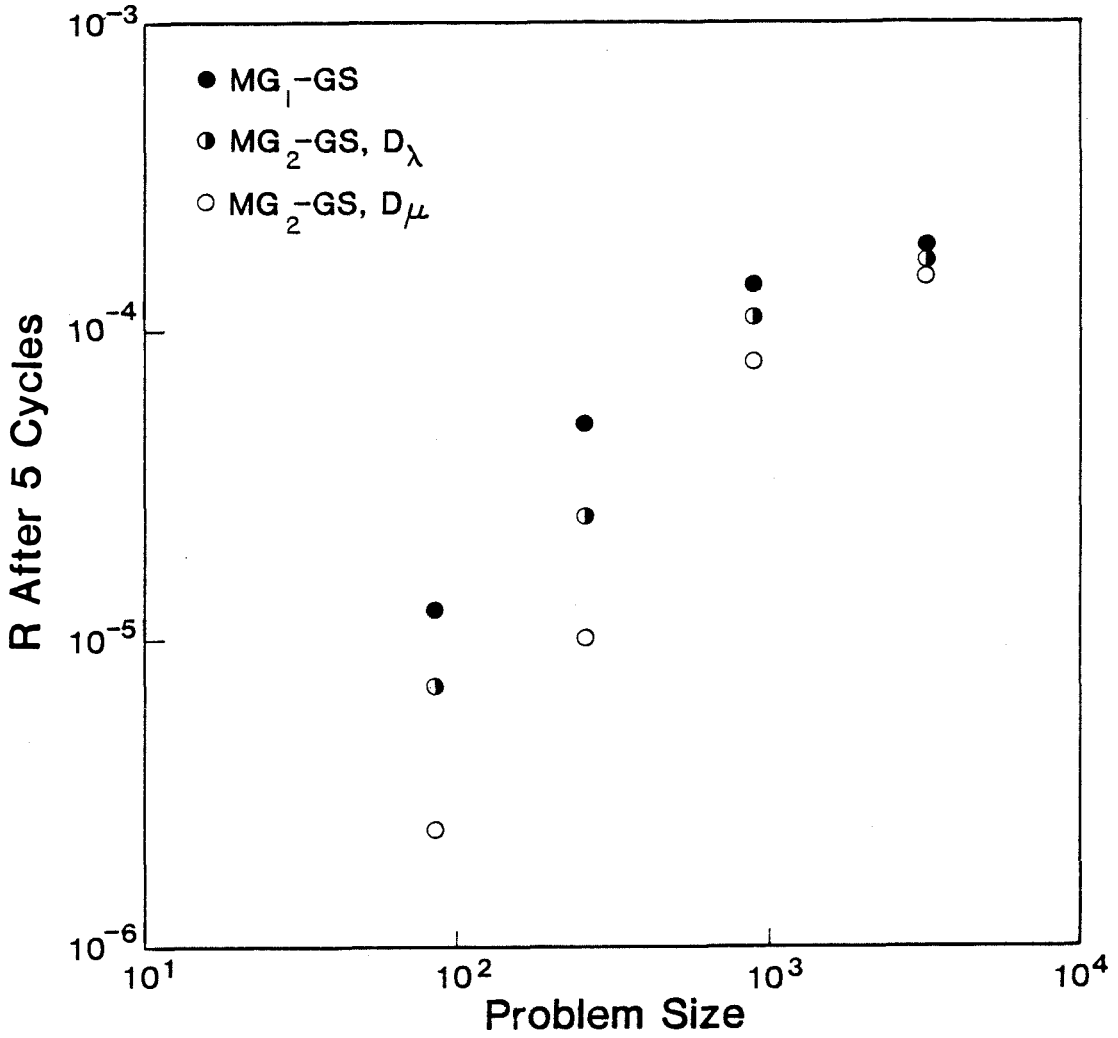


Figure 5.23: Convergence of the MG-GS ($\nu_1 = \nu_2 = 5$) Methods on the Nonuniform Meshes for Different Problem Sizes.

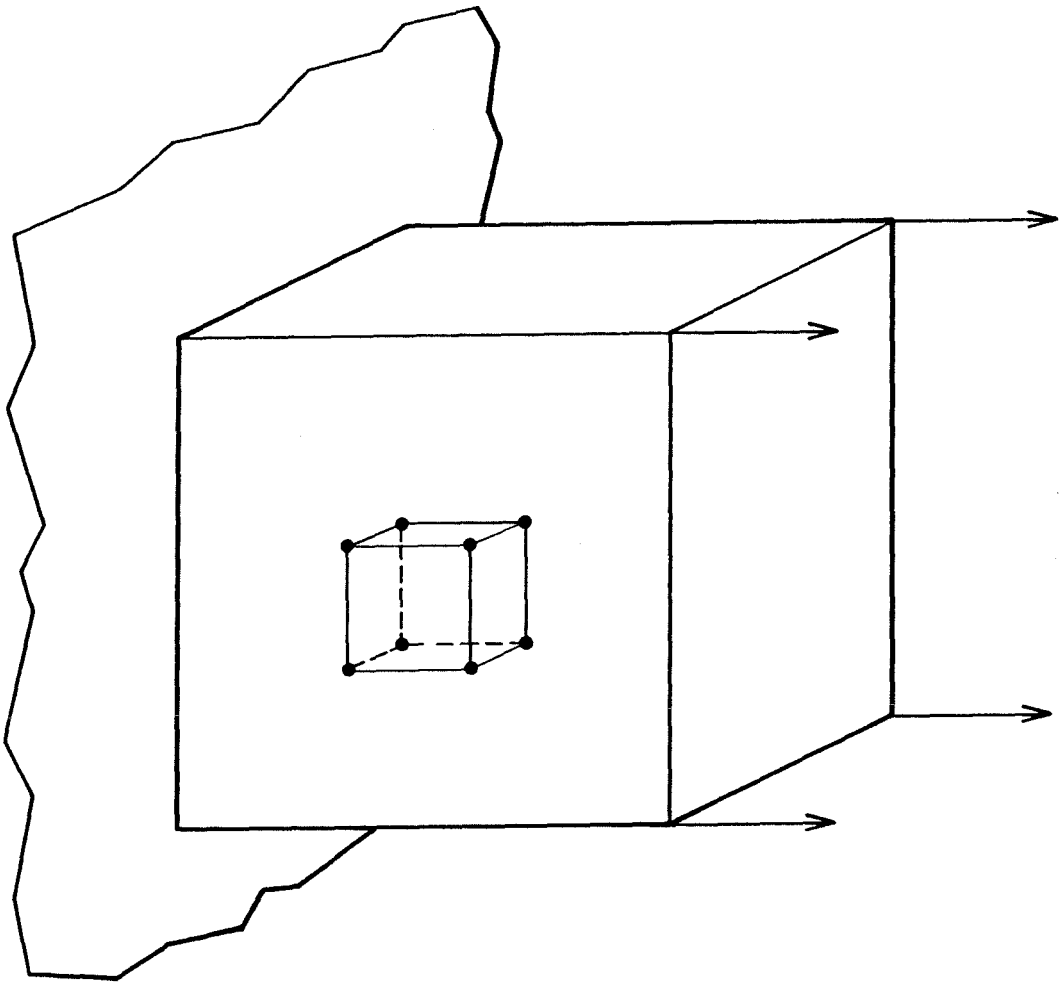


Figure 5.24: Three Dimensional Test Problem.

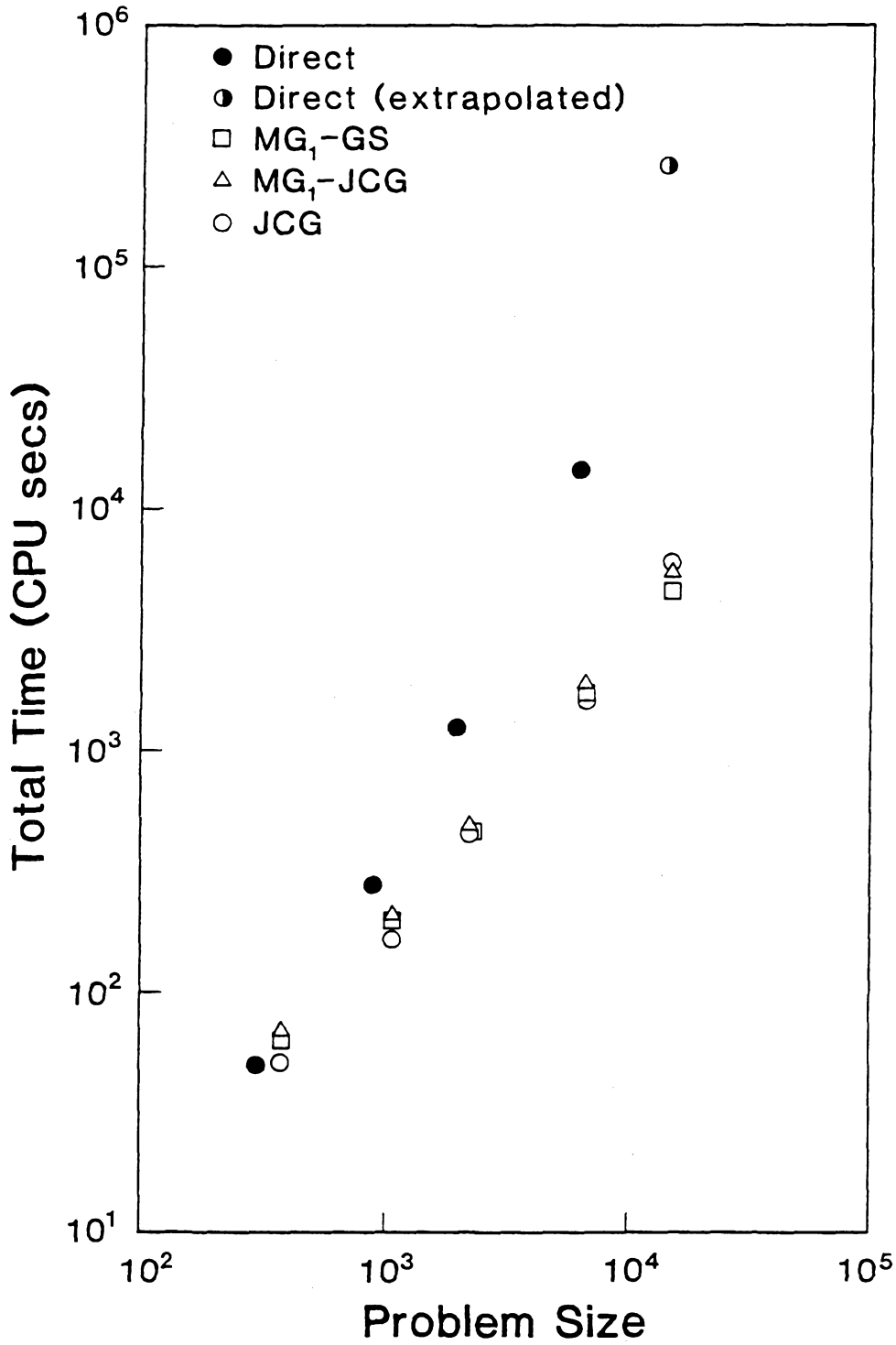


Figure 5.25: Total Computation Time for the Three Dimensional Test Problem.

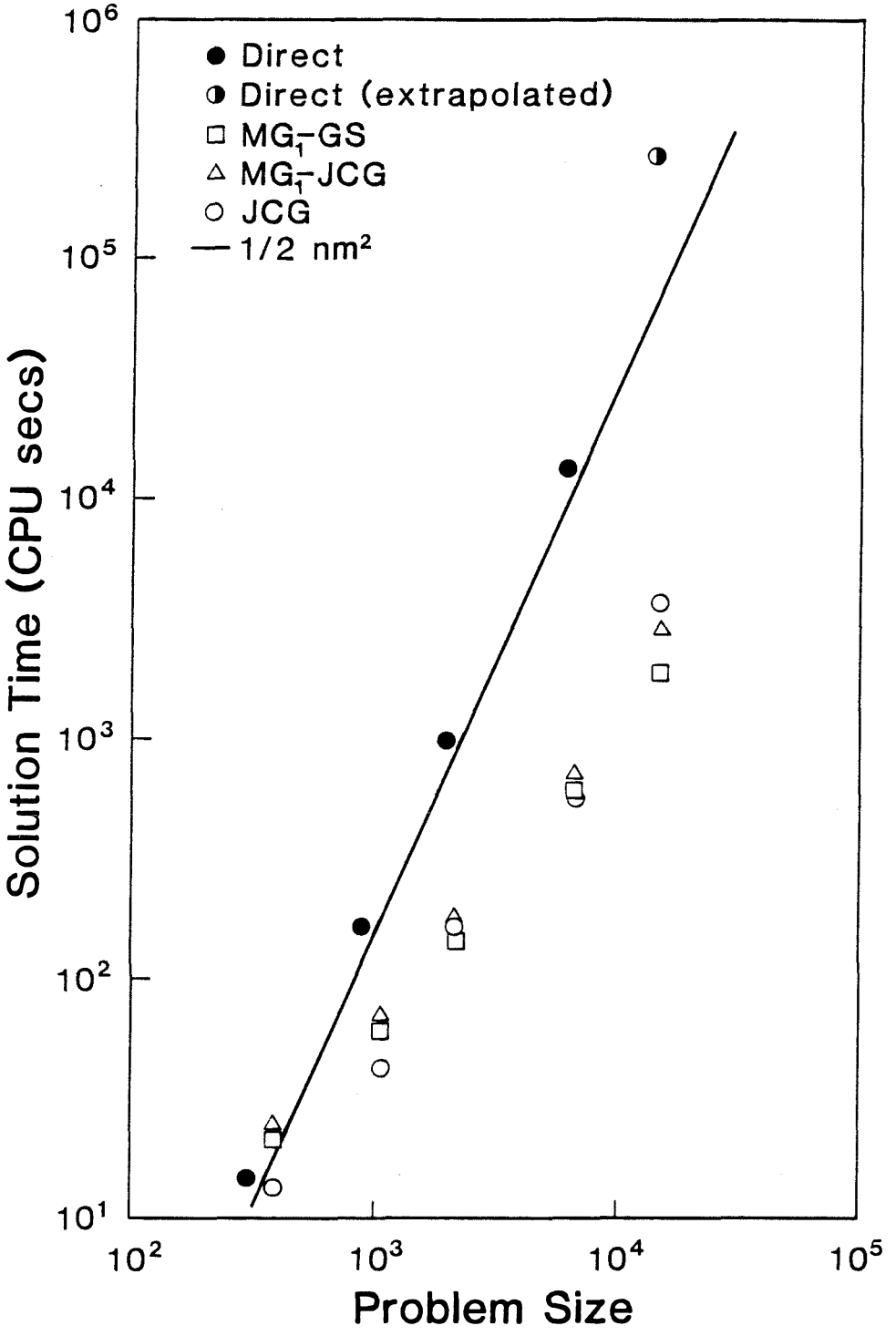


Figure 5.26: Solution Time for the Three Dimensional Test Problem.

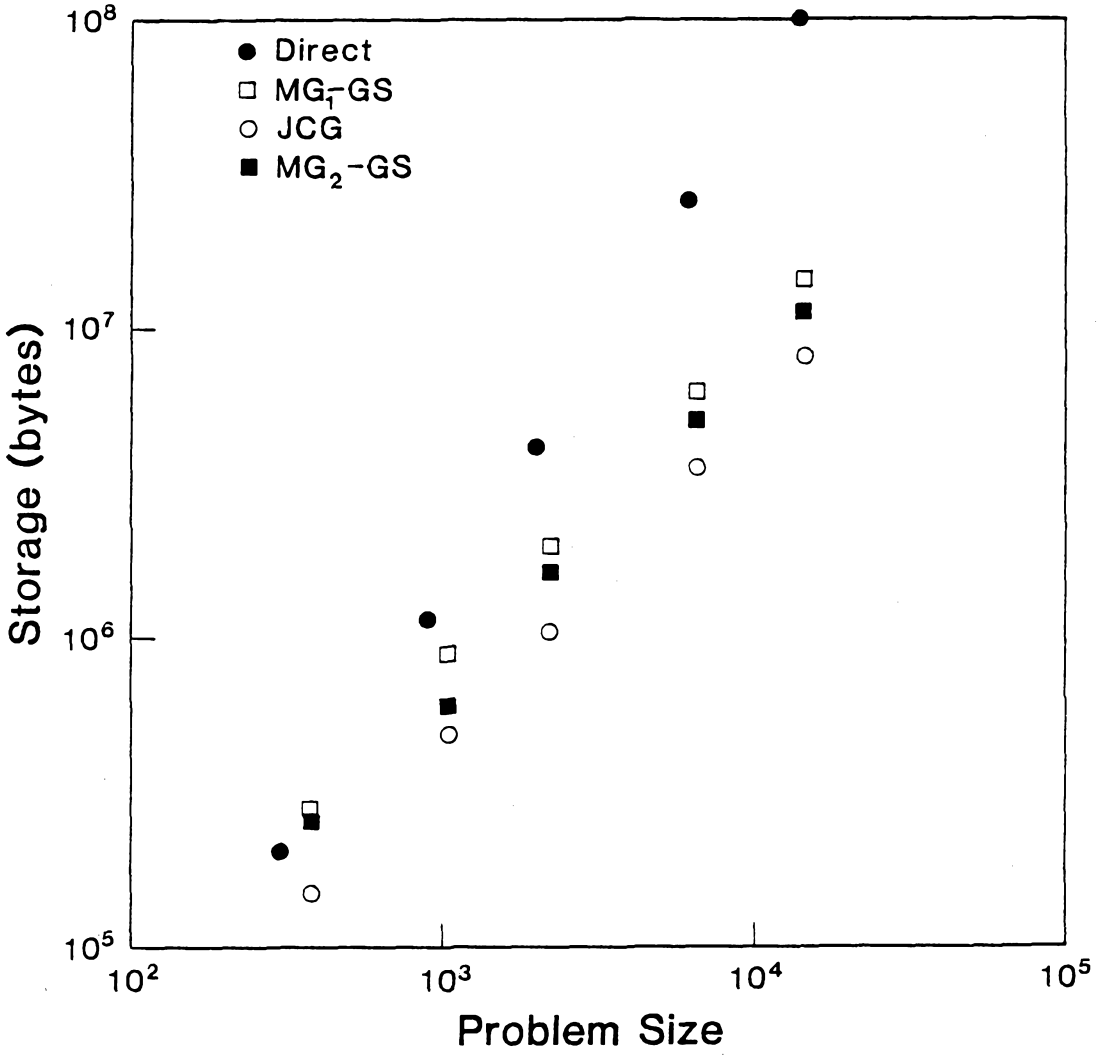


Figure 5.27: In-Core Storage Required for the Three Dimensional Test Problem.

Part II

The Solution of Some Large Solid Mechanics Problems of Practical Interest

Chapter 6

The Elastostatic State Near a Three Dimensional Edge Crack

Part II of this thesis describes the way in which the multigrid method was used to solve two classes of problems of practical engineering interest, and also presents the results that were obtained. In this chapter, the finite element method is used with the multigrid method to investigate the elastostatic state near a three dimensional edge crack. In section 6.1, a description of the problem considered is given, together with some theoretical considerations (including a brief summary of some previous analytical and numerical studies of three dimensional linear elastic fracture mechanics problems). The details of the way in which the finite element method was used to solve this problem, and the performance of the multigrid method, are given in section 6.2. This section describes how the optimum multigrid method was determined for the problem. Section 6.3 presents a detailed discussion of the elastostatic state near the crack front.

Before proceeding with the remainder of the chapter, the reader is referred to the discussion of the plane problem in linear elastostatics given in appendix A. This appendix explains the terms plane strain, the residual problem of plane

strain, generalized plane stress, and plane stress, which are all used in the following discussions.

6.1 Problem Description and Theoretical Considerations

The loading and geometry of the crack problem that was investigated is shown in figure 6.1. A square plate with a crack through half of its mid-plane was loaded with uniform tractions normal to the crack face applied at opposite ends of the plate. This is sometimes called the single edge notch test specimen [44]. The crack was assumed to be mathematically sharp with a straight crack front. Figure 6.1 shows the dimensions of the plate, the loading conditions, and the coordinate system, with its origin situated at the middle of the crack front. The material was assumed to be a linear elastic, homogeneous, isotropic solid with Poisson's ratio 0.3 and Young's modulus 207 GN/m^2. The effect of body forces was ignored. The loading conditions corresponded to mode I fracture. Note that the surfaces $x_3 = \pm 0.125$ m are equivalent to π_α in the three dimensional plane problem discussed in appendix A, and that the surfaces $x_\alpha = \pm 1$ m are the same as the lateral surface $\mathcal{B}$ of the cylinder shown in figure A.1. (The subscripts i, j have the range 1,2,3, and subscripts α, β, γ have the range 1,2 in this thesis.) The plane $x_3 = 0$ corresponds to the plane π_0.

Classical linear elasticity theory shows that the leading term of the plane strain or generalized plane stress asymptotic stress field near the crack tip for

mode I fracture is given by [43]

$$\left.\begin{aligned} \sigma'_{11} = \bar{\sigma}_{11} &= \tfrac{K_I}{\sqrt{2\pi r}} \cos\tfrac{\theta}{2}(1 - \sin\tfrac{\theta}{2}\sin\tfrac{3\theta}{2}) \\ \sigma'_{22} = \bar{\sigma}_{22} &= \tfrac{K_I}{\sqrt{2\pi r}} \cos\tfrac{\theta}{2}(1 + \sin\tfrac{\theta}{2}\sin\tfrac{3\theta}{2}) \\ \sigma'_{12} = \bar{\sigma}_{12} &= \tfrac{K_I}{\sqrt{2\pi r}} \sin\tfrac{\theta}{2}\cos\tfrac{\theta}{2}\cos\tfrac{3\theta}{2} \end{aligned}\right\} + O(1),\ r \to 0 \text{ on } \pi_0, \qquad (6.1)$$

where K_I is the mode I stress intensity factor. Following the notation used in appendix A, $[u'_i, \epsilon'_{ij}, \sigma'_{ij}]$, $[\bar{u}_i, \bar{\epsilon}_{ij}, \bar{\sigma}_{ij}]$, and $[\hat{u}_i, \hat{\epsilon}_{ij}, \hat{\sigma}_{ij}]$ denote the plane strain, generalized plane stress, and plane stress solutions associated with the solution of the plane problem, respectively. Studies of the two dimensional version of the single edge notch test specimen [44] show that K_I for this geometry is given by

$$K_I = 3.0\sigma\sqrt{\pi a}, \qquad (6.2)$$

where σ is the applied traction and a is the crack length. This allows comparisons to be made between the computed three dimensional elastostatic solution and the two dimensional plane problems.

The first two dominant terms of the two dimensional elastic stress field near the crack tip are given by

$$\sigma'_{\alpha\beta} = \bar{\sigma}_{\alpha\beta} = \frac{K_I}{\sqrt{2\pi r}} f_{\alpha\beta}(\theta) + T\delta_{1\alpha}\delta_{1\beta} + o(1),\ r \to 0 \text{ on } \pi_0 \qquad (6.3)$$

where the functions $f_{\alpha\beta}$ are given in equation (6.1), T is a constant, and the remaining terms depend on r to powers of $\frac{1}{2}$ or higher [38], [60]. The nonsingular terms in equation (6.3) need to be included if a realistic comparison is to be made between the three and two dimensional problems. Since no expressions are available for these terms for the particular geometry considered, the full elastic two dimensional stress field was obtained by using the finite element method to numerically solve the generalized plane stress version of the plate problem.

The stress intensity factor can be used with the plate thickness, t, to define parameters that can be used to non-dimensionalize quantities of interest. Stresses and displacements will be made dimensionless by division by $\frac{K_I}{\sqrt{t}}$ and $\frac{\nu K_I \sqrt{t}}{E}$ respectively; strains will be normalized through division by $\frac{K_I}{E\sqrt{t}}$. However, this does not imply that field quantities are independent of Poisson's ratio. Note that all distances, such as values of x_3 and r, are normalized through division by t. In the above, t is chosen to be the length scale of the particular problem considered.

Several attempts have been made in the past to obtain analytical and numerical solutions to three dimensional crack problems. These investigations have mainly considered the problem of the normal intersection of a plane crack with a free surface. The spherical coordinate system (ρ, ϕ, ψ), centered at the crack-surface intersection, and shown in figure 6.2, is introduced to compare the numerical solution of the cracked plate problem with some of these solutions.

Benthem [9] attempted to solve the problem of a quarter-infinite crack in a half-space by assuming a solution of the form

$$\sigma_{ij} = \rho^{\lambda} f_{ij}(\lambda, \phi, \psi), \tag{6.4}$$

where λ depends on Poisson's ratio. For $\nu = 0.3$, Benthem found that the dominant value of λ was -0.452.

A plate of finite thickness which contains a finite, through the thickness center crack was considered by Folias [21]. Folias not only examined the nature of the stresses at the crack-surface intersection, but also in the plate's interior. In the interior of the plate, all of the stresses were found to be singular of order $\frac{1}{2}$, except for σ_{23} and σ_{31}, which were bounded. In the vicinity of the crack-surface

intersection, the stresses and the displacements were proportional to $\rho^{-\frac{1}{2}-2\nu}$ and $\rho^{\frac{1}{2}-2\nu}$, respectively. This strong singularity in the stresses and the possibility of unbounded displacements at the crack-surface intersection has caused some discussion [8,22].

Three dimensional crack problems have also been investigated using combined analytical and numerical schemes. Bazant [7] solved the problem of a crack in a half-space using a method that consisted of a separation of variables of the form

$$u_i = \rho^{\mu} F_i(\phi, \psi), \tag{6.5}$$

coupled with a numerical method for a boundary value problem on a unit sphere around the crack-surface intersection point. The strengths of the stress singularities were found to agree closely with Benthem's.

In [14], the finite element method was used to solve the problem of a through the thickness center crack in a rectangular plate. Singular elements were used near the crack front that produced singular stress behavior of the form

$$\sigma_{ij} = O(r^{-\frac{1}{2}}) \text{ as } r \to 0 \tag{6.6}$$

in the plate's interior, and

$$\sigma_{ij} = O(\rho^{\lambda}) \text{ as } \rho \to 0 \tag{6.7}$$

at the crack-surface intersection. The singularity exponent λ was determined by treating it as an additional parameter to aid in the minimization of potential energy. For $\nu = 0.3$, λ was found to be -0.445.

Despite a great deal of work, it is fair to say that the elastostatic state near a three dimensional crack is not as well understood as the two dimensional

problem. None of the published analytical solutions have uniqueness proofs, and so separable solutions such as equation (6.4) are not strictly justified. Therefore the detailed numerical solution presented in this chapter (which is unbiased either by separability assumptions or by the use of singular elements) has great value as a means of understanding the nature of the elastostatic state near a three dimensional crack.

6.2 The Finite Element Solution of the Cracked Plate Problem

This section describes the way in which the finite element method was used with the multigrid method to solve the cracked plate problem. The cracked plate is a good test problem not only because of its engineering value, but also because it requires a mesh that has a large number of small elements near the crack front to capture the behavior of the singular stresses and strains. Consequently, the mesh will have a large number of degrees-of-freedom and will produce an ill-conditioned stiffness matrix (resulting from the small size of elements close to the crack front). The symmetry of the problem allows only one quarter of the plate to be modeled (e.g. the quarter given by $x_2 \geq 0$ and $x_3 \geq 0$). The boundary conditions that were applied to the finite element mesh covering this region consisted of the specified tractions and specified zero displacements that represented the symmetry conditions and prevented a rigid body motion of the mesh.

These boundary conditions are shown in figure 6.3. A uniform traction of

1 MN/m^2 acting in the positive x_2-direction was applied to the upper surface of the mesh since this was the load applied to the cracked plate. The symmetry of the problem required that the nodes on the back surface of the mesh (i.e. on the $x_3 = 0$ plane) were free to move in the x_1- and x_2-directions but were fixed in the x_3-direction, and that the nodes on the right-hand side of the lower surface (i.e. on the $x_2 = 0$ plane with $x_1 \geq 0$) were free to move in the x_1- and x_3-directions but were fixed in the x_2-direction. These two sets of displacement boundary conditions implied that the nodes on the non-negative x_1-axis were free to move only in the x_1-direction. In order to prevent a rigid body motion of the mesh (which would have produced a singular stiffness matrix), one of the points on the non-negative x_1-axis had to be fixed in all directions. As a matter of convenience, this point was chosen to be the origin of the coordinate system (i.e. the point $x_1 = x_2 = x_3 = 0$).

Two meshes of different sizes were used to represent the plate. The first mesh, which had 25,515 degrees-of-freedom, was used to determine the optimum multigrid parameters (i.e. γ, ν_1 and ν_2) for this particular problem. A second, larger mesh (61,659 degrees-of-freedom) was then used to obtain a detailed numerical solution of the problem. The performance of the multigrid method when used with these two meshes is now discussed.

6.2.1 The Preliminary Investigation

The mesh that was used to evaluate the multigrid method is shown in figure 6.4, together with the three coarse meshes used in the multigrid solution scheme. These meshes each had 25,515, 3,735, 621, and 126 degrees-of-freedom, and were

automatically generated by repeatedly subdividing the elements in the coarsest mesh. $3 \times 3 \times 3$ Gauss integration was used on all terms when computing the finest mesh element stiffness matrices.

The problem was solved using the MG_1-GS method; various values of γ, ν_1, and ν_2 were used in an attempt to find the optimum multigrid strategy. Table 6.1 shows the number of multigrid cycles required for convergence of the MG_1-GS method using different values of these parameters. It is, however, more meaningful to compare the speed of these different schemes by examining the computational time (in terms of CPU secs) taken for convergence of the methods since the time taken by each cycle depends on γ, ν_1, and ν_2. Table 6.1 and figure 6.5 shows the effect the different values of γ, ν_1, and ν_2 had on the speed of convergence of the MG_1-GS multigrid method. This suggests that this problem is best solved using the multigrid method by setting $\gamma = 3$ and $\nu_1 = \nu_2 = 5$.

The choice of ν_1 and ν_2 indicates that the error before interpolation and after restriction between meshes is effectively smoothed in five cycles of Gauss-Seidel relaxation. The sensitivity of the speed of convergence on the choice of γ can be attributed to the bending behavior of the coarse mesh elements. A coarse mesh eight node brick element will be too stiff when subjected to a bending deformation. This means that the estimate of the low frequency error on the fine mesh given by a coarse mesh will be too small; however, this can be improved by performing more multigrid cycles on the coarse mesh to solve the coarse mesh correction equation (equation (4.2)). Thus higher values of γ can be expected to reduce the total number of multigrid cycles required to solve the problem. Table 6.1 shows that this is indeed the case. The additional cost per cycle of the $\gamma = 4$

and 5 schemes offsets their lower number of cycles when compared to the $\gamma = 3$ scheme.

Three coarse meshes were used in the multigrid solution so that the coarsest mesh stiffness matrix could be factorized cheaply. This was done in only 0.01% of the time required for the convergence of the fastest multigrid scheme (1.45×10^4 CPU secs for $\gamma = 3$, $\nu_1 = \nu_2 = 5$). The times taken to calculate the finest mesh stiffness matrix and the coarse mesh stiffness and interpolation matrices were 30% and 4% of the time required by the fastest multigrid scheme, respectively. This demonstrates that the extra effort required to compute the data associated with the coarse meshes was not excessive.

The problem was also solved on the finest mesh using the Jacobi preconditioned conjugate gradient method (JCG) so that the multigrid method could be compared with another fast iterative method. Figure 6.6 shows the convergence behavior of this method; a total of 635 cycles were required, which took 3.37×10^4 CPU secs. Thus the JCG method took 2.3 times as long to converge as the fastest multigrid method. However, the total time (i.e. the solution time plus the time needed to compute the various matrices required by the solution scheme) of the JCG method was 2.0 times the total time of the fastest multigrid method. This reflects the extra computational time associated with the coarse mesh data. The in-core storage required by all the multigrid methods was 2.4×10^7 bytes; the JCG method needed 1.4×10^7 bytes of storage (60% of the multigrid storage). This difference is due to the storage required by the coarse meshes.

Summarizing the data presented in this section, it is apparent that the speed of the MG_1-GS multigrid method when applied to the cracked plate problem

is sensitive to the choice of γ, ν_1, and ν_2 The choice $\gamma = 3$, $\nu_1 = \nu_2 = 5$ seems to produce the fastest method. This method is about twice as fast as the JCG method, but the JCG method requires only 60% of the in-core storage needed by the multigrid method.

6.2.2 The Final Investigation

An even finer mesh was generated to investigate the elastostatic state near the crack front. This mesh and the three coarse meshes used in the multigrid solution scheme are shown in figure 6.7; the number of degrees-of-freedom in each mesh was 61,659, 8,559, 1,305, and 234. The MG_1-GS multigrid method with $\gamma = 3$ and $\nu_1 = \nu_2 = 5$ was chosen since this had been found to be the fastest method in the preliminary investigation. Figure 6.8 shows the convergence behavior of this method. It was found that 80 multigrid cycles were required, which took 1.72×10^5 CPU secs; the total time was 1.85×10^5 CPU secs (about 51 CPU hours). The in-core storage that was needed was 6.1×10^7 bytes.

The finest mesh used to investigate the elastostatic state near the crack front shown in figure 6.7 consisted of 61,659 degrees-of-freedom and 18,432 brick elements, and had a minimum element dimension to mesh thickness (half the plate thickness) ratio of $\frac{1}{16}$. Therefore the closest center Gauss point (the point at which element stresses and strains can be most accurately computed) was 2.2% of the plate thickness away from the crack front. $3 \times 3 \times 3$ Gauss integration was used on all terms when computing the element stiffness matrices.

The sequence of meshes shown in figure 6.7 were also used to estimate the discretization error present in the final computed displacements. The crack

problem was solved on the three meshes that had 16, 8, and 4 elements through the thickness. Figure 6.9 shows $u_3(r, \theta = 0°, x_3 = t/2)$, the out-of-plane displacements of the free surface along the line $\theta = 0°$, computed using each of these three meshes. The difference between the results obtained using the meshes with 16 and 8 elements through the thickness is small. The theoretical convergence rate of the displacements computed with the linear brick elements is quadratic (which means that the error will be reduced by a factor of four for a single mesh refinement). However, the singularities present at the crack front violate the smoothness of the displacements required to yield this rate, and therefore lower the convergence rate [55]. This reduction in the convergence rate is discussed further in chapter 7. If a linear rate of convergence is assumed, the discretization error present in the solution obtained on the finest mesh is 2.3% at $r/t = 0.25$. The discretization error present in the stresses and strains is expected to be higher since these quantities are computed from gradients of the displacements.

A two dimensional mesh was also constructed so that the numerical solution of the generalized plane stress version of the crack problem could be obtained. The mesh consisted of 1,152 four-node, linear quadrilateral plane elements; the two dimensional mesh was exactly the same as the $x_1 - x_2$ plane of the finest three dimensional mesh shown in figure 6.7. This meant that the two and three dimensional numerical solutions could be easily compared. 3×3 Gauss integration was used on all terms when computing the two dimensional element matrices.

6.3 Discussion of the Elastostatic State Near the Crack Front

This section describes in detail the features of the elastostatic state near the crack front obtained from the finite element model discussed in section 6.2.2. The issues that are examined are as follows.

1. Examination of the variation of field quantities through the plate thickness. These variations are an essential part of the three dimensional elastostatic crack problem.

2. Study of the similarities and differences between the two and three dimensional solutions. This involves studying the nature of the stress singularities near the crack front.

3. The residual problem of plane strain (see appendix A), which is solved numerically. The field quantities associated with this problem are discussed in relation to 1 and 2 above.

4. Comparison with the previous analytical and numerical attempts to solve three dimensional crack problems discussed in section 6.1. This requires computing the strengths of the stress singularities near the crack front and the crack-surface intersection.

5. Investigation of regions where the three dimensional solution can be approximated by the plane strain and plane stress solutions discussed in appendix A.

6. Inspection of the nature of the out-of-plane displacement, $u_3(r, \theta, x_3 = t/2)$, of the free surface of the plate. This is of some importance since comparisons with experimental [45] and theoretical [62] results can be made.

A common practice in the numerical study of fracture mechanics is the use of special singular elements to model the singularity present near the crack front [63]. It will be noted that the eight node brick element used in this study has no such special behavior. This can be considered advantageous in this three dimensional situation since, as was mentioned in section 6.1, no complete analytical results are available. There is, therefore, no possibility that the imposition of an incorrect singularity could influence the numerically obtained elastostatic state.

It should be noted that in the displacement based finite element method, the quantities that are directly computed are the nodal displacements. The stresses and strains within each element are then computed using these nodal displacements. For the three dimensional brick element used, the stresses and strains are most accurately computed at each element's center Gauss integration point [6]. By referring to the detail of the mesh shown in figure 6.10, it can be seen that the only radial line (i.e. line of constant θ) along which stresses and strains can be accurately evaluated near, and in front of, the crack front is the line with $\theta = 45°$ (i.e. at points such as A). Therefore the discussions of the variations of the stresses and strains are confined to the plane $\theta = 45°$. Stresses and strains at points on this plane that do not coincide with an element's center Gauss point (i.e. points distant from the crack front) were evaluated by taking the average of the values computed at points such as those denoted by B in figure 6.10. The

same procedure was followed when these quantities were computed from the two dimensional generalized plane stress model discussed in section 6.2.2. Using a similar argument, the only line of constant ϕ and ψ near the crack-surface intersection along which field quantities can be computed in front of the crack is given by $\phi = \psi = 45°$ (see figure 6.2).

The results obtained from the three dimensional finite element model are now discussed in detail.

6.3.1 Field Quantities Through the Plate Thickness

Figures 6.11, 6.12, and 6.13 show the variations of the stresses and strains through the plate thickness along the plane $\theta = 45°$ for different values of r/t. The nine values of r/t are: 0.022, 0.066, 0.110, 0.155, 0.199, 0.243, 0.287, 0.331, and 0.535. Note that the through thickness shear strain distribution is exactly the same as the shear stress distribution shown in figure 6.12. The plots of the stresses show that the in-plane stresses σ_{11} and σ_{22} are almost constant through the plate thickness except for a sharp decrease near the plate surface. These stresses appear to increase without bound as the crack front is approached (see also sections 6.3.2 and 6.3.3). At the free surface the extrapolated values are unclear. The reduced amplitudes of the in-plane tensile stresses near the free surface are consistent with the experimentally observed tendency of cracks in plates to grow initially at the center, and are also in agreement with the decay in the energy release rate as the free surface is approached observed in [14]. The value of σ_{33} at the free surface (excluding the crack-surface intersection point) is seen to be its required value of zero. However, σ_{33} at other points through the plate thickness increases toward

the crack front, apparently without limit. These results seem to agree well with the finite element calculations (using singular elements) reported in [39]. Note that for $r/t \simeq 0.5$, σ_{11} and σ_{22} are constant and σ_{33} is zero through the plate thickness, which indicates that generalized plane stress conditions exist at this distance from the crack front.

The plots of the shear stresses in figure 6.12 show that through the plate thickness, σ_{12} is almost constant apart from a variation near the free surface. It, too, appears to increase without bound towards the crack front. The out-of-plane shear stresses, σ_{23} and σ_{31}, have a value of zero at the center plane of the plate. This is due to symmetry of the problem. These stresses also have a value of zero at the plate's free surface ($x_3/t = 0.5$), since this surface is traction free. Between these two planes the behavior is somewhat complicated. The σ_{23} stress component may have an infinite spike at the crack-surface intersection. This is discussed further in section 6.3.3. The behavior of σ_{31} is even more complicated. A mesh having a finer discretization near the crack-surface intersection would provide a more detailed description of the nature of the σ_{23} and σ_{31} distribution in that region. It should be noted that both of these out-of-plane shear stresses are small compared to the normal stresses, and so are sometimes considered negligible through the plate's thickness, as is stated in [39].

The plots of the normal strains shown in figure 6.13 show that ϵ_{11} and ϵ_{22} behave similarly to σ_{11} and σ_{22} except for the curious increase in ϵ_{11} before it decreases near the free surface. Both of these quantities are constant through the thickness at $r/t \simeq 0.5$. The magnitude of ϵ_{33} reaches a maximum value and then decreases slightly as the crack front is approached in the interior of the

plate. These values have not yet converged to a constant value. Closer to the free surface, ϵ_{33} increases as the crack front is approached, apparently without limit.

The observation that, at the crack front, ϵ_{33} may have finite, but nonzero, values through the plate thickness means that a degree of plane strain (see appendix A) of one at the crack tip in the interior of the plate does not imply that plane strain conditions exist there. This will be important when the behavior of the degree of plane strain is examined near the crack front in section 6.3.4.

Figure 6.14 shows the through thickness variations of the normal stresses present in the solution of the residual problem of plane strain (see appendix A) for the same values of r/t used in figures 6.11, 6.12, and 6.13. The residual problem can be thought of as containing all the three dimensional effects present in the original cracked plate problem. The in-plane normal stresses, σ''_{11} and σ''_{22}, appear to increase without limit toward the crack front (see also section 6.3.2), and, interestingly, the through thickness distribution changes sign as the free surface is approached. In the interior of the plate, the normal in-plane stresses of the residual problem are tensile, whereas near the surface, they become compressive. This is again consistent with the experimentally observed tendency of cracks to grow in the center (the tunneling effect). As is evident from figure 6.14, the changes in sign of σ''_{11} and σ''_{22} occur at different values of x_3/t that depend on r/t. Possibly, the crossover point approaches the free surface as the crack front is neared, producing a complicated singularity.

The other normal stress, σ''_{33}, should become unbounded in tension in the plate's interior at the crack front since σ''_{11} and σ''_{22} appear to be unbounded in tension in this region and ϵ''_{33} (which is the same as ϵ_{33}) appears to be finite,

although this behavior of σ''_{33} is not yet obvious in figure 6.14. At the free surface on the crack front, σ''_{33} is unbounded in compression, as is required by the traction boundary condition of the residual problem.

An important point that should be stressed when examining the solution to the residual problem of plane strain is that the results shown in figure 6.14 indicate that three dimensional effects are present at the crack front in the original problem, even near the center plane of the plate.

6.3.2 Comparison Between the Two and Three Dimensional Solutions

Figures 6.15, 6.16, and 6.17 show the variation in σ_{22} along the plane $\theta = 45°$ obtained from the three dimensional numerical solution. The values plotted are computed at the center Gauss points of the elements nearest the center plane of the plate ($x_3/t = 0.016$), nearest the plane $x_3/t = \frac{1}{4}$ ($x_3/t = 0.234$), and nearest the free surface of the plate ($x_3/t = 0.484$). Also shown are the generalized plane stress values, $\bar{\sigma}_{22}$, computed using the finite element method and using the asymptotic result given by equation (6.1), together with the σ''_{22} values for the residual problem of plane strain (which are simply the difference between the three and two dimensional numerical solutions, see appendix A). It should again be noted that the in-plane stresses in the corresponding plane strain problem are exactly the same as in the generalized plane stress problem. The stresses are plotted in both normal and log-log form.

The three dimensional values of σ_{22} closely follow the computed two dimensional $\bar{\sigma}_{22}$ values at distances from the crack front greater than $\frac{1}{2}t$. Closer to the crack front, σ_{22} slightly exceeds $\bar{\sigma}_{22}$ in the plate's interior, but is smaller

nearer the free surface. The reduced amplitude of σ_{22} near the free surface was also observed in figure 6.11. It can also be seen that the two dimensional finite element values of $\bar{\sigma}_{22}$ agree with the leading term of the two dimensional asymptotic result at distances less than $\frac{1}{8}t$ from the crack front; the small difference is due to discretization error. The better agreement of σ_{22} with the leading term of the two dimensional asymptotic result is just by chance and would not exist in a finer mesh.

The log-log plots shown in figures 6.15, 6.16, and 6.17 indicate that, like the two dimensional stress $\bar{\sigma}_{22}$, the three dimensional stresses σ_{22} and σ''_{22} behave as $r^{-\frac{1}{2}}$ near the crack front in the interior of the plate. The slight deviation of the value of σ_{22} from this $r^{-\frac{1}{2}}$ variation at the point closest to the crack front is probably due to discretization error. Since σ''_{22} behaves as $r^{-\frac{1}{2}}$, the three dimensional solution, σ_{22}, cannot converge to the two dimensional solution, $\bar{\sigma}_{22}$, as the crack front is approached. This shows that even near the mid-plane the residual problem contributes to the three dimensional elastostatic state as the crack front is approached, a clear indication that plane strain conditions are not achieved there. A more detailed discussion of the singular nature of the stresses in the plate's interior can be found in section 6.3.3.

The fineness of the mesh does not permit determination of how close to the free surface the $r^{-\frac{1}{2}}$ behavior of σ''_{22} holds. If the σ''_{22} crossover point shown in figure 6.14 (see section 6.3.1) moves close to the free surface as the crack front is approached, then σ''_{22} at $x_3/t = 0.484$ may approach infinity in tension rather than in compression as it appears to be doing in figure 6.17. Since the mesh fineness is not such to determine the behavior of σ''_{22} at $x_3/t = 0.484$ near the

crack front, this quantity has been omitted from the log-log plot in figure 6.17.

6.3.3 Comparison With Previous Three Dimensional Solutions

In this section, the nature of the stress singularities near the crack front in the interior of the plate and near the crack-surface intersection will be compared with the three dimensional solutions discussed in section 6.1.

6.3.3.1 The Stresses in the Plate's Interior

Figures 6.18 and 6.19 show the behavior of all six stress components along the plane $\theta = 45°$ from the three dimensional numerical solution. The stresses are computed at the center Gauss points nearest the center plane of the plate ($x_3/t = 0.016$) and near the plane $x_3/t = \frac{1}{4}$ ($x_3/t = 0.234$), and are plotted in normal and log-log form. All of the stresses appear to be singular near the crack front along the plane $\theta = 45°$, with the exception of the σ_{23} component.

Using the data shown in figures 6.18 and 6.19, the strength of the singularity in the variation of all six stress components with the radial coordinate r was estimated. This was done using the values of the stresses computed at the center Gauss points of the second and third elements from the crack front. The stresses calculated at the element closest to the crack front were considered less accurate since the discretization error in the numerical solution is higher there. This error was observed in figures 6.15, 6.16, and 6.17, and noted in section 6.3.2. This method was validated when the singularity exponent was computed using the two dimensional (plane strain, generalized plane stress) numerical solution.

In this case, the second and third elements gave values of the singularity exponent closer to the theoretical value of $-\frac{1}{2}$ than the first and second elements did along the plane $\theta = 45°$.

Table 6.2 shows the values of the singularity exponent computed as described above using the two and three dimensional numerical solutions. The exponents were calculated along the plane $\theta = 45°$ at $r/t = 0.09$, and, for the three dimensional solution, at two different locations in the interior of the plate ($x_3/t = 0.016$ and 0.234). There is good agreement between the values of the exponents for some of the individual three dimensional stress components at the two different values of x_3/t. Also, the exponents calculated for the three dimensional σ_{11}, σ_{22} and σ_{12} stresses agree well with the corresponding values computed using the two dimensional results. The difference between the computed singularity exponents for the two dimensional stresses shown in table 6.2 and the theoretical value of $-\frac{1}{2}$ indicates the error present in the three dimensional values. However, the results suggest that the singularity exponents may depend on the individual stress components, and therefore indicate that the three dimensional stresses near the crack front cannot be separated into the form

$$\sigma_{ij} = r^{\lambda} f_{ij}(\theta, x_3). \tag{6.8}$$

However, it should be emphasized that the resolution of the finite element mesh may not be fine enough to capture the behavior given in equation (6.8). The apparent lack of separability of the stresses near the crack front in the plate's interior makes the idea of a three dimensional stress intensity factor redundant. However, figures 6.11, 6.15, 6.16, and 6.17 show that the amplitude of σ_{22} (the dominant tensile stress near the crack front in the plate's interior) does decrease

as the free surface is approached.

The nature of the stress singularities near the crack front in the interior of the plate discussed above can be compared with the solution given in [21]. Folias claims that, in the plate's interior, the $\sigma_{11}, \sigma_{22}, \sigma_{33}$, and σ_{12} stress components are singular of order $-\frac{1}{2}$, and that σ_{23} and σ_{31} are bounded. Figures 6.18 and 6.19 show that the numerical results presented in this paper agree with this claim qualitatively since the $\sigma_{11}, \sigma_{22}, \sigma_{33}$, and σ_{12} stresses are singular and σ_{23} appears to be bounded in the plate's interior. However, σ_{31} is observed to be singular along the plane $\theta = 45°$, and table 6.2 shows that the values of the singularity exponent computed using the numerical results disagree with those given in [21].

6.3.3.2 The Stresses Near the Crack-Surface Intersection

In this section, the nature of the stresses are examined in the vincinity of the crack-surface intersection. It was stated above that the only line of constant ϕ and ψ (see figure 6.2) in front of the crack along which all field quantities can be examined is given by $\phi = \psi = 45°$. Figure 6.20 shows the variation in all the stress components along this line. The values of the stresses are shown in normal and log-log form. Table 6.3 shows the estimates of the strength of the singularity in the stress components with the spherical radius ρ, i.e. λ in equation (6.4). These values were calculated using the stresses computed at the center Gauss points of the second and third elements from the crack-surface intersection (i.e. at a distance $\rho/t = 0.11$ from the crack-surface intersection).

There are differences in the nature of the stress components along the line of constant ϕ and ψ. Although all the stresses appear to be singular at the crack-

surface intersection, there are substantial differences in the values of λ shown in table 6.3. This suggests that there is no justification in assuming a separable solution of the form given in equation (6.4). There is considerable disagreement between the values of λ shown in table 6.3 and those presented in [7], [9], and [14] (which were -0.452, -0.452, and -0.445, respectively). It is interesting to note that the value of λ for the σ_{33} stress component given in table 6.3 is close to -1.1, which was predicted in [21] for $\nu = 0.3$, see section 6.1. However, this is the only stress component for which there is such agreement with this analytical solution. It should again be stated that the results presented in table 6.3 may be inaccurate due to the resolution of the finite element mesh.

6.3.4 Regions of Plane Stress and Plane Strain

Figure 6.21 shows the degree of plane strain, $\frac{\sigma_{33}}{\nu(\sigma_{11}+\sigma_{22})}$ (see appendix A), computed along the plane $\theta = 45°$ at the elements nearest the plate's center plane ($x_3/t = 0.016$), the plane $x_3/t = \frac{1}{4}$ ($x_3/t = 0.234$), and the free surface ($x_3/t = 0.484$). This shows again that conditions of generalized plane stress are approximately reached at distances greater than $\frac{1}{2}t$ away from the crack front.

The curves plotted in figure 6.21 indicate that there is no significant region near the crack front over which the degree of plane strain is approximately one, but appear to give it an extrapolated value of one at the crack front. It would be wrong to conclude that this shows plane strain conditions exist at the crack front in the interior of the plate. As stated in appendix A, the ratio $\frac{\sigma_{33}}{\nu(\sigma_{11}+\sigma_{22})}$ may have a value of one if ϵ_{33} is bounded and if σ_{11} and σ_{22} are singular. The results shown in figures 6.15, 6.16, 6.17, 6.18, and 6.19 show that σ_{11} and σ_{22} are

indeed singular at the crack front in the plate's interior, while the distribution of ϵ_{33} shown in figure 6.13 indicates that ϵ_{33} is bounded and nonzero there. Thus, the extrapolated degree of plane strain of one at the crack front does not imply the existence of plane strain conditions there.

Figure 6.22 shows the variation of the degree of plane strain through the thickness of the plate along the plane $\theta = 45°$ for several values of r/t. The nine values of r/t are: 0.022, 0.066, 0.110, 0.155, 0.199, 0.243, 0.287, 0.331, and 0.535. This plot shows that, as the crack tip is approached, the ratio $\frac{\sigma_{33}}{\nu(\sigma_{11}+\sigma_{22})}$ approaches a constant value through the plate thickness in the plate's interior away from the free surface. A finer mesh might indicate that the degree of plane strain converges to a value of one throughout the plate thickness as the crack front is approached, except possibly at the free surface. Again, this would not imply the existence of plane strain conditions. Figure 6.22 also illustrates that generalized plane stress conditions are achieved throughout the thickness at a distance $r/t \simeq 0.5$ from the crack front.

6.3.5 The Out-of-Plane Displacements of the Free Surface

Figure 6.23 shows the behavior of $u_3(r, \theta = 45°, x_3 = t/2)$ for the three dimensional mesh evaluated at the nodal points on the free surface along the radial line $\theta = 45°$. Also shown in figure 6.23 is the variation of u_3 computed using the full field numerical plane stress solution as well as the leading term of the asymptotic plane stress solution. The plane stress displacements were computed using the expression for $\hat{u}_3$ given in appendix A, equation (A.15). Figure 6.24 shows a surface plot of the computed three dimensional u_3 distribution over the

entire free surface.

For $\theta = 45°$, it can be seen that the computed three dimensional u_3 values are the same as the numerically computed full field plane stress results at distances from the crack front greater than $\frac{1}{2}t$. For $r \leq \frac{1}{2}t$, the deviation of the three dimensional result from its full field plane stress counterpart indicates the region of dominance of three dimensional effects in the vicinity of the crack front. The plane stress finite element solution follows the plane stress asymptotic solution out to a distance from the crack front of $\frac{1}{8}t$. The disagreement between the asymptotic and full field numerical plane stress solutions clearly demonstrates the effect of lower order nonsingular terms at distances of $r \geq \frac{1}{8}t$. It can be seen that figures 6.23 and 6.24 show that the maximum magnitude of the out-of-plane displacement on the free surface occurs at the crack front. The results shown in figure 6.9 indicate that the three dimensional u_3 value at the crack front on the free surface is bounded.

Finally, some of the basic features of the results presented above are found to be in good agreement with recent preliminary experimental and theoretical work on the subject [45,62]. In particular, the observation that plane stress conditions prevail at distances from the crack front greater than half the specimen thickness is common to all three investigations. On the other hand, there are differences between the u_3 displacements of the free surface presented in [45,62], and the results shown here in figure 6.23. The differences become noticeable very close to the crack front and may be due to the variety of simplifying assumptions that characterize each of these works.

6.3.6 Conclusions

1. In the interior of the plate, the stresses σ_{11}, σ_{22}, σ_{33}, σ_{12}, and σ_{31} are singular, whereas σ_{23} is bounded along the plane $\theta = 45°$. The strengths of the singularities in the plate's interior appear to depend on the stress components.

2. All of the stress components near the crack-surface intersection are singular. The nature of the singularities appear to depend on the stress components.

3. The agreement between the numerically measured strengths of the stress singularities and previous analytical and numerical work in the plate's interior and near the crack-surface intersection was poor. It should be noted that this could possibly be due to the resolution of the finite element mesh.

4. The out-of-plane displacements at the plate's free surface have their maximum magnitude at the crack front. At distances greater than $\frac{1}{2}t$ from the crack front (along the plane $\theta = 45°$), these displacements are the same as the corresponding plane stress values that were calculated numerically.

5. Conditions of generalized plane stress are reached at a distance of $\frac{1}{2}t$ from the crack front in the interior of the plate.

6. No region in which pure plane strain conditions exist was found. This is evident from the study of the residual problem of plane strain.

7. The solution to the residual problem of plane strain has a noticeable effect near the crack front in the interior of the plate. This demonstrates that the three dimensional elastostatic state in this region is not well approximated by the two dimensional plane strain solution. The solution to the residual

problem produces σ''_{11} and σ''_{22} stresses that are tensile in the plate's interior and compressive near the free surface.

8. The range of dominance of the leading term of the two dimensional asymptotic solution is found to be $r/t \leq \frac{1}{8}$. There are significant differences between the two dimensional full field numerical and asymptotic values of stresses and displacements further from the crack front. This demonstrates the importance of the nonsingular terms in the asymptotic two dimensional solution.

γ	ν_1, ν_2	Number of Multigrid Cycles	Solution Time (CPU secs)
1	5	61	4.28×10^4
2	5	25	1.85×10^4
3	5	18	1.45×10^4
4	5	17	1.46×10^4
5	5	17	1.58×10^4
3	1	83	2.17×10^4
3	5	18	1.45×10^4
3	10	15	2.19×10^4

Table 6.1: Number of Multigrid Cycles and Solution Times Required in the Preliminary Investigation of the Cracked Plate Problem (25,515 degrees-of-freedom).

σ_{ij}	$x_3/t = 0.016$	$x_3/t = 0.234$	Two Dimensional
σ_{11}	-0.591	-0.602	-0.604
σ_{22}	-0.518	-0.525	-0.513
σ_{33}	-1.072	-1.267	
σ_{12}	-0.593	-0.596	-0.615
σ_{23}	0.932	0.870	
σ_{31}	-0.923	-1.453	

Table 6.2: Measured Strengths of the Singularities in the Stress Components in the Plate's Interior ($\theta = 45°, r/t = 0.09$).

σ_{ij}	$\phi = \psi = 45°$
σ_{11}	-0.491
σ_{22}	-0.450
σ_{33}	-1.088
σ_{12}	-0.594
σ_{23}	-0.439
σ_{31}	-0.364

Table 6.3: Measured Strengths of the Singularities in the Stress Components Near the Crack-Surface Intersection ($\phi = \psi = 45°, \rho/t = 0.11$).

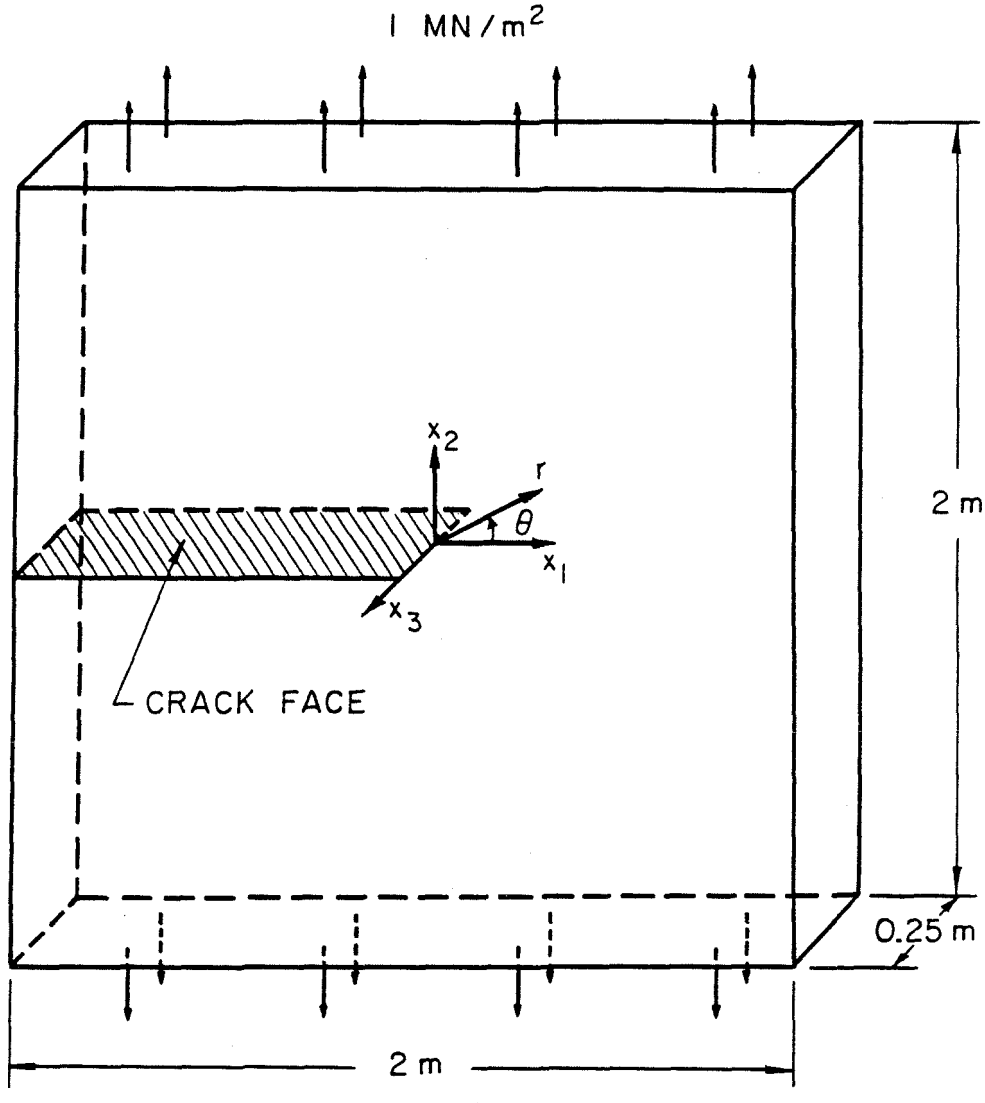


Figure 6.1: The Cracked Plate.

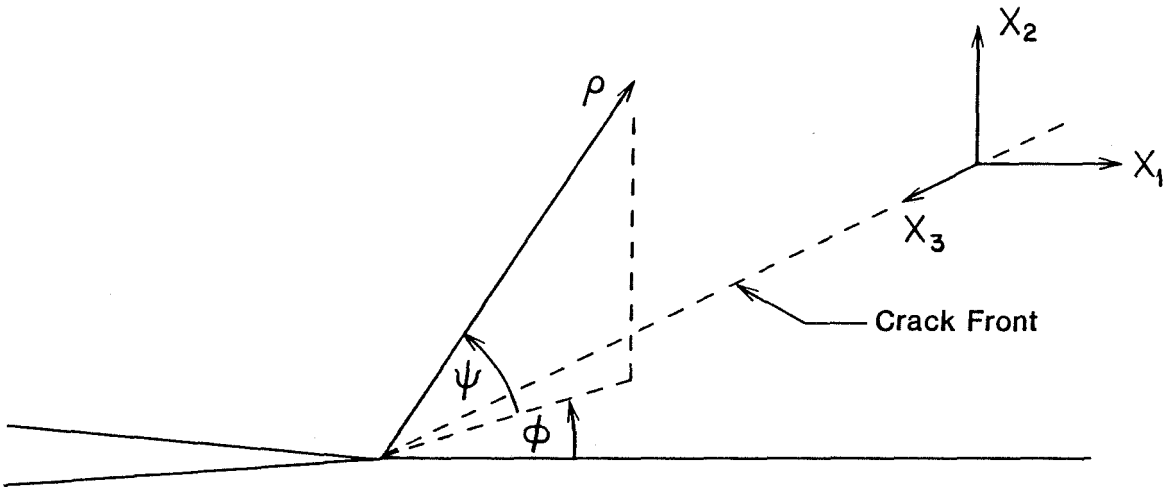


Figure 6.2: The Spherical Coordinate System (ρ, ϕ, ψ) Centered at the Crack-Surface Intersection.

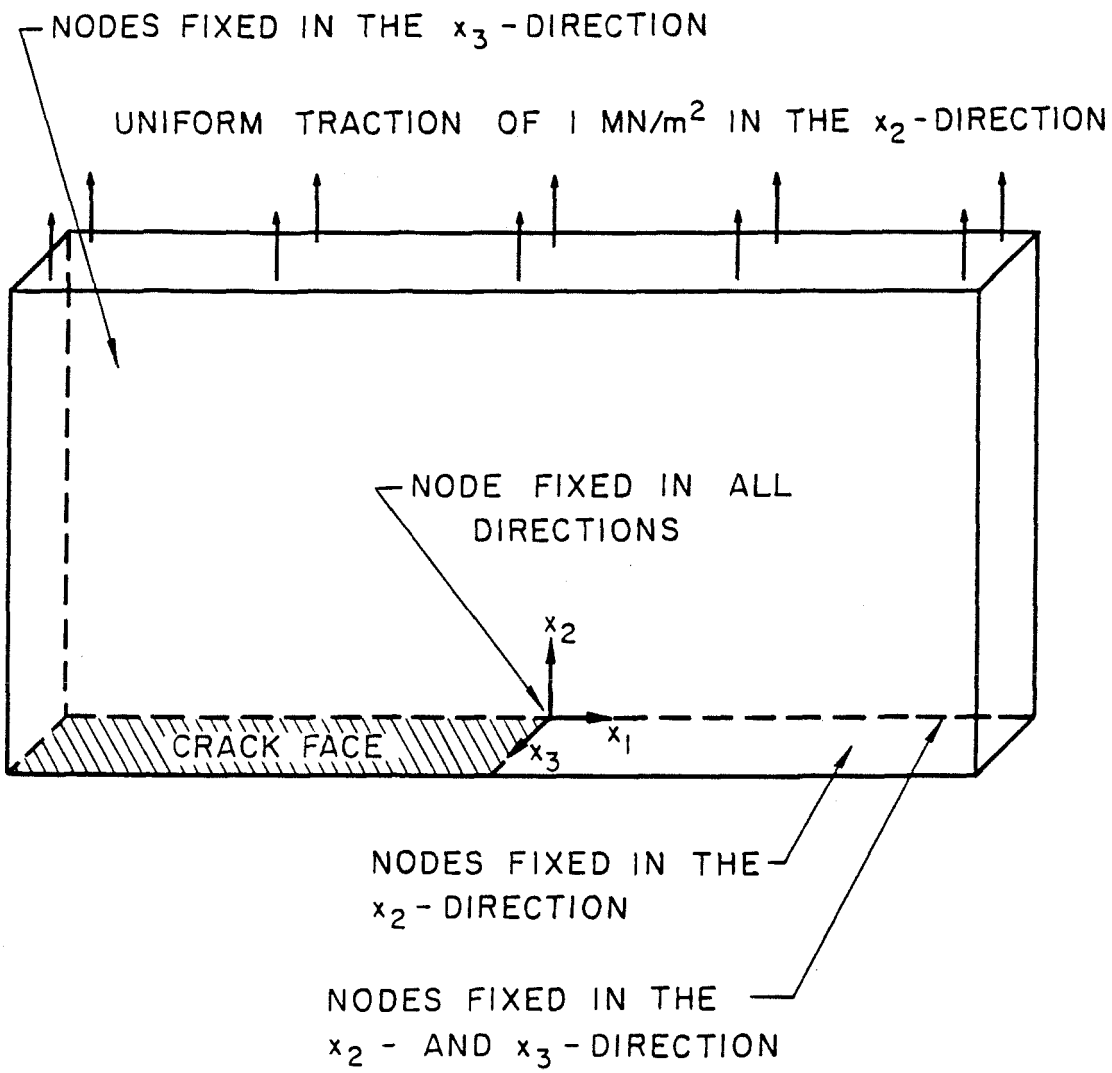


Figure 6.3: The Boundary Conditions That Were Applied to the Finite Element Models of the Cracked Plate.

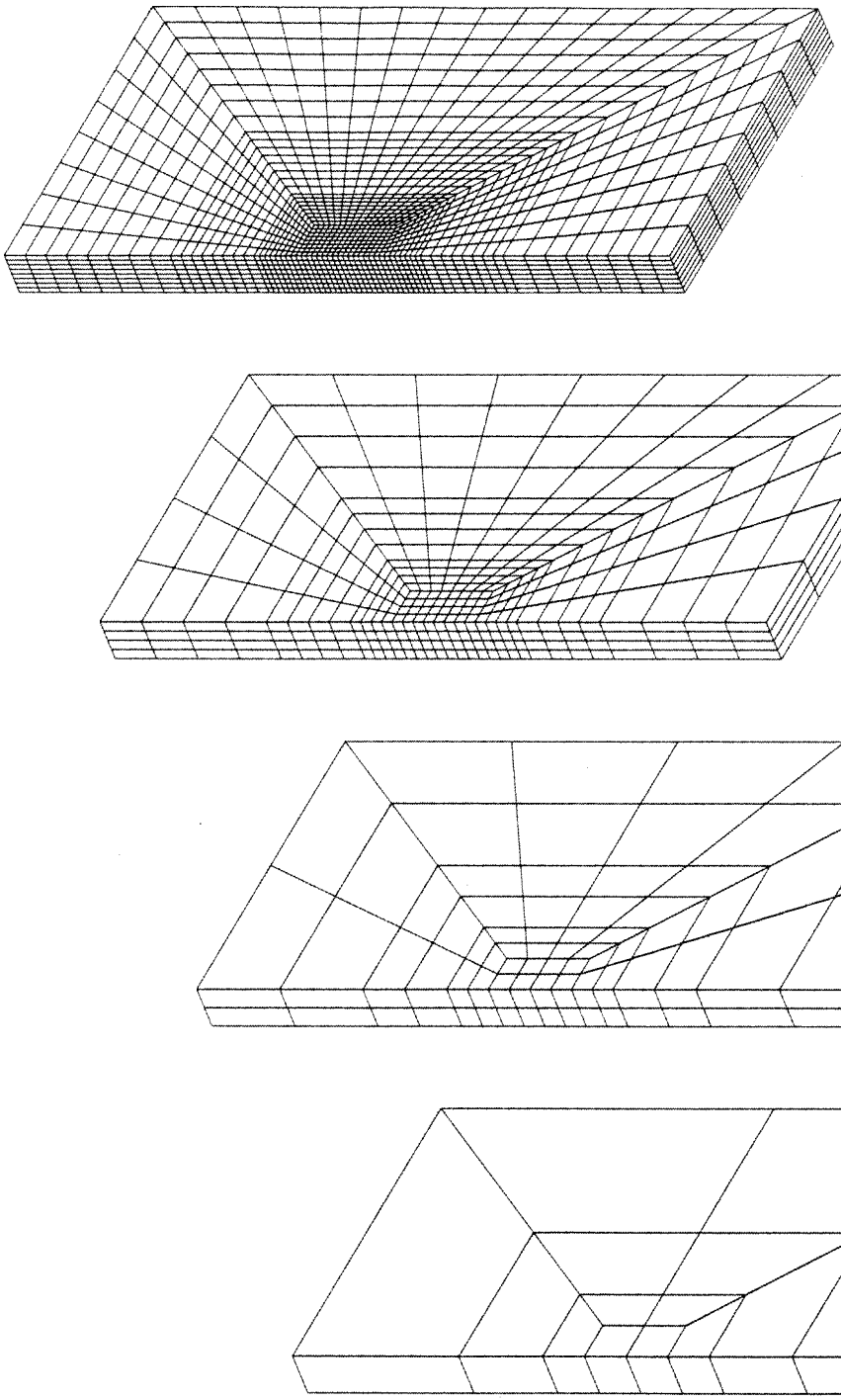


Figure 6.4: The Finite Element Meshes Used in the Preliminary
Problem.

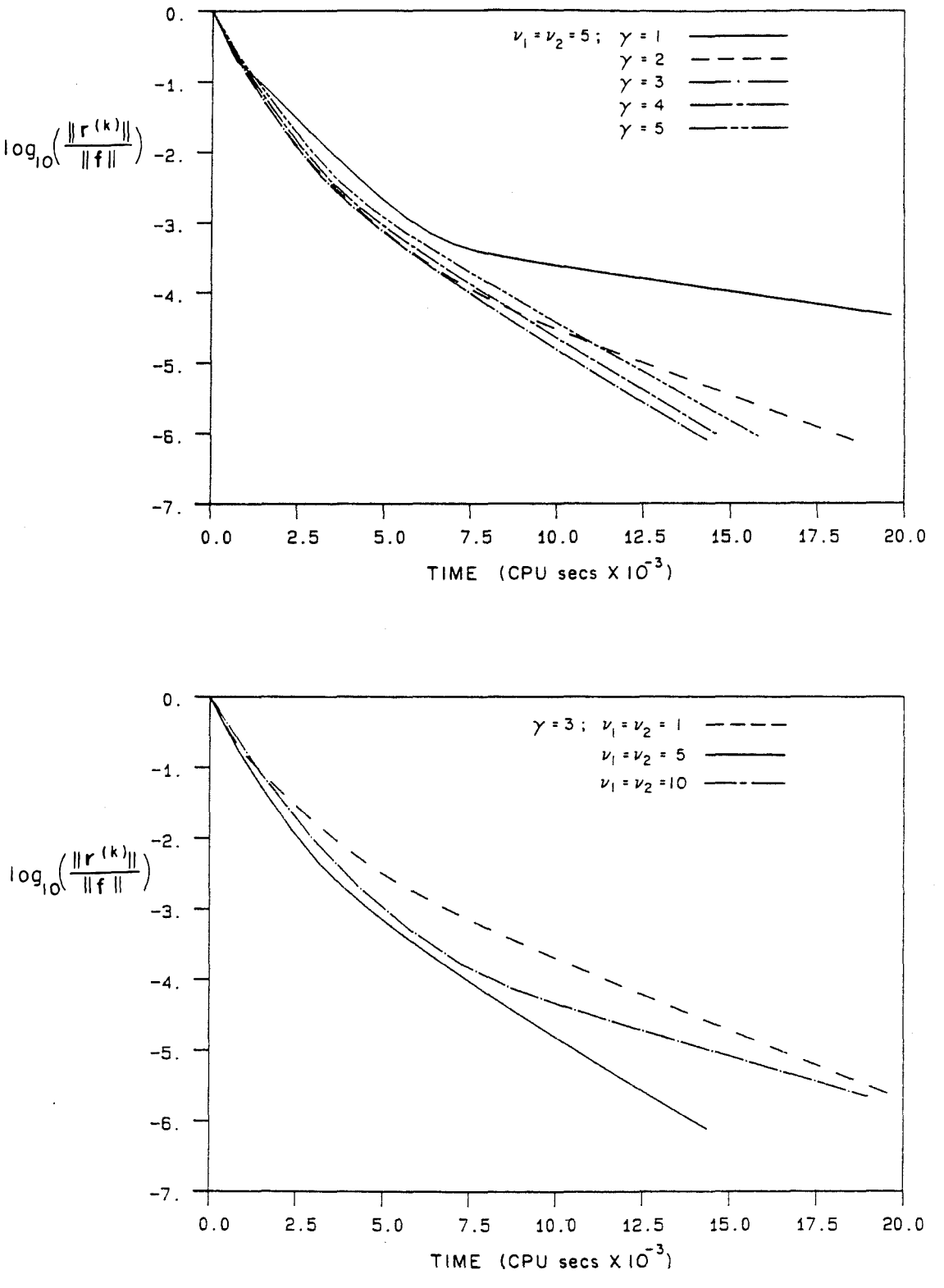


Figure 6.5: The Convergence of the MG_1-GS Method in the Preliminary Investigation of the Cracked Plate for Different Values of γ, ν_1, and ν_2 (25,515 degrees-of-freedom).

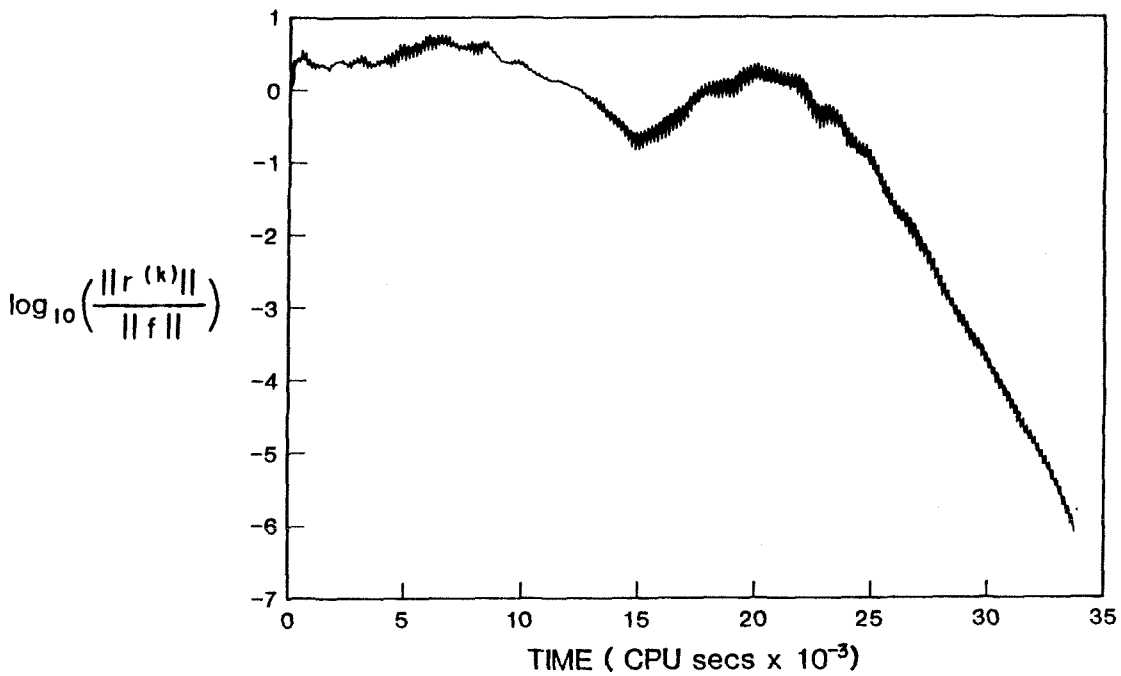
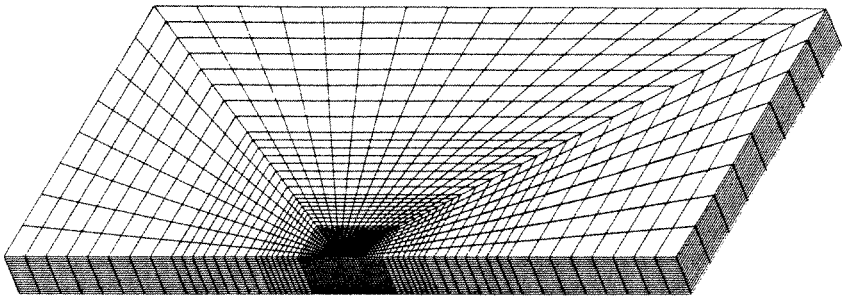
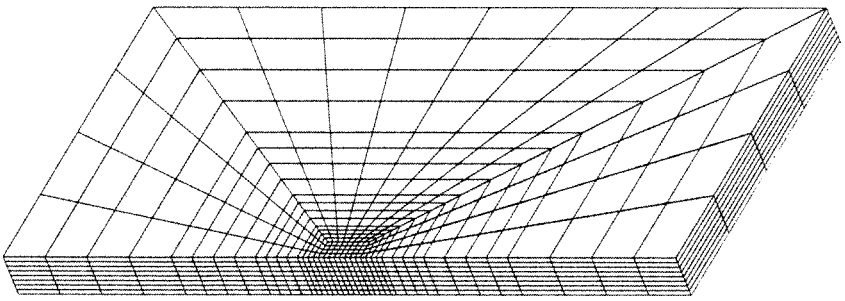
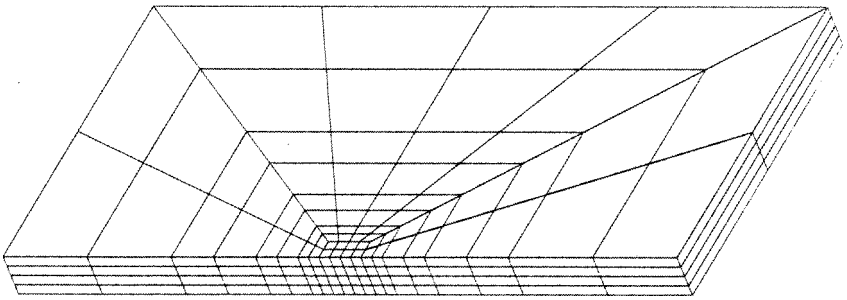
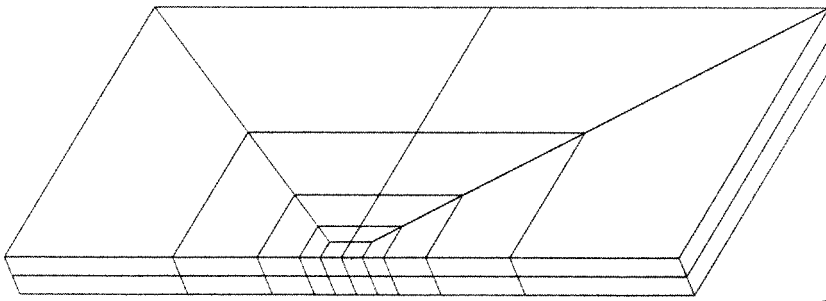


Figure 6.6: The Convergence of the JCG Method Used in the Preliminary Investigation of the Cracked Plate Problem (25,515 degrees-of-freedom).









ure 6.7: The Finite Element Meshes Used in the Final Inves

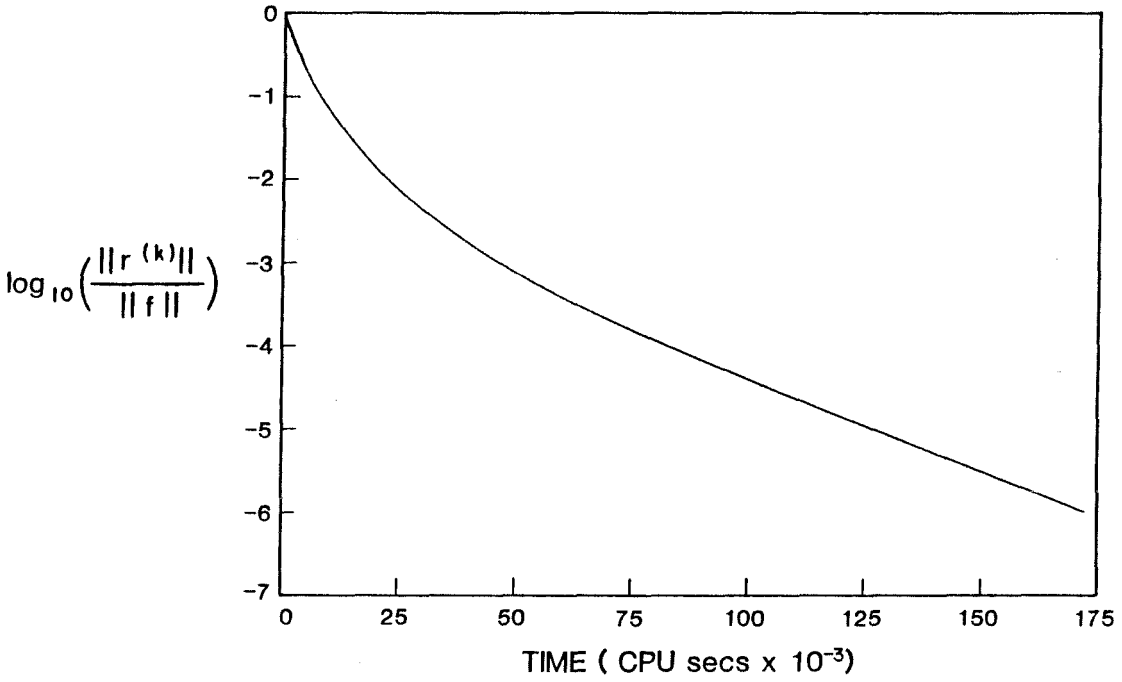


Figure 6.8: Convergence of the MG_1-GS ($\gamma = 3$, $\nu_1 = \nu_2 = 5$) Method Used in the Final Investigation of the Cracked Plate Problem (61,659 degrees-of-freedom).

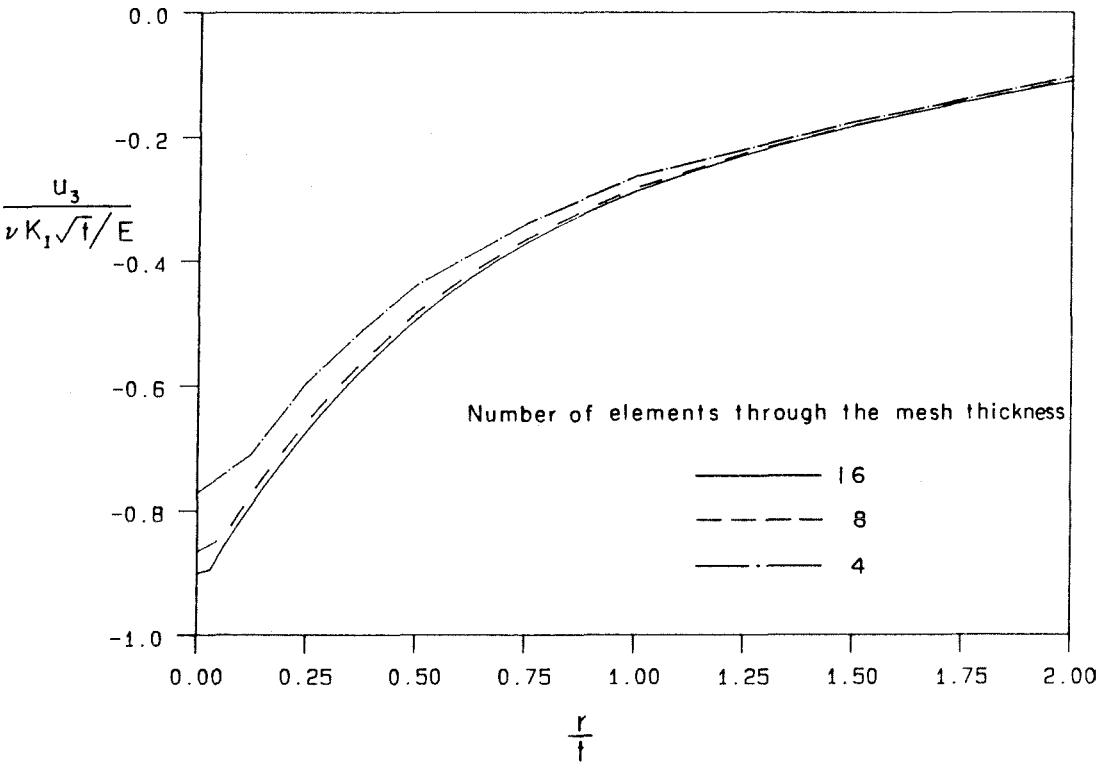


Figure 6.9: Convergence of the Finite Element Results for the Cracked Plate Problem for $u_3(r, \theta = 0°, x_3 = t/2)$ With Respect to the Element Size.

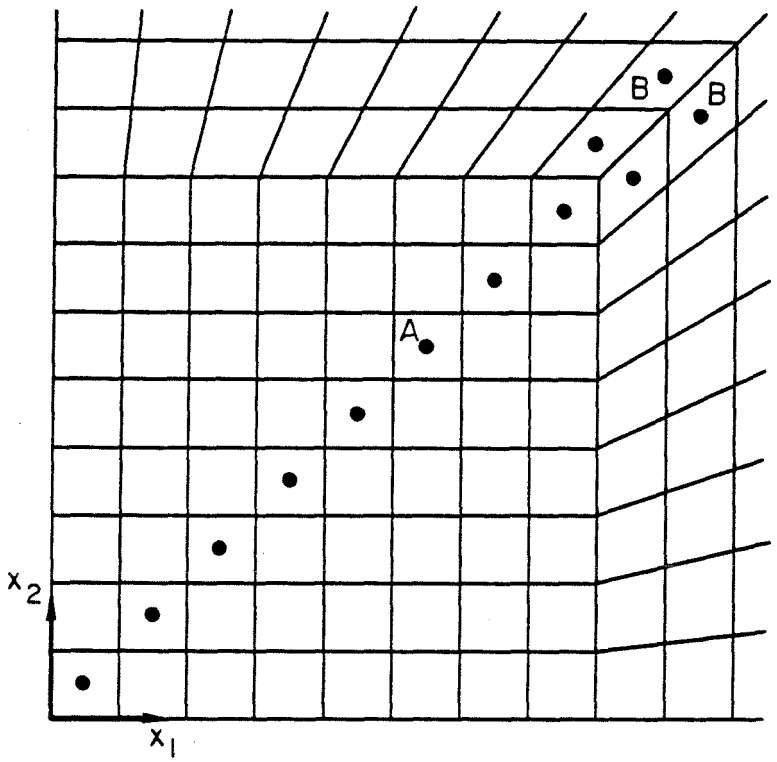


Figure 6.10: Gauss Point Locations Used to Compute the Stresses and Strains on the $\theta = 45°$ Plane.

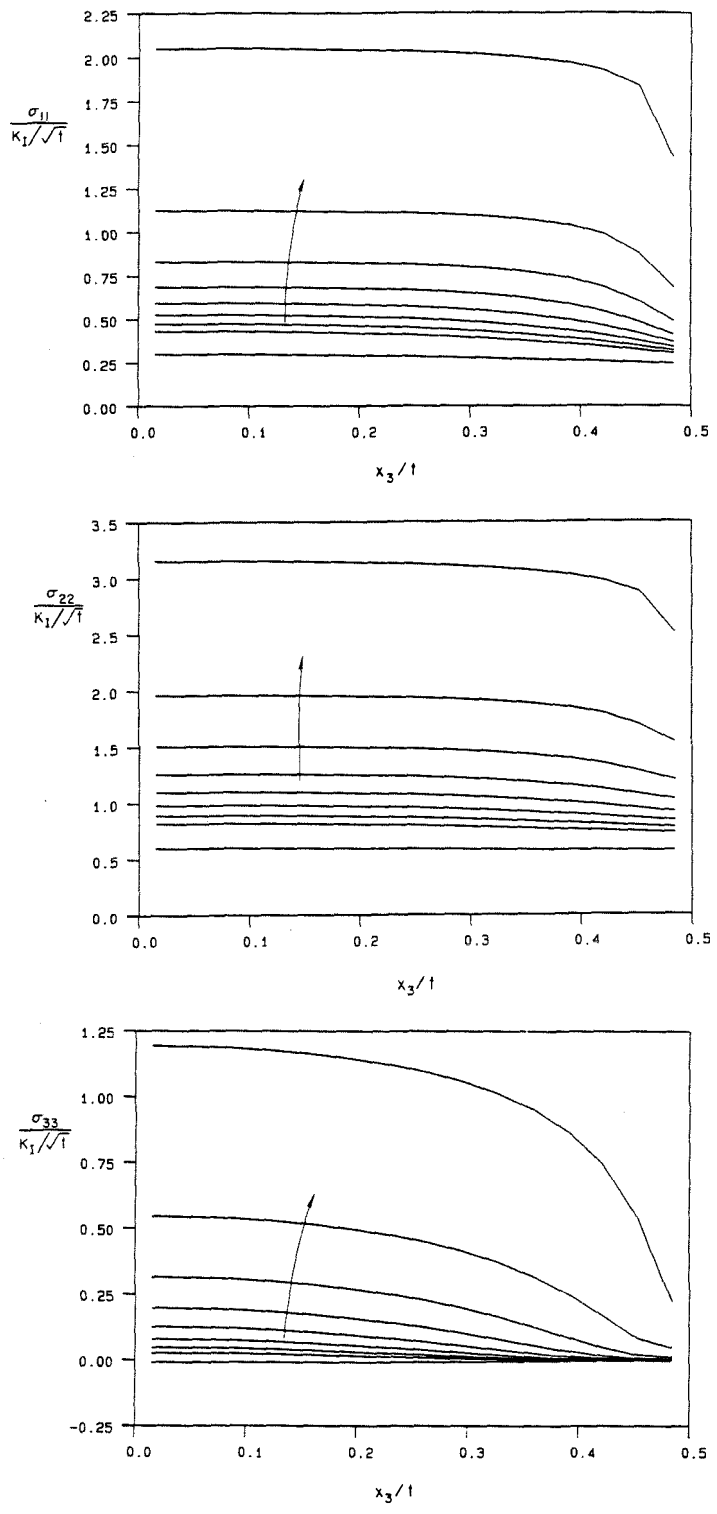


Figure 6.11: Normal Stresses Through the Plate Thickness: $\theta = 45°$, $r/t = 0.022$, 0.066, 0.110, 0.155, 0.199, 0.243, 0.287, 0.331, 0.535 (the arrows indicate the direction of decreasing r/t values).

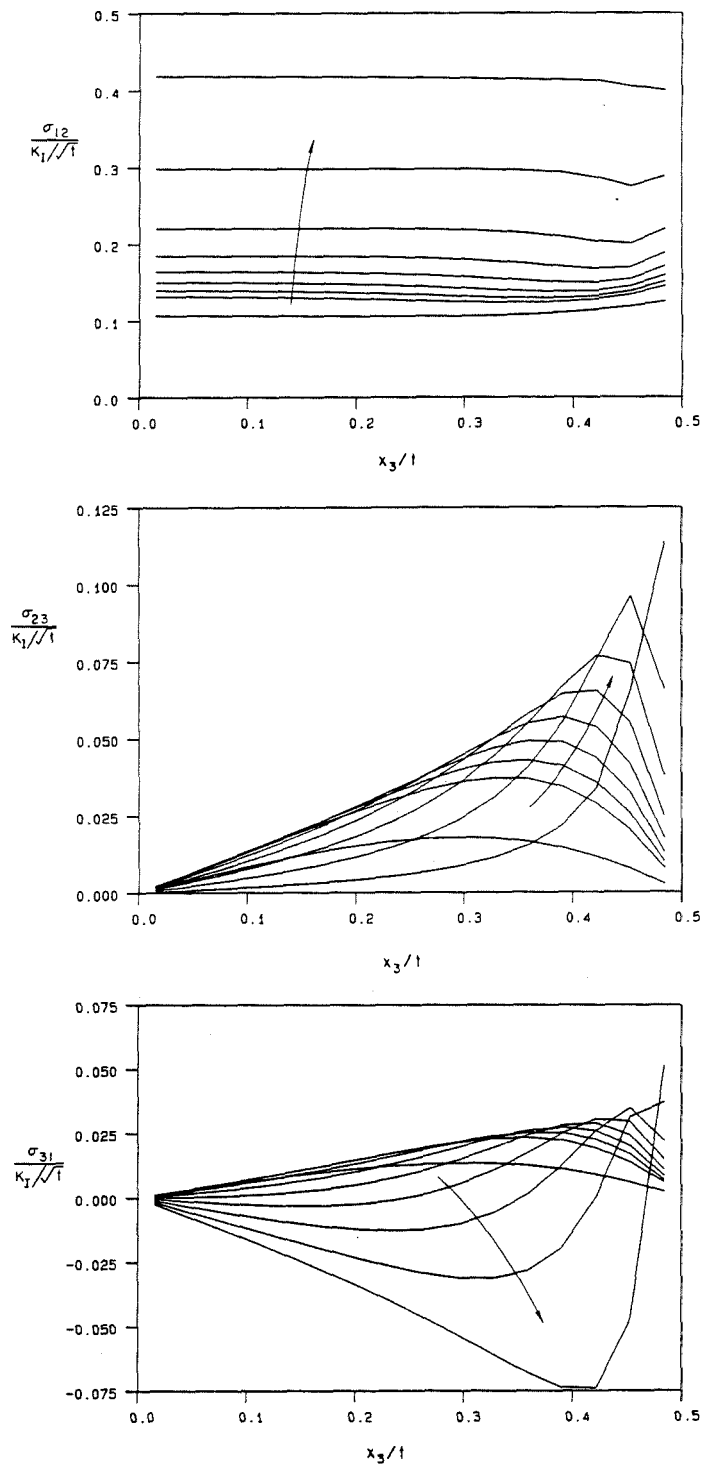


Figure 6.12: Shear Stresses Through the Plate Thickness: $\theta = 45°$, $r/t = 0.022$, 0.066, 0.110, 0.155, 0.199, 0.243, 0.287, 0.331, 0.535 (the arrows indicate the direction of decreasing r/t values).

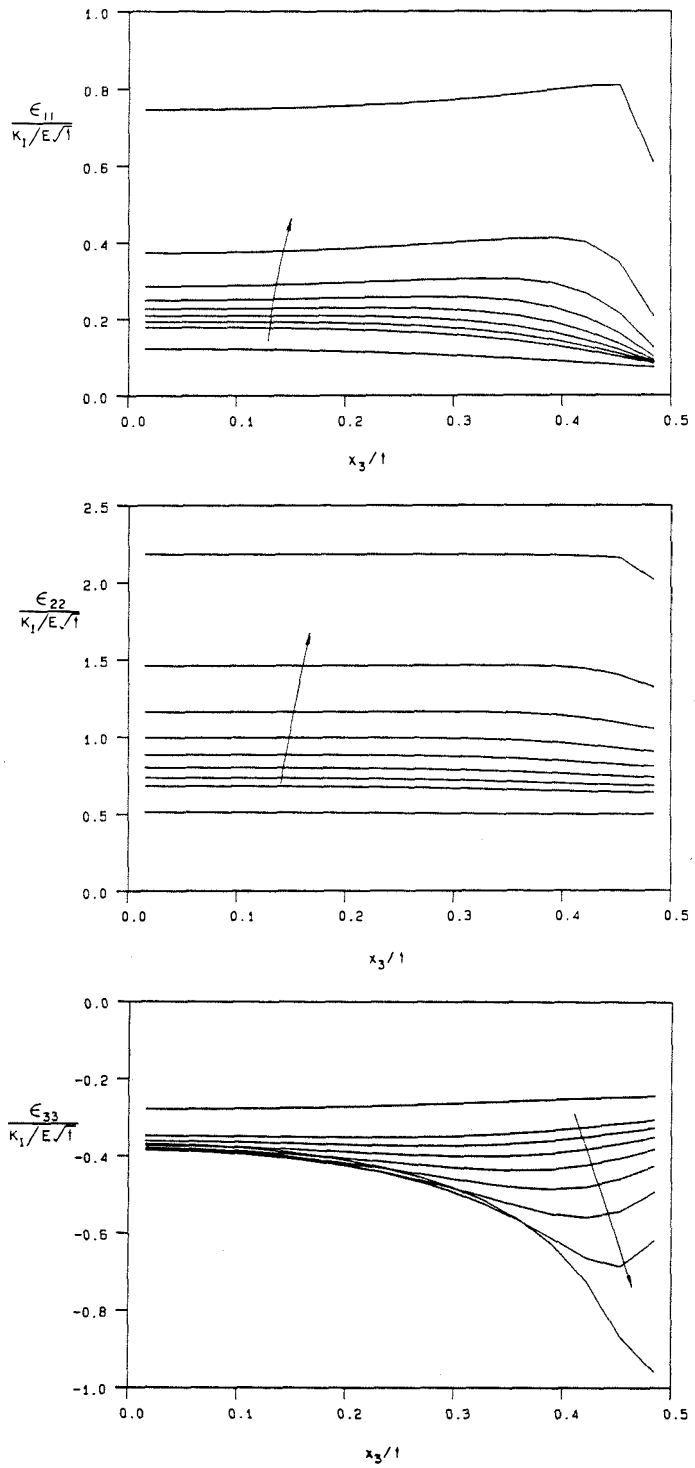


Figure 6.13: Normal Strains Through the Plate Thickness: $\theta = 45°$, $r/t = 0.022$, 0.066, 0.110, 0.155, 0.199, 0.243, 0.287, 0.331, 0.535 (the arrows indicate the direction of decreasing r/t values).

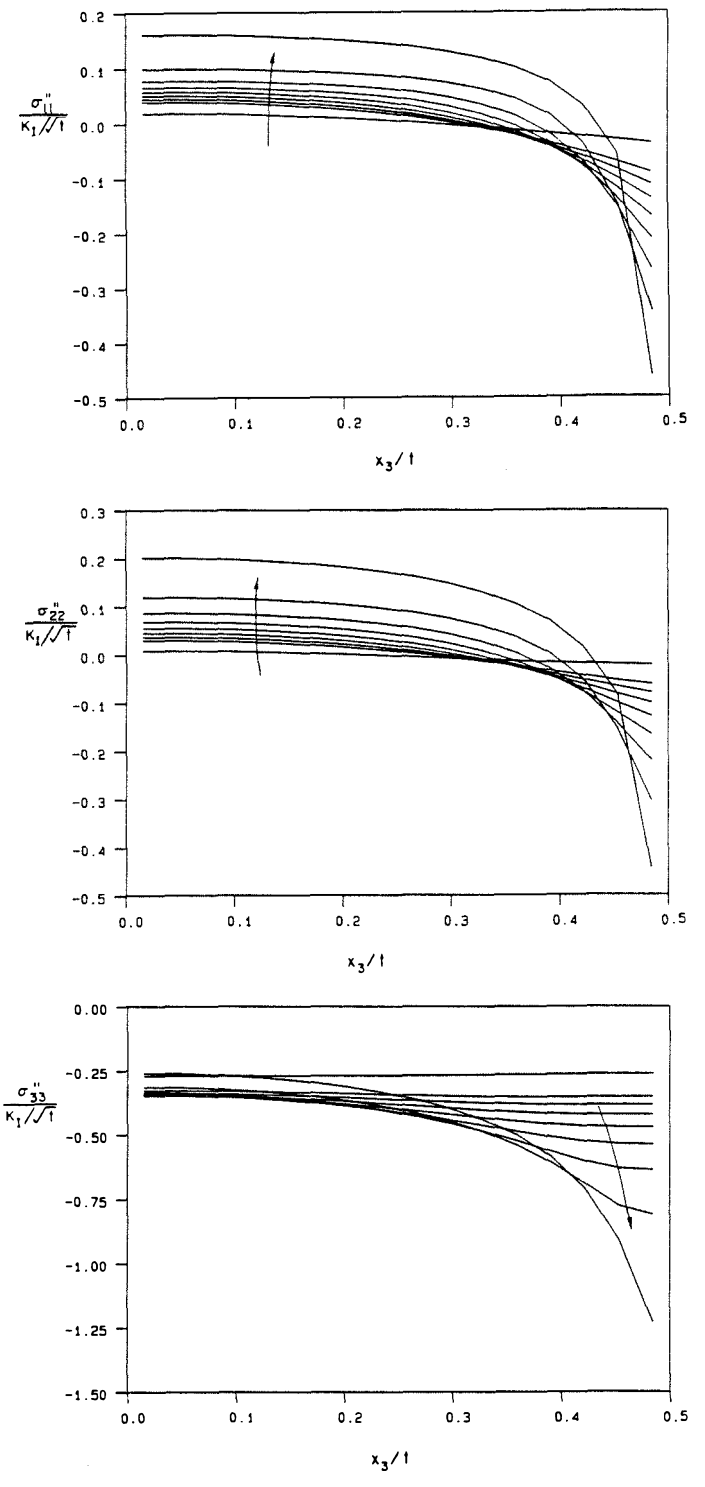


Figure 6.14: Normal Stresses for the Residual Problem Through the Plate Thickness: $\theta = 45°$, $r/t = 0.022$, 0.066, 0.110, 0.155, 0.199, 0.243, 0.287, 0.331, 0.535 (the arrows indicate the direction of decreasing r/t values).

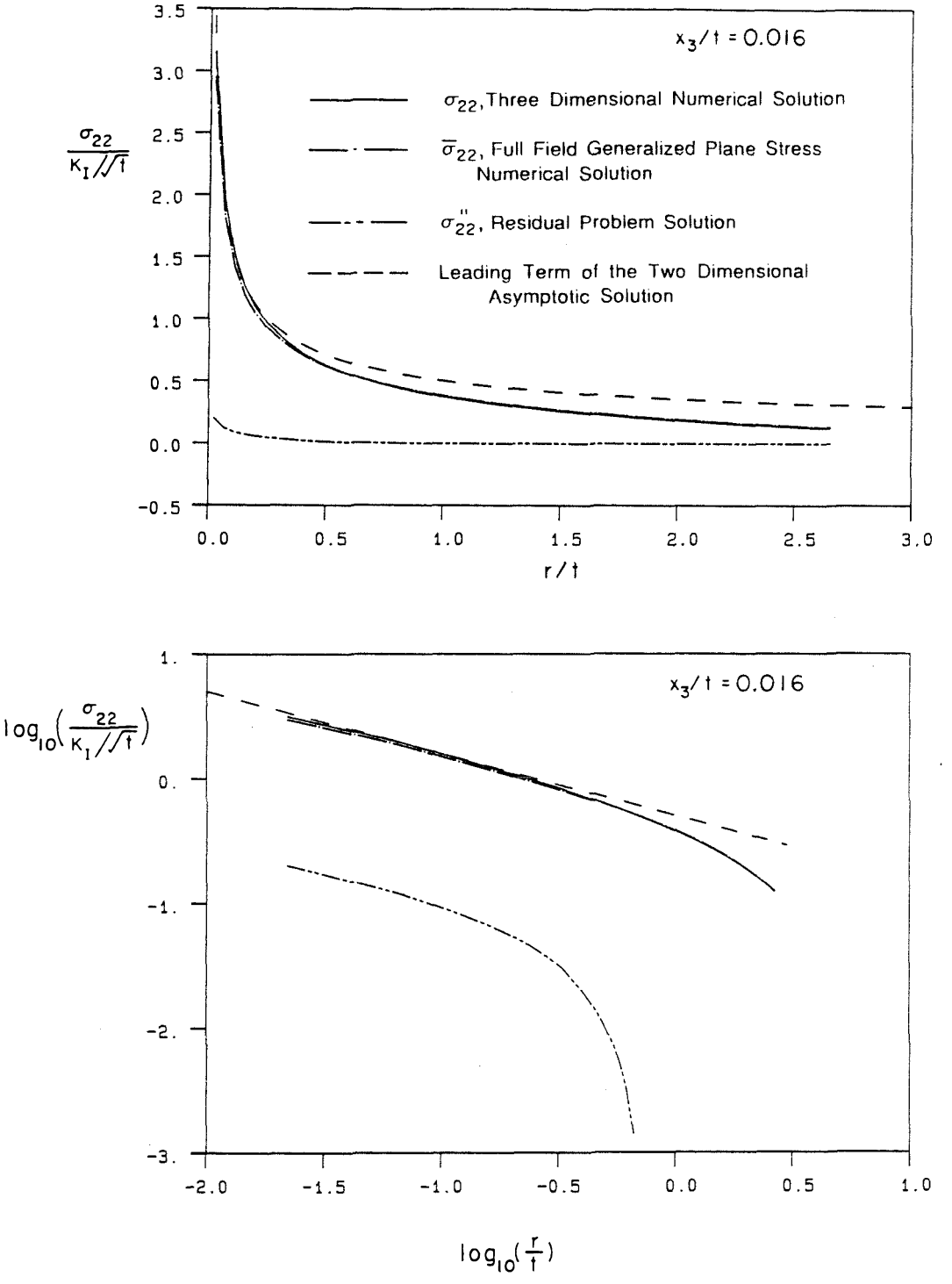


Figure 6.15: x_2-Direction Normal Stresses Near the Center Plane ($x_3/t = 0.016$); $\theta = 45°$.

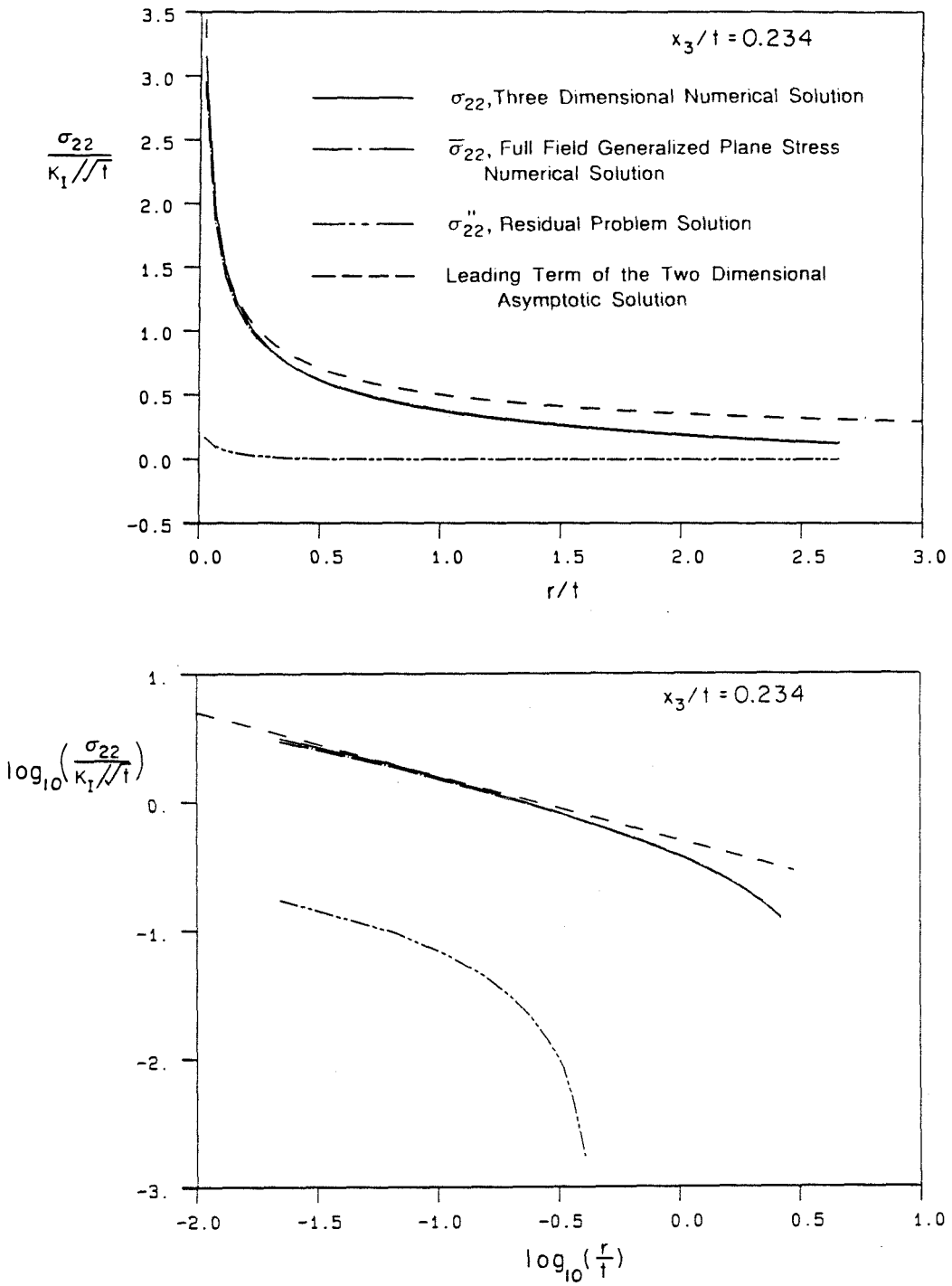


Figure 6.16: x_2-Direction Normal Stresses Near the Plane $x_3/t = \frac{1}{4}$ $(x_3/t = 0.234)$; $\theta = 45°$.

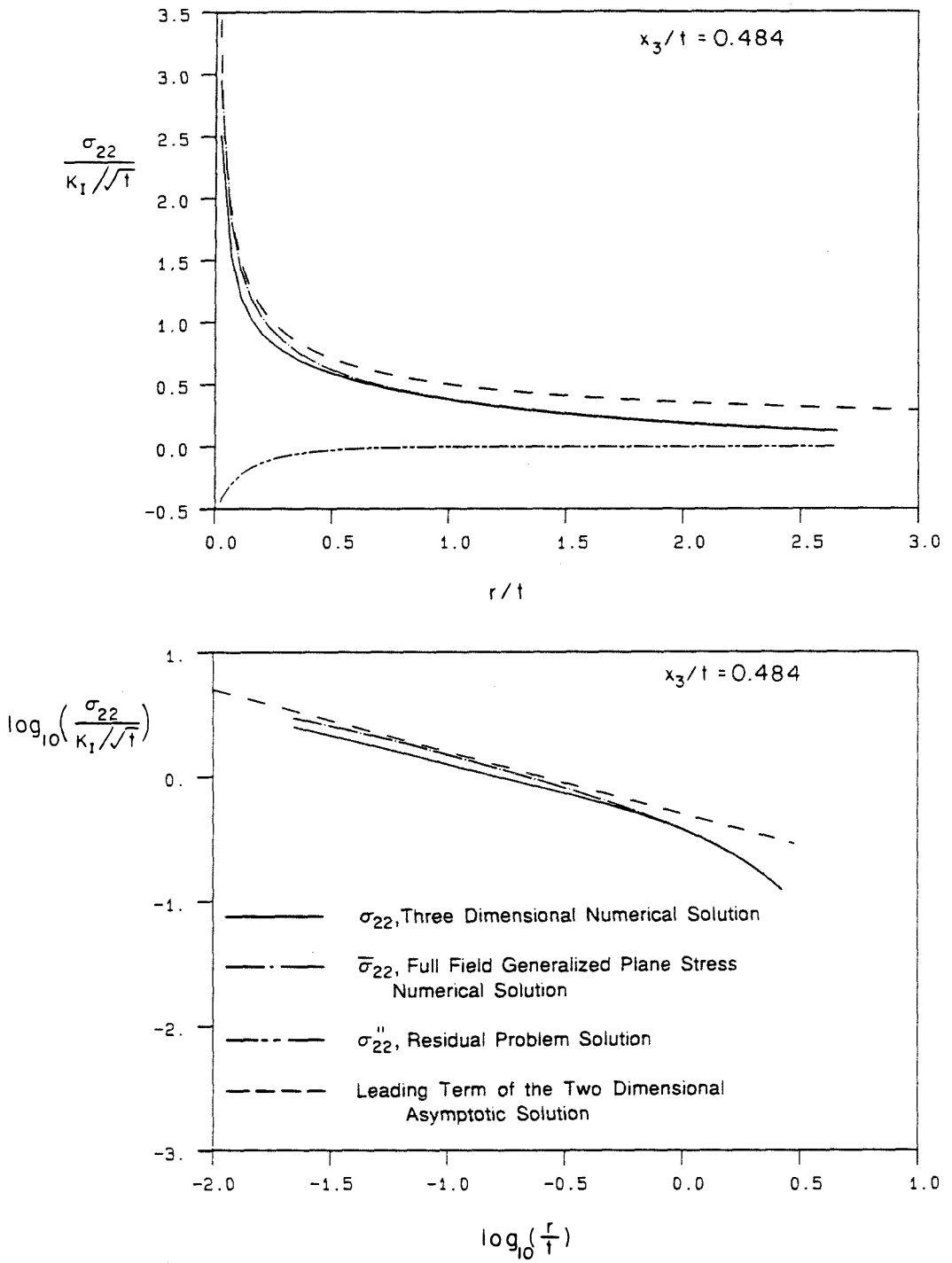


Figure 6.17: x_2-Direction Normal Stresses Near the Free Surface ($x_3/t = 0.484$); $\theta = 45°$.

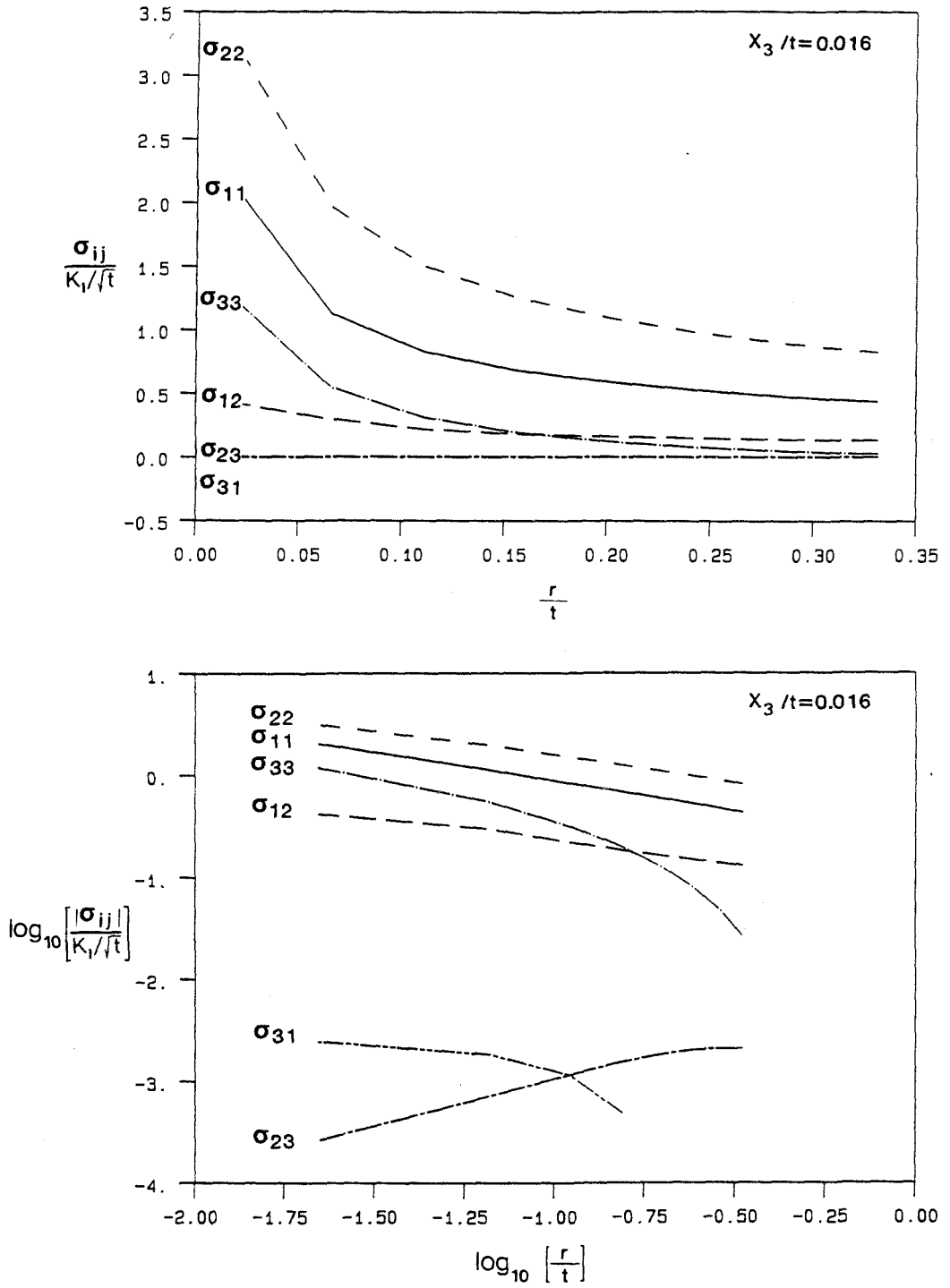


Figure 6.18: Stress Components in the Plate's Interior ($x_3/t = 0.016$, $\theta = 45°$).

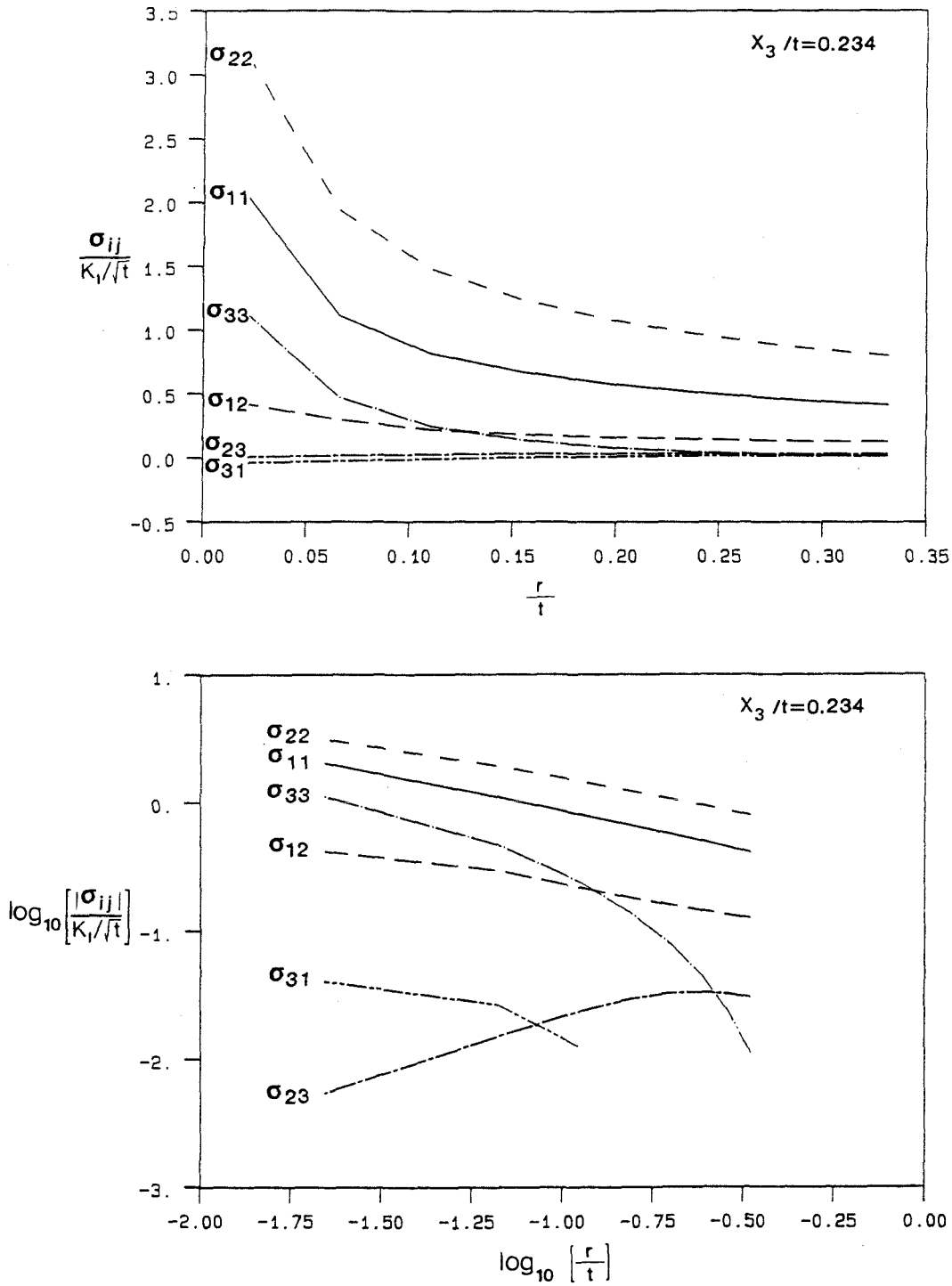


Figure 6.19: Stress Components in the Plate's Interior ($x_3/t = 0.234$, $\theta = 45°$).

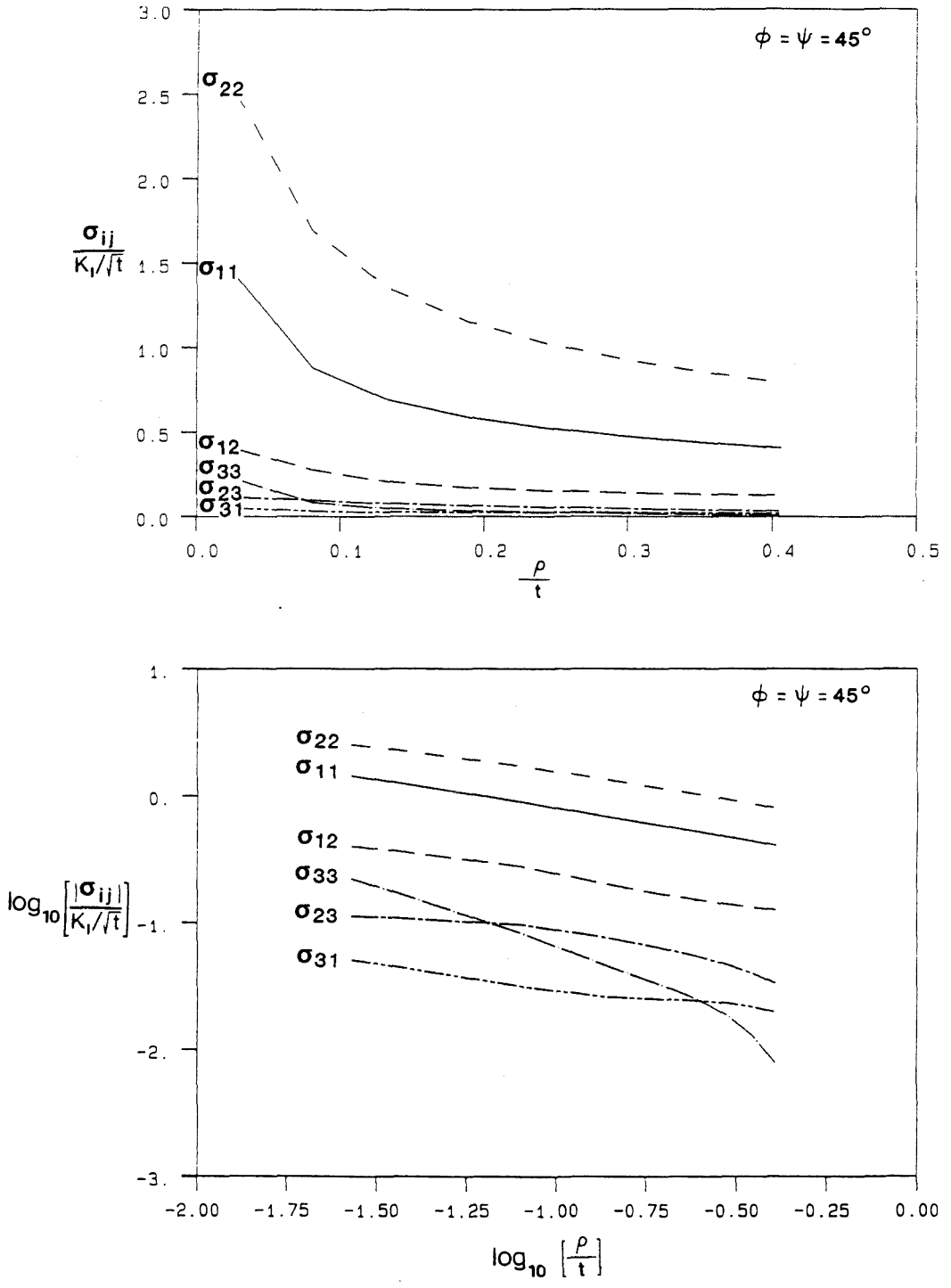


Figure 6.20: Stress Components Near the Crack-Surface Intersection ($\phi = \psi = 45°$).

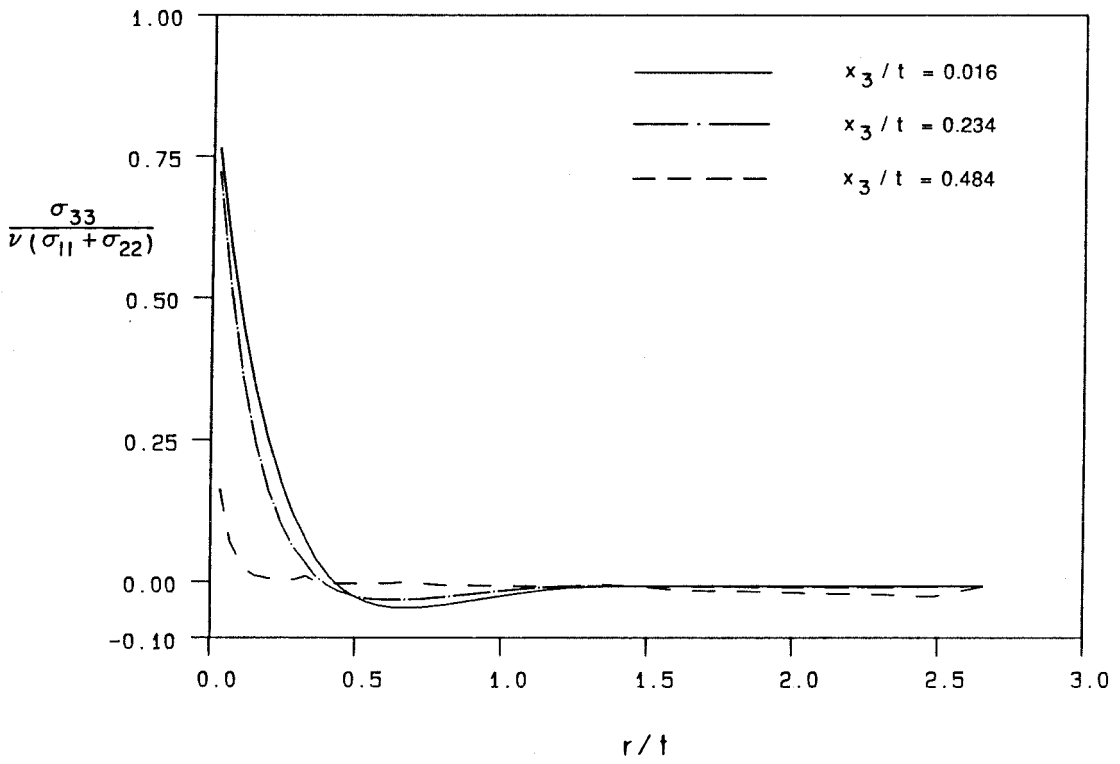


Figure 6.21: Degree of Plane Strain Along the Plane $\theta = 45°$.

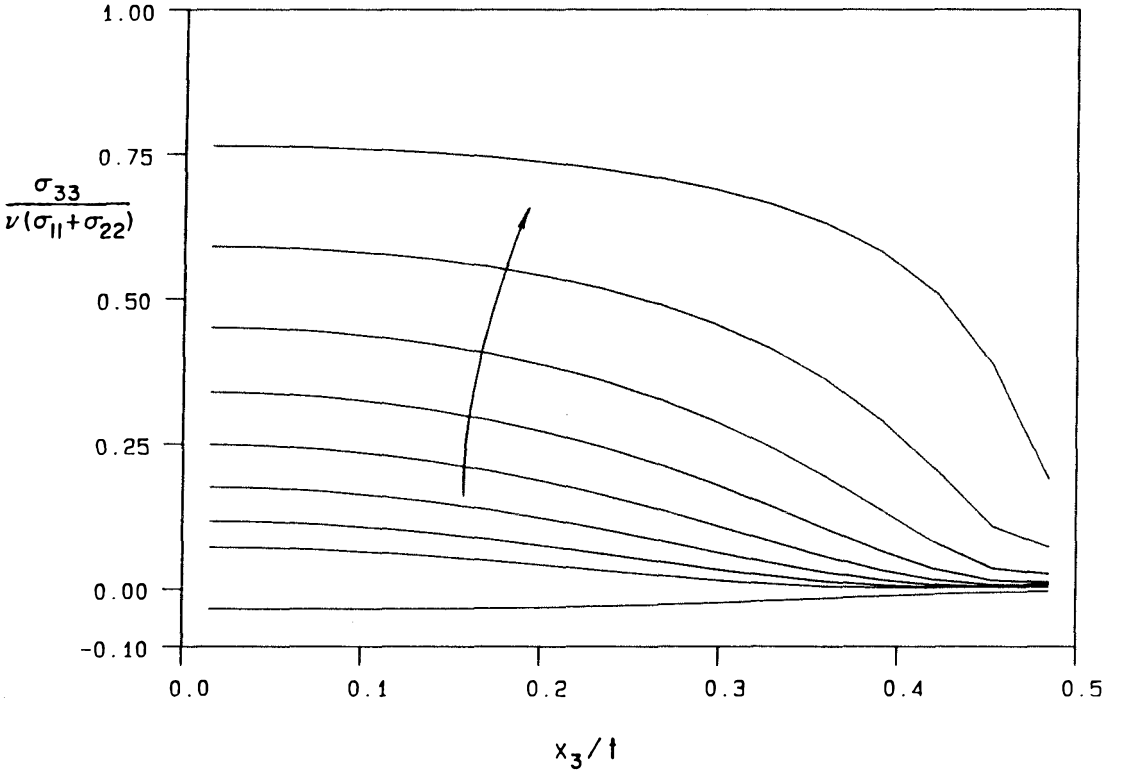


Figure 6.22: Degree of Plane Strain Through the Plate Thickness: $\theta = 45°$, $r/t =$ 0.022, 0.066, 0.110, 0.155, 0.199, 0.243, 0.287, 0.331, 0.535 (the arrow indicates the direction of decreasing r/t values).

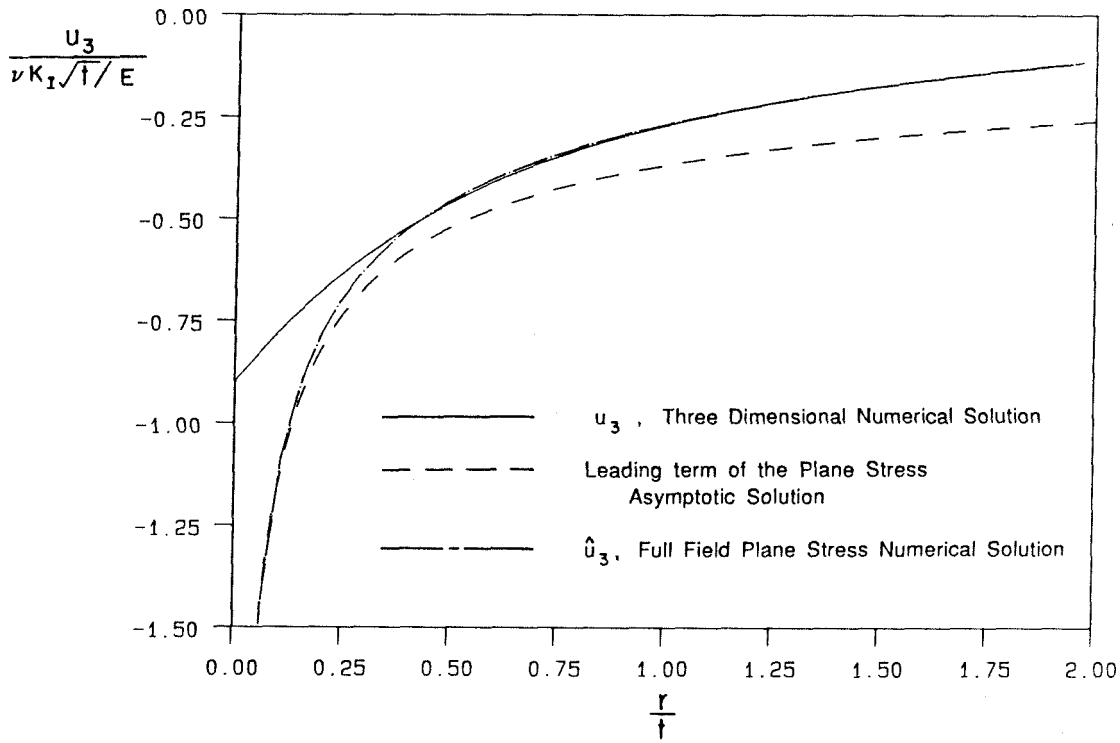


Figure 6.23: x_3−Displacement at the Free Surface ($\theta = 45°, x_3/t = 0.5$).

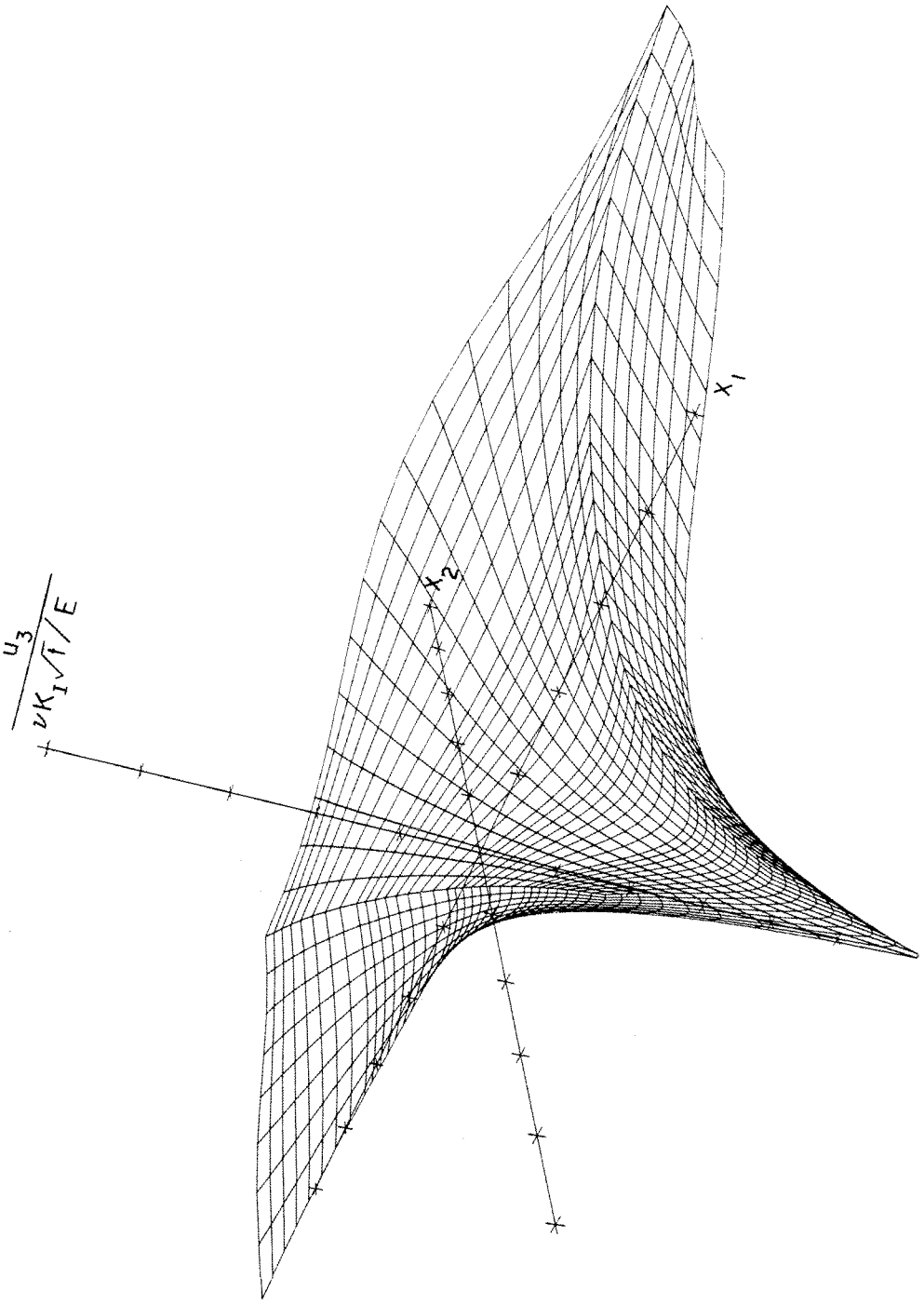


Figure 6.24: Surface Plot of the x_3-Displacement at the Free Surface ($x_3/t = 0.5$)

Chapter 7

Relationships Between the Average Offset and the Stress Drop for Two and Three Dimensional Faults

This chapter describes how the finite element method was used with the multigrid method to solve another linear elastic problem of engineering importance. This problem consisted of determining the relationships between the average offset and the stress drop for two and three dimensional faults. Section 7.1 describes the class of problems that was considered, and gives a summary of previously published results. In section 7.2, the analytical solution for a circular fault in an infinite region is used to establish the accuracy of the finite element technique. The performance of the multigrid method when applied to this problem is also discussed. The problems of rectangular strike-slip faults and two dimensional strike-slip and dip-slip faults in a half-space are numerically solved in sections 7.3 and 7.4, respectively. The results presented in section 7.2 enable the accuracy of these solutions to be estimated. Once again, the ability of the multigrid method to solve these problems is considered in these sections.

The main feature of the work presented in this chapter is that significant differences were seen between the numerical solutions for the rectangular faults (which had an error of about 1%) and some previously reported solutions which were obtained by a different numerical technique [10].

7.1 Problem Description and Previously Published Results

When a fault mechanism is modeled using linear elastostatic theory, the relationship between the average offset, $\overline{D}$, and the stress drop, $\Delta\sigma$, is of primary importance. A relation of the form

$$\Delta\sigma = C\mu\frac{\overline{D}}{l} \tag{7.1}$$

is generally used [34], where μ is the shear modulus of the medium, l is a representative fault dimension, and C is a nondimensional shape factor. For simple geometries, C can be found using analytical techniques; otherwise numerical solutions are required.

Figure 7.1 shows the geometry of the problem which is of primary interest in this chapter. A vertical plane crack, or fault, of width W and length L is centered at the origin O of a rectangular cartesian coordinate system. The fault is buried at a depth d in a half-space that is occupied by a linear elastic, homogeneous, isotropic solid with Poisson's ratio ν and shear modulus μ. Note that the limits $\frac{d}{W} = 0$ and $\frac{d}{W} \to \infty$ correspond to surface breaking and infinitely buried faults, respectively. Throughout this chapter ν is 0.25. The fault is loaded by applying uniform tractions of equal magnitude but opposite direction on its

two faces. Tractions applied in the $\pm x_1-$direction produce strike-slip faulting; tractions applied in the $\pm x_2-$direction produce dip-slip faulting. The magnitude of the applied tractions is usually called the stress drop across the fault, $\Delta\sigma$. The other faulting parameter given in equation (7.1) is the average offset, $\overline{D}$, which is computed from

$$\overline{D} = \frac{1}{A}\int_A (u_\alpha^+ - u_\alpha^-)dA, \tag{7.2}$$

where A is the area of the fault, and $u_\alpha^\pm$ represent the $x_\alpha-$displacement components of the two faces of the fault, $\alpha = 1$ or 2.

No analytical results are available for rectangular faults; however, there is an analytical solution for the case of an infinitely buried elliptical fault [19]. For the special case of an infinitely buried circular fault of radius R, the displacements of the two faces of the fault are given by

$$\left.\begin{aligned} u_1^\pm = \pm\tfrac{R\Delta\sigma}{\mu\eta}(1 - \tfrac{x_1^2}{R^2} - \tfrac{x_2^2}{R^2})^{\frac{1}{2}} = \pm\tfrac{R\Delta\sigma}{\mu\eta}(1 - \tfrac{r^2}{R^2})^{\frac{1}{2}}, \\ u_2 = 0, \\ u_3 = \alpha x_1, \\ \text{where } \alpha = \tfrac{\pi(1-2\nu)\Delta\sigma}{8\mu(1-\nu)\eta}, \\ \eta = \tfrac{\pi(2-\nu)}{4(1-\nu)}, \\ r = \sqrt{x_1^2 + x_2^2}. \end{aligned}\right\} \tag{7.3}$$

Using equation (7.1) and equation (7.3), and taking the representative fault dimension l in equation (7.1) as the fault radius, C is given as

$$C = \frac{7\pi}{16} \tag{7.4}$$

for $\nu = 0.25$. There are several features of the solution given in (7.3) that are worth noting. The displacements u_1 are seen to have an axisymmetric distribution, while the values of u_2 are, interestingly, zero all over the fault surface.

The linear variation of u_3 with x_1 indicates that the circular fault surface rotates about the x_2-axis like a rigid disk. The results given in equations (7.3) and (7.4) will be used later to evaluate the accuracy of the three dimensional finite element analysis.

Special cases of the rectangular fault for which there are analytical solutions are the two dimensional cases obtained as the long axis of the fault approaches infinity. With reference to figure 7.1, taking $\frac{L}{W} \rightarrow \infty$ simplifies the strike-slip and dip-slip conditions to those of anti-plane shear and plane strain, respectively. For a two dimensional, vertically orientated strike-slip fault, l is taken as W, and C is given by

$$C = \frac{4}{\pi} \tag{7.5}$$

for $\frac{d}{W} \rightarrow \infty$, and by

$$C = \frac{2}{\pi} \tag{7.6}$$

for $\frac{d}{W} = 0$ [37]. The corresponding analytical solution for dip-slip faults exists only for $\frac{d}{W} \rightarrow \infty$, and is given by

$$C = \frac{16}{3\pi} \tag{7.7}$$

[53]. The expressions for C given in equations (7.5) and (7.6) are independent of Poisson's ratio, while that in equation (7.7) is for Poisson's ratio 0.25.

Numerical Green's function solutions for C have been reported for vertically oriented, rectangular strike-slip faults at various depths of burial [10]. A range of aspect ratios $\frac{L}{W}$ were considered including the two dimensional limit. Results are presented in table 7.1 in brackets. The values of C at the two dimensional limit agree with the analytical values given above for the surface breaking

and infinitely buried faults. However, the results for the faults of finite length may be questioned through the following argument [29]. For the $\frac{L}{W} = 1$ and 2 strike-slip faults that break the surface, the values for C given in [10] show that C decreases by more than a factor of two as the value of $\frac{L}{W}$ is doubled. The rate of decrease of C should be less than the rate of increase of $\frac{L}{W}$. This is due to the behavior of C as $\frac{L}{W}$ becomes small, and the fact that C is nonzero for $\frac{L}{W} \rightarrow \infty$. For small values of $\frac{L}{W}$, the rectangular strike-slip fault experiences the same conditions as an infinitely long dip-slip fault that is infinitely buried. A consequence of this is that C for the strike-slip fault will be given by $\frac{1.70}{L/W}$ as $\frac{L}{W}$ becomes small, since 1.70 is the value of C for an infinitely long, and infinitely buried, dip-slip fault. In other words, C will be proportional to $\frac{1}{L/W}$ for a strike-slip fault as $\frac{L}{W}$ becomes small. Since C is nonzero for $\frac{L}{W} \rightarrow \infty$, the rate at which C varies with $\frac{L}{W}$ is expected to be less than -1 for nonzero values of $\frac{L}{W}$.

There is another reason for questioning the previously published results. Comparison of the data for the infinitely buried square fault given in [10] and the solution for the circular fault in equation (7.4) shows the square fault to be less stiff than a circular fault of the same area. This is contrary to what might be expected due to the extra constraint imposed by the corners of the square fault. In this comparison, the fault stiffness is taken as the total force per unit average displacement, and is given by

$$k = \frac{\Delta\sigma A}{\bar{D}} = \frac{C\mu A}{l}. \tag{7.8}$$

Using $C = 2.1$ for the square fault (the bracketed value in table 7.1) and equation (7.4) for the circular fault, the stiffness of the square fault is 0.86 that of the circular fault of the same area. More will be said on this later in section 7.3.3.

7.2 A Circular Fault in an Infinite Region

Before attempting to solve the rectangular fault problems, a finite element mesh was constructed to solve the infinitely buried circular fault problem discussed previously. When an acceptable agreement between the numerical results and the analytical solution had been obtained, similar meshes (in terms of both overall and element dimensions) were constructed to compute C for rectangular strike-slip faults. Two dimensional versions of these meshes were then used to calculate C for two dimensional strike-slip and dip-slip faults. This section presents the meshes used to solve the circular fault problem, comments on the performance of the multigrid method, and discusses the agreement between the numerical and analytical results.

7.2.1 Application of the Finite Element Method

By considering the symmetry of the circular fault problem, only one eighth of the infinite region had to be discretized. This was chosen to be the octant with $x_i \geq 0,\ i = 1,2,3$. Whereas the actual domain extended to infinity, the finite element mesh was truncated a large distance away from the origin (ten fault radii). All nodes were fixed on this artificial boundary. Thus the region that was modeled with linear brick elements was a cube with sides ten fault radii in length. Figure 7.2 shows the coarsest mesh used in the multigrid solution of the circular fault problem. The three finer meshes that were also used were formed by repeatedly subdividing the coarsest mesh (each element was divided into eight). This resulted in a series of four meshes that each had 58,419, 8,139, 1,251, and 228 degrees-of-freedom (17,408, 2,176, 272, and 34 elements). Figures 7.3, 7.4, and 7.5

show details of the fine meshes near the fault. The smallest element dimension of the finest mesh was 2.75% of the fault radius; $3 \times 3 \times 3$ Gauss integration was used when forming the finest mesh stiffness matrix. Figure 7.6 shows the displacement boundary conditions that were applied to the finite element model to describe the loading and symmetry conditions.

7.2.2 Performance of the Multigrid Method

As in the case of the crack problem discussed in chapter 6, the MG_1-GS multigrid method was used to solve the circular fault problem. Table 7.2 shows the number of multigrid cycles and the solution times required for the solution of this problem for various values of γ, ν_1, and ν_2. Figure 7.7 shows the convergence behavior of these MG_1-GS methods.

The optimal method was given by $\gamma = 3$ and $\nu_1 = \nu_2 = 10$. The reason for this particular choice of the parameter γ was discussed in section 6.2.1. It is interesting to note that the MG_1-GS method is much less sensitive to the choice of γ for the circular fault problem than for the crack problem; this is due to the absense of any significant bending deformation in the former problem. The times required to form the finest mesh stiffness matrix and the coarse mesh stiffness and interpolation matrices were 43% and 7% of the solution time of the fastest MG_1-GS method. This once again shows that the computational effort associated with the coarse meshes was fairly small.

The Jacobi preconditoned conjugate gradient method (JCG) was also used to solve the circular fault problem using the finest mesh shown in figure 7.5. The convergence behavior of the method is shown in figure 7.8. 310 cycles of iteration

were required, which took 4.00×10^4 CPU secs. Therefore the solution time of the JCG method was 1.6 that of the fastest MG_1-GS scheme. However, the total time required by the JCG method was 5.06×10^4 CPU secs, which was 1.4 times as long as the fastest MG_1-GS method. The JCG method needed 3.25×10^7 bytes of storage, which is 56% of the multigrid storage requirement (5.77×10^7 bytes).

The data presented in this section once again demonstrates the effectiveness of the multigrid method when used to solve a large linear elastic solid mechanics problem. Although the optimal MG_1-GS method appears to be given by $\gamma = 3$, $\nu_1 = \nu_2 = 10$, the parameters that were used in the production runs for the solution of the fault problems discussed in sections 7.3 and 7.4 were $\gamma = 1$, $\nu_1 = \nu_2 = 5$. This was done because it was considered too expensive to determine the optimum parameters for each of the different geometries considered.

7.2.3 Numerical Results

One of the benefits of using the multigrid method is that the coarse meshes are available to solve the problem under consideration. This allows the convergence property of the finite element meshes to be observed. Improved numerical results can then be obtained by extrapolation using the measured convergence rate. Figures 7.9 and 7.10 show plots of u_1 and u_3 versus x_1 along the x_1-axis, respectively, computed for the circular fault using the three finest meshes. Also shown is the analytical solution given in equation (7.3). The meshes are identified by means of a typical mesh size; the finest mesh is referred to as mesh 1, the next coarsest as mesh 2, etc. The plots of u_1 and u_3 clearly indicate that the numerical results are converging to the analytical solution as the meshes are refined. Figure 7.10

demonstrates the rapid decay in the u_3–displacements away from the fault. Figure 7.11 shows the values of C computed from the numerical results obtained from the three meshes and the analytical result given in equation (7.4). The numerical values of u_1 were integrated using 3×3 Gauss integration. The error in the value of C computed on the finest mesh was 2.8% of the exact solution. This computed value of C was improved by linearly extrapolating the results computed using the two finest meshes. This results in a new value of C of 1.37, which gives an error of only 0.3% of the exact solution. Thus, a simple linear extrapolation between the values of C computed on the two finest meshes reduced the error to less than 1%.

The near linear convergence of C can be explained in the following way. There are two sources of error that need to be considered: the discretization error present in the computed displacements, and the error that is associated with the numerical integration of these displacements to obtain C. The theoretical convergence rate of the displacements computed with the linear brick elements used in the finite element model of the circular fault is quadratic (which means that the error will be reduced by a factor of four for a single mesh refinement). However, the singularity present at the edge of the fault violates the smoothness of the displacements required to yield this rate, and therefore lowers the rate of convergence. This reduced rate applies even in places where the solution is smooth [55]. Figure 7.12 shows the values of u_1 computed at the center of the circular fault (where the solution is smooth) using the three finest meshes and the analytical solution given in equation (7.3). This indicates that the error in the displacements is decreasing at a rate slower than quadratic. In fact, the error in u_1 computed on the finest mesh is 42% of the error on the next finest mesh, suggesting that

the computed displacements are converging at a rate which is nearly linear. A similar observation has been reported in a solution of a two dimensional crack problem where linear quadrilateral elements were used [55], further supporting the nearly linear rate of convergence seen for the circular fault.

The numerical integration of the displacements necessary to compute C also leads to an error. It can be shown for the meshes used in this study that the error present in the numerical integration of u_1 is reduced by a factor of about 0.3 as the mesh size is halved. This was done by assuming that the computed nodal displacements were exact and by making use of the analytical solution for u_1 given in equation (7.3). This gives an upper bound on the convergence rate of C, which is faster than linear.

This discussion shows that it is reasonable to use a linear convergence rate when computing an improved value of C by extrapolation. The accuracy of the extrapolated value obtained for the circular fault shows that the artificial boundary was placed sufficiently far (ten fault radii) from the center of the fault.

7.3 Rectangular Faults in a Half-Space

7.3.1 Application of the Finite Element Method

In view of the accuracy of the results discussed in the previous section, the meshes used for the circular fault served as a basis for the meshes used to solve the rectangular, vertically oriented, strike-slip fault problems described in section 7.1. The values of C were computed for faults with $\frac{L}{W} = 1$ and 2, and at depths

$\frac{d}{W} = 0$, 0.02, 0.165, and ∞. The meshes used to model these geometries had the same basic features as those shown in figures 7.2, 7.3, 7.4, and 7.5; i.e. the elements near the fault had similar dimensions, and the artificial boundary was placed ten times half the width of the fault away from the center of the fault. The applied boundary conditions were similar to those shown in figure 7.6. The degrees-of-freedom in the meshes for the four depths of burial numbered 90,891, 121,827, 105,099, and 58,419, respectively. The larger numbers for the surface breaking and finitely buried faults were due to the need to model one quarter of the domain, rather than the one eighth possible for the infinitely buried fault.

The detail of the fault region for the meshes with $\frac{L}{W} = 1$ and $\frac{d}{W} = 0, 0.02, 0.165$, and ∞ are shown in figures 7.13, 7.14, 7.15, and 7.16. The meshes used to model the faults with $\frac{L}{W} = 2$ were generated by modifying these meshes. For example, figure 7.17 shows the detail of the finest mesh used for the $\frac{L}{W} = 2$, $\frac{d}{W} = 0.165$ fault. This mesh was formed by moving all of the nodes in the coarsest mesh used for the $\frac{L}{W} = 1$, $\frac{d}{W} = 0.165$ fault one unit in the x_1-direction (except those on the $x_2 - x_3$ plane). This coarsest mesh was then uniformly subdivided three times to produce the mesh shown in figure 7.17.

7.3.2 Performance of the Multigrid Method

The solution of the rectangular strike-slip fault problems discussed above enables a great deal of data concerning the performance of the multigrid method to be collected. This information is summarized in tables 7.3 and 7.4. Table 7.3 shows the number of MG_1-GS ($\gamma = 1$, $\nu_1 = \nu_2 = 5$) cycles required to solve the $\frac{L}{W} = 1$ fault problems, together with the solution and total times, and the storage

requirements of the method. The corresponding data for the $\frac{L}{W} = 2$ fault is shown in table 7.4. These tables not only show the performance of the multigrid method when it was used to solve the fault problem on the finest meshes, but also on the next two coarsest meshes. This data demonstrates the effectiveness of the method when it was used to solve large solid mechanics problems on a VAX 11/750 (for example, the $\frac{L}{W} = 1$, $\frac{d}{W} = 0.165$ fault with 105,099 degrees-of-freedom was solved in about 27 CPU hours).

Tables 7.3 and 7.4 show that the number of cycles required for convergence of each fault problem was not independent of the problem size, n. Figures 7.18 and 7.19 show plots of the number of multigrid cycles versus $\log_{10} n$ for the $\frac{L}{W} = 1$ and 2 faults, respectively. These graphs show that the number of cycles were proportional to the logarithm of the problem size. However, this can be considered a weak dependence on n, especially given the large range of n. Tables 7.3 and 7.4 also demonstrate that the storage requirements of the multigrid method are linearly proportional to n.

Another interesting feature of the data presented in this section is that the fault problems with $\frac{L}{W} = 2$ required significantly more multigrid cycles (approximately twice as many) for convergence than those with $\frac{L}{W} = 1$. Since the elements in the meshes used to model the $\frac{L}{W} = 2$ faults had a larger aspect ratio than the elements in the $\frac{L}{W} = 1$ meshes, they would have been subjected to a greater bending deformation. As was noted in chapters 5 and 6, the bending behavior of the coarse meshes can significantly effect the convergence of the multigrid method.

7.3.3 Numerical Results

As in the case of the circular fault, values of C were computed from the finest mesh and the next finest mesh for each fault geometry. A linear extrapolation was then performed to obtain an estimate of C, which is expected to have an error of less than 1% of the exact value. Table 7.1 and figure 7.20 show these estimated values for fault lengths of $\frac{L}{W} = 1$ and 2. These results exceed those values given in [10]; for example, for the faults that break the surface, the values of C presented in this chapter are higher by 20% for $\frac{L}{W} = 1$ and by 58% for $\frac{L}{W} = 2$. The present results are thought to be correct since verification of the computational technique was performed on the circular fault. The square fault stiffness is now 1.05 that of the circular fault, a more reasonable value (see section 7.1). Note also that the values for C computed in the thesis decrease by a factor of less than 2 as $\frac{L}{W}$ is increased from 1 to 2 for all values of $\frac{d}{W}$, which is physically realistic. It is possible that the previous formulation omitted the nature of the stress singularities present at the corners of the rectangular fault. This would explain the agreement in the results for $\frac{L}{W} \to \infty$, since the effect of the singularities would be reduced as the length of the fault is increased.

Figure 7.21 shows the distribution of slip (the u_1-displacement component) on the rectangular faults at various depths of burial. The main feature of the distributions shown in figure 7.21 is the insensitivity of the u_1-displacement component to the depth of burial. The faults at depths $\frac{d}{W} = 0.02$, 0.165, and ∞ all have similar distributions of slip; the distribution changes only when the faults break the surface (i.e. $\frac{d}{W} = 0$). This is reflected in the sharp drops in C as the faults break the surface shown in figure 7.20.

7.4 Two Dimensional Faults in a Half-Space

7.4.1 Application of the Finite Element Method

Two dimensional versions of the meshes used for the rectangular faults were generated to compute C for two dimensional (i.e. $\frac{L}{W} \rightarrow \infty$), vertically oriented, strike-slip and dip-slip faults. These meshes were exactly the same as the $x_2 - x_3$ plane of the three dimensional meshes. Linear quadrilateral finite elements were employed. The strike-slip fault was modeled by applying tractions parallel to the x_1-axis and imposing anti-plane shear conditions, while the dip-slip fault was modeled by applying tractions in the x_2-direction and imposing plane strain conditions. Since a two dimensional region was being modeled (which required substantially fewer degrees-of-freedom than the three dimensional region), two further mesh refinements were possible to examine the accuracy of the computed values of C. These two refinements were applied to the two dimensional versions of the meshes used to model the $\frac{d}{W} \rightarrow \infty$ faults (see figure 7.16). This produced meshes that had 41,538 and 20,769 degrees-of-freedom to model the dip-slip and the strike-slip faults, respectively.

The discussion in this chapter has been confined to vertical faults (i.e. faults with a dip of 90°). The case of a dip-slip fault that has a small dip is also of interest. A two dimensional mesh (shown in figure 7.22) was constructed using linear quadrilateral and triangular elements to model a dip-slip fault with a 10° dip angle. The smallest element dimension was about 3% of the width of the fault; the artificial boundary was placed a distance of ten fault widths away from the fault. In all of the problems discussed previously in this chapter, the

symmetry of the geometry precluded any opening of the fault. However, this is not the case for the 10° dip-slip fault. Therefore, in order to keep the fault closed (which is physically more realistic), relative motion perpendicular to the fault was prevented between nodal points on opposite sides of the fault. This meant that the fault remained closed and that only slip along the fault was allowed.

7.4.2 Performance of the Multigrid Method

The analysis of the two dimensional vertical strike-slip and dip-slip faults allows an interesting comparison to be made concerning the performance of the multigrid method. The anti-plane shear conditions that were used to model the strike-slip faults assume that the only nonzero displacement is the out-of-plane component, u_3. The equations of linear elasticity then reduce to

$$\nabla^2 u_3 = 0 \tag{7.9}$$

[35], i.e. the strike-slip fault problems require the solution of Laplace's equation for u_3. There is, therefore, a fundamental difference in the nature of the equations to be solved for the dip-slip and strike-slip problems.

Tables 7.5 and 7.6 give the same data on the performance of the multigrid method when applied to the two dimensional faults that was presented in tables 7.3 and 7.4 for the three dimensional faults discussed in the previous section. It should be noted that the meshes used for the two types of faults buried at different depths were exactly the same; the difference in the number of degrees-of-freedom in each mesh is due to the fact that conditions of plane strain require two unknowns per node (u_1 and u_2), whereas anti-plane shear conditions require only one (u_3). The multigrid method performs considerably better when it is used

to solve the strike-slip problems, since the number of cycles required is less than for the corresponding dip-slip fault problems. The number of cycles required for convergence of the method is independent of the problem size for the strike-slip faults, with the exception of the $\frac{d}{W} = 0.02$ fault. On the other hand, the number of cycles do depend on the problem size for the $\frac{d}{W} = 0.165$ and 0.02 dip-slip faults. However, the results presented in chapter 5 indicate that it is possible that the number of cycles would be independent of n if further refinement of the meshes were performed. The data shown in tables 7.5 and 7.6 for the $\frac{d}{W} \rightarrow \infty$ faults once again demonstrates the speed of the multigrid method. The 41,538 degrees-of-freedom dip-slip problem was solved in about 100 CPU minutes on the VAX 11/750; the 20,769 degrees-of-freedom strike-slip problem took only 30 CPU minutes.

The mesh shown in figure 7.22 that was used to model the 10° dip-slip fault is the only mesh in this thesis that incorporates linear triangular elements. These elements were used to represent the acute angle at the fault-surface intersection. An alternative way of doing this would be to use a linear quadrilateral element, and place two of its nodes at the fault-surface intersection point. However, this was found to severely affect the performance of the multigrid method. To demonstrate this, the 10° dip-slip problem was solved on each of the two meshes shown in figures 7.23 and 7.24, using a two mesh MG_1-GS method with $\nu_1 = \nu_2 = 5$. The mesh shown in figure 7.23 that included triangular elements consisted of 428 degrees-of-freedom, and required 12 multigrid cycles for a converged solution (which took 24 CPU secs on the VAX 11/750). The mesh that consisted entirely of quadrilateral elements shown in figure 7.24 had 434 degrees-of-freedom, and required 22 multigrid cycles for convergence (44 CPU secs), which is 83% more than

the other mesh. This increase is probably due to the way in which coarse mesh triangular and quadrilateral elements near the fault-surface intersection were subdivided to form fine mesh elements. This is shown in figure 4.8 for the triangular elements, and in figure 7.25 for the quadrilaterals when two nodes are placed at the same location. These figures show that the fine mesh quadrilateral elements near the fault-surface intersection have a larger aspect ratio than the corresponding triangular elements. This means that the coarse mesh will be less effective in capturing the coarse mesh correction in this region due to the extra bending stiffness of the coarse mesh quadrilateral elements. Thus the performance of the multigrid method will be degraded. This problem does not arise when triangular elements are employed.

7.4.3 Numerical Results

The analytical values of C given in equations (7.5) and (7.7) for two dimensional faults in an infinite region were used to examine the accuracy of the computed values of C. The two further mesh refinements discussed in section 7.4.1 allowed closer observation of the convergence behavior of C. Figure 7.26 shows the values of C computed for the two dimensional strike-slip and dip-slip faults using meshes of increasing refinement. A mesh size of 1 corresponds to the same refinement present in the three dimensional meshes having a mesh size of 1. Also shown are the analytical results, which are denoted by a mesh size of 0.

As in the three dimensional case discussed previously, the singularities present at the edge of the fault are expected to reduce the convergence of the displacements from their theoretical quadratic rate. This is evident in figure

7.26; however, it also appears that the numerical results for both the strike-slip and dip-slip faults are converging to values of C that are larger than the analytical solutions. Linear extrapolation using the results from the meshes with size $\frac{1}{4}$ and $\frac{1}{2}$ yield values of C that have an error of 0.4% and 0.6% of the exact solution for the strike-slip and dip-slip faults, respectively. This very small error is due primarily to the effect of the artificial boundary, which was ten times half of the fault width from the center of the fault. Although the artificial boundary was the same distance away for the three dimensional faults examined previously, its effect on the solution near the fault should be less in three dimensions than in two, due to the faster decay of the displacements away from the fault in three dimensions. Figure 7.26 shows that a linear extrapolation using the meshes with size 1 and 2 produce values of C that have an error of 0.2% and 0.1% of the exact solution for the strike-slip and dip-slip faults, respectively. These improved results can be attributed to a fortunate cancelling of errors.

The above discussion suggests that it is still reasonable to calculate C for each two dimensional fault geometry from a linear extrapolation using the values of C computed on the two meshes with mesh sizes 1 and 2. The values of C computed in this way are expected to have an error of less than 1% of the analytical solution. Table 7.1 and figure 7.20 gives these values of C for the two dimensional strike-slip and dip-slip faults at various depths of burial. The dip-slip fault stiffness exceeds that of the strike-slip stiffness, becoming more pronouced as the depth of burial increases.

Figure 7.27 shows the deformed mesh that was used to model the 10° dip-slip fault for conditions of reverse and normal faulting. It can be seen that there

is significant vertical displacement directly above the buried end of the fault. C was computed as 0.48 (which is lower than for the vertical dip) by extrapolating the results from the mesh shown in figure 7.22 and a coarser one. The low value of C is due to the increased flexiblity of the upper portion of the fault.

7.5 Summary of the Numerical Results

For vertically oriented, rectangular, strike-slip faults, C ranges from 2.04 (surface breaking) to 2.55 (infinitely buried) for $\frac{L}{W} = 1$, from 1.26 (surface breaking) to 1.83 (infinitely buried) for $\frac{L}{W} = 2$, and from 0.65 (surface breaking) to 1.28 (infinitely buried) for $\frac{L}{W} \rightarrow \infty$, which is the two dimensional case, where L and W are the fault length and width, respectively. The values for $\frac{L}{W} = 1$ and 2 are 20-58% higher than some previously reported values, but are believed to be correct since verification of the computational technique was performed by examining the circular fault problem. Vertical dip-slip faults are stiffer than corresponding strike-slip ones, especially at deeper burial, as C varies from 0.67 (surface breaking) to 1.70 (infinitely buried) for $\frac{L}{W} \rightarrow \infty$. Shallow dips reduce C due to the increased flexiblity of the upper portion of the fault; C equals 0.48 for a two dimensional dip-slip fault with a 10° dip.

$\frac{L}{W}$	Fault Type	$\frac{d}{W} = 0$	$\frac{d}{W} = 0.02$	$\frac{d}{W} = 0.165$	$\frac{d}{W} \to \infty$
1	strike-slip	2.04	2.35	2.48	2.55
		[1.7]	[1.9]	[2.0]	[2.1]
2	strike-slip	1.26	1.60	1.75	1.83
		[0.8]	[1.1]	[1.3]	[1.4]
∞	strike-slip	0.65	0.99	1.16	1.28
		[0.6]	[1.0]	[1.1]	[1.3]
∞	dip-slip	0.67	1.10	1.41	1.70

Table 7.1: Values of C from this thesis for three dimensional rectangular ($\frac{L}{W} = 1, 2$) strike-slip, and two dimensional ($\frac{L}{W} \to \infty$) strike-slip and dip-slip faults oriented vertically at various depths of burial ($\frac{d}{W} = 0, 0.02, 0.165$, and ∞). Values in brackets [ ] are from [10].

γ	ν_1, ν_2	Number of Multigrid Cycles	Solution Time (CPU secs)
1	5	16	2.72×10^4
2	5	14	2.50×10^4
3	5	13	2.49×10^4
4	5	13	2.69×10^4
5	5	13	2.90×10^4
3	1	62	3.90×10^4
3	5	13	2.49×10^4
3	10	7	2.46×10^4

Table 7.2: Number of MG_1-GS Cycles and Solution Times Required for the Solution of the Circular Fault Problem (58,419 degrees-of-freedom).

$\frac{d}{W}$	n	Number of Cycles	Solution Time (CPU secs)	Total Time (CPU secs)	Storage (bytes)
	90,891	24	6.51×10^4	8.53×10^4	9.07×10^7
0	12,423	18	5.99×10^3	8.44×10^3	1.18×10^7
	1,845	12	3.68×10^2	6.60×10^2	1.74×10^6
	121,827	70	2.59×10^5	2.86×10^5	1.22×10^8
0.02	16,515	47	2.08×10^4	2.41×10^4	1.59×10^7
	2,415	28	1.11×10^3	1.52×10^3	2.36×10^6
	105,099	23	7.32×10^4	9.65×10^4	1.06×10^8
0.165	14,247	17	6.56×10^3	9.44×10^3	1.37×10^7
	2,085	13	4.69×10^2	8.10×10^2	2.04×10^6
	58,419	23	3.91×10^4	5.13×10^4	5.77×10^7
∞	8,139	15	2.24×10^3	3.73×10^3	7.57×10^6
	1,251	12	2.32×10^2	4.17×10^2	1.11×10^6

Table 7.3: Performance of the MG_1-GS ($\gamma = 1$, $\nu_1 = \nu_2 = 5$) Multigrid Method for the $\frac{L}{W} = 1$ Strike-Slip Faults.

$\frac{d}{W}$	n	Number of Cycles	Solution Time (CPU secs)	Total Time (CPU secs)	Storage (bytes)
	90,891	49	1.33×10^5	1.53×10^5	9.07×10^7
0	12,423	36	1.18×10^4	1.43×10^4	1.18×10^7
	1,845	20	5.94×10^2	8.88×10^2	1.74×10^6
	121,827	153	5.51×10^5	5.78×10^5	1.22×10^8
0.02	16,515	96	4.24×10^4	4.58×10^4	1.59×10^7
	2,415	47	1.83×10^3	2.23×10^3	2.36×10^6
	105,099	53	1.68×10^5	1.92×10^5	1.06×10^8
0.165	14,247	39	1.50×10^4	1.78×10^4	1.37×10^7
	2,085	25	8.64×10^2	1.20×10^3	2.04×10^6
	58,419	55	9.22×10^4	1.05×10^5	5.77×10^7
∞	8,139	38	6.06×10^3	7.60×10^3	7.57×10^6
	1,251	20	3.79×10^2	5.64×10^2	1.11×10^6

Table 7.4: Performance of the MG_1-GS ($\gamma = 1$, $\nu_1 = \nu_2 = 5$) Multigrid Method for the $\frac{L}{W} = 2$ Strike-Slip Faults.

$\frac{d}{W}$	n	Number of Cycles	Solution Time (CPU secs)	Total Time (CPU secs)	Storage (bytes)
	1,745	8	4.99×10^1	1.38×10^2	9.07×10^5
0	457	8	1.23×10^1	3.65×10^1	2.38×10^5
	125	8	2.86×10^0	1.07×10^1	6.53×10^4
	2,609	13	1.21×10^2	2.58×10^2	1.36×10^6
0.02	681	11	2.53×10^1	6.01×10^1	3.55×10^5
	185	10	5.19×10^0	1.58×10^1	9.70×10^4
	2,337	9	7.54×10^1	1.97×10^2	1.22×10^6
0.165	609	9	1.89×10^1	5.03×10^1	3.18×10^5
	165	9	4.32×10^0	1.39×10^1	8.66×10^4
	20,769	8	6.14×10^2	1.77×10^3	1.08×10^7
	5,265	8	1.59×10^2	4.66×10^2	2.74×10^6
∞	1,353	8	3.84×10^1	1.08×10^2	7.04×10^5
	357	8	9.74×10^0	2.88×10^1	1.86×10^5
	99	9	2.56×10^0	8.98×10^0	5.19×10^4

Table 7.5: Performance of the MG_1-GS ($\gamma = 1$, $\nu_1 = \nu_2 = 5$) Multigrid Method for the Two Dimensional Strike-Slip Faults.

$\frac{d}{W}$	n	Number of Cycles	Solution Time (CPU secs)	Total Time (CPU secs)	Storage (bytes)
	3,490	12	2.52×10^2	4.15×10^2	1.65×10^6
0	914	11	5.64×10^1	9.53×10^1	4.33×10^5
	250	11	1.27×10^1	2.40×10^1	1.19×10^5
	5,218	36	1.20×10^3	1.44×10^3	2.47×10^6
0.02	1,362	27	2.10×10^2	2.66×10^2	6.46×10^5
	370	17	2.95×10^1	4.54×10^1	1.76×10^5
	4,674	23	6.79×10^2	9.02×10^2	2.22×10^6
0.165	1,218	17	1.18×10^2	1.70×10^2	5.78×10^5
	330	15	2.38×10^1	3.82×10^1	1.57×10^5
	41,538	11	4.00×10^3	6.02×10^3	1.97×10^7
	10,530	11	9.65×10^2	1.49×10^3	4.99×10^6
∞	2,706	11	1.77×10^2	2.99×10^2	1.28×10^6
	714	11	4.43×10^1	7.40×10^1	3.39×10^5
	198	11	9.99×10^0	1.91×10^1	9.40×10^4

Table 7.6: Performance of the MG_1-GS ($\gamma = 1$, $\nu_1 = \nu_2 = 5$) Multigrid Method for the Two Dimensional Dip-Slip Faults.

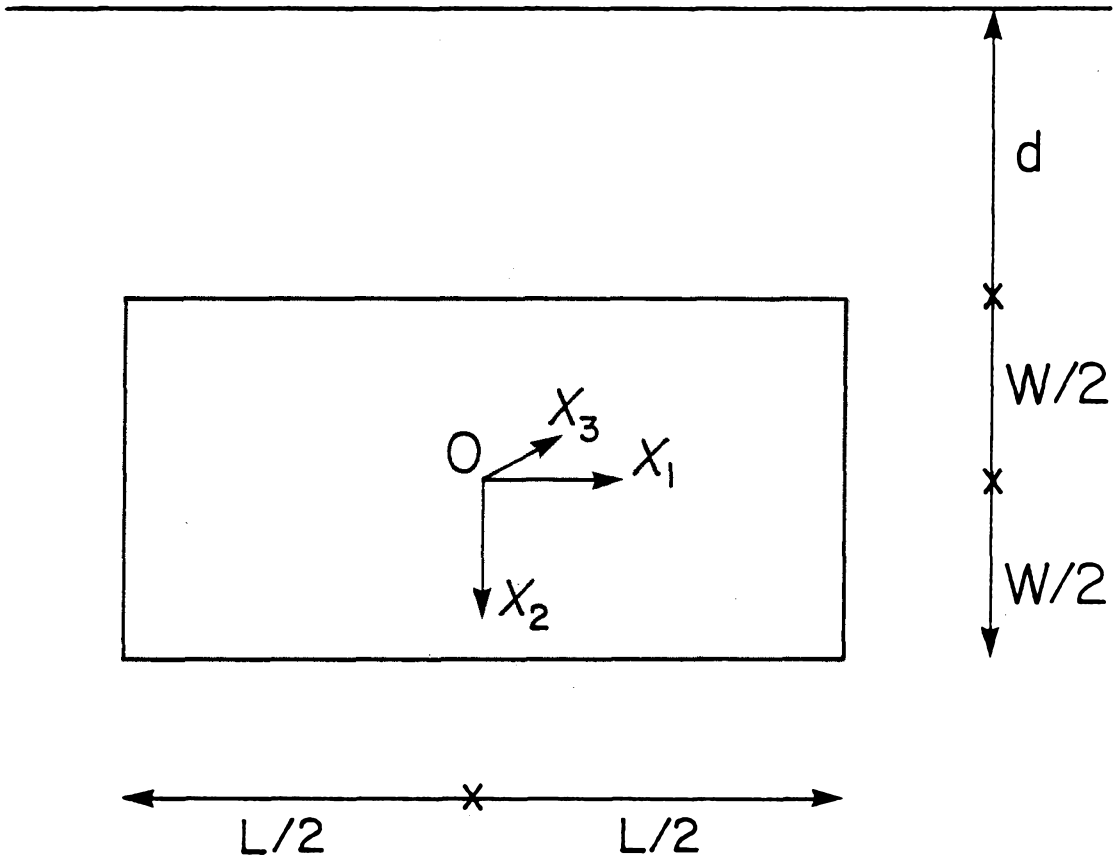


Figure 7.1: The Geometry of a Vertically Oriented Rectangular Fault.

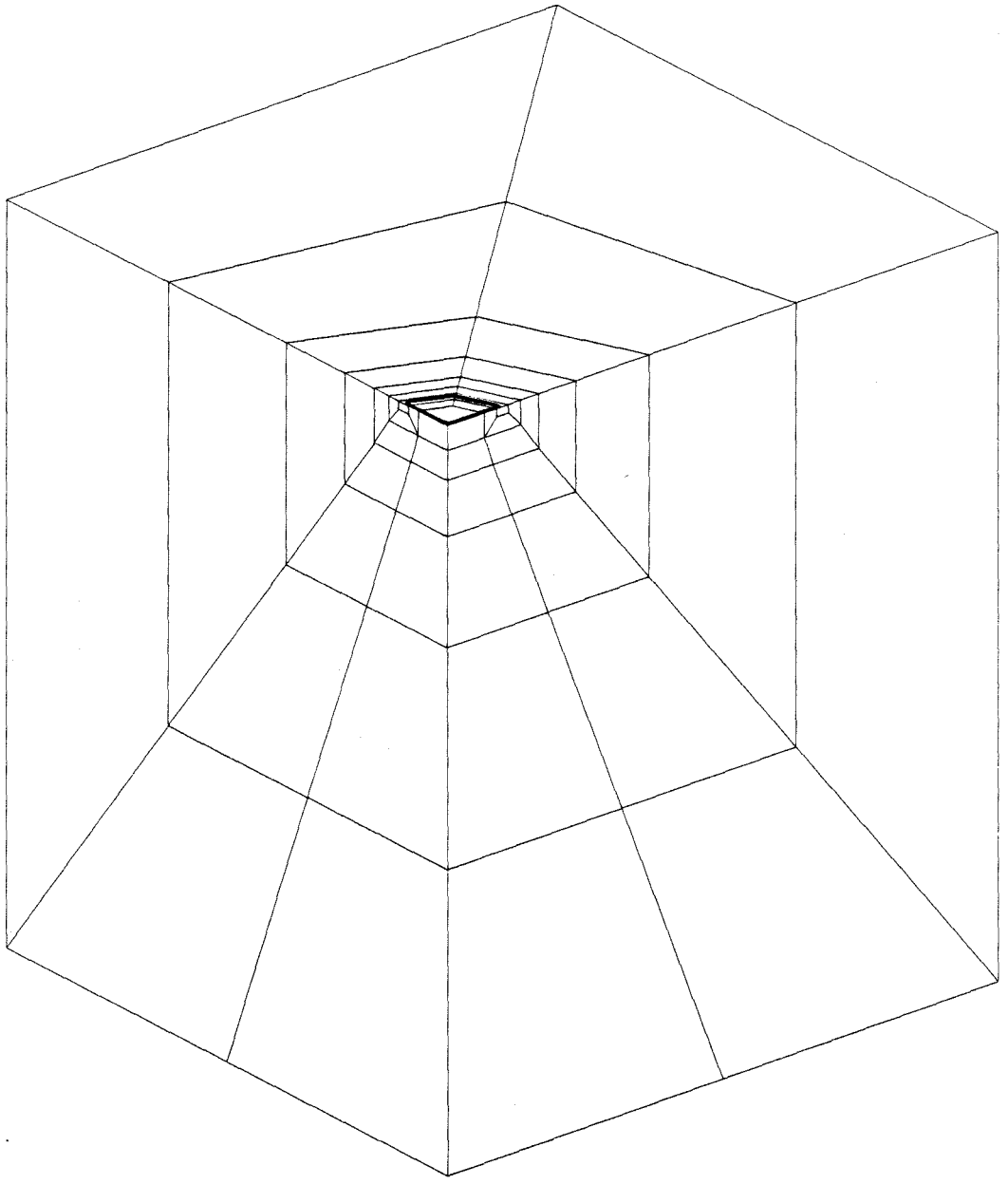


Figure 7.2: The Coarsest Mesh for the Infinitely Buried Circular Fault (the bold line indiates the edge of the fault).

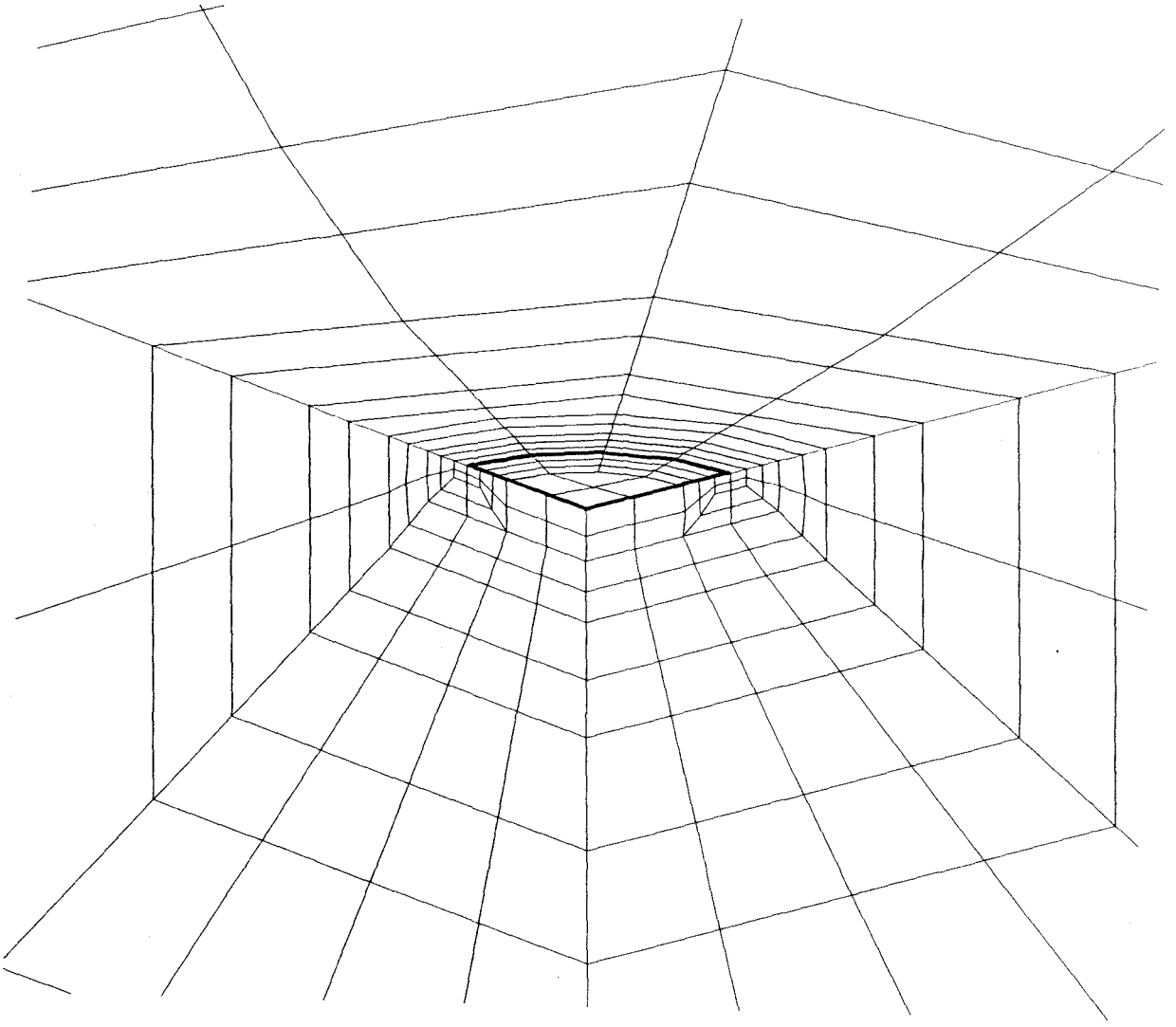


Figure 7.3: Detail of the Mesh With 1,251 Degrees-of-Freedom Used to Model the Infinitely Buried Circular Fault (the bold line indiates the edge of the fault).

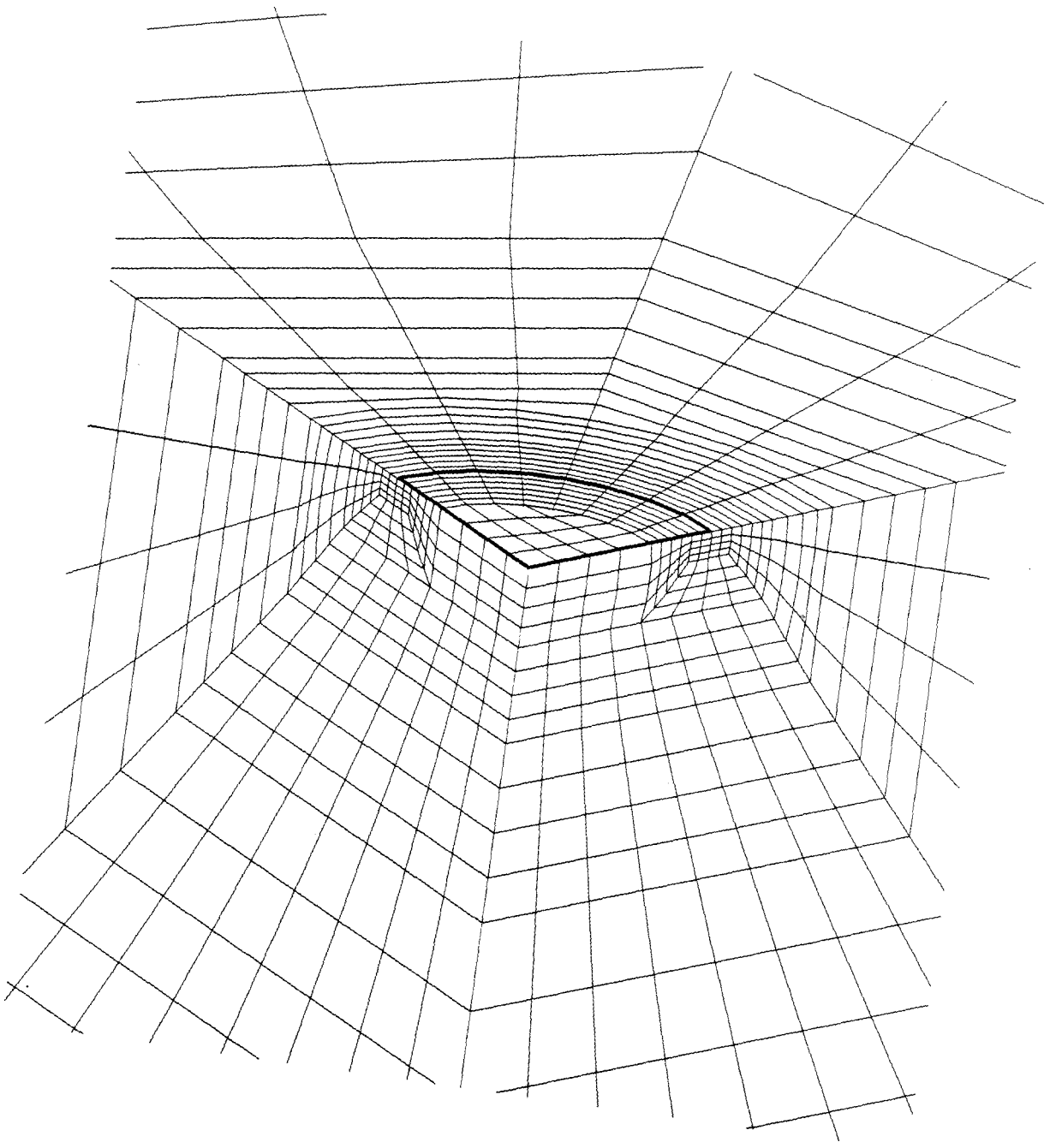


Figure 7.4: Detail of the Mesh With 8,139 Degrees-of-Freedom Used to Model the Infinitely Buried Circular Fault (the bold line indiates the edge of the fault).

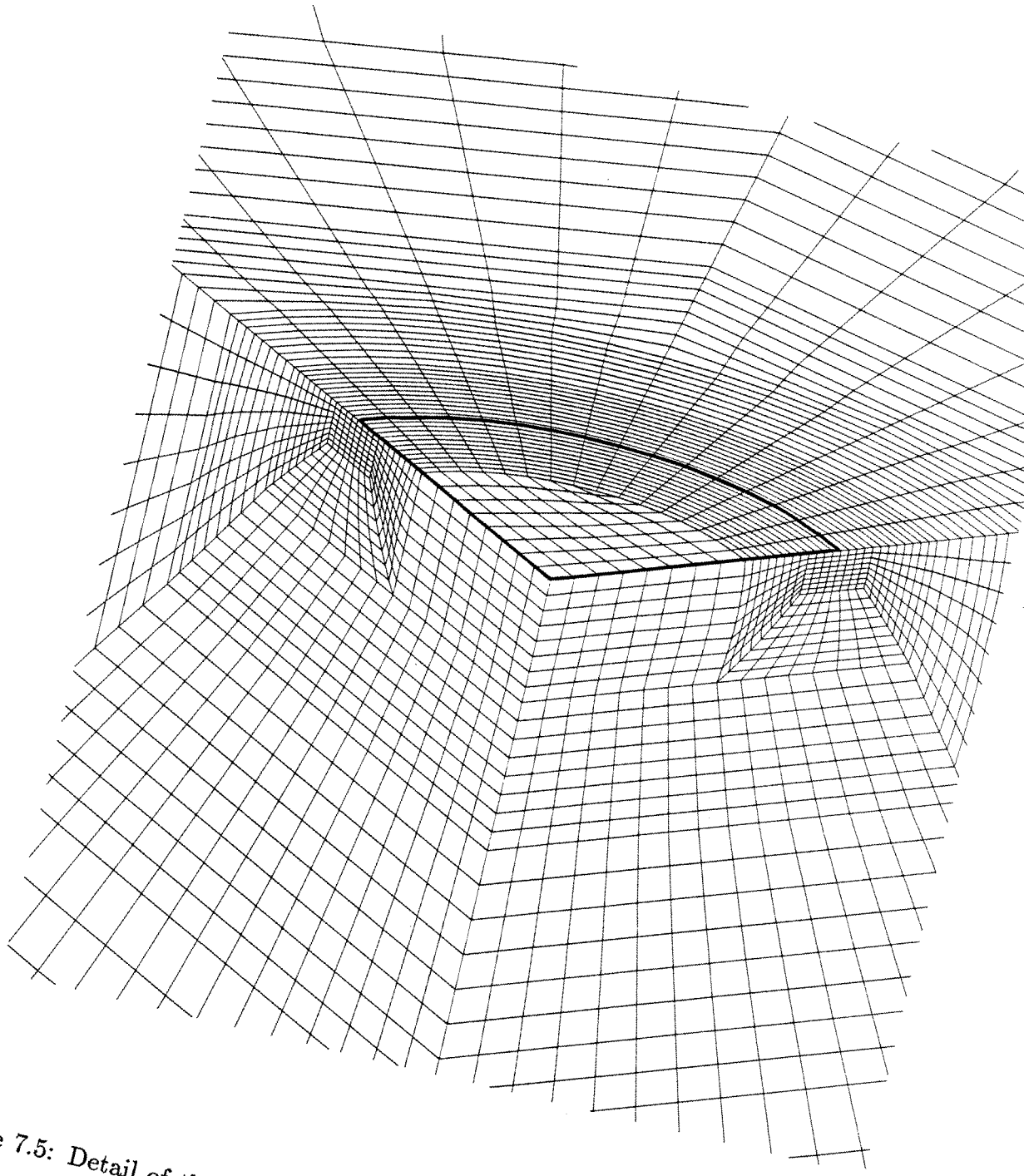


Figure 7.5: Detail of the Mesh With 58,419 Degrees-of-Freedom Used to Model the Infinitely Buried Circular Fault (the bold line indiates the edge

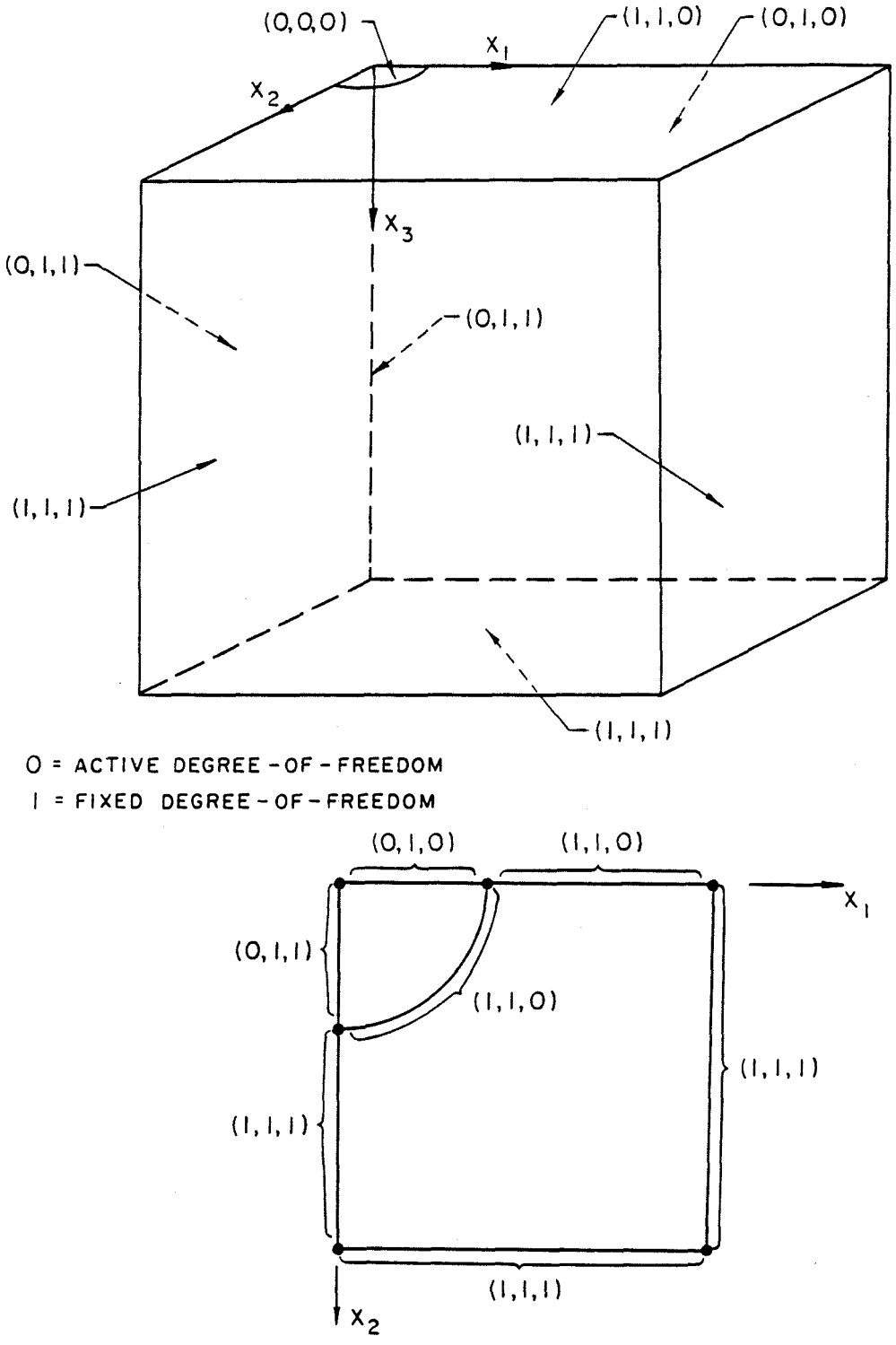
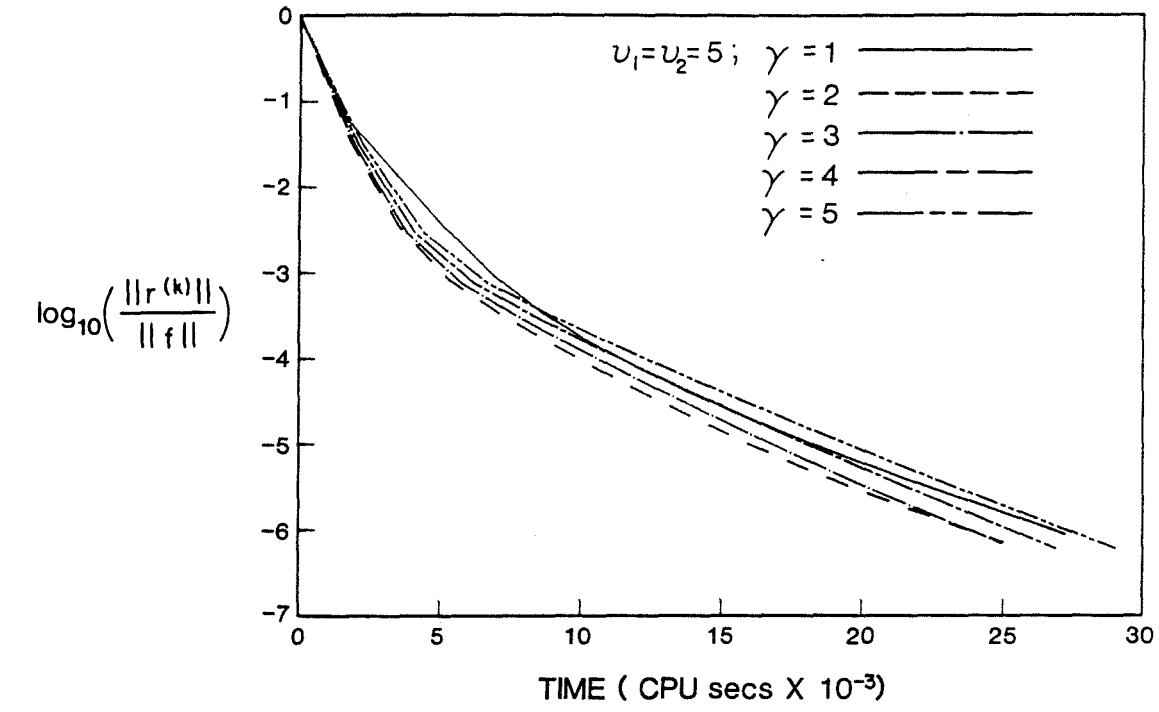


Figure 7.6: The Boundary Conditions for the Infinitely Buried Circular Fault.



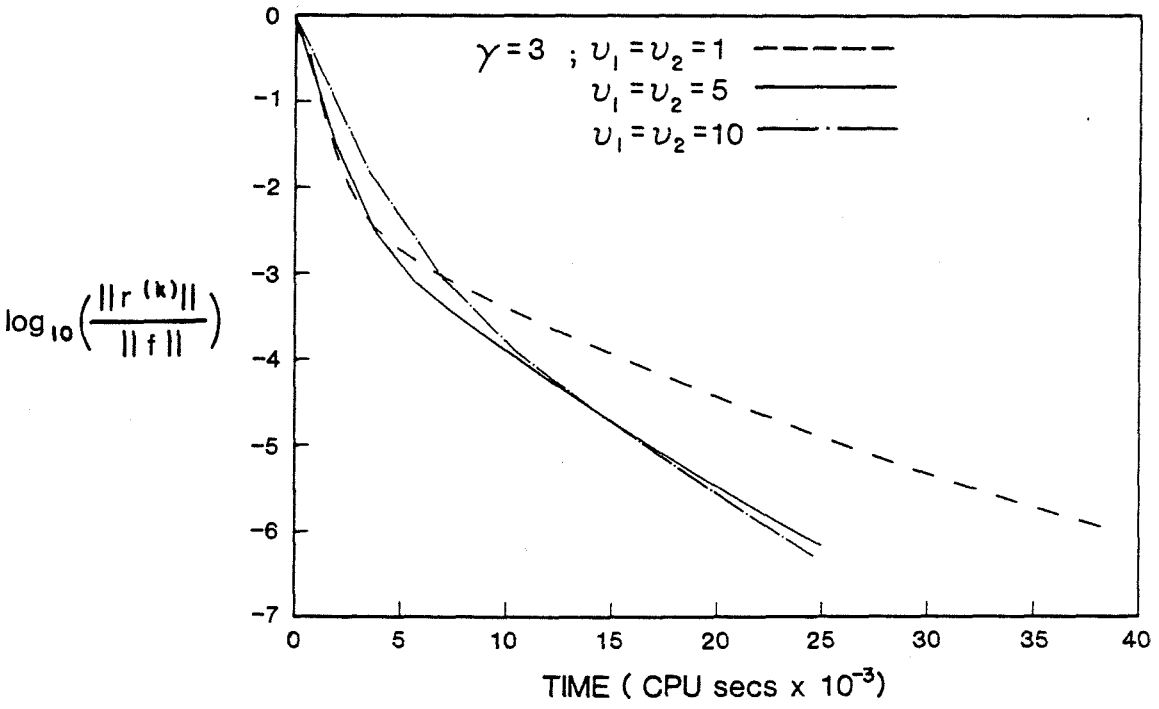


Figure 7.7: The Convergence of the MG_1-GS Method for the Infinitely Buried Circular Fault for Different Values of γ, ν_1, and ν_2 (58,419 degrees-of-freedom).

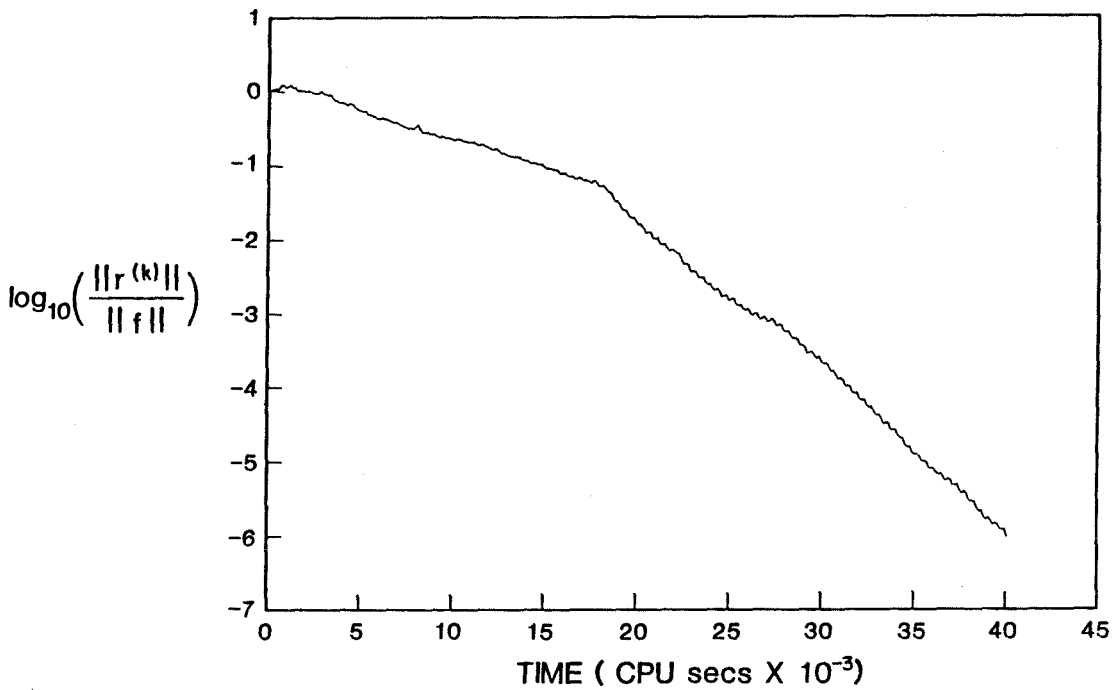


Figure 7.8: The Convergence of the JCG Method for the Infinitely Buried Circular Fault (58,419 degrees-of-freedom).

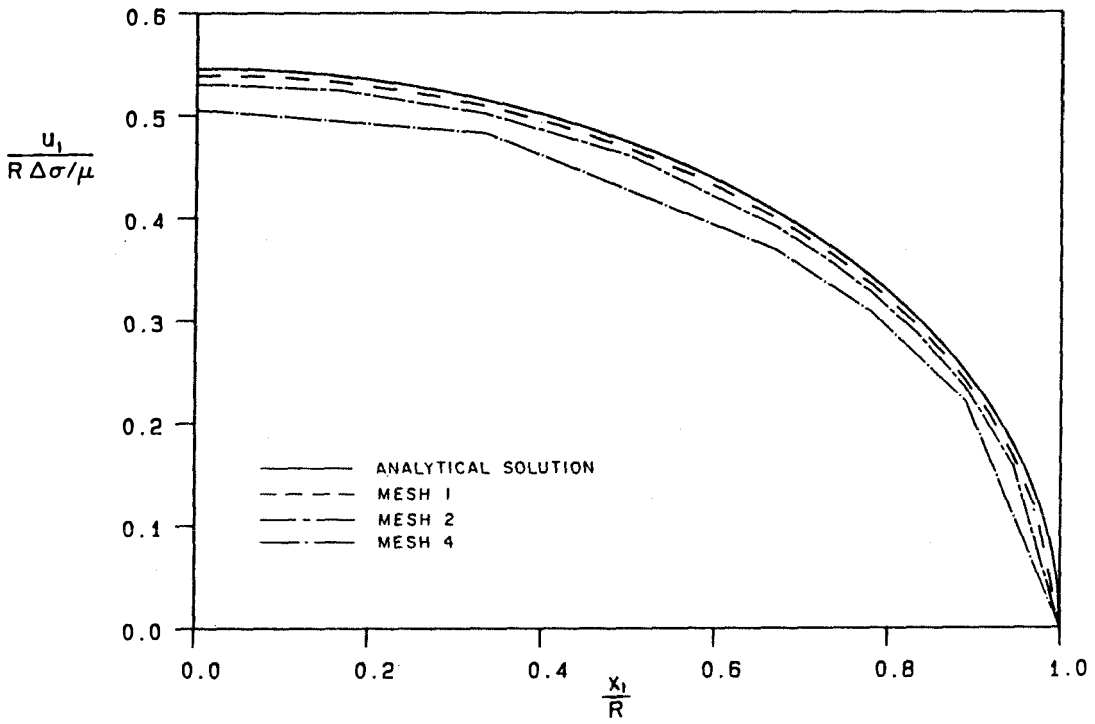


Figure 7.9: u_1-Displacement Versus x_1 Along the x_1-Axis for the Infinitely Buried Circular Fault.

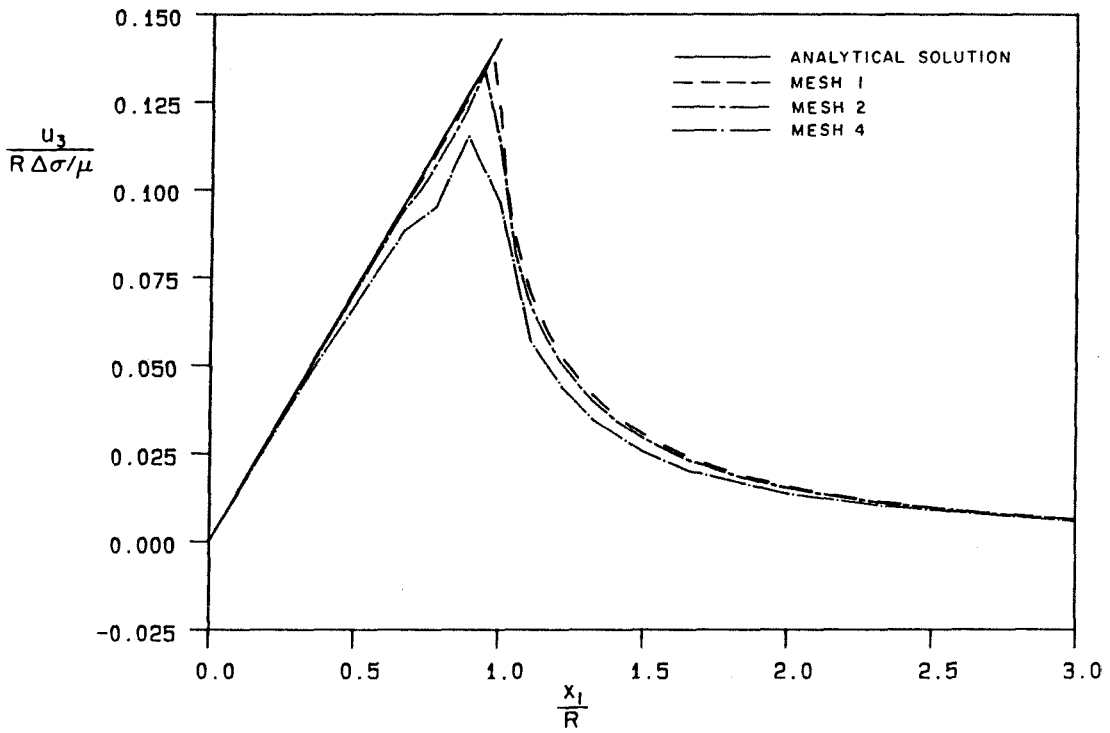


Figure 7.10: u_3-Displacement Versus x_1 Along the x_1-Axis for the Infinitely Buried Circular Fault.

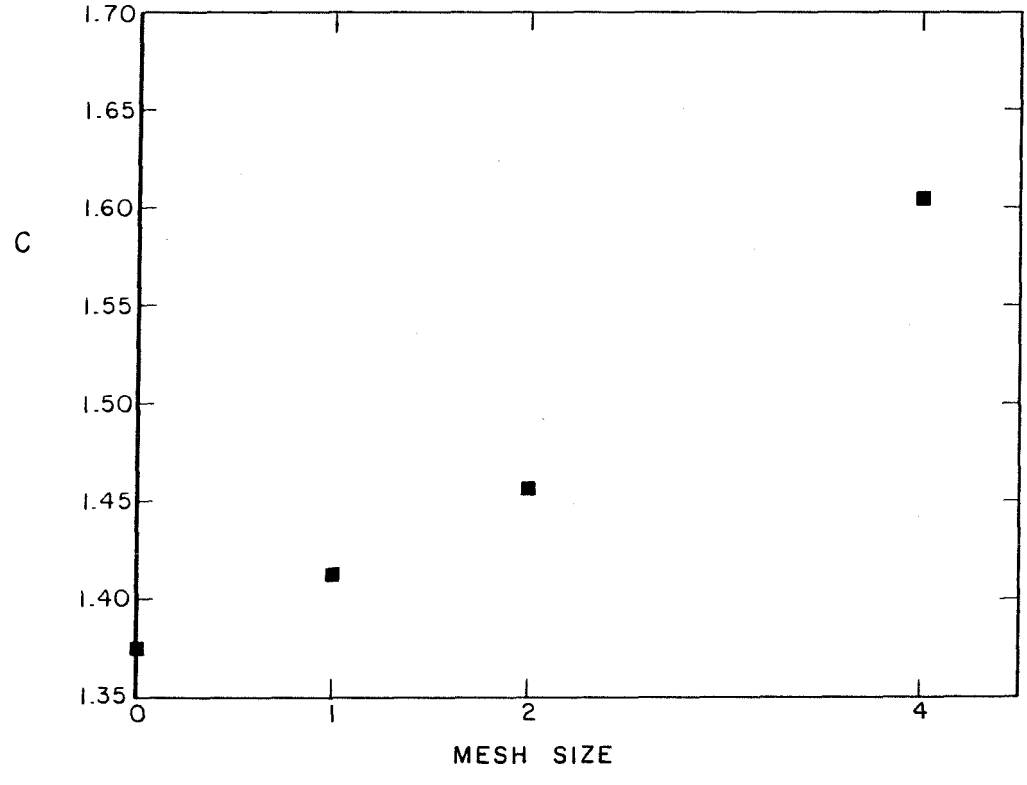


Figure 7.11: Computed and Analytical Values for C for the Infinitely Buried Circular Fault.

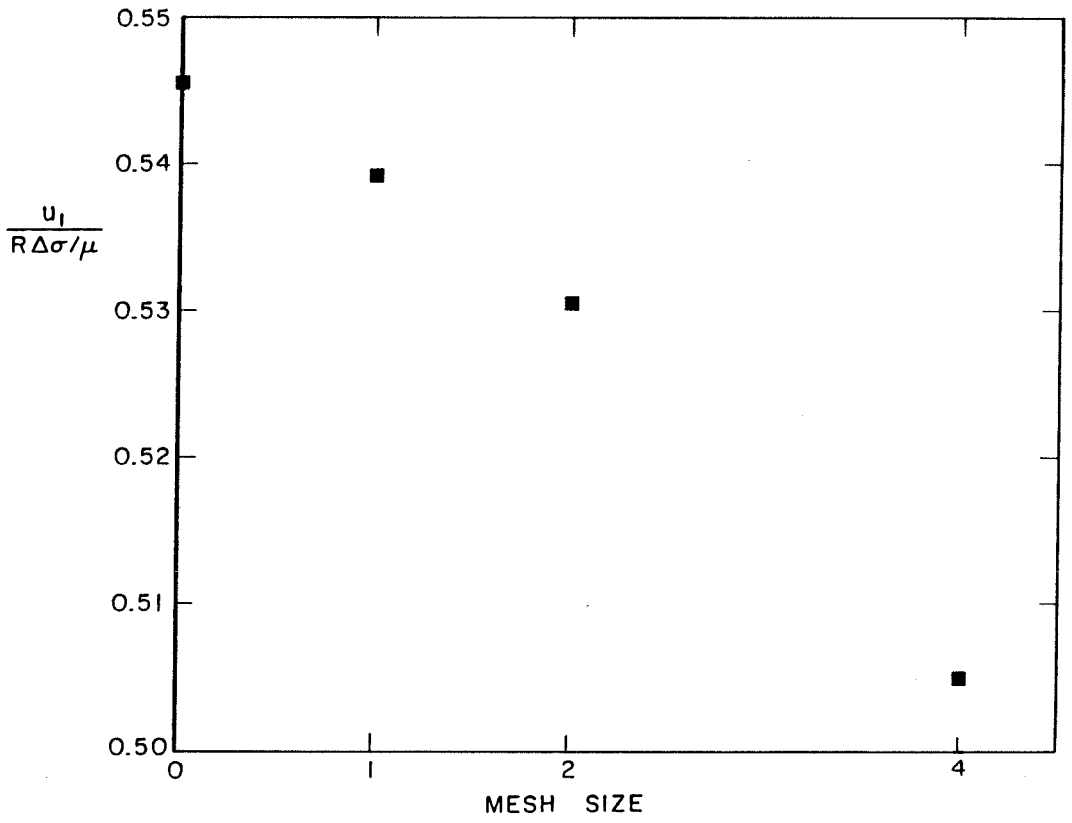


Figure 7.12: Computed and Analytical Values of the u_1—Displacement at the Center of the Infinitely Buried Circular Fault.

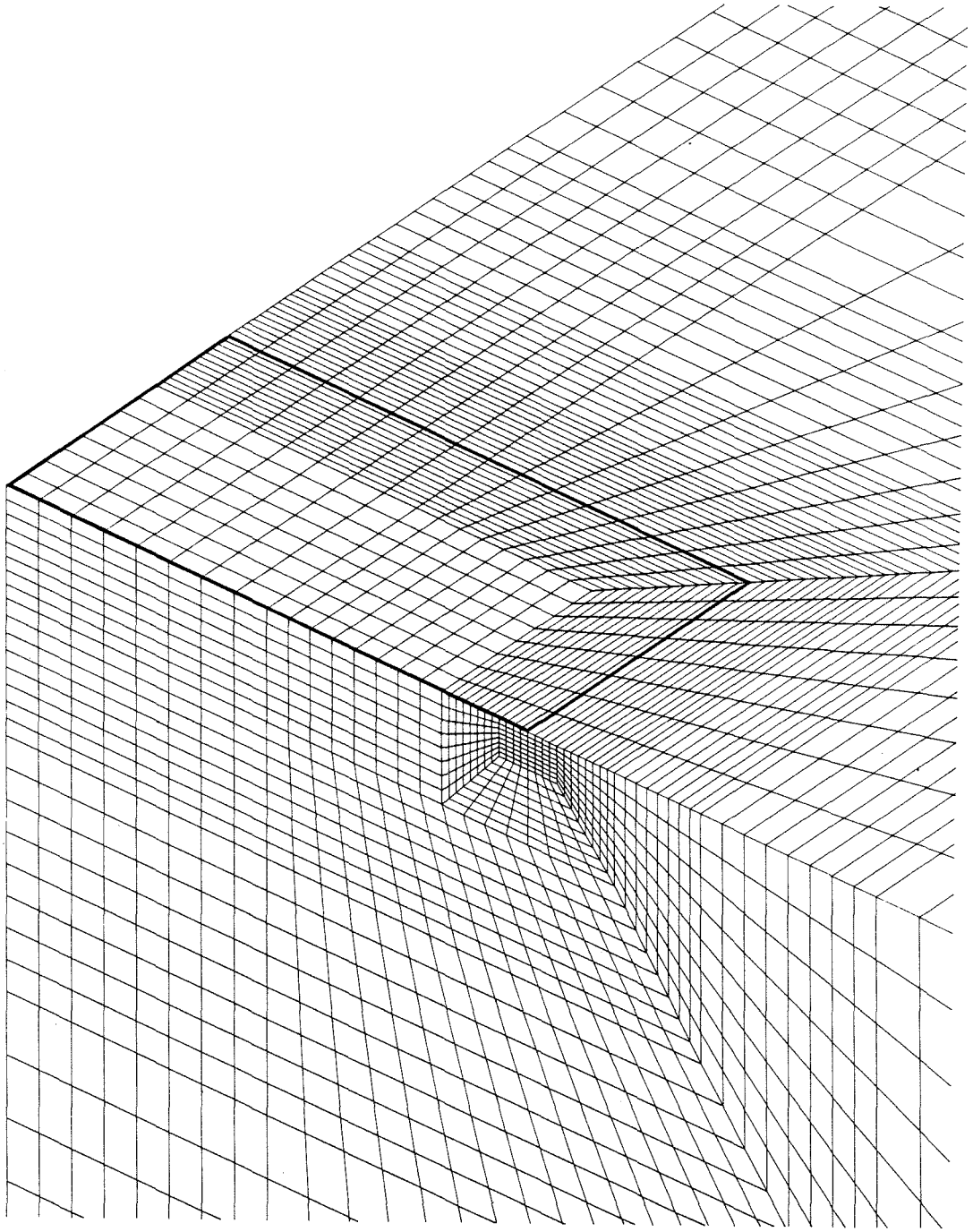


Figure 7.13: Detail of the Finest Mesh Used to Model the $\frac{L}{W} = 1$, $\frac{d}{W} = 0$ Strike-Slip Fault (the bold line indiates the edge of the fault); 90,891 degrees-of-freedom.

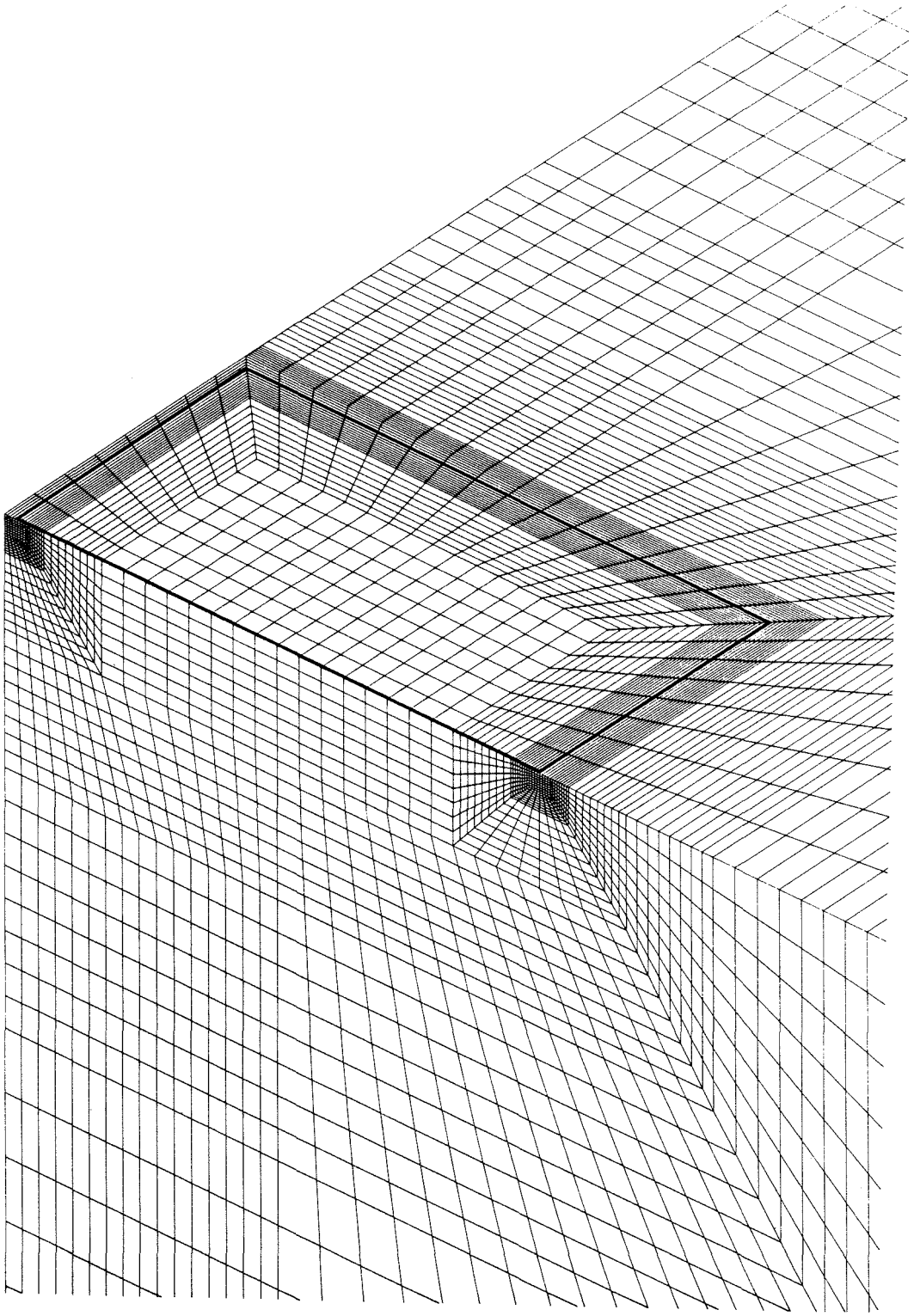


Figure 7.14: Detail of the Finest Mesh Used to Model the $\frac{L}{W} = 1$, $\frac{d}{W} = 0.02$ Strike-Slip Fault (the bold line indiates the edge of the fault); 121,827 degrees-of-freedom.

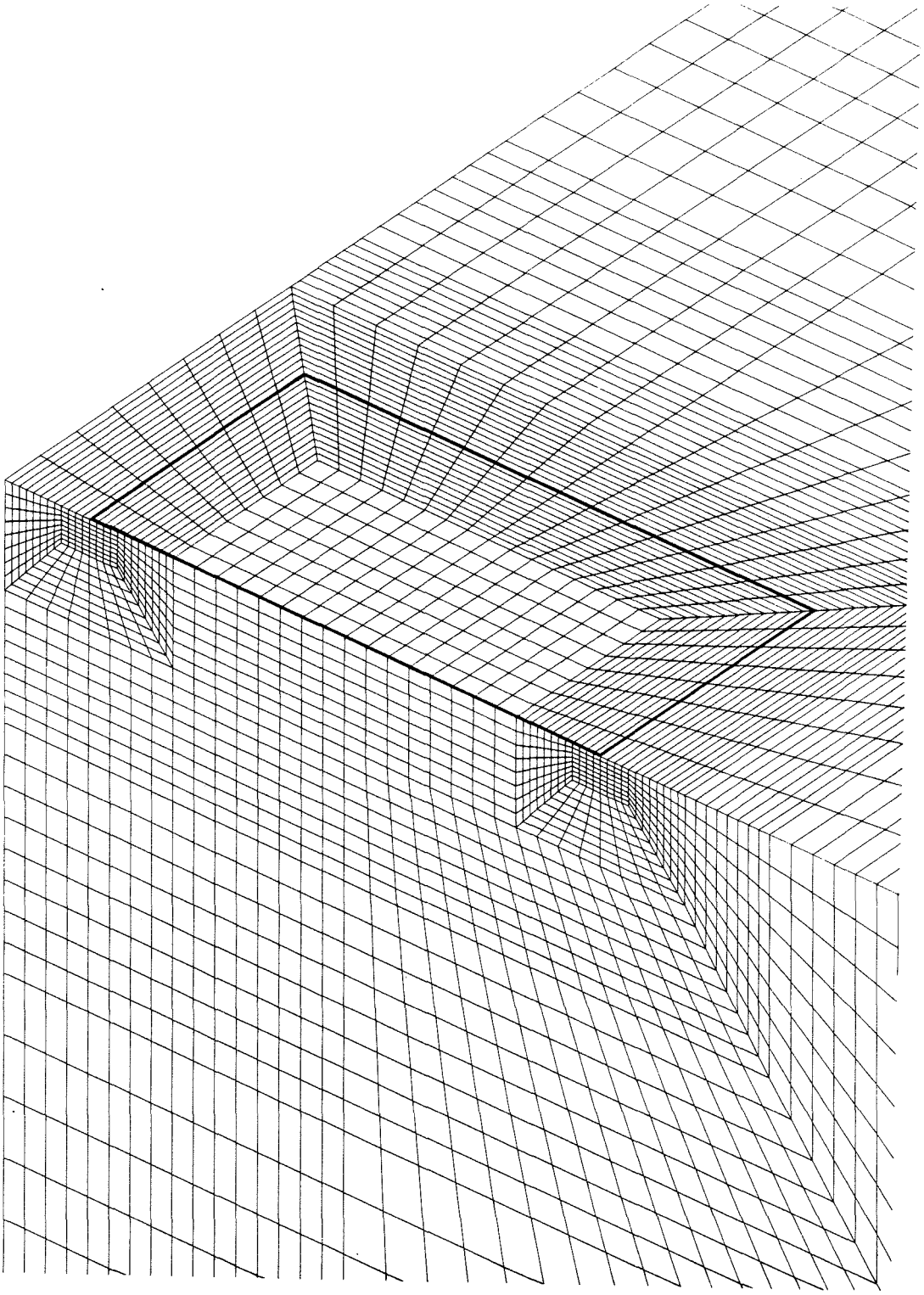


Figure 7.15: Detail of the Finest Mesh Used to Model the $\frac{L}{W} = 1$, $\frac{d}{W} = 0.165$ Strike-Slip Fault (the bold line indiates the edge of the fault); 105,099 degrees-of-freedom.

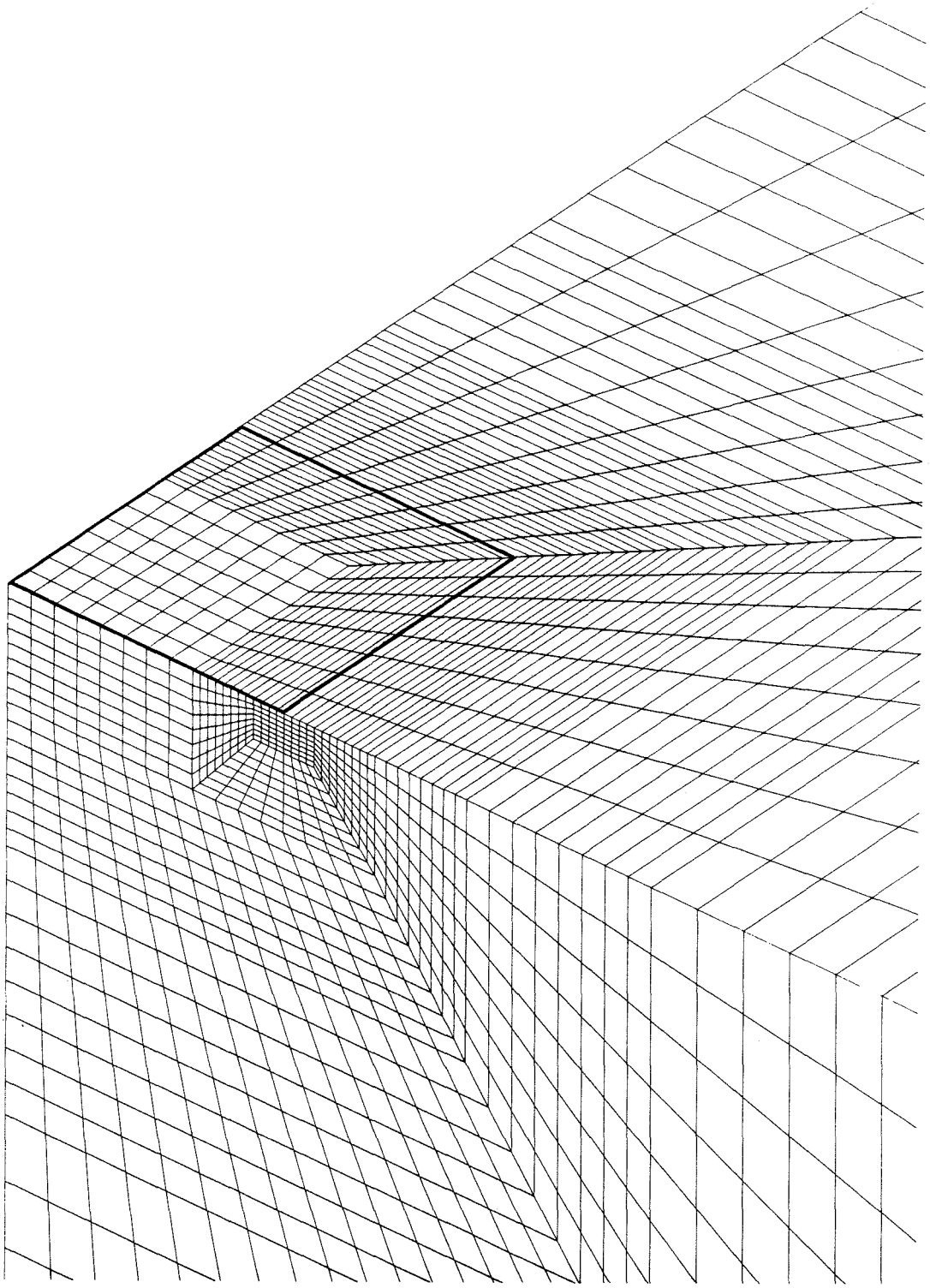


Figure 7.16: Detail of the Finest Mesh Used to Model the $\frac{L}{W} = 1$, $\frac{d}{W} \rightarrow \infty$ Strike-Slip Fault (the bold line indiates the edge of the fault); 58,419 degrees-of-freedom.

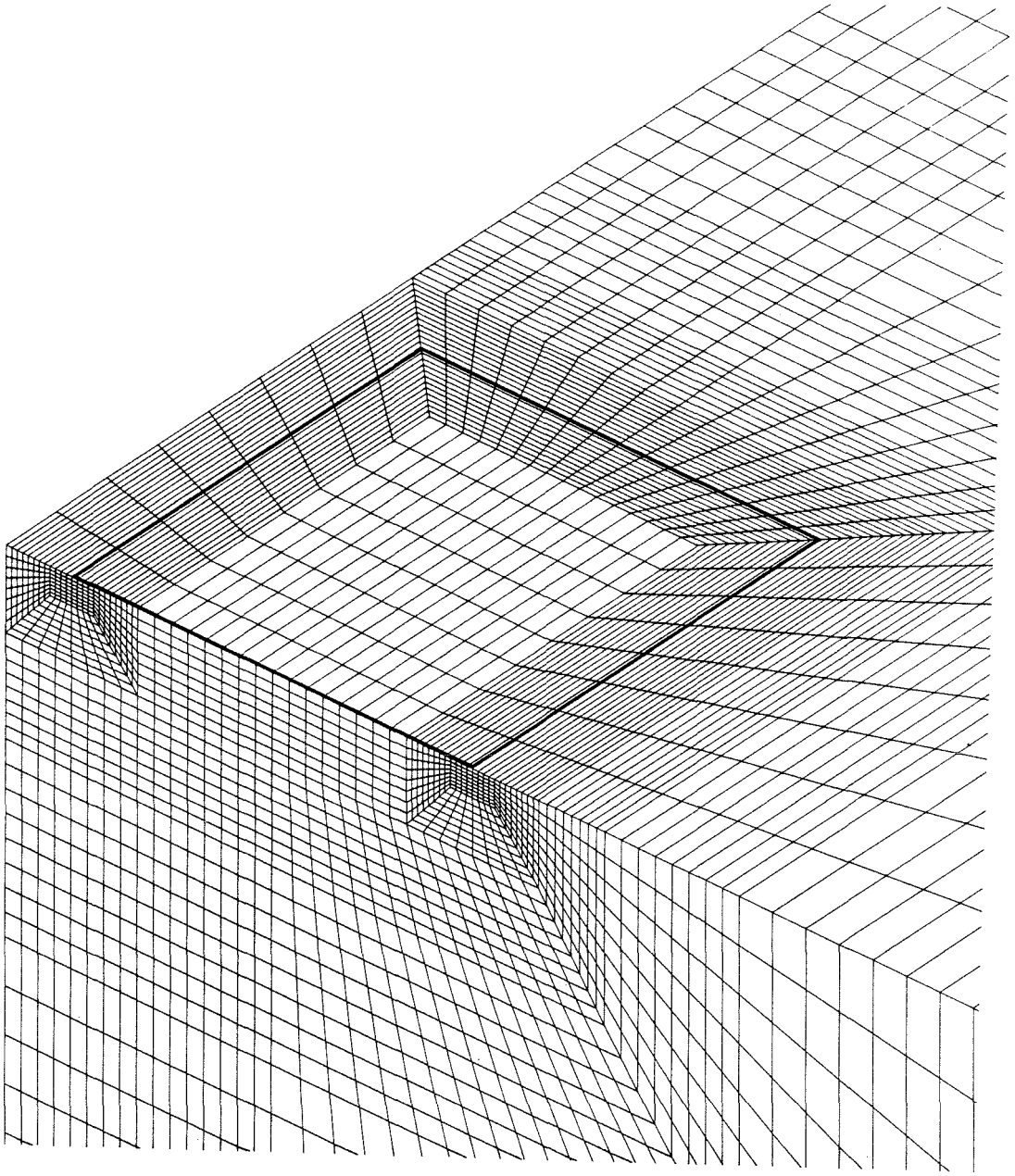


Figure 7.17: Detail of the Finest Mesh Used to Model the $\frac{L}{W} = 2$, $\frac{d}{W} = 0.165$ Strike-Slip Fault (the bold line indiates the edge of the fault); 105,099 degrees-of-freedom.

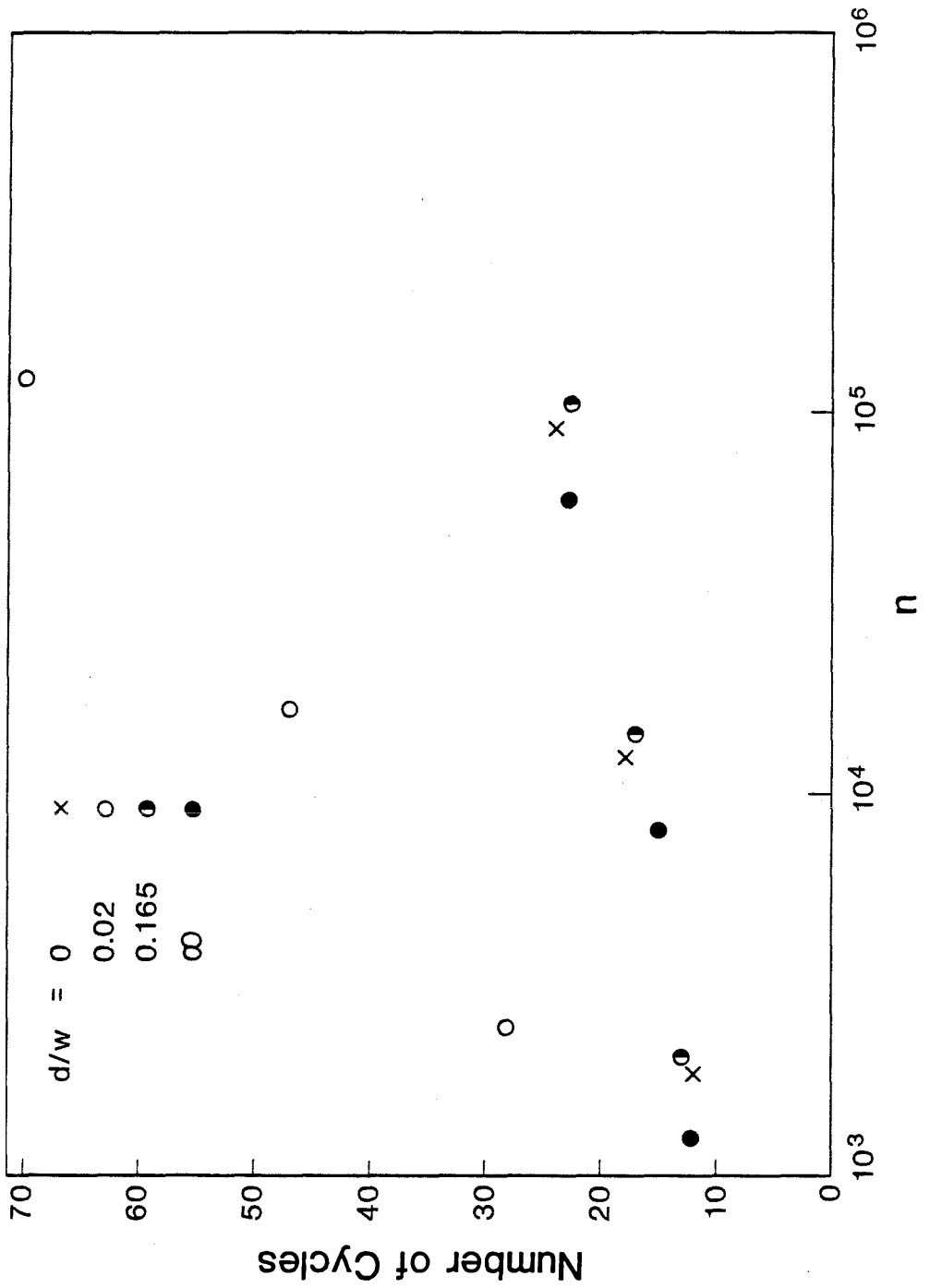


Figure 7.18: Number of MG_1-GS Cycles Versus $\log_{10} n$ for the $\frac{L}{W} = 1$ Strike-Slip Faults.

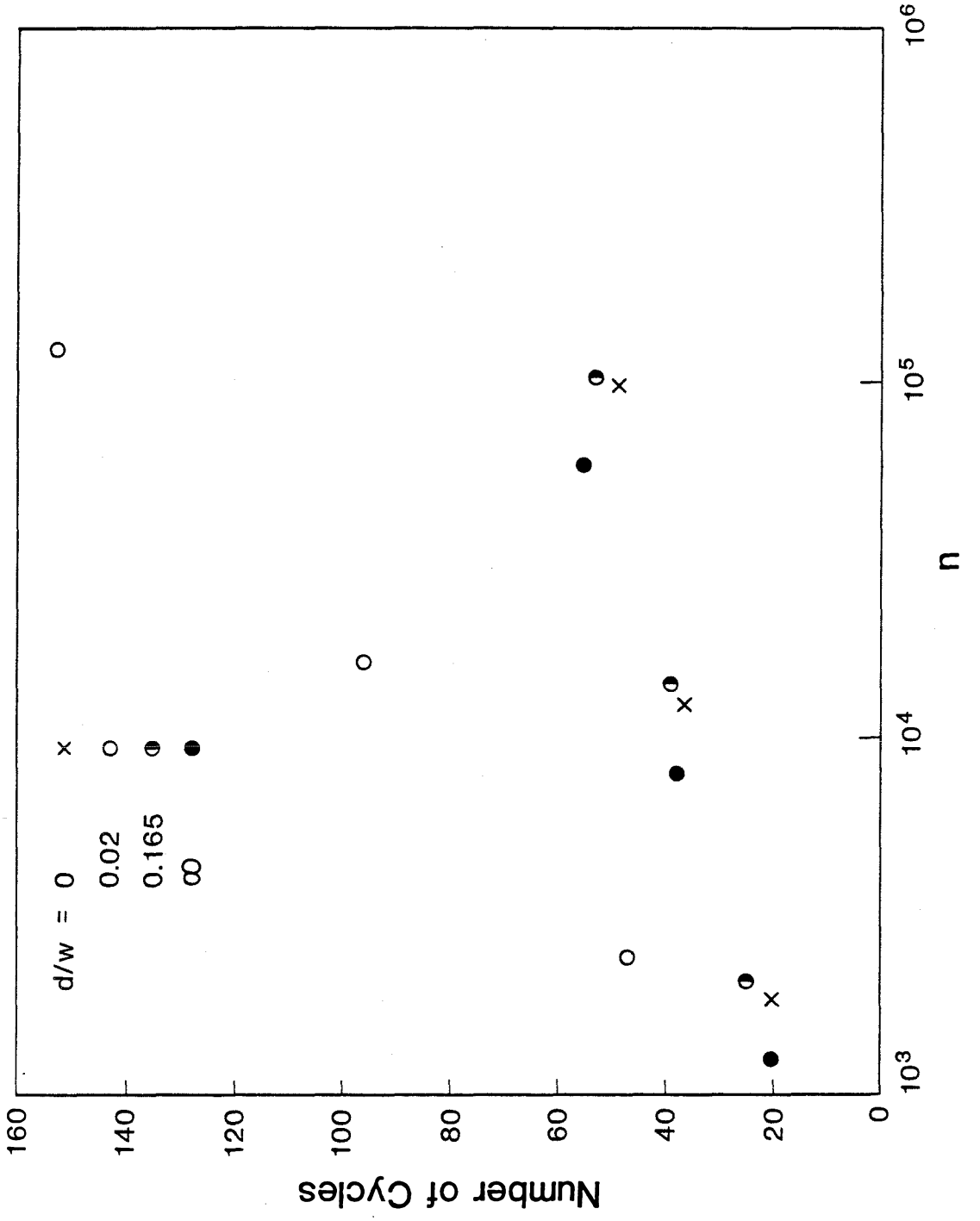


Figure 7.19: Number of MG_1-GS Cycles Versus $\log_{10} n$ for the $\frac{L}{W} = 2$ Strike-Slip Faults.

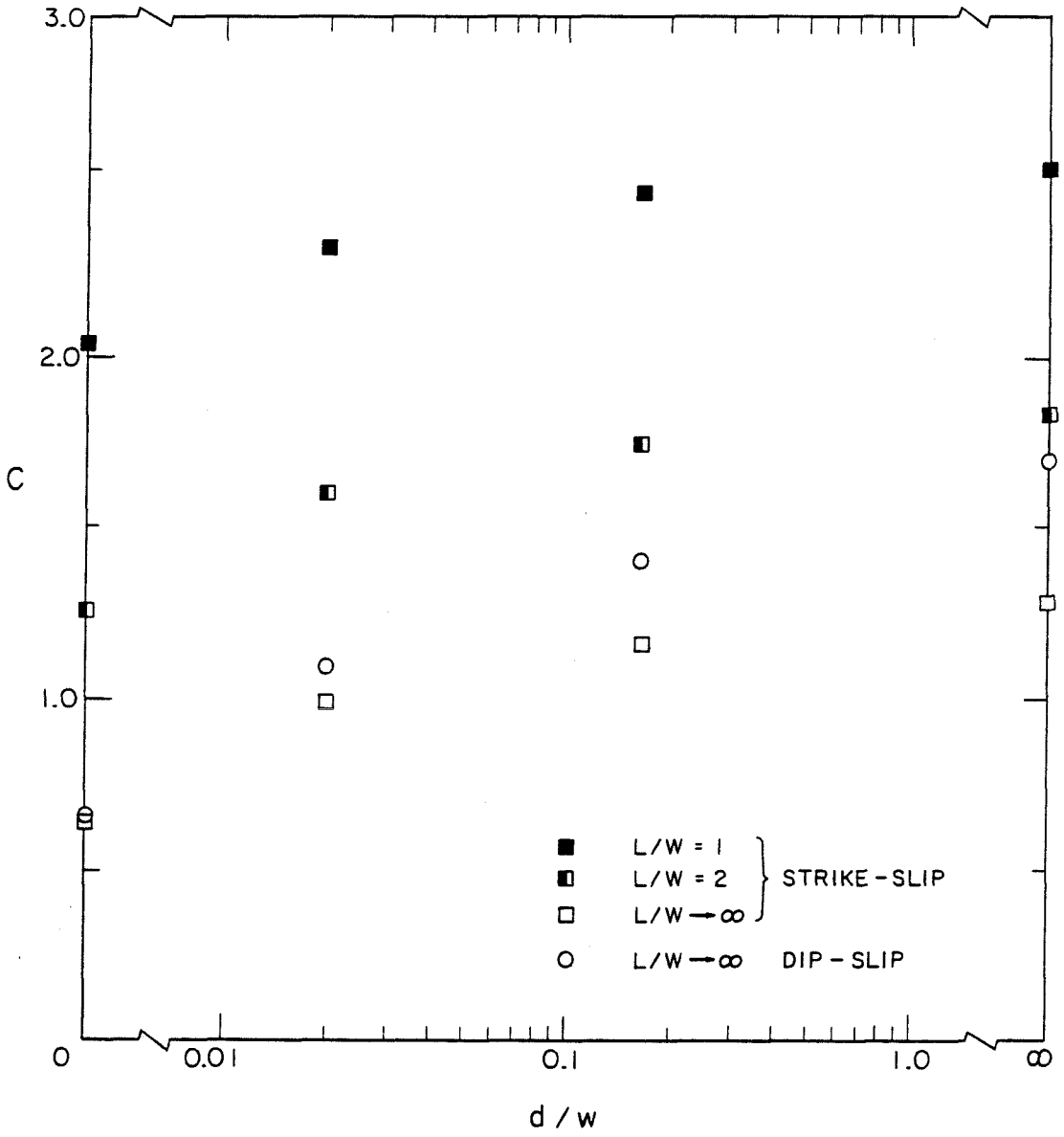


Figure 7.20: Values of C for Rectangular Faults in a Half-Space.

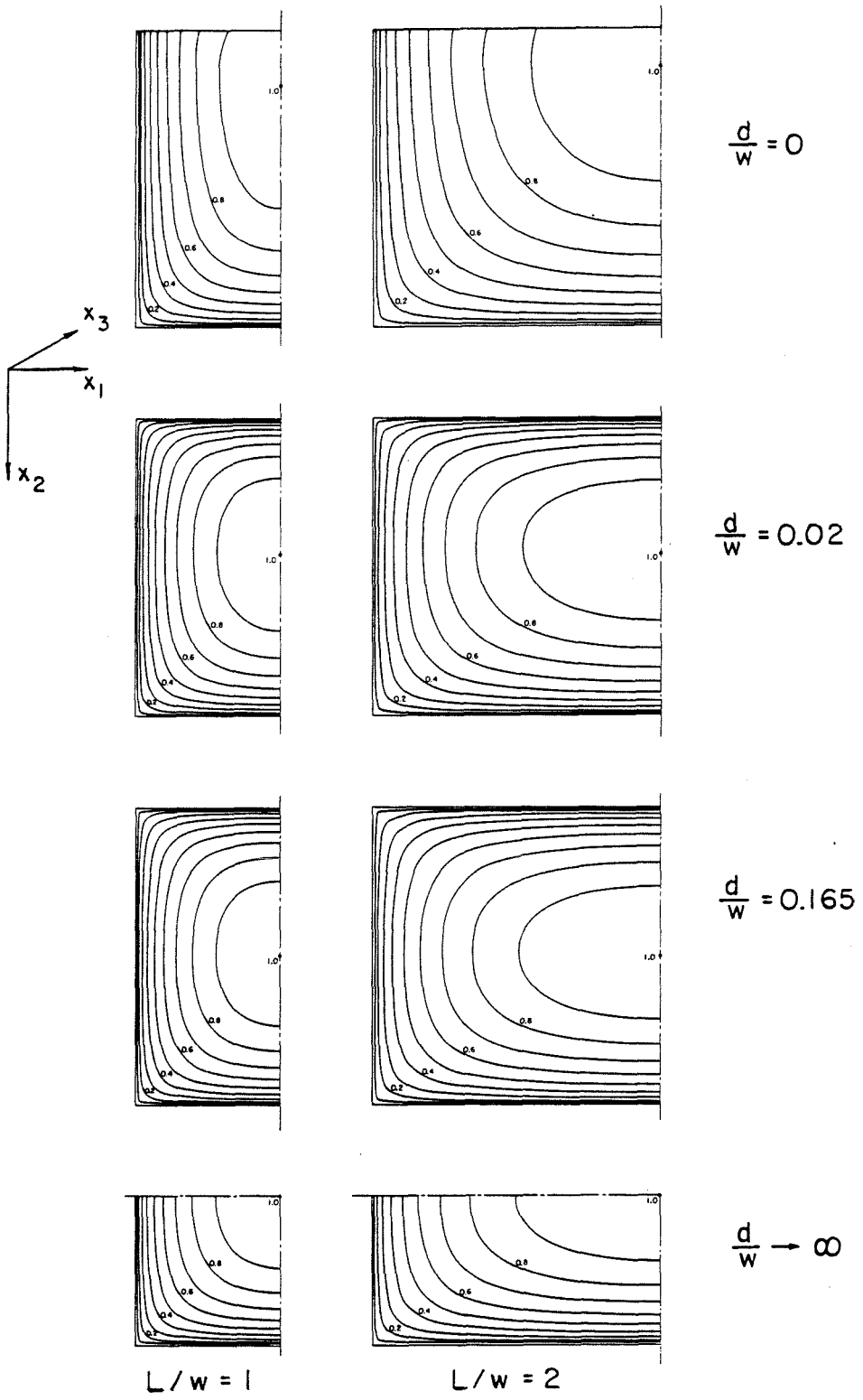


Figure 7.21: Distribution of Slip for the Rectangular Faults (the contours represent values of u_1 as fractions of the maximum u_1−displacement for each fault).

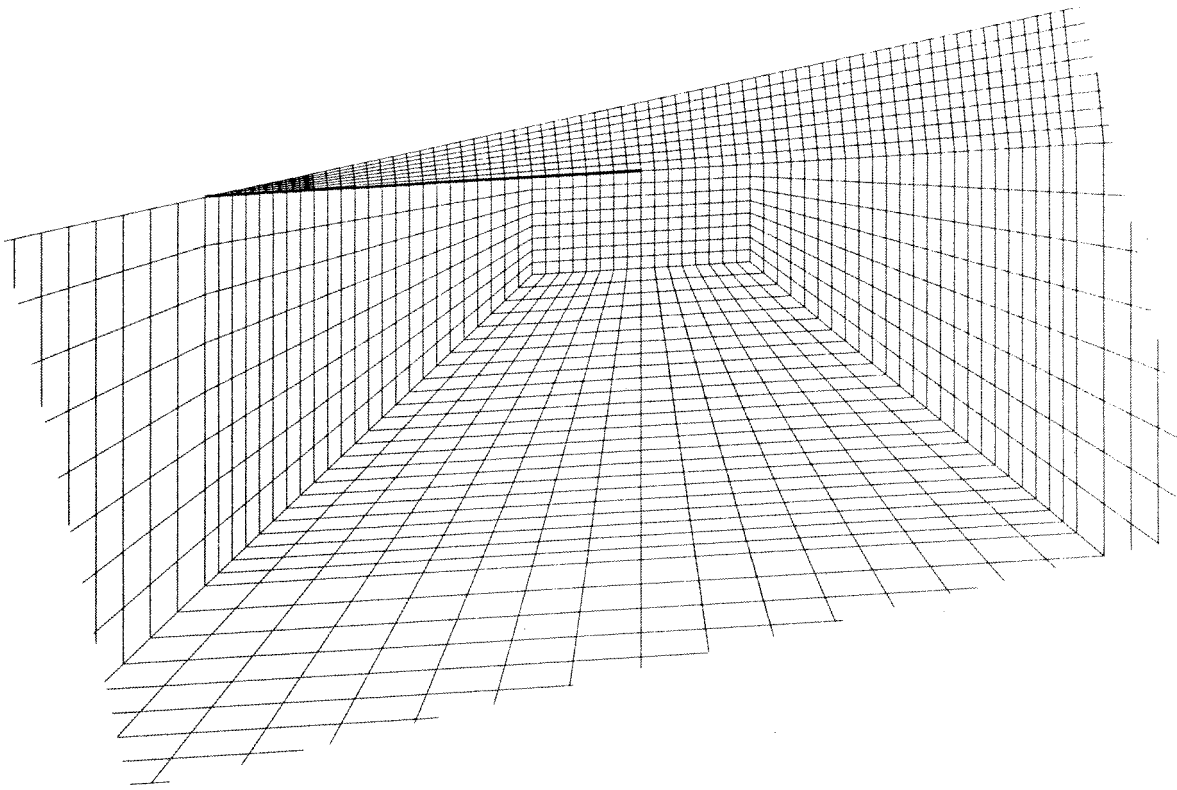


Figure 7.22: Detail of the Mesh Used to Model the Two Dimensional
line indiates the position of the fault).

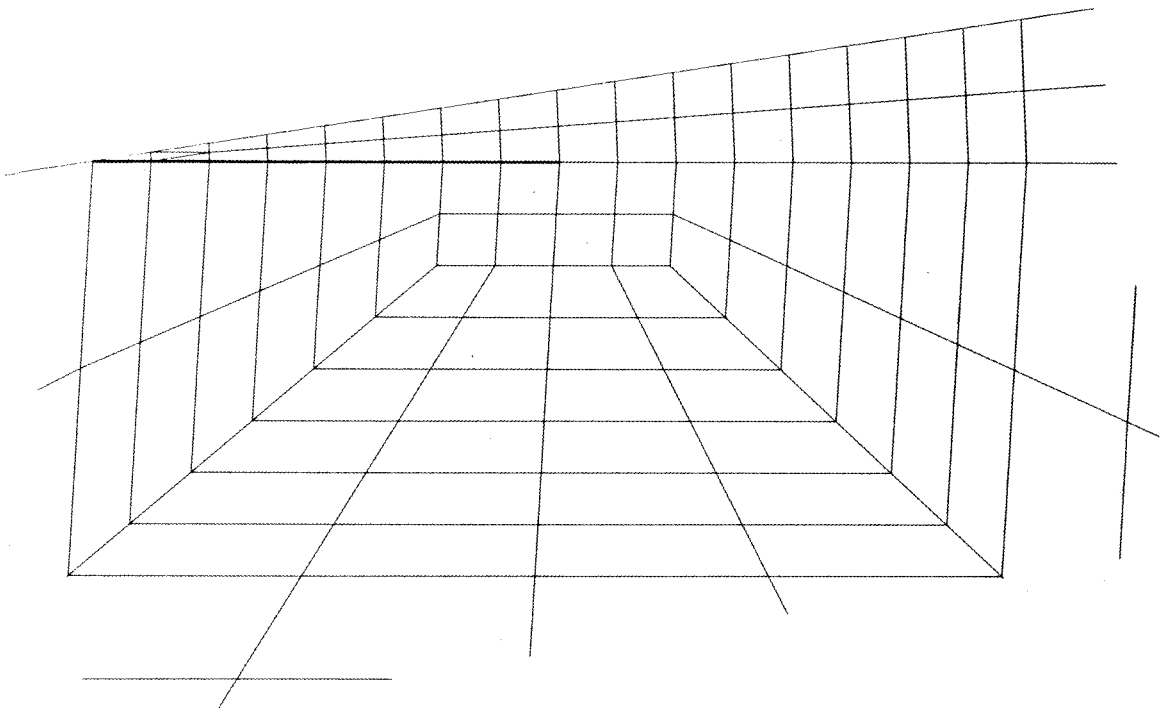


Figure 7.23: Detail of the Mesh Used to Model the Two Dimensional 10° Dip
Fault Using Linear Quadrilateral and Triangular Elements (the bold line ind
fault).

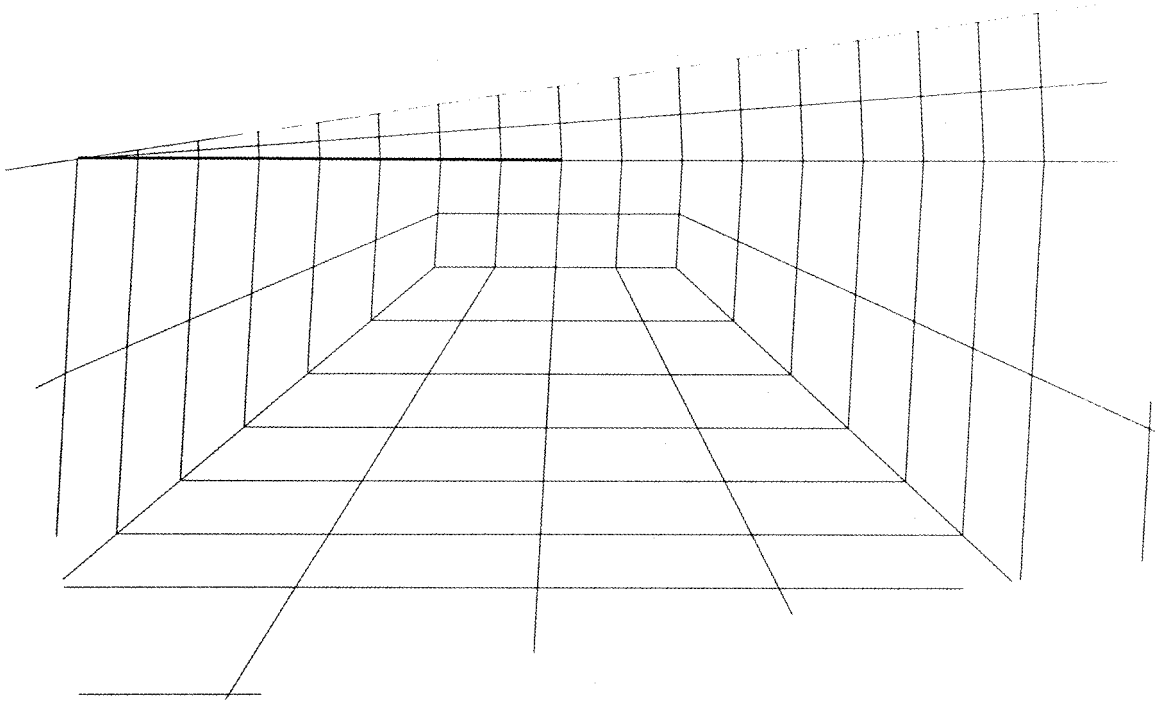


Figure 7.24: Detail of the Mesh Used to Model the Two Dimensional 10° Dip-S
Fault Using Linear Quadrilateral Elements (the bold line indiates the positio

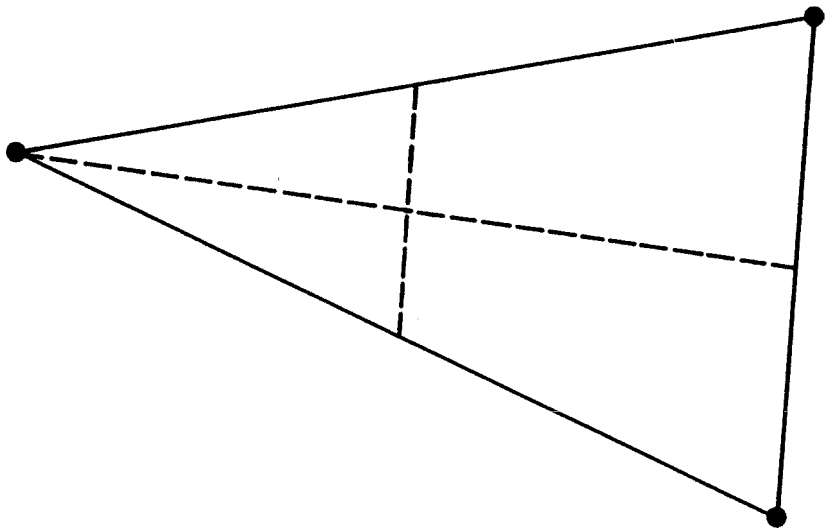


Figure 7.25: Formation of Fine Mesh Elements by Subdividing a Linear Quadrilateral Element.

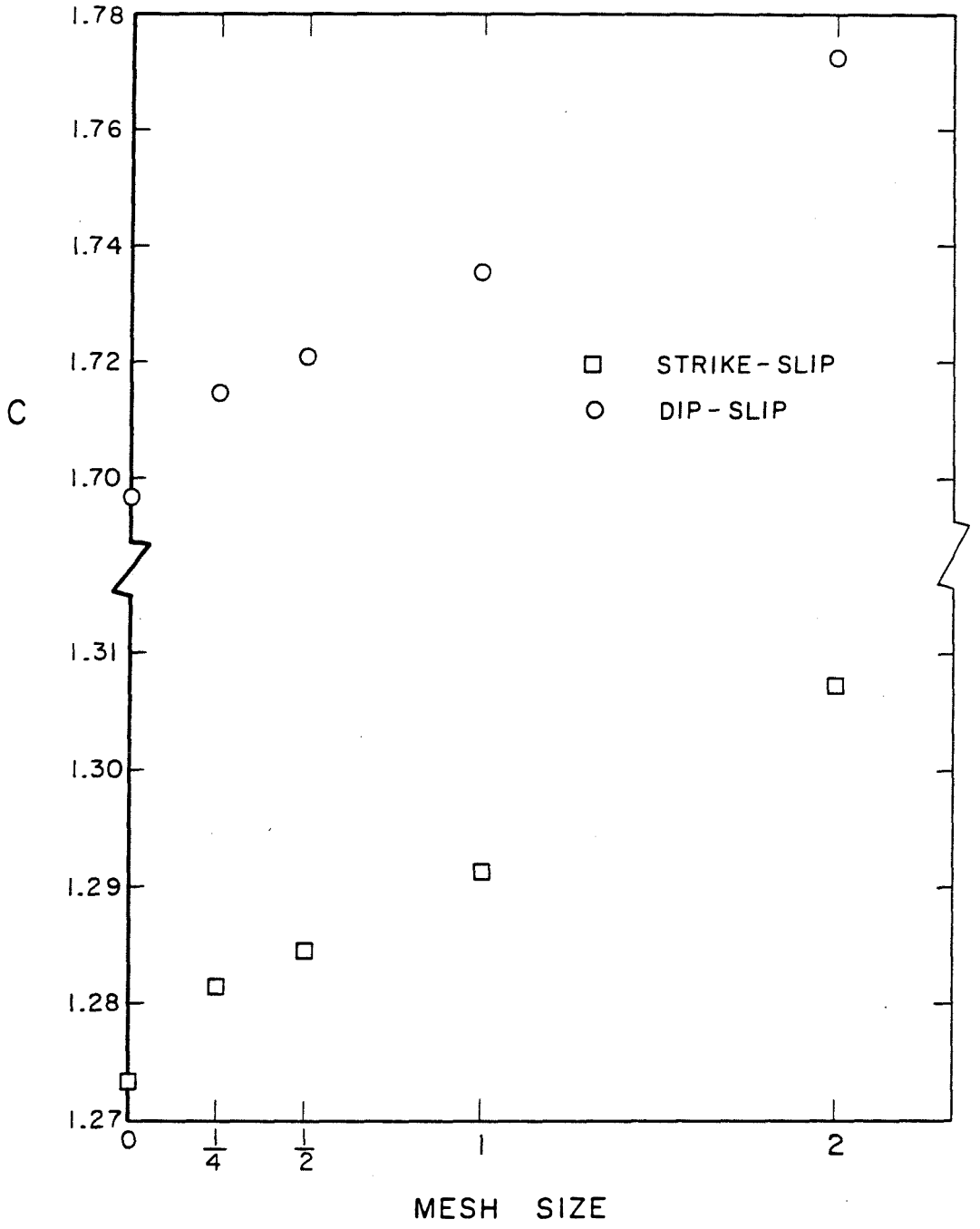


Figure 7.26: Computed and Analytical Values of C for the Two Dimensional Infinitely Buried Faults.

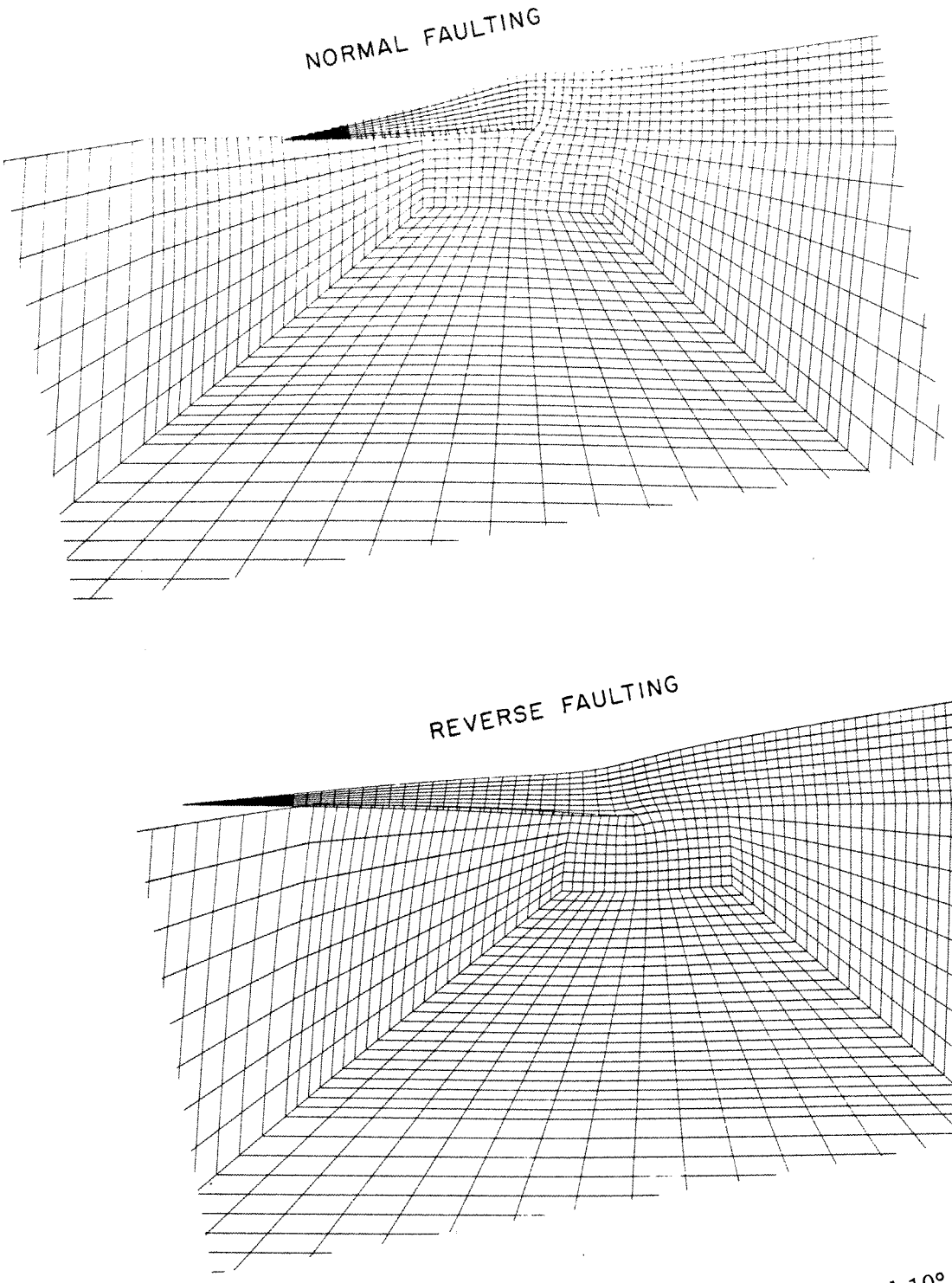


Figure 7.27: Deformed Meshes for the Two Dimensional 10° Dip-Slip

Part III

Closing Comments and Remarks

Chapter 8

Discussion of the Performance of the Multigrid Method

This chapter summarizes the performance of the multigrid method when applied to the solid mechanics problems discussed in chapters 5, 6, and 7. The work presented in this thesis is then put into perspective by briefly discussing similar work in the area of solid mechanics. This not only includes other applications of the multigrid method, but also the solution of some large three dimensional problems that have appeared in the literature. Finally, the multigrid algorithm is compared to the element-by-element (EBE) preconditioned conjugate gradient method [30,31,32,61], which may be considered to represent the state-of-the-art in iterative methods for solid mechanics problems.

8.1 Summary of the Multigrid Method's Performance

The simple two dimensional problems that were examined in chapter 5 demonstrate two important features of the multigrid method. First of all, the method was found to perform poorly when conditions of near incompressibilty are en-

countered. This is due to a combination of a reduction in the smoothing effect of the Gauss-Seidel relaxation and coarse mesh locking. The use of Jacobi preconditioned conjugate gradient iteration (JCG) to smooth the error, and the assembly of the coarse mesh stiffness matrix using a reduced integration scheme were found to partially relieve these problems. Secondly, the bending behavior of the linear brick and quadrilateral elements was found to slow the convergence of the multigrid algorithm. This was caused by the inability of the coarse mesh to accurately capture a low frequency error that consists of a bending deformation. The performance of the coarse mesh improved when the coarse mesh stiffness matrix was assembled using reduced integration on the $\mathbf{D}_\lambda^e$ and $\mathbf{D}_\mu^e$ terms of the element stiffness matrices, although this technique was less effective for nonuniform meshes.

The bending behavior of the coarse meshes was also found to be responsible for producing solution times that were not proportional to the problem size, n, when nonuniform meshes were used (see section 5.2). However, the linear dependence on n was recovered as the problem size increased.

The results discussed in section 5.3 show that the computational effort (in terms of CPU time and storage) required by the multigrid method is substantially less than that of a direct method. Also, the performance of the multigrid method is comparable to that of the JCG method for the well-conditioned three dimensional test problem that was considered.

The solution of the problems of engineering interest discussed in chapters 6 and 7 highlight the power and the usefulness of the multigrid method. In particular, the solution of the preliminary crack (section 6.2.1) and the circular fault (section 7.2) problems demonstrate that the multigrid method is faster than JCG

iteration. The performance of the two methods when applied to these problems is summarized in table 8.1. Another feature of the multigrid method that was demonstrated in these chapters is its sensitivity to the choice of the parameters γ, ν_1, and ν_2. For example, it was found that $\gamma = 3$ gave the fastest multigrid schemes for both of these problems; this can be attributed to the bending behavior of the coarse meshes (higher values of γ mean that more multigrid cycles are used on the coarse mesh to solve the coarse mesh correction equation (4.2)). However, the circular fault problem was less sensitive to the choice of γ than the crack problem, since substantially more bending deformation was present in the latter case.

The two dimensional faults in a half-space examined in section 7.4 enabled a comparison to be made between the ability of the multigrid method to solve Laplace's equation (anti-plane shear conditions) and the equations of plane strain (see appendix A). It was found that the method performed better when applied to Laplace's equation, requiring up to 36% fewer cycles of iteration for problems with identical meshes ($\frac{d}{W} = 0.02$ two dimensional faults; see tables 7.5 and 7.6).

The most important theoretical property of the multigrid method is that it enables problems to be solved in a computational effort that is only linearly proportional to the size of the problem. This means that the storage requirements grow linearly with the problem size (since only the nonzero terms of the matrices are stored, see section 4.2.7), and that the number of multigrid cycles required for the solution of a given problem is independent of the number of unknowns (since the number of operations required by each cycle is proportional to the problem size). This property was shown to be true for the simple problems considered in

chapter 5 with uniform meshes, and with nonuniform meshes if the discretization is fine enough.

However, when the multigrid method was used to solve the problems discussed in chapters 6 and 7, it was found that the number of cycles did depend on the problem size. In particular, section 7.3.2 shows that the number of cycles required for the solution of the rectangular fault problems was proportional to $\log_{10} n$. It should be noted that the number of cycles required by some of the two dimensional faults discussed in section 7.4 was independent of n.

Figures 8.1, 8.2, and 8.3 show log-log plots of the multigrid solution time, the total computation time (the solution time plus the time required to compute all of the necessary interpolation and stiffness matrices), and the storage requirement versus the problem size for seven of the problems discussed in chapter 5, 6, and 7. Figure 8.1 shows that the times required for the solution of the three dimensional test problem and the two dimensional fault problems were proportional to n, but that this was not true for the other problems. However, these solution times were only proportional to n to the power of about 1.3 or less. The discussion in section 5.4 suggests that the linear dependence on n may be recovered if the discretization is increased (i.e. as n becomes large). When the total time is considered (see figure 8.2), the speed of the multigrid method is seen to be proportional to n to the power of about 1.2 or less. This reduction in the strength of the dependency on n is due to the fact that the time required to calculate the necessary matrices is itself proportional to n (see section 4.2.7). It is apparent from figures 8.1 and 8.2 that the constant of proportionality in the relationships between the problem size and the solution and total computation

times is problem dependent. This once again demonstrates the influence of the bending behavior of the coarse meshes on the number of multigrid cycles required. The storage required by the multigrid algorithm shown in figure 8.3 is seen to be proportional to n. In this case, the constant of proportionality depends only on the dimension of the problem; more storage is required by the three dimensional problems because there are more nonzero terms in each row of the stiffness and interpolation matrices, and because more data is needed by the brick element to describe its connectivity.

8.2 The Multigrid Method in Solid Mechanics

This section attempts to place the work described in this thesis in context by discussing similar work in solid mechanics. It was mentioned in section 4.1 that hardly any work has been done in the past concerning the application of the multigrid method to solid and structural mechanics. The relatively small problems discussed in [5] (up to 2,000 degress-of-freedom) demonstrate that the method can be used to quickly solve two dimensional linear elastic problems, although no attempt was made to examine and explain its weaknesses. However, one observation that is common to this thesis and [5] is that the multigrid algorithm performs better when applied to the solution of Laplace's equation compared to the governing equations of two dimensional elasticity. The work of Brand reported in [11] confirms the observation made in this thesis that the bending behavior of the coarse meshes does not enable the coarse mesh correction to be accurately captured. However, Brand improves the performance of the multigrid method by using a preconditioned conjugate gradient method to smooth the error (which

was found to be ineffective for the two dimensional crack problem discussed in section 5.1.3), whereas this thesis describes the use of a coarse mesh stiffness matrix assembled using reduced integration (section 5.1.3) and increased values of γ (sections 6.1.2 and 7.2.2) to attempt to solve this problem. To the author's knowledge, no previously published work has attempted to use the multigrid method to solve nearly incompressible elasticity problems (section 5.1.2), or large linear elastic problems of practical interest (chapters 6 and 7).

There are several examples of the solution of large solid mechanics problems in the literature. In [32], an element-by-element preconditioned conjugate gradient method is used on a Cray X-MP/48 supercomputer to solve problems that have up to 104,504 degrees-of-freedom. This work is compared to the multigrid method in the next section. The group incomplete relaxation scheme was used in [18] to perform an elastic analysis of a dam. The largest problem that was solved consisted of 34,908 degrees-of-freedom. An interesting feature of this work is the use of the solution to the problem on a coarser mesh as the initial approximation to improve the convergence of the iteration method. Another example of the solution of a large solid mechanics problem is reported in [41]; an analysis of a complicated composite led to the solution of a problem that had 100,000 degrees-of-freedom on a Cyber 205 computer. This very brief discussion of other numerical solutions to solid mechanics problems demonstrates that the problems considered in this thesis can certainly be considered large by today's standards, and that their solution on a VAX 11/750 is a considerable achievement.

8.3 A Comparison Between the Multigrid and the Element-by-Element Preconditioned Conjugate Gradient Methods

In this section, as a final attempt to demonstrate the importance of the multigrid method, its performance is compared to that of the element-by-element (EBE) preconditioned conjugate gradient method [30,31,32,61]. This procedure has been found to be an effective preconditioner for the conjugate gradient method when applied to solid mechanics problems. In [32], the method is used to solve several problems using a Cray X-MP/48 computer. In particular, a three dimensional point load problem is solved using a series of uniform cubic meshes, which is similar to the three dimensional test problem discussed in section 5.3. Furthermore, this point load problem is also solved using the JCG method. The data on the performance of these two iterative algorithms given in [32] enables the EBE and the multigrid methods to be compared. This can be done by dividing the storage and CPU time requirements of the two methods by the corresponding data for the JCG iteration. This normalization procedure should help to account for the differences in performance between the VAX 11/750 and the Cray X-MP/48, and thereby give a reasonable comparison between the EBE and multigrid methods.

Figure 8.4 shows the storage required by the EBE and multigrid methods normalized by the requirement of the JCG iteration. Figure 8.5 shows a similar plot of the time required by the two methods. The data shown in figure 8.5 for the EBE method represents the solution time (i.e. the time required to solve $\mathbf{Kx} = \mathbf{f}$), whereas the multigrid data shows the total computation time (i.e. the

solution time plus the time to assemble all of the necessary matrices). This was done because the computation of the coarse mesh data is an important feature of the multigrid method. No data is available regarding the total time required by the EBE method.

Figures 8.4 and 8.5 show that the performance of the multigrid method is comparable to that of the EBE scheme. The multigrid storage requirement is less than for the EBE method, although both methods require more storage than the JCG iteration. Although the multigrid method generally requires more CPU time (see figure 8.5), it is interesting to note that it appears to gain an advantage over the EBE method as the problem size is increased.

It should be emphasized that this discussion is not designed to determine whether the multigrid method is better than the EBE method. It does, however, demonstrate that the performance of both of these schemes is comparable when applied to two similar well-conditioned problems. Both methods have been successfully applied to practical problems in solid mechanics (see chapters 6 and 7 of this thesis, and [32]); more such applications are needed before any definitive comparisons can be made. An important application area which requires further work is that of structural mechanics. It should be noted that the EBE iteration has been implemented to solve nonlinear solid mechanics problems, and so at present can be considered to be a more mature solution technique.

Problem	n	Method	Solution Time (CPU secs)	Total Time (CPU secs)	Storage (bytes)
Crack	25,515	MG_1-GS	1.45×10^4	1.93×10^4	2.43×10^7
		JCG	3.37×10^4	3.81×10^4	1.36×10^7
Fault	58,419	MG_1-GS	2.46×10^4	3.61×10^4	5.77×10^7
		JCG	4.00×10^4	5.06×10^4	3.25×10^7

Table 8.1: Computational Effort Required to Solve the Preliminary Crack (section 6.2.1) and the Circular Fault (section 7.2) Problems Using the MG_1-GS and the JCG Methods.

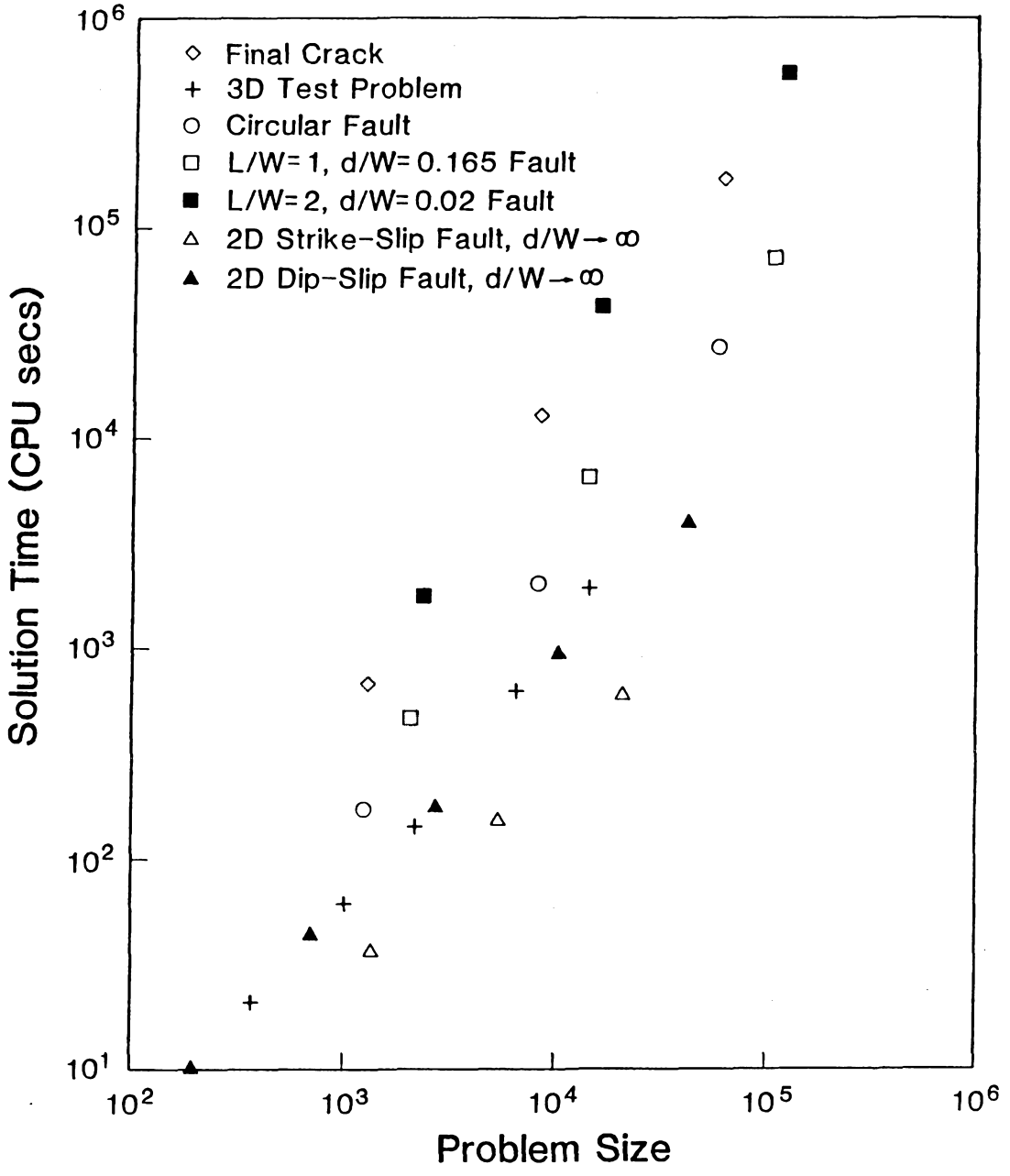


Figure 8.1: Solution Times for the Multigrid Method When Applied to Different Problems (each symbol represents refinements of a particular mesh).

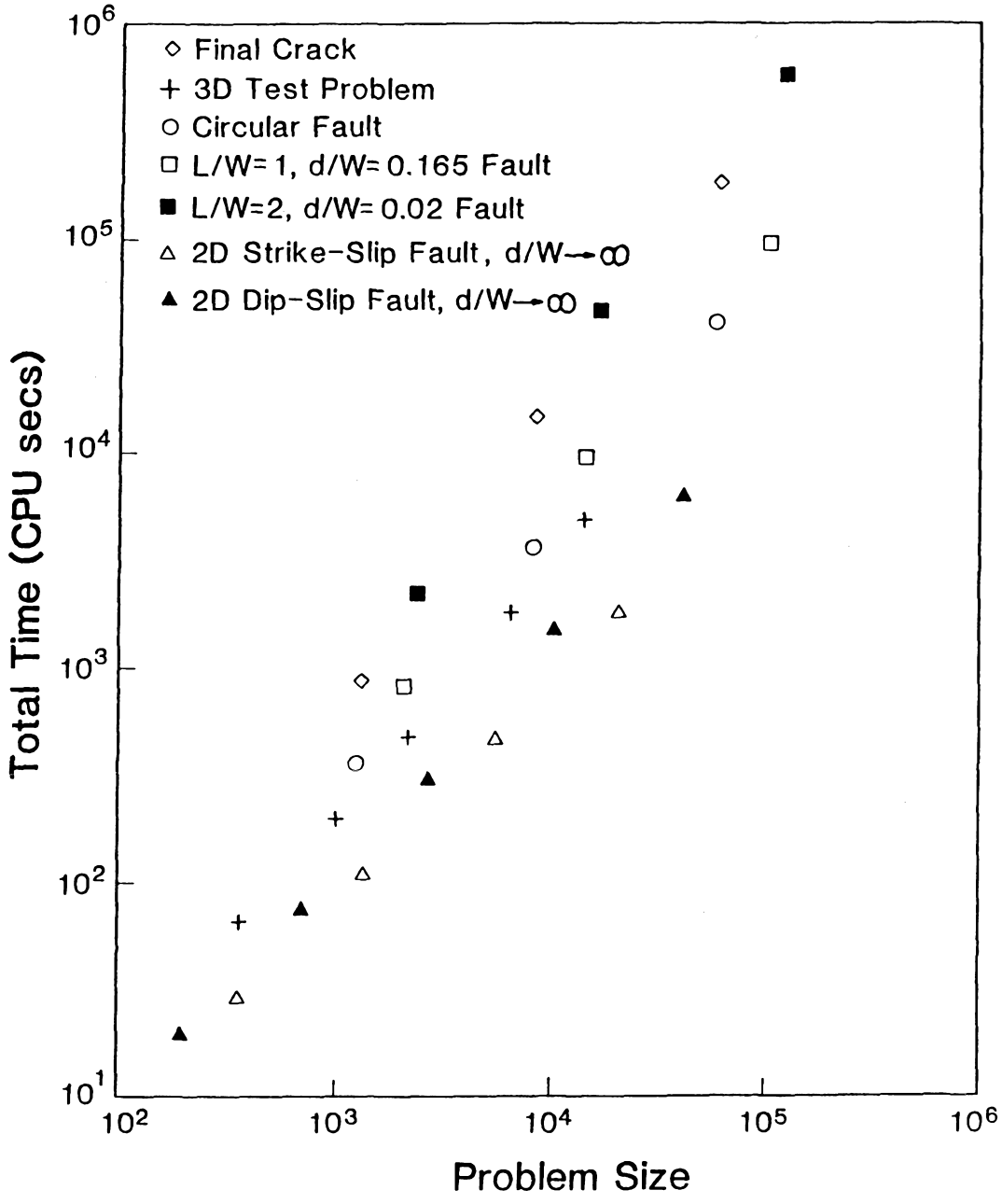


Figure 8.2: Total Times for the Multigrid Method When Applied to Different Problems (each symbol represents refinements of a particular mesh).

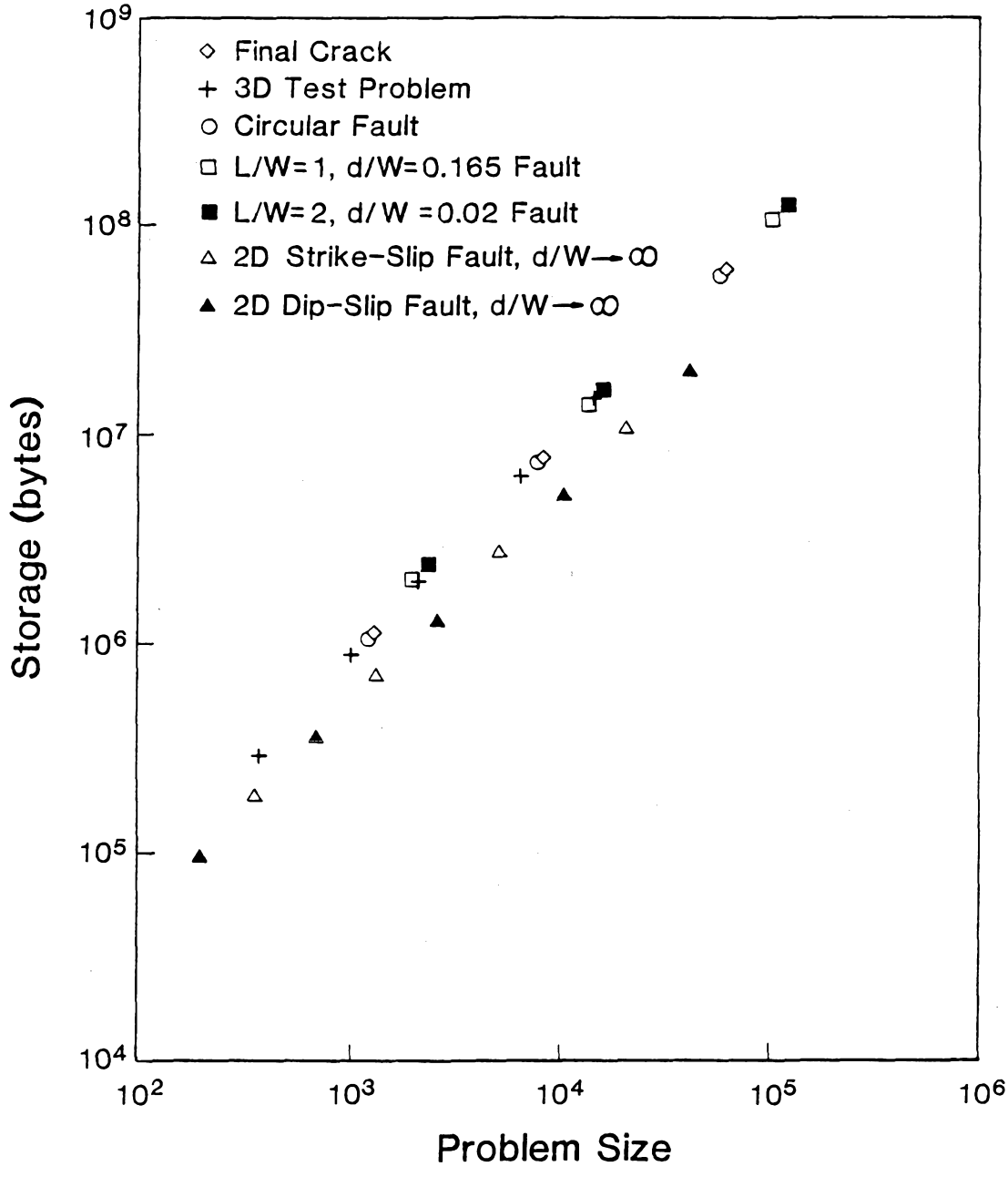


Figure 8.3: Storage Requirements of the Multigrid Method When Applied to Different Problems (each symbol represents refinements of a particular mesh).

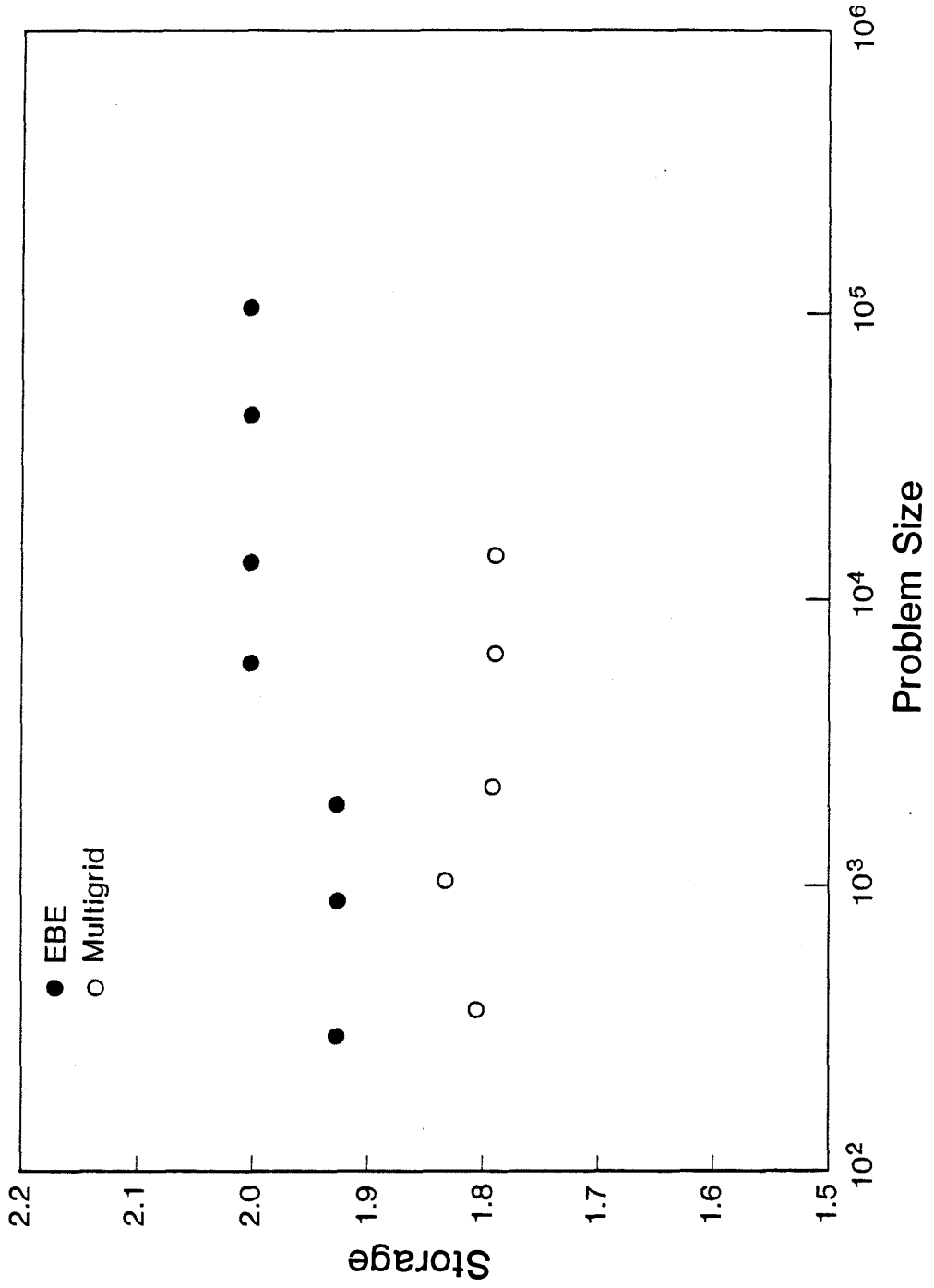


Figure 8.4: A Comparison Between the Storage Requirements of the Multigrid Method and the EBE Method (the data is normalized with respect to the JCG method).

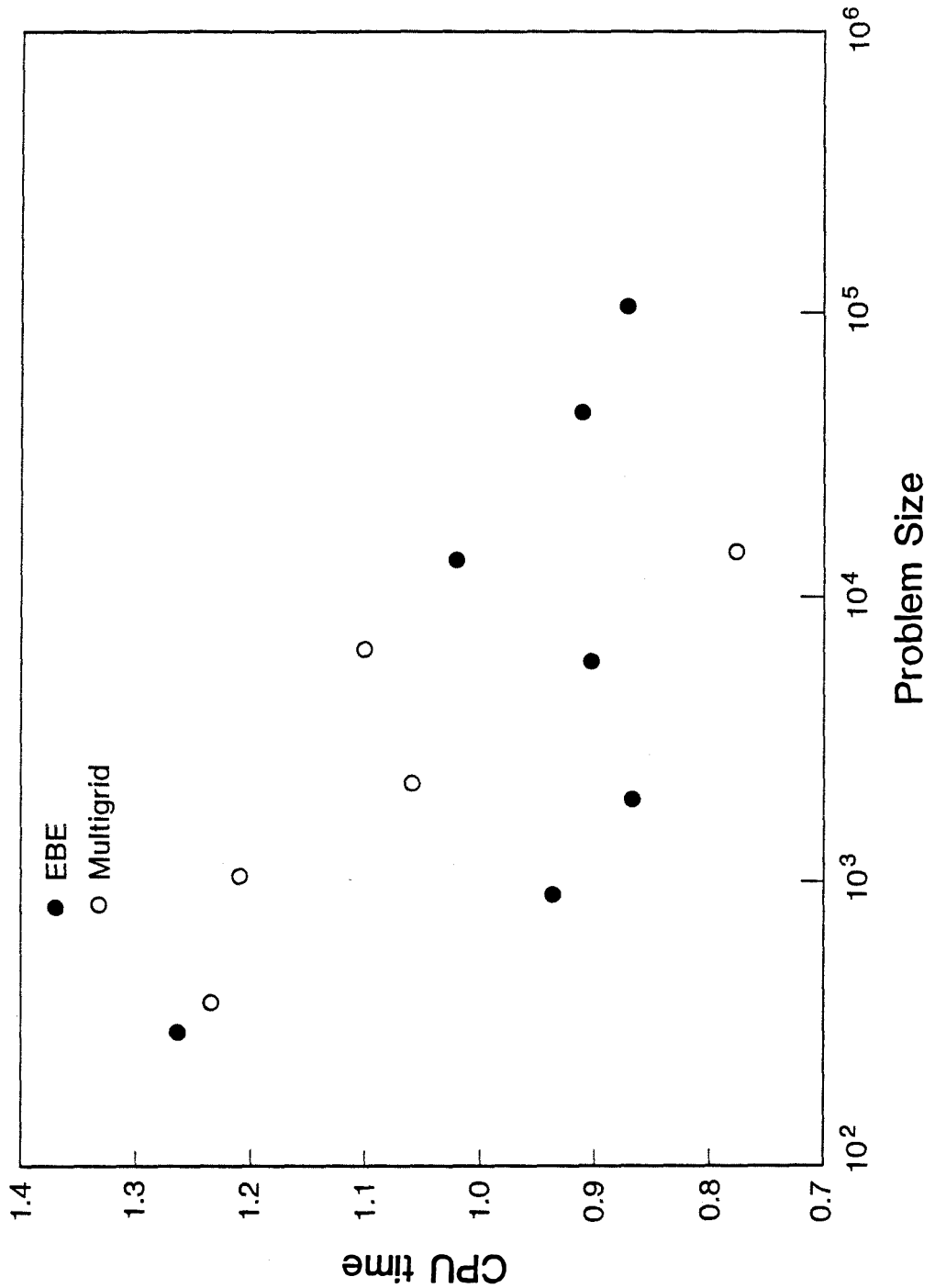


Figure 8.5: A Comparison Between the CPU Time of the Multigrid Method (total time) and the EBE Method (solution time) (the data is normalized with respect to the JCG method).

Chapter 9

Suggestions for Future Work

The objective of this chapter is to briefly outline some of the author's ideas for future research regarding the multigrid method. The work described in this thesis has concentrated on linear elasticity, which is the simplest class of solid mechanics problems. There is considerable potential for the development of multigrid algorithms to solve nonlinear problems in solid mechanics (e.g. plasticity, nonlinear elasticity, and contact problems). Although nonlinear multigrid algorithms have been developed for use in the field of fluid mechanics, the nonlinearities encountered are geometric rather than constitutive (which are more common in solid mechanics). It is therefore possible that considerable algorithm development will be necessary before the multigrid method can be used to solve nonlinear solid mechanics problems.

Another area of interest is that of structural mechanics, in which problems involving trusses, beams, plates, and shells are solved. Some work in this area has been reported in [24], in which the eigenvalues and eigenvectors of a rectangular plate were obtained using the multigrid method. The inherent ill-conditioning of problems involving beams and shells may present difficulties for the multigrid

method. One of the more common classes of problems encountered in structural mechanics is that of eigenvalue analysis. The algorithms developed in [27] for this purpose may have a wide range of application, not only for linear structural dynamics, but also for linearized buckling analysis [6]. The use of the constraints described in section 4.2.2 may be particularly useful in the area of structural mechanics, since it may be difficult to explicitly define the coarse mesh elements.

One area that has received considerable attention in fluid mechanics is the integration of adaptive mesh refinement techniques with the multigrid method [4,12,13]. In the field of solid mechanics, automatic mesh refinement techniques have been developed to produce a near optimal mesh on which to solve a given problem [36]. A natural extension of this research would be to use the sequence of meshes generated by the refinement algorithm as the coarse meshes for the multigrid method. The engineering analyst would only have to specify the boundaries of the problem that was to be solved; the solution would then be computed on an optimal mesh using the multigrid method.

One of the current multigrid research topics that may have applications in the area of solid and structural mechanics is the algebraic multigrid method [46]. This technique attempts to construct coarse meshes by examining the equations associated with the finest mesh. The feature of this method is that no coarse meshes are physically constructed. Therefore it could be useful for structural mechanics problems, since, as was noted above, the coarse meshes may not be easy to define.

Finally, the author believes that the multigrid method has great potential as a production code for the solution of solid mechanics problems on state-of-

the-art supercomputers (e.g. the Cray machines) and concurrent computers (e.g. hypercubes). The computations performed on a VAX 11/750 discussed in this thesis have demonstrated the power of the multigrid method; even larger problems could be solved if the method was efficiently implemented on a supercomputer such as a Cray. The multigrid algorithm described in chapter 4 has successfully been programed on the Caltech hypercube [15,16], although the memory capacity of this prototype machine limited the size of problem that could be solved.

Chapter 10

Conclusions

- The multigrid method works extremely well for linear solid mechanics problems. Its performance is superior to a direct method, is better than the Jacobi preconditioned conjugate gradient method when applied to practical problems, and is comparable to the element-by-element preconditioned conjugate gradient method when applied to a well-conditioned test problem.

- The storage requirements of the multigrid method are linearly proportional to the problem size. This observation is true for all iterative methods, since only the nonzero terms of the stiffness matrix are required. The constant of proportionality depends only on the dimension of the problem.

- The solution times of the multigrid method were found to be linearly proportional to the problem size if uniform meshes were used. This was not true for most of the problems that were solved with nonuniform meshes. However, it was found that the solution times for some simple two dimensional problems with nonuniform meshes did become proportional to the problem size as the discretization was increased. The constant of proportionality in the relationship between the problem size and the solution time depends on the particular problem that is being solved.

- The bending behavior of the coarse mesh elements can slow the convergence of the multigrid method. This problem was partially cured by using more multigrid cycles on the coarse mesh to compute the coarse mesh correction, or by assembling the coarse mesh stiffness matrix using a reduced integration scheme. The bending behavior of the coarse meshes was also found to be responsible for producing solution times that were not proportional to the problem size when nonuniform meshes were used.

- The multigrid method performed poorly when nearly incompressible elasticity problems were encountered. This was due to a combination of a reduction in the smoothing effect of the Gauss-Seidel relaxation and coarse mesh locking. These problems were partially relieved by using Jacobi preconditioned conjugate gradient iteration to smooth the error, and assembling the coarse mesh stiffness matrix with reduced integration.

References

[1] Akhurst, K.N. and Chell, G.G., 1983, "Methods of Calculating Stress Intensity Factors for Nozzle Corner Cracks," *International Journal of Pressure Vessels and Piping*, vol. 14, pp 227–257.

[2] Axelsson, O. and Barker, V.A., 1984, *Finite Element Solution of Boundary Value Problems*, Academic Press.

[3] Bachvalov, N.S., 1966, "On the Convergence of a Relaxation Method With Natural Constraints on the Elliptic Operator," *USSR Computational Mathematics and Mathematical Physics*, vol. 6, no. 5, pp 101–135.

[4] Bank, R.E., 1985, "A-Posteriori Error Estimates, Adaptive Local Mesh Refinement and Multigrid Iteration," pp 7–72 in *Multigrid Methods II*, Hackbusch, W. and Trottenberg, U., editors, Springer-Verlag.

[5] Barrett, K.E., Butterfield, D.M., Ellis, S.E., Judd, C.J., and Tabor, J.H., 1985, "Multigrid Analysis of Linear Elastic Stress Problems," pp 263–282 in *Multigrid Methods for Integral and Differential Equations*, Paddon, D.J. and Holstein, H., editors, Clarendon Press.

[6] Bathe, K.J., 1982, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall.

[7] Bazant, Z.P. and Estenssoro, L.F., 1978, "Surface Singularity and Crack Propagation," *International Journal of Solids and Structures*, vol. 15, pp 405–426.

[8] Benthem, J.P. and Koiter, W.T., 1976, Discussion, *Journal of Applied Mechanics*, vol. 43, pp 374.

[9] Benthem, J.P., 1977, "State of Stress at the Vertex of a Quarter-Infinite Crack in a Half-Space," *International Journal of Solids and Structures*, vol. 13, pp 479–492.

[10] Boore, D.M. and Dunbar, W.S., 1977, "Effect of the Free Surface on Calculated Stress Drops," *Bulletin of the Seismological Society of America*, vol. 67, pp 1661–1664.

[11] Braess, D., 1985, "On the Combination of the Multigrid Method and Conjugate Gradients," pp 52–64 in *Multigrid Methods II*, Hackbusch, W. and Trottenberg, U., editors, Springer-Verlag.

[12] Brandt, A., 1977, "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation*, vol. 31, pp 333–390.

[13] Brandt, A., 1980, "Multilevel Adaptive Computations in Fluid Dynamics," *AIAA Jounal*, vol. 18, pp 1165–1172.

[14] Burton, W.S., Sinclair, G.B., Solecki, J.S., and Swedlow, J.L., 1984, "On the Implications for LEFM of the Three-Dimensional Aspects in Some Crack/Surface Intersection Problems," *International Journal of Fracture*, vol. 25, pp 3–32.

[15] Cisneros, A., 1986, "A Communications System for Irregular Local Interaction Problems on a Concurrent Computer," *Caltech Concurrent Computation Project*, Report C^3P 317, California Institute of Technology.

[16] Cisneros, A., 1986, "Irregular Regions and Multigrid Methods on the Hypercube," *Caltech Concurrent Computation Project*, Report C^3P 318, California Institute of Technology.

[17] Deconinck, H. and Hirsch, C., 1982, "A Multigrid Method for the Transonic Full Potential Equation Discretized With Finite Elements on an Arbitrary Body Fitted Mesh," *Journal of Computational Physics*, vol.48, pp 344–365.

[18] Dzyuba, K.I. and Fradkin, B.V., 1983, "Investigations of the Kurpsai Dam on the Basis of a Numerical Solution of the Three-Dimensional Problem of Elasticity Theory," *Gidrotekhnicheskoe Stroitel'stvo*, no. 2, pp 19–25.

[19] Eshelby, J.D., 1957, "The Determination of the Elastic Stress Field of an Ellipsoidal Inclusion, and Related Problems," *Proceedings of the Royal Society of London, Series A*, vol. 241, pp 376–396.

[20] Fedorenko, R.P., 1961, "A Relaxation Method for Solving Elliptic Difference Equations," *USSR Computational Mathematics and Mathematical Physics*, vol. 1, no. 4, pp 1092–1096.

[21] Folias, E.S., 1975, "On the Three-Dimensional Theory of Cracked Plates," *Journal of Applied Mechanics*, vol. 42, pp 663–674.

[22] Folias, E.S., 1976, Author's Closure, *Journal of Applied Mechanics*, vol. 43, pp 374–375.

[23] Gurtin, M.E., 1984, "Linear Theory of Elasticity," pp 1–295 in *Mechanics of Solids, vol II*, Truesdell, C., editor, Springer-Verlag.

[24] Hackbusch, W. and Hofmann, G., 1980, "Results of the Eigenvalue Problem for the Plate Equation," *Jounal of Applied Mathematics and Physics (ZAMP)*, vol. 31, pp 730–739.

[25] Hackbusch, W. and Trottenberg, U., 1981, editors, *Multigrid Methods*, Springer-Verlag.

[26] Hackbusch, W. and Trottenberg, U., 1985, editors, *Multigrid Methods II*, Springer-Verlag.

[27] Hackbusch, W., 1985, *Multi-Grid Methods and Applications*, Springer-Verlag.

[28] Hageman, L.A. and Young, D.M., 1981, *Applied Iterative Methods*, Academic Press.

[29] Heaton, T.H., 1987, *Personal Communication.*

[30] Hughes, T.J.R., Winget, J., Levit, I., and Tezduyar, T., 1983, "New Alternating Direction Procedures in Finite Element Analysis Based Upon EBE Approximate Factorizations," pp 75–109 in *Computer Methods for Nonlinear Solids and Structural Mechanics*, Atluri, S.N. and Perrone, N., editors, ASME.

[31] Hughes, T.J.R., Raefsky, A., Muller, A., Winget, J., and Levit, I., 1984, "A Progress Report on EBE Solution Procedures in Solid Mechanics," pp 18–26 in *Numerical Methods for Nonlinear Problems, vol. 2*, Taylor, C. et al., editors, Pineridge Press, Swansea.

[32] Hughes, T.J.R., Ferencz, R.M., and Hallquist, J.O., 1986, "Large-Scale Vectorized Implicit Calculations in Solid Mechanics on a CRAY X-MP/48 Utilizing EBE Preconditioned Conjugate Gradients," pp 233–277 in *Computational Mechanics - Advances and Trends*, Noor, A.K., editor, ASME.

[33] Hughes, T.J.R., 1987, *The Finite Element Method*, Prentice-Hall.

[34] Kanamori, H. and Anderson, D.L., 1975, "Theoretical Basis of Some Empirical Relations in Seismology," *Bulletin of the Seismological Society of America*, vol. 65, pp 1073–1095.

[35] Kanninen, M.F. and Popelar, C.H., 1985, *Advanced Fracture Mechanics*, Oxford University Press.

[36] Kela, A., Perucchio, R., and Voelcker, H., 1986, "Toward Automatic Finite Element Analysis," *Computers in Mechanical Engineering*, vol. 5, no. 1, pp 57–71.

[37] Knopoff, L., 1958, "Energy Release in Earthquakes," *Geophysical Journal*, vol. 1, pp 44–52.

[38] Larsson, S.G. and Carlsson, A.J., 1973, "Influence of Non-Singular Stress Terms and Specimen Geometry on Small-Scale Yielding at Crack Tips in Elastic-Plastic Materials," *Journal of the Mechanics and Physics of Solids*, vol. 21, pp 263–277.

[39] Levy, N., Marcal, P.V., and Rice, J.R., 1971, "Progress in Three-Dimensional Elastic-Plastic Stress Analysis for Fracture Mechanics," *Nuclear Engineering and Design*, vol. 17, pp 64–75.

[40] Lindley, P.B., 1978, *Engineering Design With Natural Rubber*, NR Technical Bulletin, Malaysian Rubber Producers' Research Association.

[41] Ortega, J.M. and Voigt, R.G., 1985, "Solution of Partial Differential Equations on Vector and Parallel Computers," *SIAM Review*, vol. 27, pp 149–240.

[42] Reid, J.K., 1983, "Solution of Linear Finite Element Equations," pp 163–181 in *State-of-the-Art Surveys on Finite Element Technology*, Noor, A.K. and Pilkey, W.D., editors, ASME.

[43] Rice, J.R., 1968, "Mathematical Analysis in the Mechanics of Fracture," pp 191–311 in *Fracture: An Advanced Treatise, vol. II, Mathematical Fundamentals*, Liebowitz, H., editor, Academic Press.

[44] Rooke, D.P. and Cartwright, D.J., 1976, *Compendium of Stress Intensity Factors*, HMSO.

[45] Rosakis, A.J. and Ravi-Chandar, K., 1986, "On Crack-Tip Stress State: an Experimental Evaluation of Three-Dimensional Effects," *International Journal of Solids and Structures*, vol. 22, pp 121–134.

[46] Ruge, J. and Stuben, K., 1985, "Efficient Solution of Finite Difference and Finite Element Equations by Algebraic Multigrid," pp 169–212 in *Multigrid Methods for Integral and Differential Equations*, Paddon, D.J. and Holstein, H., editors, Clarendon Press.

[47] Rybicki, E.F. and Benzley, S.E, editors, 1975, *Computational Fracture Mechanics*, ASME.

[48] Schroder, W. and Hanel, D., 1985, "A Comparison of Several MG Methods for the Solution of the Time-Dependent Navier-Stokes Equations," pp 272–

284 in *Multigrid Methods II*, Hackbusch, W. and Trottenberg, U., editors, Springer-Verlag.

[49] Schwarz, H.R., Rutishauser, H., and Stiefel, E., (translated by Hertelendy, P.), 1973, *Numerical Analysis of Symmetric Matrices*, Prentice-Hall.

[50] Sokolnikoff, I.S., 1983, *Mathematical Theory of Elasticity*, Krieger.

[51] Southwell, R.V., 1940, *Relaxation Methods in Engineering Science*, Oxford University Press.

[52] Southwell, R.V., 1946, *Relaxation Methods in Theoretical Physics*, Clarendon Press.

[53] Starr, A.T., 1928, "Slip in a Crystal and Rupture in a Solid Due to Shear," *Proceedings of the Cambridge Philosophical Society*, vol. 24, pp 489–500.

[54] Sternberg, E., 1986, unpublished lecture notes, California Institute of Technology.

[55] Strang, G. and Fix, G.J., 1973, *An Analysis of the Finite Element Method*, Prentice-Hall.

[56] Stuben, K. and Trottenberg, U., 1982, "Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications," pp 1–176 in *Multigrid Methods*, Hackbusch, W. and Trottenberg, U., editors, Springer-Verlag.

[57] Todd, J., 1977, *Basic Numerical Mathematics, vol. 2: Numerical Algebra*, Birkhauser.

[58] Turing, A.M., 1948, "Rounding-Off Errors in Matrix Processes," *Quarterly Journal of Mechanics and Applied Mathematics*, vol. 1, pp 287–308.

[59] Varga, R.S., 1962, *Matrix Iterative Analysis*, Prentice-Hall.

[60] Williams, M.L., 1957, "On the Stress Distribution at the Base of a Stationary Crack," *Journal of Applied Mechanics*, vol. 24, pp 109–114.

[61] Winget, J.M. and Hughes, T.J.R., 1985, "Solution Algorithms for Nonlinear Transient Heat Conduction Analysis Employing Element-by-Element Iterative Strategies," *Computer Methods in Applied Mechanics and Engineering*, vol. 52, pp 711–815.

[62] Yang, W. and Freund, L.B., 1985, "Transverse Shear Effects for Through-Cracks in an Elastic Plate," *International Journal of Solids and Structures*, vol. 21, pp 977–994.

[63] Zienkiewicz, O.C., 1983, *The Finite Element Method*, McGraw-Hill.

Appendix A

The Plane Problem in Linear Elastostatics

The purpose of this appendix is to review the plane problem in linear elastostatics, and thereby clarify the terms plane strain, generalized plane stress, and plane stress [23,50,54]. This is important since the three dimensional crack problem examined in chapter 6 is an example of the plane problem. The study of regions in which the associated plane strain, generalized plane stress, and plane stress solutions hold is an important issue that arises when the elastostatic state near a crack is investigated.

The plane problem is the three dimensional elastostatic problem shown in figure A.1. A cylinder occupying the region $\mathcal{R}$ is centered at the origin O of a rectangular cartesian coordinate system, with its generators parallel to the x_3-axis and with end faces π_1 and π_2 at $x_3 = \pm h/2$. The lateral surface and the middle cross-section of the cylinder are denoted by $\mathcal{B}$ and π_0, respectively. The cylinder is composed of a linear elastic, homogeneous, isotropic solid with Poisson's ratio ν and Young's modulus E. The plane problem of elastostatics consists of finding an elastostatic state, $\mathcal{S} = [u_i, \epsilon_{ij}, \sigma_{ij}]$, on the cylinder such that $\mathcal{S}$ satisfies the boundary conditions $s_\alpha = s^*_\alpha(x_1, x_2)$, $s_3 = 0$ on $\mathcal{B}$ and $s_i = 0$ on π_α. For the pur-

poses of this discussion, it will be assumed that body forces vanish. The symbols u_i, ϵ_{ij}, σ_{ij}, and s_i denote the cartesian components of the displacement vector, the symmetric part of the displacement gradient tensor (the infinitesimal strain tensor), the stress tensor, and the traction vector, respectively. The subscripts i, j have the range 1,2,3, and subscripts α, β, γ have the range 1,2. The usual summation convention will be implied by repeated subscripts. The superscript $*$ means that the quantity has a prescribed value. An important feature of the plane problem is that the applied surface tractions on $\mathcal{B}$ are independent of x_3 and parallel to the $x_1 - x_2$ plane. This motivates the study of two dimensional versions of this problem.

A.1 Plane Strain

The plane strain solution associated with the solution $\mathcal{S}$ of the original three dimensional problem will be denoted by $\mathcal{S}' = [u_i', \epsilon_{ij}', \sigma_{ij}']$, and is obtained by making the following kinematic assumption:

$$\left.\begin{array}{c} u_\alpha' = u_\alpha'(x_1, x_2) \\ u_3' = 0 \end{array}\right\} \text{on } \mathcal{R}. \tag{A.1}$$

This assumption, together with the assumption that $\mathcal{S}'$ is an elastostatic state on $\mathcal{R}$, implies that $\epsilon_{ij}' = \epsilon_{ij}'(x_1, x_2)$ and $\sigma_{ij}' = \sigma_{ij}'(x_1, x_2)$. Also

$$\left.\begin{array}{c} \epsilon_{\alpha\beta}' = \frac{1}{2}(u_{\alpha,\beta}' + u_{\beta,\alpha}'), \quad \epsilon_{i3}' = 0 \\ \sigma_{\alpha 3}' = 0, \quad \sigma_{33}' = \nu\sigma_{\alpha\alpha}' \end{array}\right\} \text{on } \mathcal{R}, \tag{A.2}$$

and

$$\sigma_{\alpha\beta}' = \frac{\mathrm{E}}{2(1+\nu)}\left(\frac{2\nu}{1-2\nu}\delta_{\alpha\beta}u_{\gamma,\gamma}' + u_{\alpha,\beta}' + u_{\beta,\alpha}'\right) \text{ on } \pi_0, \tag{A.3}$$

where the subscript $,\alpha$ denotes differentiation with respect to x_α. In the absence of body forces, the equilibrium equation becomes

$$\sigma'_{\alpha\beta,\beta} = 0 \text{ on } \pi_0, \tag{A.4}$$

and the traction boundary conditions on $\mathcal{B}$ reduce to

$$\sigma'_{\alpha\beta} n_\beta = s^*_\alpha \text{ on } \partial\pi_0. \tag{A.5}$$

Equations (A.3), (A.4), and (A.5) define a two dimensional boundary value problem for u'_α and $\sigma'_{\alpha\beta}$ which has a unique solution.

An important feature, given in equation (A.2), of the plane strain solution is

$$\sigma'_{33} = \nu\sigma'_{\alpha\alpha}. \tag{A.6}$$

Thus the plane strain solution, in general, violates the traction free conditions imposed on the ends π_α of the cylinder. However, strictly two dimensional plane strain conditions may be generated in the cylinder by applying tractions to the ends of the cylinder such that

$$\left.\begin{aligned} \sigma'_{\alpha 3} &= 0 \\ \sigma'_{33} &= \nu\sigma'_{\alpha\alpha} \end{aligned}\right\} \text{ on } \pi_\alpha. \tag{A.7}$$

Consequently, the full three dimensional solution to the original three dimensional plane problem can then be obtained using the principle of superposition and by solving the so called residual problem of plane strain. This consists of finding an elastostatic state, $\mathcal{S}'' = [u''_i, \epsilon''_{ij}, \sigma''_{ij}]$, obtained by applying on π_α the negative of the tractions required to generate the plane strain solution, i.e. applying tractions such that

$$\left.\begin{aligned} \sigma''_{\alpha 3} = 0,\ \sigma''_{33} = -\nu\sigma'_{\alpha\alpha} \text{ on } \pi_\alpha, \\ s''_i = \sigma''_{ij} n_j = 0 \text{ on } \mathcal{B}. \end{aligned}\right\} \tag{A.8}$$

This residual problem is a problem of the Saint-Venant type and thus no simpler than the three dimensional plane problem. Note that the solution to the plane problem is equal to the plane strain solution associated with the plane problem plus the solution to the residual problem of plane strain (i.e. $\mathcal{S} = \mathcal{S}' + \mathcal{S}''$).

A.2 Generalized Plane Stress

The generalized plane stress solution associated with the solution of the three dimensional plane problem characterizes the thickness averages of $u_\alpha, \epsilon_{\alpha\beta}, \sigma_{\alpha\beta}$ under the approximative assumption

$$\sigma_{33} = 0 \text{ on } \mathcal{R}. \tag{A.9}$$

The thickness average, $\bar{f}$, of a function f on $\mathcal{R}$ is defined by

$$\bar{f}(x_1, x_2) = \frac{1}{h} \int_{-\frac{h}{2}}^{\frac{h}{2}} f(x_1, x_2, x_3) dx_3, \quad \forall (x_1, x_2) \in \pi_0. \tag{A.10}$$

Using the above definition and assumption, the following relations can be shown to hold:

$$\left.\begin{aligned} \bar{\epsilon}_{\alpha\beta} &= \tfrac{1}{2}(\bar{u}_{\alpha,\beta} + \bar{u}_{\beta,\alpha}) \text{ on } \pi_0, \\ \bar{\sigma}_{\alpha\beta} &= \tfrac{\mathrm{E}}{2(1+\nu)} \left(\tfrac{2\nu}{1-\nu} \delta_{\alpha\beta} \bar{u}_{\gamma,\gamma} + \bar{u}_{\alpha,\beta} + \bar{u}_{\beta,\alpha} \right) \text{ on } \pi_0, \\ \bar{\sigma}_{\alpha\beta,\beta} &= 0 \text{ on } \pi_0, \\ \bar{\sigma}_{\alpha\beta,\beta} n_\beta &= s^*_\beta \text{ on } \partial\pi_0. \end{aligned}\right\} \tag{A.11}$$

Equation (A.11) is a complete two dimensional boundary value problem for $\bar{u}_\alpha$, $\bar{\epsilon}_{\alpha\beta}$, $\bar{\sigma}_{\alpha\beta}$. The solution of this problem, $\bar{\mathcal{S}} = [\bar{u}_\alpha, \bar{\epsilon}_{\alpha\beta}, \bar{\sigma}_{\alpha\beta}]$, is called the generalized plane stress solution associated with the plane problem. It should be

noted that this does not satisfy the three dimensional field equations, nor does it give any information about the remaining unknowns, e.g. u_3.

It can be seen that, in terms of the stress components, the mathematical formulations of the plane strain and generalized plane stress solutions are identical. The governing differential equations and boundary conditions differ only in the appearance of the barred symbols. It can be shown that for a simply connected body in the absence of body forces, the in-plane stresses in the two solutions are exactly the same [23].

A.3 Plane Stress

The plane stress solution associated with the solution of the plane problem consists of an elastostatic state, $\hat{\mathcal{S}} = [\hat{u}_\alpha, \hat{\epsilon}_{\alpha\beta}, \hat{\sigma}_{\alpha\beta}]$, which is obtained in the following way. Assume

$$\hat{\sigma}_{3i} = 0 \text{ on } \mathcal{R}, \tag{A.12}$$

and suppose that $\hat{\mathcal{S}}$ is symmetric about the plane $x_3 = 0$, i.e.

$$\left.\begin{aligned} \hat{u}_\alpha(x_1, x_2, -x_3) &= \hat{u}_\alpha(x_1, x_2, x_3) \\ \hat{u}_3(x_1, x_2, -x_3) &= -\hat{u}_3(x_1, x_2, x_3) \end{aligned}\right\} \text{on } \mathcal{R}, \tag{A.13}$$

and that

$$\frac{1}{h}\int_{-\frac{h}{2}}^{\frac{h}{2}} \hat{s}_\alpha dx_3 = s^*_\alpha \text{ on } \partial\pi_0, \tag{A.14}$$

where

$$\hat{s}_\alpha(x_1, x_2, x_3) = \hat{\sigma}_{\alpha\beta}(x_1, x_2, x_3) n_\beta(x_1, x_2).$$

It can be shown [54] that $\hat{\mathcal{S}}$ admits the representation on $\mathcal{R}$:

$$\left.\begin{aligned}
\hat{u}_\alpha(x_1,x_2,x_3) &= \bar{u}_\alpha(x_1,x_2) + \tfrac{\nu}{2\mathrm{E}}\phi(x_3)\bar{\sigma}_{\beta\beta,\alpha}(x_1,x_2),\\
\hat{u}_3(x_1,x_2,x_3) &= -\tfrac{\nu}{\mathrm{E}}x_3\bar{\sigma}_{\alpha\alpha}(x_1,x_2),\\
\hat{\epsilon}_{ij} &= \tfrac{1}{2}(\hat{u}_{i,j}+\hat{u}_{j,i}),\\
\hat{\sigma}_{\alpha\beta}(x_1,x_2,x_3) &= \bar{\sigma}_{\alpha\beta}(x_1,x_2) + \tfrac{\nu}{2(1+\nu)}\phi(x_3)\bar{\sigma}_{\gamma\gamma,\alpha\beta}(x_1,x_2),\\
\hat{\sigma}_{3i} &= 0,\\
\text{where } \phi(x_3) &= x_3^2 - \tfrac{h^2}{12},\ (-\tfrac{h}{2}\le x_3\le\tfrac{h}{2}).
\end{aligned}\right\} \qquad (A.15)$$

The plane stress solution satisfies the field equations and the boundary conditions for the ends of the cylinder, but in general violates the boundary conditions for the boundary $\mathcal{B}$, which are met only in the mean. It does, however, give information about the unknowns that do not appear in the generalized plane stress solution (e.g. u_3). It should be noted that the plane stress solution is not a true two dimensional solution since the field quantities depend on x_3.

The purpose of this appendix was to explain fully the terms plane strain, generalized plane stress, and plane stress. This is relevant to the study of the elastostatic state near a three dimensional edge crack because it is useful to investigate regions where the two dimensional plane strain or generalized plane stress states hold.

The normal strain in the original three dimensional problem, ϵ_{33}, is given by

$$\epsilon_{33} = \frac{1}{\mathrm{E}}\left[\sigma_{33} - \nu(\sigma_{11}+\sigma_{22})\right]. \qquad (A.16)$$

Therefore, a convenient measure of the degree of plane strain has been taken to

be the ratio

$$\frac{\sigma_{33}}{\nu(\sigma_{11}+\sigma_{22})} = \frac{\mathrm{E}\epsilon_{33}}{\nu(\sigma_{11}+\sigma_{22})} + 1 \tag{A.17}$$

[39]. This will have a value of zero when $\sigma_{33} = 0$, which corresponds to conditions of generalized plane stress and plane stress. The degree of plane strain will have a value of one when $\epsilon_{33} = 0$ (which corresponds to plane strain conditions). However, if ϵ_{33} is bounded and nonzero at a stress singularity (which does not correspond to plane strain conditions), then the ratio will still have a value of one. This ratio is used in chapter 6 to examine the elastostatic state near a crack front. In addition, a comparison between the out-of-plane displacements u_3 and $\hat{u}_3$, the plane stress value given in equation (A.15), is used to identify regions where plane stress conditions are achieved.

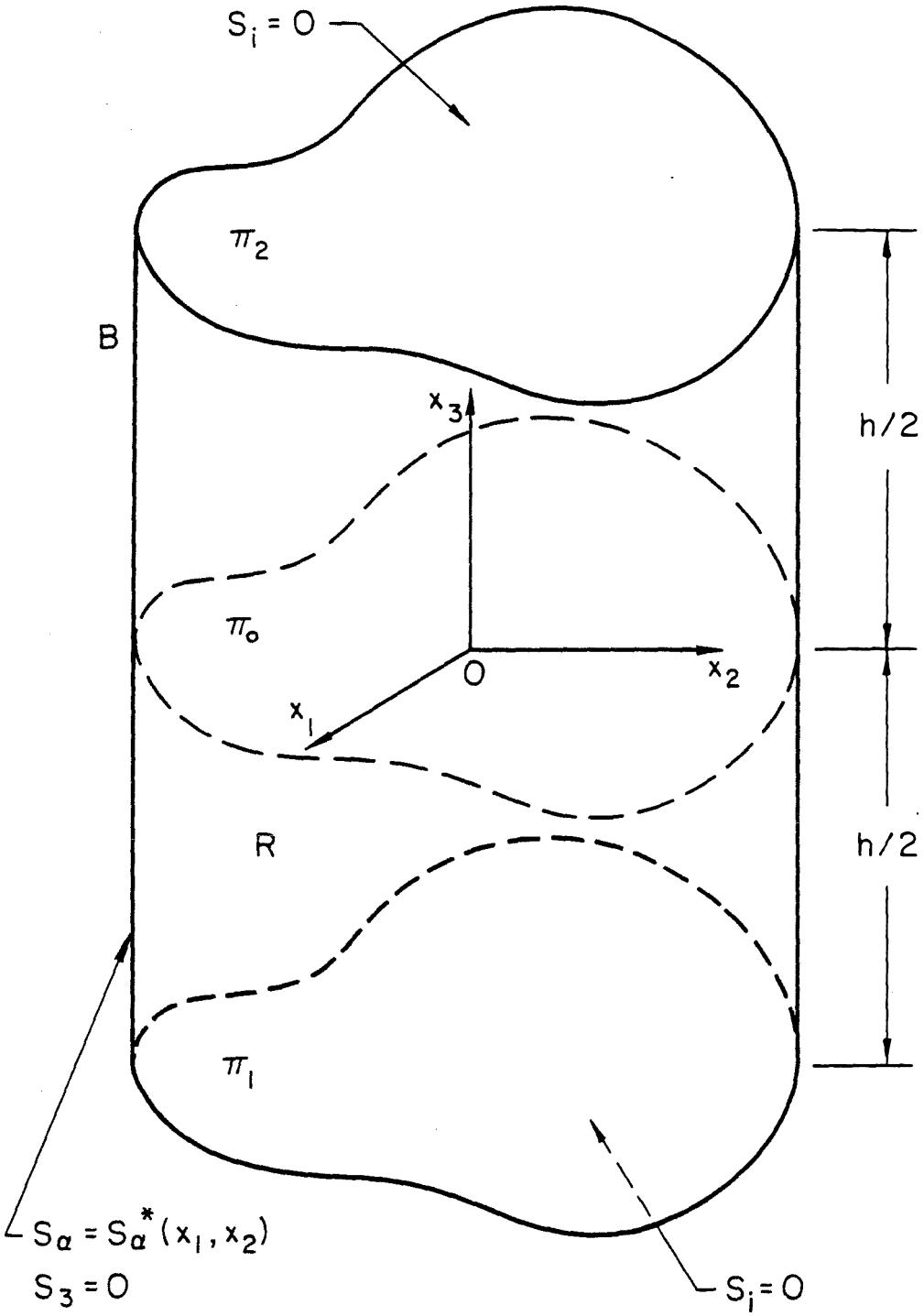


Figure A.1: The Plane Problem.