Continuum damage model for compressive fatigue of high strength concrete.

Shaalan, Ali Mohamed Kamel Ali

Civil Engineering Department

December 1998

Abstract

An anisotropic elasto-damage model for predicting the response of concrete subjected to monotonic and fatigue loading is presented in this research. The model utilizes a concrete appropriate damage-effect tensor M in constructing the constitutive equations using the concept of multiple bounding surfaces. One of the salient features is the introduction of a bounding surface that constrains the elliptical movement of the limit fracture surface. The model after calibration is shown to predict the monotonic, comprehensive uniaxial stress strain path for concrete of various strengths as well as S-N curves depicting the fatigue response of concrete for uniaxial and biaxial loading. A concept of residual strength determination based on continuum damage formulation is also introduced.

Continuum Damage Model for Comprehensive Fatigue of High Strength Concrete

by

Ali Mohamed Kamel Ali Shaalan

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

CIVIL ENGINEERING

December, 1998

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

NOTE TO USERS

The original document received by UMI contains pages with indistinct print. Pages were microfilmed as received.

This reproduction is the best copy available

UMI

CONTINUUM DAMAGE MODEL FOR COMPRESSIVE FATIGUE OF HIGH STRENGTH CONCRETE

BY

Ali Mohamed Kamel Ali Shaalan

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

In

CIVIL ENGINEERING

December, 1998

UMI Number: 1393217

UMI Microform 1393217
Copyright 1999, by UMI Company. All rights reserved.

This microform edition is protected against unauthorized
copying under Title 17, United States Code.

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
DHAHRAN 31261, Saudi Arabia

COLLEGE OF GRADUATE STUDIES

This thesis, written by **Ali Mohamed Kamel Shaalan** under the direction of his Thesis Advisor and approved by his Thesis Committee, has been presented to and accepted by the Dean of the College of Graduate Studies, in partial fulfillment of the requirements for the degree of MASTER **OF SCIENCE** in **CIVIL ENGINEERING** (Structures)

Thesis Committee

Prof. **Mohammed H. Baluch**
Thesis Adviser

Dr. **Ali H. Al-Gadhib**
Co-Advisor

Dr. **Husain J. Al-Gahtani**
Member

Dr. **Maher A. Bader**
Member

Dr. **Sahel N. AbdulJauwad**
Department Chairman

Dr. **Abdullah M. Al-Shehri**
Dean, College of Graduate Studies

Date: 26 December, 1998

بسم الله الرحمن الرحيم

This thesis
is dedicated
to my parents, wife,
daughter, son
and to those
who do good deeds
for the sake of
ALLAH

ACKNOWLEDGMENT

Praise and thanks be to Almighty Allah for His limitless helping and guidance; and peace and prayers be upon his Prophet.

Acknowledgement is due to King Fahd University of Petroleum and Minerals for support of this study through its enormous computing and laboratory facilities.

My expression of thanks to Prof. Mohammed H. Baluch and Dr. Ali H. Al-Gadhib, the major advisor and the co-advisor of my thesis committee, does not arise from convention rather, from my heart-felt gratitude for their constant counsel, matchless patience and extraordinary attentiveness.

I am deeply indebted to Dr. Husain J. Al-Gahtani and Dr. Maher A. Bader, the other two-committee members, for their cooperation and valuable suggestions.

It would be a form of unforgivable negligence if I do not mention my wife. She sacrificed her time and effort and devoted her support during the preparation of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

THESIS ABSTRACT

FULL NAME OF STUDENT	:	**ALI MOHAMED KAMEL SHAALAN**
TITLE OF STUDY	:	CONTINUUM DAMAGE MODEL FOR COMPRESSIVE FATIGUE OF HIGH STRENGTH CONCRETE
MAJOR FIELD	:	Civil Engineering
DATE OF DEGREE	:	December, 1998

An anisotropic elasto-damage model for predicting the response of concrete subjected to monotonic and fatigue loading is presented in this research. The model utilizes a concrete appropriate damage-effect tensor M in constructing the constitutive equations using the concept of multiple bounding surfaces. One of the salient features is the introduction of a bounding surface that constrains the elliptical movement of the limit fracture surface. The model after calibration is shown to predict the monotonic, compressive uniaxial stress strain path for concrete of various strengths as well as S-N curves depicting the fatigue response of concrete for uniaxial and biaxial loading. A concept of residual strength determination based on continuum damage formulation is also introduced.

MASTER OF SCIENCE DEGREE

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

Dhahran, Saudi Arabia

December, 1998

خلاصة الرسالة

الأسم الكامل : علي محمد كامل علي شعلان

عنوان الرسالة : النمذجة التلفية المتصلة لإجهادات الضغوط الكلالية للخرسانة عالية الإجهاد.

التخصص : هندسة إنشائية.

في هذا البحث تم عرض النمذجة التلفية المرنة المتباينة الخواص للتنبؤ بتصرف الخرسانة تحت تأثير الحمل المطرد وحمل الكلال، النموذج يستخدم مصفوفة تأثير التلف (م) في بناء العلاقة الأساسية بإستخدام طريقة الأسطح المقيدة المتعددة. إحدى الصفات البارزة تقديم المسار المقيد الذى يحدد الحركة البيضاوية للسطح المحدد للتهشم. النموذج بعد المعايرة يظهر تنبؤ بمسار العلاقة بين الإجهاد والانفعال للحمل المطرد في حالة الضغط الاحادي المحور لخرسانات ذات اجهادات قصوى مختلفة وكذلك منحنى العلاقة بين قيمة الحمل وعدد مرات التحميل المتكررة الذي يصف استجابة الكلال في الخرسانة تحت تأثير الحمل الأحادى والحمل الثنائي المحور ، وكذلك تم تعريف طريقة تحديد القوة المتبقية مبنية علي أساس علاقات التلف المتصل.

درجة الماجستير في العلوم

جامعة الملك فهد للبترول والمعادن

الظهران – المملكة العربية السعودية

ديسمبر ١٩٩٨

CHAPTER 1

INTRODUCTION

The increased use of concrete as a primary structural material in building complex structures necessitates the development of sophisticated analysis capabilities for accurately predicting the response of such structures whenever subjected to different types of loading including monotonic loading, unloading, reloading and in particular fatigue loading. Analytical approaches fail to analyze structures in which geometry or loading is complex, necessitating the use of alternate techniques to be employed for structures of complex nature or loading. Numerical techniques are powerful alternatives.

The numerical technique of finite element is firmly established and has been used extensively in solving engineering problems in different disciplines. For linear analysis, the technique is widely employed with confidence. However concrete is no more linear isotropic and elastic and its stiffness already has degraded significantly, the trend is to go for ultimate design. The nonlinear nature of the problem is not captured unless the deformation history is traced until failure occurs, which requires a nonlinear finite element analysis.

The crucial part in structural analysis modeling is the constitutive relations and the associated material damage modeling. No matter how sophisticated the method of solution is, it cannot increase the accuracy of an unrealistic model.

1.1 LITERATURE REVIEW

Concrete belongs to the typical class of composite materials since it consists of stiff inclusions embedded in a weaker matrix material. Due to its heterogeneous microstructure, failure mechanisms are of a complex nature. Microvoids initiate at the interface between the cement matrix and the aggregate particles of concrete specimens leading to local stiffness degradation.

In recent years, there has been a proliferation of models for predicting short-term rate independent stress-strain behavior of concrete. For example concrete has been treated as elastic isotropic or hypoelastic material (see Ref. [24], [31], [32] and [33]). However this type of material modeling ignores the path dependence of deformation and is suited only for monotonic loading. In the early stages of constitutive modeling, plasticity models which have been originally used to predict ductile behaviors of metals, have been used to predict the nonlinear response of reinforced concrete shells.

A number of plasticity models for concrete ranging from one parameter to five parameter models have become available. These models are known as Drucker-Prager[36], Ottosen[37], and Willam and Warnke[40], respectively. Some of these models have been incorporated into a finite element coding for analysis of concrete structures Ref. [28] and [29].

Desai [34] introduced a model based on the use of a function expressed as polynomial form in terms of the three invariants of stress. The author has demonstrated that the majority of the aforementioned models can be degenerated from this model as special cases and thus unifying the existing models. Moreover, this model gives a single continuous surface (or function) which includes both yielding and failure surfaces (or function) contained in the cap model. This model of Desai's has been used along with nonuniform hardening plasticity by Dafalios and Popov [33] to predict behavior of concrete.

In the past two decades, the continuum damage mechanics approach has emerged as a state variable framework for the description of distributed material damage including material stiffness degradation, microcracking initiation, and damage-induced anisotropy. Damage mechanics has also been introduced to describe the inelastic behavior of brittle materials such as concrete, rock and ceramics.

A continuum damage mechanics theory was first used by Kachanov [25] to model the creep rupture of metals. In this model, damage was defined as the area density of voids in a known cross-section. The damage was treated as a kinematics variable, the growth of which resulted in gradual degradation of the material. In the early stages, the continuum damage models treated material as isotropic and therefore they adopted a scalar damage approach as in Lemaitre [27], in which a model of isotropic ductile plastic damage based on a continuum damage variable using effective stress concept and principle of thermodynamics is derived. The damage is linear with equivalent strain and shows a large influence of triaxiality through the use of a damage equivalent stress.

Suaris and Shah [26] consider flat microcracks where a vectorial representation is adopted instead of scalar for the damage variable. They consider a free energy function, which is dependent on the coupled invariant of strain and damage, and derive the constitutive equations by using the thermodynamic restrictions.

Continuum damage mechanics is concerned with the modeling of material deterioration on a phenomenological level. The bulk of existing continuum damage formulations is restricted to isotropic damage evolution resulting in the degradation of the elasticity modulus as a function of a scalar damage parameter by exploiting the notion of effective stress Lemaitre and Chaboche [44] and a recent work by Marotti de Sciarra [41] provide a useful review in the evolution of such models.

Several formulations have been proposed to extend the concept of effective stress to anisotropic damage model (see Ortiz [13], Simo and Ju [43], Yazdani and Schreyer [42] and Carol et al. [46]). Recently, Simo et al[50], Govindjee et al. [47] and Meschke et al. [49]) have proposed an anisotropic damage model, using the principal of maximum dissipation for defining the evolution of the compliance tensor.

Representation of anisotropic behavior of brittle materials has recently been addressed in the context of kinematic softening plasticity theory by Lourenco et al. [48]. In this formulation the degradation of the post-cracking residual strength is controlled by a scalar internal variable. Also a tensorial damage formulation which is similar in terminology and notation to the well known theory of elasto-plasticity has been proposed (Ignaclo et al.) [51]

An anisotropic damage model of concrete has been developed by Ramn and De Bost [52] based on microplane model approach which has been developed recently by Bazant and Prat[53], and Carol and Bazant [54].

The advancement and growth in concrete construction industries in response to the demand of high quality concrete has led to the production concrete of compressive strength of the order of 120 MPa through the use of selective admixtures and additives. In contrast, the understanding pertaining to behavior of high-strength concrete and failure mechanisms, especially under cyclic loading, has not kept pace. It is only in recent years that fatigue of concrete has received increasing attention, being essentially stimulated by the increased use of slender concrete structures with repetitive live load possibly being the major portion of the total load [1].

Fatigue, unfortunately, is a property of concrete that is least understood, especially with regard to modeling. Continuum damage modeling (CDM) has given new impetus to the constant search for improvement in constitutive modeling of complex bimodular material [2,3]. With sufficient insight gained into CDM for both brittle and ductile fractures [4,6], research has been initiated into incorporation of CDM models into finite element model for solution of problems of engineering interest [7,9]. Suaris et al. [4] have developed a damage model for monotonic and cyclic behavior of concrete where elastic potential is introduced in terms of principal stresses and damage dependent compliance tensor. The evolution of damage is calculated by tracking the movement of the loading surface in its approach towards the bounding surface, a concept originally introduced in elasto-plasticity by Dafalias [10]. The limit fracture surface (which set the threshold of damage), the loading surface and the

bounding surface are all expressed in terms of the strain energy release rate R_i. Ref. [11] has extended the stress control model of Ref. [4], making provisions for strain control under general loading and incorporation into a finite element code to solve a variety of problems including strip loading of rectangular prism and Brazilian test.

The Suaris model [4] presents an elegant approach to the formulation of the elasto-damage problem, but has certain inherited deficiencies. The first is the underestimated of concrete strength by approximately 25% in biaxial compression as obtained from Kupfer experiments [12]. The second limitation is that the Suaris model when applied to cyclic loading of concrete simulates the experimental S-N curve only in the range of the low-cyclic fatigue which correspond to high cycling stress ratio σ/fc'. Further, the material damage parameters are calibrated for one specific concrete compressive strength, thus limiting the use of the model.

Nonlinearity in concrete is attributed to the development of microcracks and microvoids which tend to destroy the interface bond between the cement matrix and aggregate and/or destruction of the material grains themselves, affecting the elastic properties and imparting anisotropy to the material [13]. Continuum damage models have been developed to account for such anisotropic damage for an initially isotropic material [14]. Based on the use of damage-effect tensor M, constitutive equations of anisotropic damage have been developed [6], using the hypothesis of elastic strain energy equivalence introduced in [15]. A symmetric stiffness matrix results in contrast to the postulate of equivalence of strain which leads to asymmetry of the stiffness matrix [16]. The concept of the M-tensor is an attractive one that has been applied to predict damage and behavior of metals [5-7]. It is only recently that state-

of-the-art constitutive equations expressed in terms of the M-tensor for concrete have been in Ref. [17] where essential features of concrete such as degradation of elastic properties, strain softening, gain in strength under confinement and different behavior in tension and compression have been captured effectively.

In addition to the data presented in Ref. [4] for fatigue of concrete in uniaxial compression, use has been made of experimental data given by Su and Hsu [1] for uniaxial and biaxial compression fatigue loading of concrete to calibrate parameters governing the size of the limit fracture surface.

1.2 OBJECTIVES AND SCOPE OF WORK

I. To formulate the constitutive elasto-damage relations in principal frame based on the effective C-tensor concept presented in Ref. [17] for use in concrete.

II. To calibrate the material damage parameters β and D and to simulate the uniaxial and biaxial compression behavior for concretes of varying strengths for monotonic loading.

III. To further generalize the model to predict the fatigue life of concrete in uniaxial and biaxial compression loading. This requires a hypothesis governing the movement of the limit fracture surface and a calibration of the size parameter R_o embedded in the description of the limit fracture surface.

IV. To predict the S-N curve for high strength concrete under uniaxial and biaxial compression loading.

V. To present an exploratory approach for simulating residual strength of concrete subjected to inherent damage resulting from uniaxial cyclic compression loading.

CHAPTER 2

ELASTO-DAMAGE LAW

In this chapter, the fundamental aspects of elasto-damage law are deliberately outlined. The theoretical preliminaries of the damage mechanics will be derived by using the bounding surface technique. The concepts of the damage effective tensor and constitutive elasto-damage equation for monotonic and cyclic loading are derived. Then the residual strength of concrete is defined.

2.1 MONOTONIC LOADING

2.1.1 DAMAGE EFFECT TENSOR

Damage variable may be considered as an internal state variable, which characterize the irreversible deterioration of a material point in accordance with the thermodynamic formulation [18]. Based on the theory of isotropic continuum damage mechanics, the effective Cauchy stress tensor $\bar{\underset{\sim}{\sigma}}$ is related to the usual Cauchy stress tensor $\underset{\sim}{\sigma}$ by

$$\bar{\underset{\sim}{\sigma}} = \left\{\frac{1}{1-\omega}\right\}\underset{\sim}{\sigma} \tag{2.1}$$

where ω is a scalar measure for damage. However in general, the internal state variable may be portrayed through a damage effect tensor, as introduced by Leckie and Onat [19]. For anisotropic damage, the effective stress can be expressed in a generalized form as:

$$\bar{\underset{\sim}{\sigma}} = M(\omega):\underset{\sim}{\sigma} \tag{2.2}$$

where the symbol (:) means tensorial product contracted on two indices and $M(\omega)$, known as damage-effect tensor, is a linear symmetric operator represented by a fourth order tensor. There are many possible forms of the generalized damage-effect tensor $\mathbf{M}_{ij}$. However, one obvious criterion for development of such damage-effect tensor is that it should reduce to a scalar for isotropic damage. This reduction should be possible not only in a principal coordinate system but also in any coordinate system

There are many possible forms of the generalized damage-effect tensor which obey the stated criterion, with some of this forms defined in [5-7] for analysis of metals. However, as far as concrete behavior is concerned, one form of the M tensor which satisfy the stipulated criterion has been introduced in Ref. [17] and takes the following form:

$$M_{IJ} = \begin{bmatrix} \frac{(1-\beta\omega_1)}{(1-\alpha\omega_1)(1-\beta\omega_2)(1-\beta\omega_3)} & 0 & 0 \\ 0 & \frac{(1-\beta\omega_2)}{(1-\beta\omega_1)(1-\alpha\omega_2)(1-\beta\omega_3)} & 0 \\ 0 & 0 & \frac{(1-\beta\omega_3)}{(1-\beta\omega_1)(1-\beta\omega_2)(1-\alpha\omega_3)} \end{bmatrix} \tag{2.3}$$

where ω_i, i = 1, 2, 3, are the principal damage components. The parameters α and β are introduced to calibrate and account for the different behavior of concrete in tension and compression. It is obvious that for isotropic damage, $\omega_1 = \omega_2 = \omega_3 = \omega$ and Equation (2.3) can be readily reduced to a scalar.

For undamage state, the linear elastic constitutive relation is:

$$\underset{\sim}{\varepsilon}^e = C : \underset{\sim}{\sigma} \tag{2.4}$$

Where $\underset{\sim}{\varepsilon}^e$ is the elastic strain tensor and C is the elastic compliance tensor given by

$$[C] = \frac{1}{E_o}\begin{bmatrix} 1 & -\nu & -\nu \\ -\nu & 1 & -\nu \\ -\nu & -\nu & 1 \end{bmatrix} \tag{2.5}$$

However, if the material is in a damaged state, then the elasto-damage constitutive equation can be written as

$$\underset{\sim}{\varepsilon} = \overline{C} : \underset{\sim}{\sigma} \tag{2.6}$$

Where $\overline{C}$ is the effective compliance matrix and $\underset{\sim}{\varepsilon}$ is the elasto-damage strain tensor. In the hypothesis of elastic energy equivalence stated in [15], the complementary elastic energy for a damaged material is the same in form as that for an undamaged material, except that the Cauchy stress $\underset{\sim}{\sigma}$ is replaced by the Cauchy effective stress $\underset{\sim}{\overline{\sigma}}$ in the energy

formulation. Accordingly, the complementary energy per unit volume $\rho\Lambda$ (ρ is material mass density) for undamaged and damaged states may be written as:

$$\rho\Lambda(\sigma,0)=\frac{1}{2}\underset{\sim}{\sigma}^{T}\underset{\sim}{\varepsilon}^{e}=\frac{1}{2}\underset{\sim}{\sigma}^{T}:C:\underset{\sim}{\sigma} \tag{2.7}$$

$$\rho\Lambda(\sigma,\omega)=\frac{1}{2}\underset{\sim}{\bar{\sigma}}^{T}:C:\underset{\sim}{\bar{\sigma}}$$

$$=\frac{1}{2}\underset{\sim}{\sigma}^{T}:\left(M^{T}:C:M\right):\underset{\sim}{\sigma}$$

$$\rho\Lambda(\sigma,\omega)=\frac{1}{2}\underset{\sim}{\sigma}^{T}:\bar{C}:\sigma \tag{2.8}$$

where

$$\bar{C}=M^{T}:C:M \tag{2.9}$$

Upon substitution of equation (2.3) and (2.5) into equation (2.9), one may write the components of the effective compliance matrix explicitly as:

$$\bar{C}=\begin{bmatrix}\bar{C}_{11} & \bar{C}_{12} & \bar{C}_{13}\\ \bar{C}_{21} & \bar{C}_{22} & \bar{C}_{23}\\ \bar{C}_{31} & \bar{C}_{32} & \bar{C}_{33}\end{bmatrix} \tag{2.10}$$

where,

$$\bar{C}_{11}=\frac{(1-\beta\omega_{1})^{2}}{E_{o}(1-\alpha\omega_{1})^{2}(1-\beta\omega_{2})^{2}(1-\beta\omega_{3})^{2}} \tag{2.10a}$$

$$\bar{C}_{22}=\frac{(1-\beta\omega_{2})^{2}}{E_{o}(1-\beta\omega_{1})^{2}(1-\alpha\omega_{2})^{2}(1-\beta\omega_{3})^{2}} \tag{2.10b}$$

$$\overline{C}_{33}=\frac{(1-\beta\omega_3)^2}{E_o(1-\beta\omega_1)^2(1-\beta\omega_2)^2(1-\alpha\omega_3)^2} \tag{2.10c}$$

$$\overline{C}_{12}=\overline{C}_{21}=\frac{-\nu}{E_o(1-\alpha\omega_1)(1-\alpha\omega_2)(1-\beta\omega_3)^2} \tag{2.10d}$$

$$\overline{C}_{13}=\overline{C}_{31}=\frac{-\nu}{E_o(1-\alpha\omega_1)(1-\beta\omega_2)^2(1-\alpha\omega_3)} \tag{2.10e}$$

$$\overline{C}_{23}=\overline{C}_{32}=\frac{-\nu}{E_o(1-\beta\omega_1)^2(1-\alpha\omega_2)(1-\alpha\omega_3)} \tag{2.10f}$$

From (2.10) it is obvious that the thermodynamic constraint requirement $E_i\nu_{ji} = E_j\nu_{ij}$ is satisfied.

2.1.2 GENERAL EVOLUTION EQUATIONS OF DAMAGE

In order to construct a rational model accounting for damage growth, concepts are borrowed from incremental theory of plasticity in general and the bounding surface plasticity model in particular as introduced by Dafalias & Popov [10]. Plasticity bounding surface model as proposed by Dafalias requires definition of multiple surfaces in stress space. However, the fundamental surfaces in the present work are best described in strain-energy release space, as proposed by Suaris et al. [4] and given by

$$\mathrm{f}=(R_iR_i)^{1/2}-R_c/b=0 \tag{2.11}$$

$$\mathrm{F}=(\overline{R}_i\overline{R}_i)^{1/2}-R_c=0 \tag{2.12}$$

$$\mathrm{f}_0=(R_iR_i)^{1/2}-R_0=0 \tag{2.13}$$

where, f is the loading surface (LS), F is the bounding surface (BS), and f_o is a limit fracture surface (LFS) as shown in Fig. 2.1. The loading function surface f is defined in terms of thermodynamic-force conjugates, R_i, where,

$$R_i = \rho \frac{\partial \Lambda}{\partial \omega_i}\left(\sigma_{ij}, \omega_i\right) \tag{2.14}$$

$\overline{R}$, is an image point on F = 0 associated with a given point R_i on f = 0 defined by a mapping rule

$$\overline{R}_i = bR_i \tag{2.15}$$

$$b = R_C / \left(R_i R_i\right)^{1/2} \tag{2.16}$$

with the mapping parameter b is ranging from an initial value of ∞ to a limiting value of 1 on growth of loading surface to eventual coalescence with bounding surface. R_c critical energy release rates, is a parameter of the model and is calibrated to the standard uniaxial compression test.

The damage growth is determined from the loading surface f = 0 where the damage increment vector is assumed to be coaxial with the gradient of f which itself is also the unit vector n_i to the loading surface as shown in Fig. 2.1. Therefore the principal damage components may be written as:

$$d\omega_i = d\lambda \frac{\partial f}{\partial R_i} = d\lambda \; n_i \tag{2.17}$$

with k = Rc/b, equation of loading surface becomes

$$f(R_i, k) = (R_i R_i)^{1/2} - k(\bar{\omega}_P) = 0 \tag{2.18}$$

where $\bar{\omega}_P$ is the form of the accumulated damage and whose increment is defined by

$$d\bar{\omega}_P = [d\omega_i \, d\omega_i]^{1/2} = d\lambda \tag{2.19}$$

It can be shown readily from (2.17) and (2.19) that the scalar magnitude of $d\bar{\omega}_P = d\lambda$.

The satisfaction of the consistency condition df = 0, yield

$$df = \frac{\partial f}{\partial R_i} dR_i + \frac{\partial f}{\partial k} dk = 0 \tag{2.20}$$

From (2.14) one may write

$$dR_i = \frac{\partial R_i}{\partial \sigma_K} d\sigma_K + \frac{\partial R_i}{\partial \omega_j} d\omega_j \tag{2.21}$$

Also from (2.18), the incremental increase in the loading surface size may be written as

$$dk = \frac{\partial k}{\partial \bar{\omega}_P} d\bar{\omega}_P = \frac{\partial k}{\partial \bar{\omega}_P} d\lambda \tag{2.22}$$

Introducing H = $\dfrac{\partial k}{\partial \bar{\omega}_P}$ = damage modulus, it can be measured experimentally in a uniaxial compression test, and the same form assumed for a more general stress path. In the present work, H is expressed as a function of the distance between the loading and the bounding surface [4], given by

$$H = \frac{D\delta}{\langle \delta_{in} - \delta \rangle} \tag{2.23}$$

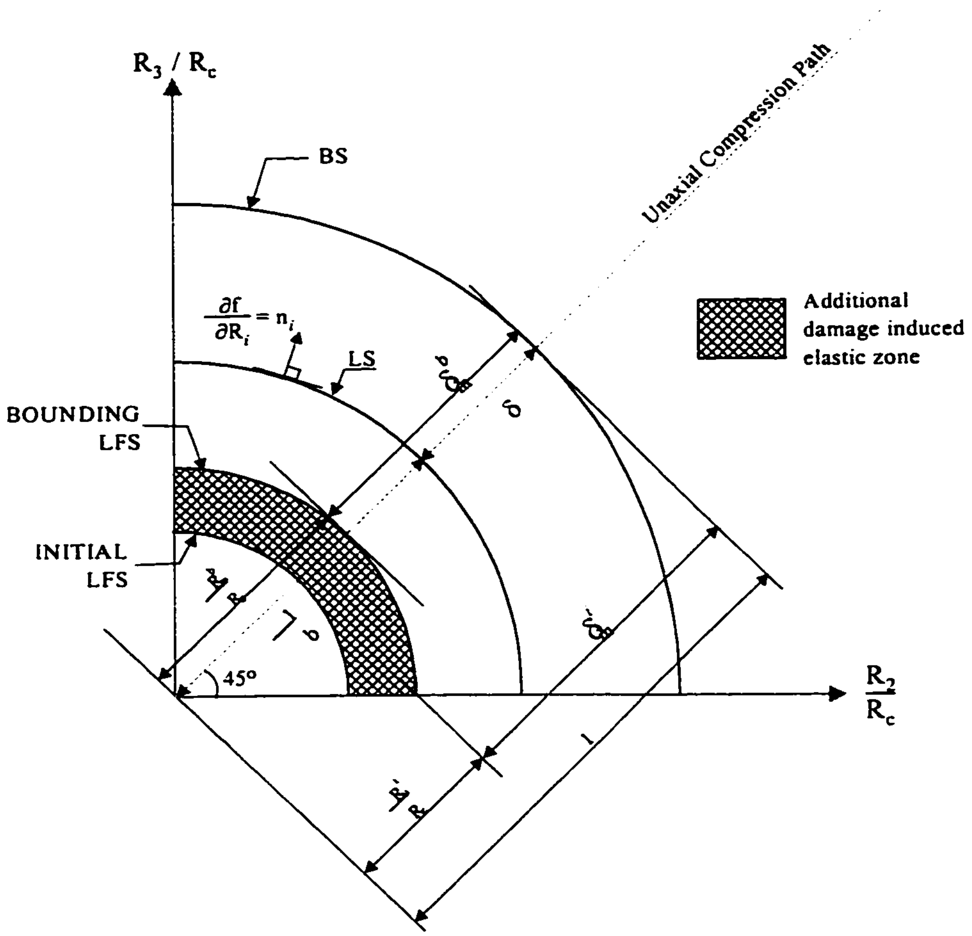

Figure 2.1 Limit Fracture, Loading and Bounding Surfaces.

Where D is a material parameter which needs to be calibrated in accordance with concrete Strength and < > are Maculay brackets that set the quantity within to zero if the argument is negative. The normalized distance δ between the loading and bounding surface is given by

$$\delta = 1 - \frac{1}{b} \tag{2.24}$$

The $\delta = \delta_{in}$ corresponds to R_o when the loading surface first crosses the limit fracture surface Fig. 2.1. Substitution of (2.21) and (2.22) into (2.20) and solving for $d\lambda$ then substituting the results into (2.19), yield

$$d\omega_k = \left[\frac{\frac{\partial f}{\partial R_J} \frac{\partial R_J}{\partial \sigma_S} d\sigma_S}{H - \frac{\partial f}{\partial R_n} \frac{\partial R_n}{\partial \omega_m} \frac{\partial f}{\partial R_m}} \right] \frac{\partial f}{\partial R_k} \tag{2.25}$$

Equation (2.25) is convenient for stress control. However for strain control; one may adopt the strain energy density for damaged material ρW defined as:

$$\rho W(\varepsilon_i, \omega_k) = \frac{1}{2} \{\sigma\}^T \{\varepsilon\} \tag{2.26}$$

where $\sigma = \sigma(\varepsilon_i, \omega_k)$

The constitutive equation defined by (2.6) may be inverted and expressed as:

$$\sigma_i = \overline{D}_{ij} \varepsilon_j \tag{2.27}$$

where $\overline{D}_{ij}$ are the components of the stiffness matrix defined by the inverse of the effective compliance matrix such that $\overline{D} = [\overline{C}]^{-1}$ using (2.27) into (2.26) yield

$$\rho W(\varepsilon_i,\omega_k)=\frac{1}{2}\{\varepsilon\}^T:\overline{D}:\{\varepsilon\} \tag{2.28}$$

The energy released rate vector R_i may now be expressed alternative to (2.14) as

$$R_j=-\rho\frac{\partial W(\varepsilon_{jk},\omega_k)}{\partial\omega_j} \tag{2.29}$$

whose increment may be written as

$$dR_i=\frac{\partial R_i}{\partial\omega_j}d\omega_j+\frac{\partial R_i}{\partial\varepsilon_l}d\varepsilon_l \tag{2.30}$$

Employing same procedure as that used for derivation of (2.25), except using equation (2.30) Instead of equation (2.21), leads to the increment of damage

$$d\omega_i=\left[\frac{\dfrac{\partial f}{\partial R_j}\dfrac{\partial R_j}{\partial\varepsilon_k}d\varepsilon_k}{H-\dfrac{\partial f}{\partial R_n}\dfrac{\partial R_n}{\partial\omega_m}\dfrac{\partial f}{\partial R_m}}\right]\frac{\partial f}{\partial R_i} \tag{2.31}$$

Which is convenient for the case of strain control.

2.1.3 CONSTITUTIVE ELASTO-DAMAGE EQUATIONS

The incremental form of the elasto-damage constitutive equation (2.6) may be expressed in indicial notation as:

$$d\varepsilon_i=\overline{C}_{ij}d\sigma_j+\sigma_l\frac{\partial\overline{C}_{il}}{\partial\omega_k}d\omega_k \tag{2.32}$$

which upon substitution of (2.25) yields,

$$d\varepsilon_i = \left[\overline{C}_{ij} + \sigma_l \frac{\partial \overline{C}_{il}}{\partial \omega_k} \left(\frac{\dfrac{\partial f}{\partial R_K} \dfrac{\partial f}{\partial R_S} \dfrac{\partial R_s}{\partial \sigma_j}}{H - \dfrac{\partial f}{\partial R_n} \dfrac{\partial R_n}{\partial \omega_m} \dfrac{\partial f}{\partial R_m}} \right) \right] d\sigma_j \tag{2.33}$$

The incremental form of equation (2.27) in indicial notation may be written as

$$d\sigma_i = \overline{D}_{ij} d\varepsilon_j + \varepsilon_l \frac{\partial \overline{D}_{il}}{\partial \omega_k} d\omega_k \tag{2.34}$$

which up on substitution of (2.31) results in

$$d\sigma_i = \left[\overline{D}_{ij} + \varepsilon_l \frac{\partial \overline{D}_{il}}{\partial \omega_k} \left(\frac{\dfrac{\partial f}{\partial R_k} \dfrac{\partial f}{\partial R_S} \dfrac{\partial R_s}{\partial \varepsilon_j}}{H - \dfrac{\partial f}{\partial R_n} \dfrac{\partial R_n}{\partial \omega_m} \dfrac{\partial f}{\partial R_m}} \right) \right] d\varepsilon_j \tag{2.35}$$

The term within the square parenthesis in equation (2.35) may be interpreted as being the elasto-damage stiffness D_{ij}^{ed}.

2.1.3.1 APPLICATIONS

2.1.3.1.1 UNIAXIAL COMPRESSION - STRESS CONTROL

For uniaxial compression, the Cauchy stress tensor reduces to a diagonal matrix or stress vector given by

$$\begin{bmatrix} -\sigma & 0 & 0 \end{bmatrix} \tag{2.36}$$

The complementary energy density $\rho\Lambda$ of equation (2.25) takes the following form in indicial or in matrix notation:

$$\rho\Lambda = \frac{1}{2}\sigma_i \overline{C}_{ij}\sigma_j = \frac{1}{2}\{\sigma\}^T\{\bar{c}\}\{\sigma\} \tag{2.37}$$

Substitution of equations (2.10) and (2.36) into (2.37) and setting $\alpha = 0$ (as it is used primarily as a parameter for matching peak strength in tension test), one obtains

$$\rho\Lambda = \frac{\sigma^2(1-\beta\omega_1)^2}{2E_o(1-\beta\omega_2)^2(1-\beta\omega_3)^2} \tag{2.38}$$

Differentiating (2.38) with respect to ω_i and substituting into (2.14), yields

$$R_1 = \frac{-\beta\sigma^2(1-\beta\omega_1)}{E_O(1-\beta\omega_2)^2(1-\beta\omega_3)^2} = 0 \qquad \text{(Since } R_i >= 0\text{)} \tag{2.39a}$$

$$R_2 = \frac{\beta\sigma^2(1-\beta\omega_1)^2}{E_O(1-\beta\omega_2)^3(1-\beta\omega_3)^2} \tag{2.39b}$$

$$R_3 = \frac{\beta\sigma^2(1-\beta\omega_1)^2}{E_O(1-\beta\omega_2)^2(1-\beta\omega_3)^3} \tag{2.39c}$$

From symmetry, $\omega_2 = \omega_3 = \omega$ and $\omega_1 = 0$ (by virtue of equation (2.39a)). Thus

$$R_2 = R_3 = \frac{\beta\sigma^2}{E_O(1-\beta\omega)^5} \tag{2.40}$$

And the loading surface of equation (2.11) becomes

$$f = \left[R_2^{\,2} + R_3^{\,2}\right]^{1/2} - \frac{R_c}{b} = 0 \tag{2.41}$$

Whose gradient may be expressed as

$$\frac{\partial f}{\partial R_i}=\left[0,\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right] \tag{2.42}$$

Differentiating R_j with respect to ω_i and σ_k and substituting the results along with (2.42) into (2.25) yields $d\omega_1=0$ *and* $d\omega_2=d\omega_3=d\omega$, given by

$$d\omega=\frac{2\sigma\beta d\sigma/E_o(1-\beta\omega)^5}{H-[5\beta^2\sigma^2/E_o(1-\beta\omega)^6]} \tag{2.43}$$

Differentiating $\overline{C}_{ij}$ of equation (2.10) with respect to ω_k and substituting the results along with (2.10) and (2.43) into (2.32), one obtains

$$d\varepsilon_1=\left\{\frac{1}{E_o(1-\beta\omega)^4}+\frac{8\beta^2\sigma^2/E_o^2(1-\beta\omega)^{10}}{H-\left[5\beta^2\sigma^2/E_o(1-\beta\omega)^6\right]}\right\}d\sigma \tag{2.44}$$

as the elasto-damage constitutive equation for uniaxial compression with the damage parameter ω being obtained by the accumulation of its increment defined in (2.43).

2.1.3.1.2 UNIAXIAL COMPRESSION - STRAIN CONTROL

For uniaxial compression, the strain energy density ρW for damaged material may be expressed from equation (2.26), (2.27), and (2.10) as

$$\rho W(\varepsilon_i,\omega_k)=\frac{1}{2}\sigma_i\varepsilon_i$$

$$=\frac{1}{2}\sigma_1\varepsilon_1=\frac{\varepsilon_1^2E_o(1-\beta\omega_2)^2(1-\beta\omega_3)^2}{2(1-\beta\omega_1)^2} \tag{2.45}$$

Note that for uniaxial $\sigma_2=\sigma_3=0$ but $\varepsilon_2=\varepsilon_3\neq 0$

Differentiating (2.45) with respect to ω_i and substituting into (2.29) yields

$$R_1 = 0 \tag{2.46a}$$

$$R_2 = \frac{\beta\varepsilon_1{}^2 E_o (1-\beta\omega_2)(1-\beta\omega_3)^2}{(1-\beta\omega_1)^2} \tag{2.46b}$$

$$R_3 = \frac{\beta\varepsilon_1{}^2 E_o (1-\beta\omega_2)^2(1-\beta\omega_3)}{(1-\beta\omega_1)^2} \tag{2.46c}$$

Again from symmetry $\omega_2 = \omega_3 = \omega$ and also $\omega_1 = 0$. Differentiating R_j with respect to ω_i and ε_k and substituting the results along with (2.42) into (2.31) yield $d\omega_1 = 0$ and $d\omega_2 = d\omega_3 = d\omega$ given by

$$d\omega = \frac{2\beta\varepsilon_1 E_o (1-\beta\omega)^3 d\varepsilon_1}{H + 3\beta^2\varepsilon_1{}^2 E_o (1-\beta\omega)^2} \tag{2.47}$$

Differentiating $\overline{D}_{ij}$ with respect to ω_k and substituting the results along with (2.47) into (2.34) yield

$$d\sigma = \left\{ E_o (1-\beta\omega)^4 - \frac{8\beta^2\varepsilon_1{}^2 E_o{}^2 (1-\beta\omega)^6}{H + 3\beta^2\varepsilon_1{}^2 E_o (1-\beta\omega)^2} \right\} d\varepsilon_1 \tag{2.48}$$

2.1.3.1.3 BIAXIAL COMPRESSION - STRESS CONTROL

$$\rho\Lambda = \frac{1}{2E_o}\begin{bmatrix} -\sigma_1 & -\sigma_2 & 0 \end{bmatrix}\left[\overline{C}\right]\begin{bmatrix} -\sigma_1 \\ -\sigma_2 \\ 0 \end{bmatrix} \tag{2.49}$$

Setting $\alpha = 0.0$ in the $\left[\overline{C}\right]$ matrix and substitute the result into (2.49), yields

$$\rho\Lambda = \frac{1}{2E_o}\left[\frac{(1-\beta\omega_1)^2\sigma_1^{\ 2}}{(1-\beta\omega_2)^2(1-\beta\omega_3)^2} - \frac{2\nu\sigma_1\sigma_2}{(1-\beta\omega_3)^2} + \frac{\sigma_2^{\ 2}(1-\beta\omega_2)^2}{(1-\beta\omega_3)^2(1-\beta\omega_1)^2}\right] \tag{2.50}$$

Also

$$R_i = \rho\frac{\partial\Lambda}{\partial\omega_i}\left(\sigma_{ij}, \omega_i\right)$$

yields

$$R_1 = \frac{1}{2E_o}\left[\frac{-2(1-\beta\omega_1)\sigma_1^{\ 2}(\beta)}{(1-\beta\omega_2)^2(1-\beta\omega_3)^2} + \frac{2\beta\sigma_2^{\ 2}(1-\beta\omega_2)^2}{(1-\beta\omega_3)^2(1-\beta\omega_1)^3}\right] \tag{2.51a}$$

$$R_2 = \frac{1}{2E_o}\left[\frac{2\beta(1-\beta\omega_1)^2\sigma_1^{\ 2}}{(1-\beta\omega_2)^3(1-\beta\omega_{3)})^2} - \frac{2\beta\sigma_2^{\ 2}(1-\beta\omega_2)}{(1-\beta\omega_3)^2(1-\beta\omega_{1)})^2}\right] \tag{2.51b}$$

$$R_3 = \frac{1}{2E_o}\left[\frac{2\beta(1-\beta\omega_1)^2\sigma_1^{\ 2}}{(1-\beta\omega_2)^2(1-\beta\omega_3)^3} - \frac{4\beta\nu\sigma_1\sigma_2}{(1-\beta\omega_3)^3} + \frac{2\beta\sigma_2^{\ 2}(1-\beta\omega_2)^2}{(1-\beta\omega_3)^3(1-\beta\omega_1)^2}\right] \tag{2.51c}$$

The evolution equation can be written in the following form:

$$d\omega_i = d\lambda\frac{\partial f}{\partial R_i} \tag{2.52}$$

And

$$d\lambda = \frac{A_2}{H - A_3} \tag{2.53}$$

Where:

$$A_2 = \frac{\partial f}{\partial R_j}\frac{\partial R_j}{\partial \sigma_k} d\sigma_k \tag{2.54}$$

$$A_3 = \frac{\partial f}{\partial R_j}\frac{\partial R_j}{\partial \omega_k}\frac{\partial f}{\partial R_k} \tag{2.55}$$

Appendix A contains all the above derivatives namely $\frac{\partial f}{\partial R_j}, \frac{\partial R_j}{\partial \sigma_k}$ and $\frac{\partial R_j}{\partial \omega_k}$.

The damage evolution equation of (2.52) can be evaluated first by computing A_2, A_3 and H and then substituting into (2.53) and (2.52).

The constitutive equation for stress control can be written as:

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{Bmatrix} = \frac{1}{E_o}\begin{bmatrix} \frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2(1-\beta\omega_3)^2} & \frac{-\nu}{(1-\beta\omega_3)^2} & \frac{-\nu}{(1-\beta\omega_2)^2} \\ \frac{-\nu}{(1-\beta\omega_3)^2} & \frac{(1-\beta\omega_2)^2}{(1-\beta\omega_3)^2(1-\beta\omega_1)^2} & \frac{-\nu}{(1-\beta\omega_1)^2} \\ \frac{-\nu}{(1-\beta\omega_2)^2} & \frac{-\nu}{(1-\beta\omega_1)^2} & \frac{(1-\beta\omega_3)^2}{(1-\beta\omega_1)^2(1-\beta\omega_2)^2} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ 0 \end{Bmatrix} \tag{2.56}$$

where

$$\varepsilon_1 = \frac{1}{E_o}\frac{(1-\beta\omega_1)^2\sigma_1}{(1-\beta\omega_2)^2(1-\beta\omega_3)^2} + \frac{-\nu\sigma_2}{E_o(1-\beta\omega_3)^2} \tag{2.57a}$$

$$\varepsilon_2 = \frac{1}{E_o}\frac{-\nu\sigma_1}{(1-\beta\omega_3)^2} + \frac{(1-\beta\omega_2)^2\sigma_2}{E_o(1-\beta\omega_3)^2(1-\beta\omega_1)^2} \tag{2.57b}$$

$$\varepsilon_3 = \frac{1}{E_o}\frac{-\nu\sigma_1}{(1-\beta\omega_2)^2} + \frac{-\nu\sigma_2}{E_o(1-\beta\omega_1)^2} \tag{2.57c}$$

The above three equations constitute the constitutive equation for stress control.

2.1.3.1.4 BIAXIAL COMPRESSION - STRAIN CONTROL

For strain control one inverts equations (2.57a) and (2.57b) which yields:

$$\sigma_1 = \left(\frac{E_o(1-\beta\omega_3)^2}{(1-\nu^2)}\right)\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2} + \varepsilon_2\nu\right) \tag{2.58a}$$

$$\sigma_2 = \frac{E_o(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\nu + \varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right) \tag{2.58b}$$

By substituting equations (2.58a) and (2.58b) into (2.51), one obtains the strain energy release components R_i in terms of strain components as:

$$R_1 = E_o\left[\frac{-(1-\beta\omega_1)(\beta)}{(1-\beta\omega_2)^2(1-\beta\omega_{3)})^2}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2} + \varepsilon_2\nu\right)\right]^2$$

$$+E_o\left[\frac{\beta(1-\beta\omega_2)^2}{(1-\beta\omega_3)^2(1-\beta\omega_1)^3}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\nu + \varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right]^2 \tag{2.59a}$$

$$R_2 = E_o\left[\frac{\beta(1-\beta\omega_1)^2}{(1-\beta\omega_2)^3(1-\beta\omega_{3)})^2}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2} + \varepsilon_2\nu\right)\right]^2$$

$$-E_o\left[\frac{\beta(1-\beta\omega_2)}{(1-\beta\omega_3)^2(1-\beta\omega_{1)})^2}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right]^2 \tag{2.59b}$$

$$R_3=E_o\left[\frac{\beta(1-\beta\omega_1)^2\sigma_1{}^2}{(1-\beta\omega_2)^2(1-\beta\omega_{3)})^3}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}+\varepsilon_2\nu\right)\right]^2$$

$$-E_o\left[\frac{2\beta\nu\sigma_1\sigma_2}{(1-\beta\omega_{3)})^3}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}+\varepsilon_2\nu\right)\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right]$$

$$+E_o\left[\frac{\beta(1-\beta\omega_2)^2\sigma_2{}^2}{(1-\beta\omega_3)^3(1-\beta\omega_{1)})^2}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right]^2 \tag{2.59c}$$

The evolution equation can be written as in (2.52) where

$$d\omega_i=d\lambda\frac{\partial f}{\partial R_i} \tag{2.60}$$

and

$$d\lambda=\frac{A_1}{H-A_3} \tag{2.61}$$

where:

$$A_1=\frac{\partial f}{\partial R_j}\frac{\partial R_j}{\partial\varepsilon_k}d\varepsilon_k \tag{2.62}$$

$$A_3=\frac{\partial f}{\partial R_j}\frac{\partial R_j}{\partial\omega_k}\frac{\partial f}{\partial R_k} \tag{2.63}$$

Appendix A contains all the above derivatives namely $\frac{\partial f}{\partial R_j}, \frac{\partial R_j}{\partial\varepsilon_k}$ and $\frac{\partial R_j}{\partial\omega_k}$.

Thus damage evolution equation (2.52) can be evaluated first by computing A_1, A_3 and H and then substitution into equation (2.60) & (2.61).

2.1.4 CALIBRATION OF MODEL PARAMETERS

The developed elasto-damage incremental laws given by equations (2.33) and (2.35) and the associated damage evolution described by equations (2.25) and (2.31) reflect a general form, valid for any monotonic loading state. Certain standard tests are utilized to calibrate the model.

In the model predicting monotonic response, there are basically five parameters that need to be calibrated using experimental data. These parameters are R_o^i (or δ_{in}), α, β, R_c and D.

The parameter R_o^i defines the initiation of microcracking which occurs at about 40% of the peak stress under uniaxial compressive loading for concrete with f_c'=5600 Psi [4]. This initiation of damage has been noted to vary with the compressive strength [21], where the damage threshold has been noted to be almost 60% of the peak stress for concrete with f_c' = 11,000 psi. For concrete strength in the range of $3000 < f_c' < 7000$ psi, a median value of R_o^i = 0.08 in-lb/in^3 was computed; whereas for the range $7000 < f_c' < 9700$ psi, R_o^i was fixed at 0.16 in-lb/in^3. Once R_o^i is determined then the corresponding δ_{in} can be obtained from equations (2.16) and (2.24).

The α parameter is used primarily for matching peak strength of concrete in direct tension, and which has been reported in Ref. [17]. It has been found that α lies in the range

$1.25 < \alpha < 6.6$ for a compressive strength variation from 4000 psi to 17400 psi. Since the model is assessed through compressive fatigue life of concrete, the value of α is immaterial and is outside the scope of the work presented herein.

The most important and critical model parameter is β which controls the damage growth rate and influences the pre-peak behavior as well as the level at which the peak stress is attained. Thus the behavior of concrete of varying compressive strength is simulated by different values of β. The variation of β (referred to as β_{st} for monotonic loading) as a function of concrete strength f_c' is shown in Fig. 2.2, where the higher the concrete strength, the lower is the value of β. This variation guarantees a slower accumulation of damage in order to attain levels of stress commensurate with increase in concrete strength.

The parameter R_c is the critical energy release rate and is the magnitude of the energy release rate vector R_i when the loading surface $f = 0$ reaches the bounding surface $F = 0$. Just as R_o is a function of concrete strength, R_c must also vary with f_c'. However, the introduction of the scaling parameters α and β obviates this requirement, and R_c is chosen to be fixed at 1.29 in-lb/in^3.

The parameter D controls the softening phase of material response in σ-ε space. In order to simulate sharper softening gradients as depicted by concretes of increasing brittleness and higher strength [22], the variation of D with concrete strength was adopted as shown in Fig. 2.2.

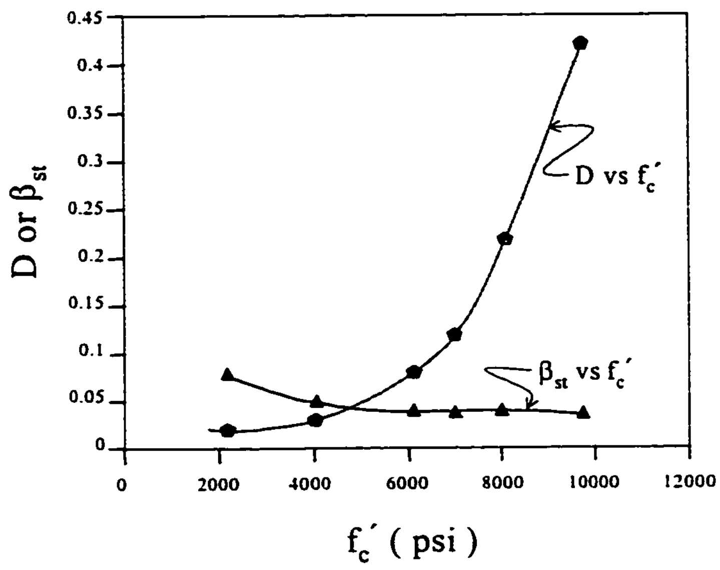

Figure 2.2 Variation of Model Parameter β_{st} and D with Compressive Strength f_c'

2.1.5 MODEL RESPONSE UNDER MONOTONIC LOADING

In order to simulate the response under monotonic loading, the strain control model of section 2.1.3.1.1 was coded into a Fortran program. Input for four model parameters (R_o , R_c, β and D) was provided as a function of concrete strength.

In as much as the model presented is elasto-damage, there is no account for any residual or plastic strain. In the σ-ε response, a plastic strain component ε^p, assumed proportional to the damage strain ε^d with constant of proportionality to be 1.5 (i.e. $\varepsilon^p = 1.5\ \varepsilon^d$) has been superposed as proposed in Ref [4]. The total strain is thus calculated as $\varepsilon = \varepsilon^{ed} + 1.5\ \varepsilon^d$, where $\varepsilon^{ed} = \varepsilon^e + \varepsilon^d$, the sum of elastic and damage strain components.

2.2 CYCLIC LOADING

The distinguishing feature of cyclic loading from monotonic loading is a rational updating of the threshold of damage with increasing number of stress cycles. For monotonic loading, the threshold of damage is identified by the limit fracture surface $f_o = 0$ and whose size is $R_o = R_o^i$ Fig. 2.1. However, for cyclic loading R_o is hypothesized to change and increase with each successive cycle and is represented as $R_o = R_o(\overline{\omega})$ where $\overline{\omega} = (\omega_i \omega_i)^{\frac{1}{2}}$ is the magnitude of the damage vector ω_i. Accordingly, a higher stress level is needed for subsequent cycles to cause extra damage beyond whatever damage was accumulated and this is done by allowing R_o to move and expand isotropically in the R-space leading to degradation of mechanical properties such as the modulus of elasticity and an overall increase in the flexibility.

2.2.1 MOVEMENT OF LIMIT FRACTURE SURFACE f_o UNDER CYCLIC LOADING

The limit fracture surface is the surface beyond which the material behaves inelastically due to initiation or propagation of crack damage. The size of the surface R_o is a function of the amount of the accumulated damage. Different functional forms describing the movement of the surface $f_o = 0$ were considered and that of an elliptical form was found to successfully predict the experimental results for cyclic loading in compression [20]. The form of the surface in $R_o - \overline{\omega}$ space may be expressed as:

$$\frac{(R_o - R_o^{\,i})^2}{(R_o^{\,b} - R_o^{\,i})^2} + \frac{(\overline{\omega} - \overline{\omega}_b)^2}{(\overline{\omega}_i - \overline{\omega}_b)^2} = 1 \tag{2.63}$$

The parameters $R_o^{\,i}$ and $\overline{\omega}_i$ correspond to the initial size of the limit fracture surface and the associated damage, respectively with $R_o^{\,b}$ and $\overline{\omega}_b$ corresponding to the bound or the limiting size of the limit fracture surface and the associated damage, respectively shown in Fig. 2.3.

The function in equation (2.63) represents an ellipse in $R_o - \overline{\omega}$ space, where R_o grows monotonically as damage $\overline{\omega}$ increases until it reaches a limiting value R_o^b, which is the size of the bounding limit fracture surface Fig 2.1. At this stage the damage will reach $\overline{\omega}_b$ as shown in Fig 2.3. It is emphasized that the limit fracture surface may reach its bounding surface whilst the loading surface $f = 0$ may still be remote from its own conjugate bounding surface $F = 0$. Consequently, further damage is deemed to occur at a fixed size of limit fracture surface R_o^b until damage reaches its limiting value $\overline{\omega}_m$ and the loading surface $f = 0$ reaches the bounding surface $F = 0$, defining incipient failure.

The experimental results of Suaris et al. [4] indicate that crack initiation in compression occurs at about 40% of the peak stress for the particular case of concrete strength $f_c' = 5600$ psi, assuming an inherent initial damage $\omega_i = 0.05$. The strain energy release rate components corresponding to this initial damage and stress level are used on equation (2.13) in order to determine the initial size of the limit fracture surface $R_o^{\,i}$. However, the initial size

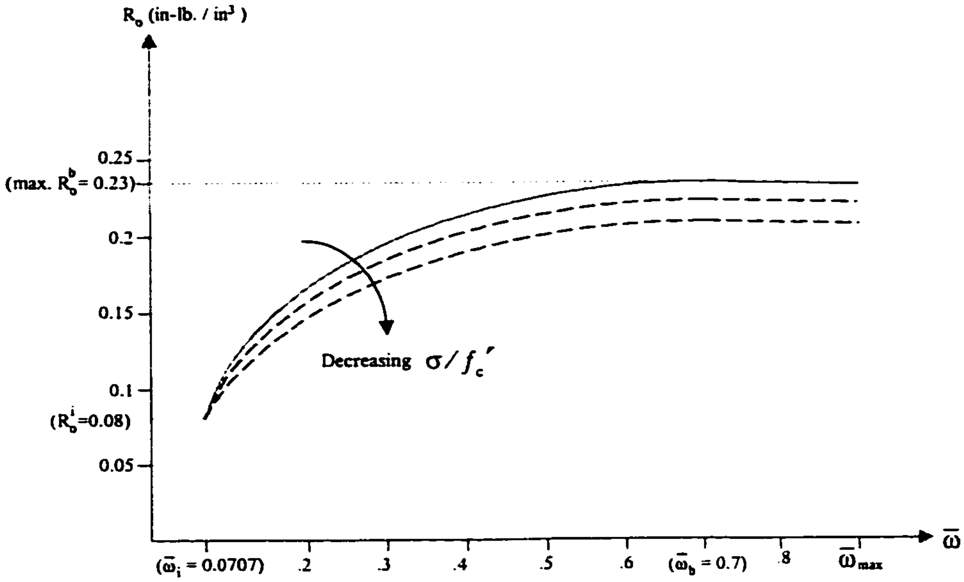

Figure 2.3 Size Evolution of Limit Fracture Surface in R_o - $\bar{\omega}$ Space

$R_o{}^i$ tends to increase with concrete strength as damage is initiated at higher levels of threshold stress [21].

Using $\overline{\omega}_i = \sqrt{2}\omega_i = 0.0707$ into equation (2.63) yields the size of the limit fracture surface R_o as a function of $\overline{\omega}$

$$R_o = R^i{}_o + (R_o{}^b - R^i{}_o)\sqrt{1 - \frac{(\overline{\omega} - \overline{\omega}_b)^2}{(0.0707 - \overline{\omega}_b)^2}} \qquad (2.64)$$

Equation (2.64) is a two-parameter model in terms of $R_o{}^b$ and $\overline{\omega}_b$ for describing the size of the limit fracture surface in $R_o - \overline{\omega}$ space. These two parameters remain to be calibrated in accordance with phenomenological data available from cyclic loading of concrete in uniaxial and biaxial compression.

2.3 RESIDUAL STRENGTH

Residual strength Is defined as the residual or remaining strength of a structural component which has been damaged due to the application of a finite number of load cycles less than the fatigue life. This is determinate by applying a monotonic load to failure to the structural component, which has inherent damage participated due to the presented applied cyclic loading. Fig. 2.4 shows the movement of surface in $\sigma - \varepsilon$ space for residual strength determination.

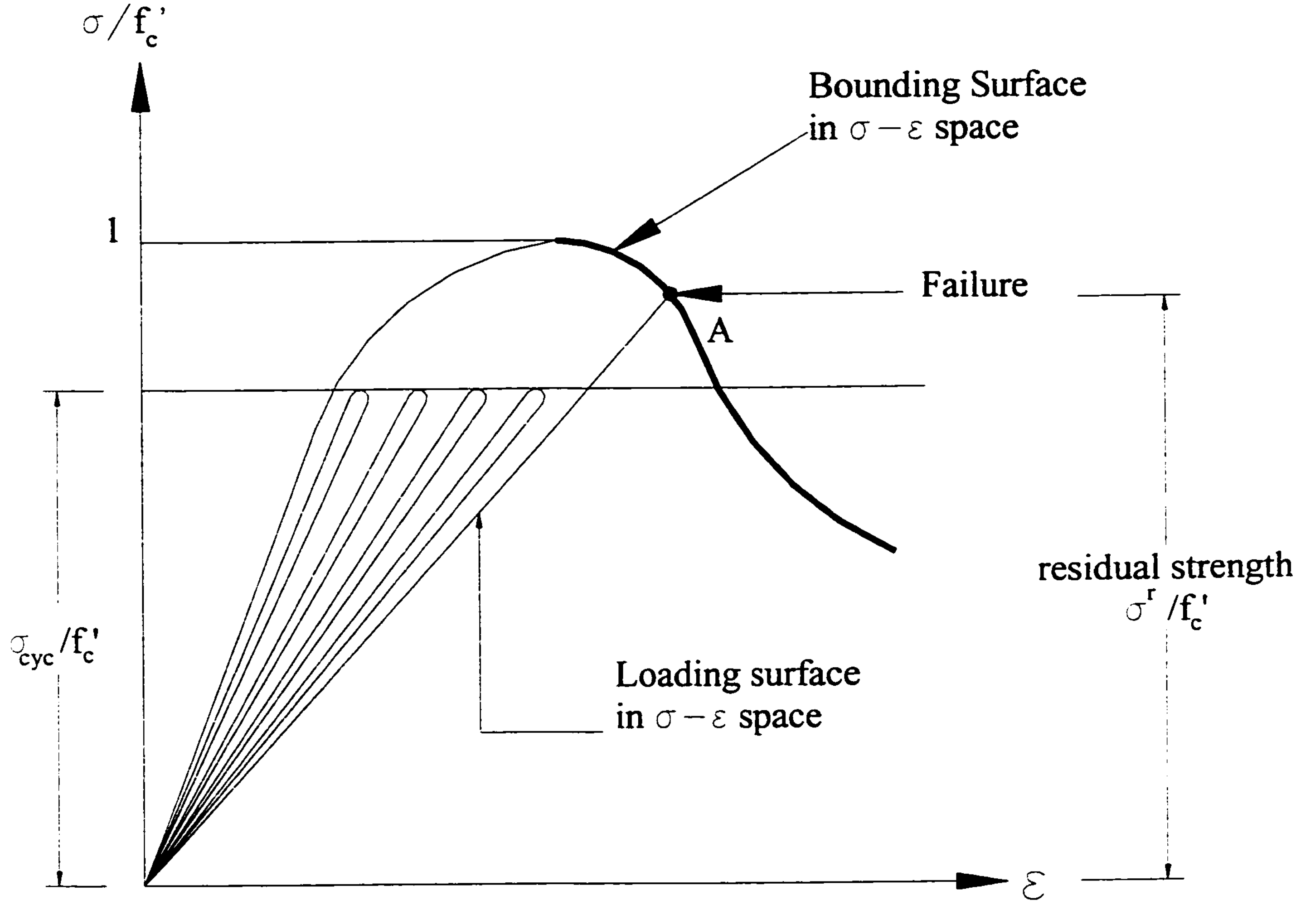

Figure 2.4 Movement of Loading Surface in σ–ε Space for Residual Strength Determination

CHAPTER 3

DEVELOPMENT OF COMPUTER SOFTWARE

This chapter presents a detailed discussion regarding the development of the computer software discussed in chapter 2. These programs are written in FORTRAN code. In the following sections, the description of the programs is presented.

3.1 UNIAXIAL MONOTONIC LOAD PROGRAM

3.1.1 TYPE OF THE PROGRAM

This program is strain control program. Strain control program depends on the increment of the strain to calculate the damage and then get the equivalent stress. Strain control program is used to calculate the damage in monotonic loading model. The stress strain curve that results from strain control testing exhibits the softening part as well as the peak stress which represent the strength of the concrete.

3.1.2 SOLUTION ALGORITHM

The flowchart in Fig. 3.1 consists of two main loops: the inner loop accounts for computation of damage increment based on the values of damage and strain energy release rate of previous increment as an initial guess. The iterative procedure is implemented until convergence is attained in terms of a constant set ω_i-R_i, followed by updating of damage and

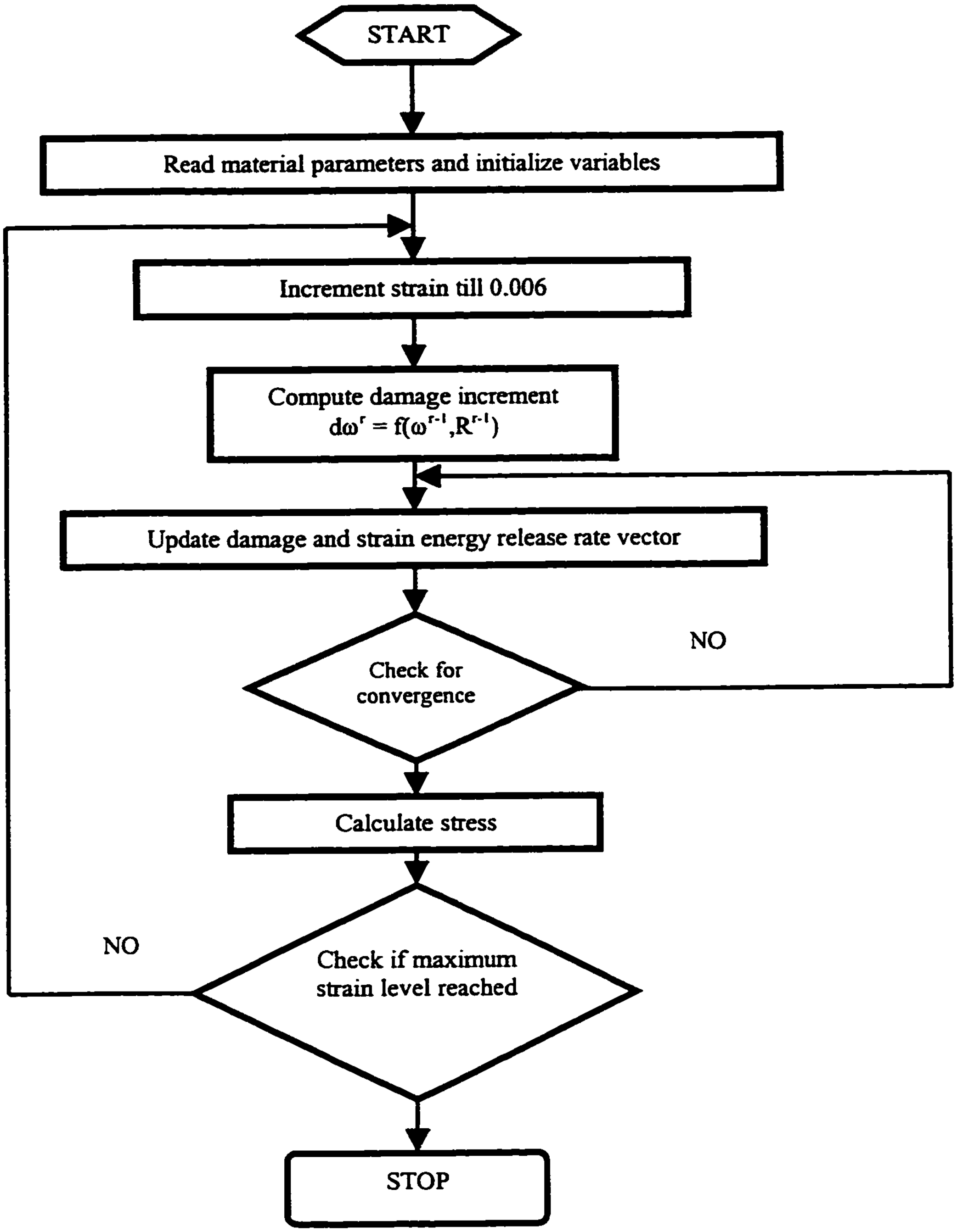

Figure 3.1 Flow Chart for Monotonic Loading Program

stress values. The outer loop is related to the incrementation of strain from zero to a maximum specified value of 0.006.

Two monotonic load programs have been developed. The first one predicts the concrete stress-strain curve corresponding to a specific maximum strength. The second calibrates the parameter β_{dyn} for uniaxial cyclic load model to yield a peak stress, which is 17% greater than the compression strength obtained from the static load test [1].

3.2 CYCLIC LOAD PROGRAM

3.2.1 TYPE OF THE PROGRAM

This program is stress control program. Stress control program uses increment of stress as an input, followed by computation for corresponding damage and the equivalent strain in cyclic loading model. This software is developed to predict the required number of cycles to failure under a concrete stress less than the concrete strength.

3.2.2 SOLUTION ALGORITHM

The general problem of response of elasto-damage material to a prescribed loading is highly nonlinear. The use of incremental form for description of state variables such as damage ω_i is necessary in order to describe their evolution. A Fortran 77-program code has been written, the flow chart of which is shown in Fig. 3.2. The code computes cumulative damage as the number of cycles N is increased.

The flow chart consists of three main do -loops: the innermost loop accounts for computation of damage increment based on the values of damage and strain energy release rate of previous increment as an initial guess. The iterative procedure is implemented until convergence is attained in terms of a constant set $\omega_i - R_i$. The intermediate loop is related to the incrementation of stress starting from zero until the prescribed stress level is reached followed by unloading to the origin; the outermost do-loop monitors the movement of the loading surface as it approaches the bounding surface. The latter effect is shown symbolically in σ-ε space in Fig. 3.3.

3.2.3 Calibration of R_o^b for cyclic loading

For uniaxial cyclic loading the evolution of the limit fracture surface has been defined in terms of the parameters R_o^b and $\overline{\omega_b}$. In general these parameters are noted to be functions of the maximum amplitude of cycling load σ / f_c' . For the range of experimental data presented in [20], for concrete with $f_c' = 6100$ psi it was noted that (assuming $\overline{\omega}_b = 0.7 =$ constant for simplicity)

$$R_0^{\,b} = 0.356\left(\frac{\sigma}{f_c'}\right) - 0.069 \qquad 0.6 \le \left(\frac{\sigma}{f_c'}\right) < 0.92 \tag{3.1a}$$

$$R_0^{\,b} = 0.25 \qquad \left(\frac{\sigma}{f_c'}\right) \ge 0.92 \tag{3.1b}$$

Fig 3.4 shows the relation between σ/f_c' and R_o^b.

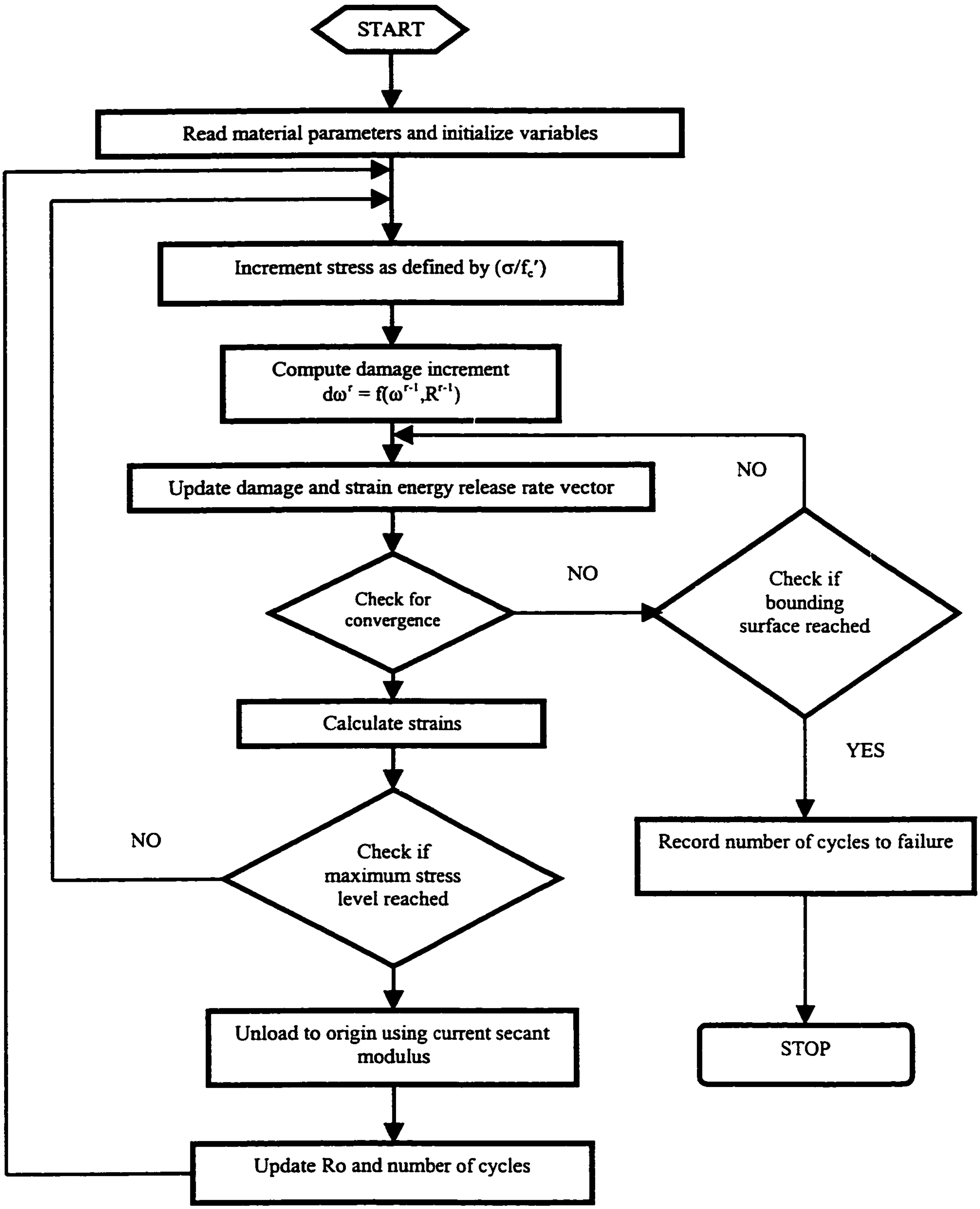

Figure 3.2 Flow Chart for Cyclic Loading Program

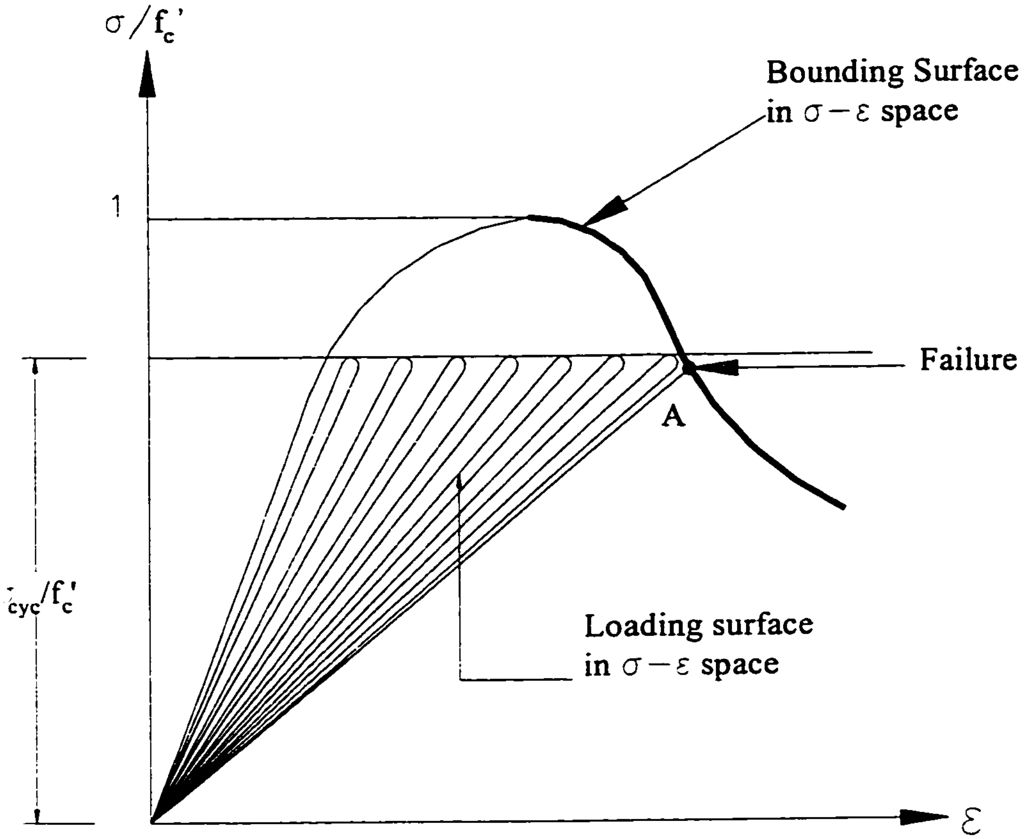

Figure 3.3 Movement of Loading Surface in σ-ε Space for Cyclic Load Determination

For biaxial cyclic loading the evolution of the limit fracture surface depends also on the ratio σ_1/σ_2 where σ_1 is the confining stress. This parameter was obtained for a range of data presented in [1] and for the same f_c' that was considered in the uniaxial cyclic loading case. The relation between σ/f_c' and R_o^b was determined as:

I. Case σ1/σ2 =0.2

$$R_0^b = 0.348\left(\frac{\sigma}{f_c'}\right) - 0.112 \qquad 0.8 \le \left(\frac{\sigma}{f_c'}\right) < 1.20 \tag{3.2a}$$

$$R_0^b = 0.27 \qquad \left(\frac{\sigma}{f_c'}\right) \ge 1.20 \tag{3.2b}$$

II. Case σ1/σ2 =0.5

$$R_0^b = 0.44\left(\frac{\sigma}{f_c'}\right) - 0.185 \qquad 0.8 \le \left(\frac{\sigma}{f_c'}\right) < 1.20 \tag{3.3a}$$

$$R_0^b = 0.318 \qquad \left(\frac{\sigma}{f_c'}\right) \ge 1.20 \tag{3.3b}$$

III. Case σ1/σ2 =1.0

$$R_0^b = 0.340\left(\frac{\sigma}{f_c'}\right) - 0.142 \qquad 0.7 \le \left(\frac{\sigma}{f_c'}\right) < 1.10 \tag{3.4a}$$

$$R_0^b = 0.28 \qquad \left(\frac{\sigma}{f_c'}\right) \ge 1.10 \tag{3.4b}$$

Figs 3.5 to 3.7 show the relation between σ/f_c' and R_o^b for the three cases of biaxial loading.

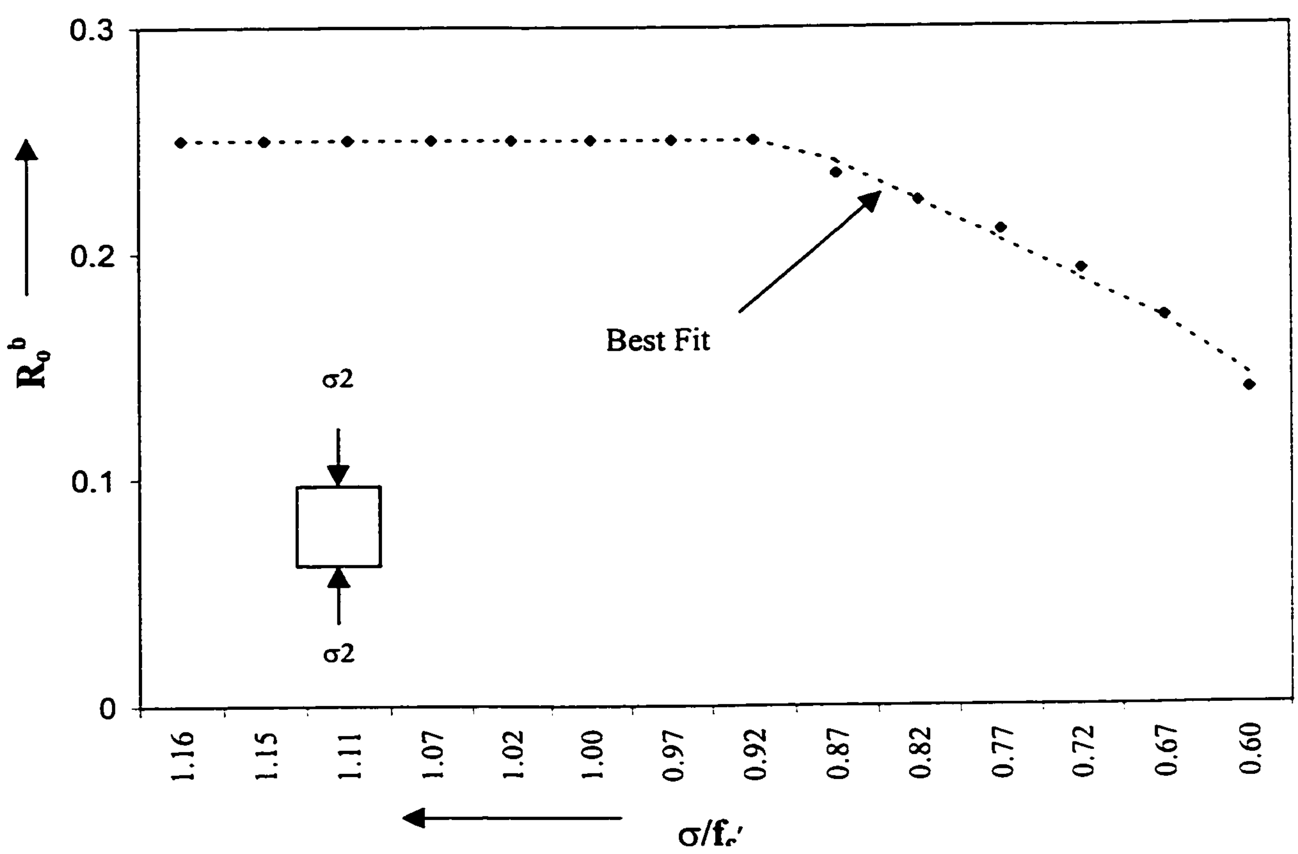

Figure 3.4 Relation Between σ/f_c' and R_o^b
for $\sigma_1/\sigma_2 = 0.0$ (Uniaxial Compression)

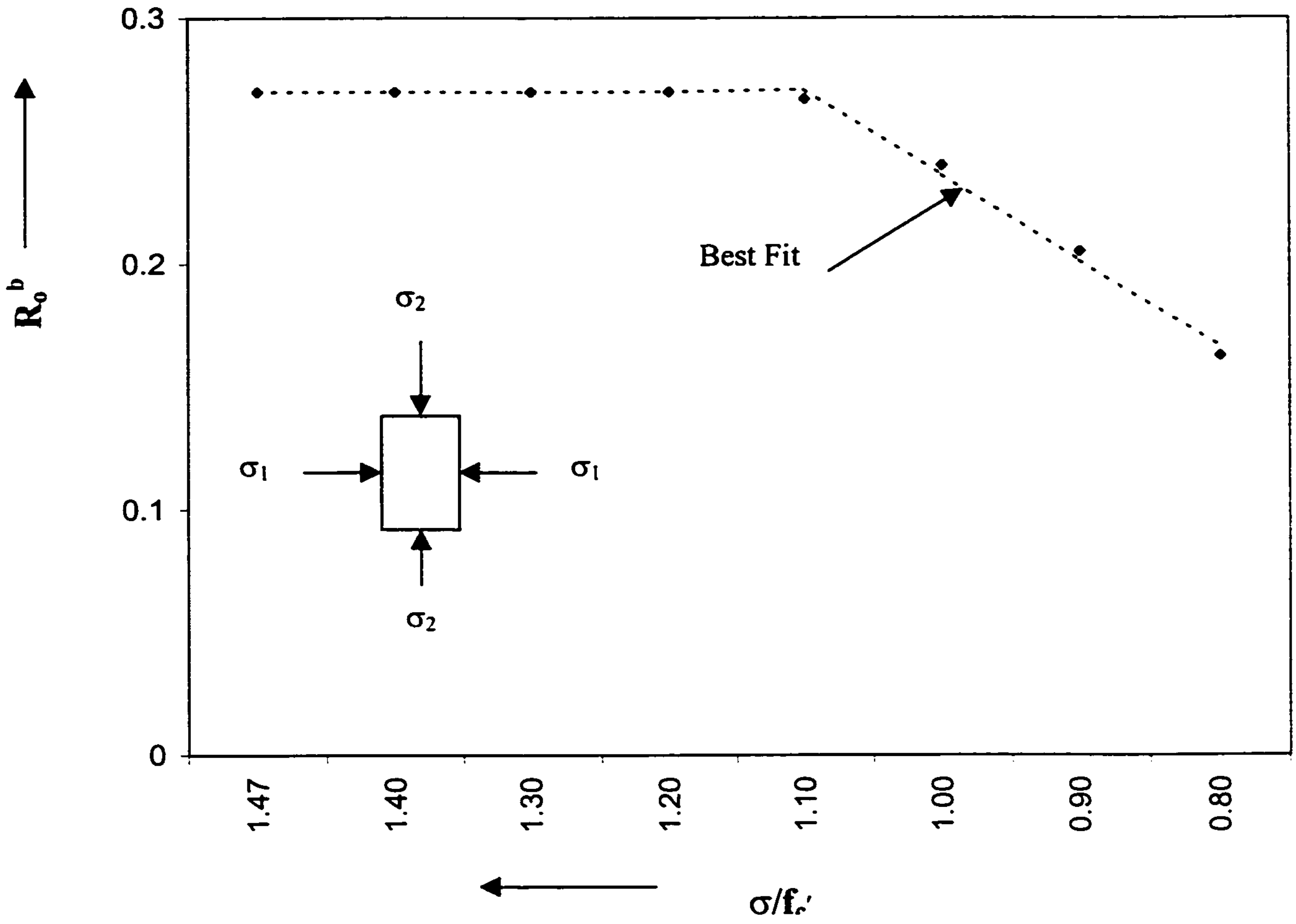

Figure 3.5 Relation Between σ/f_c' and R_o^b for $\sigma_1/\sigma_2 = 0.2$

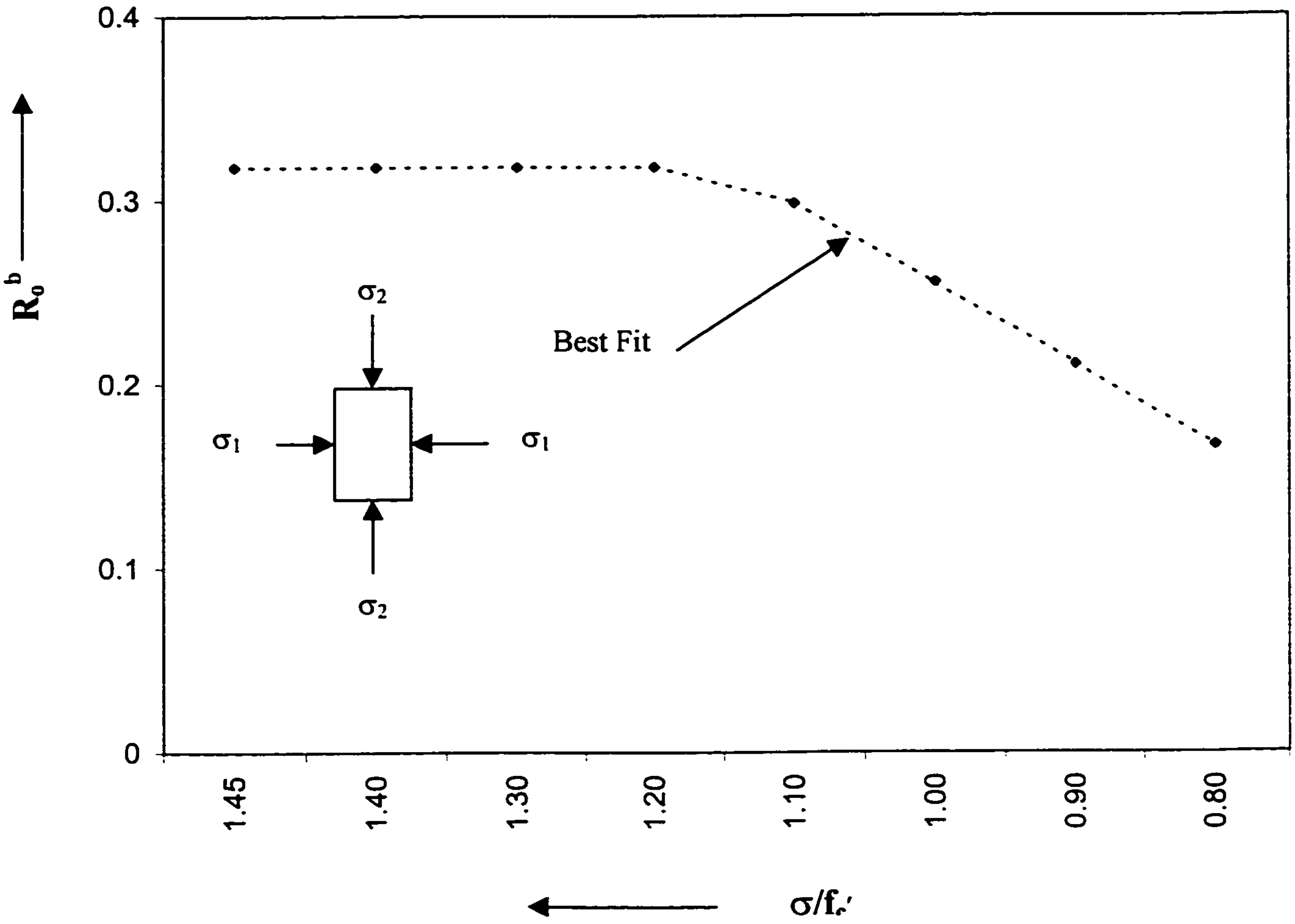

Figure 3.6 Relation Between σ/f_c' and R_o^b for $\sigma_1/\sigma_2 = 0.5$

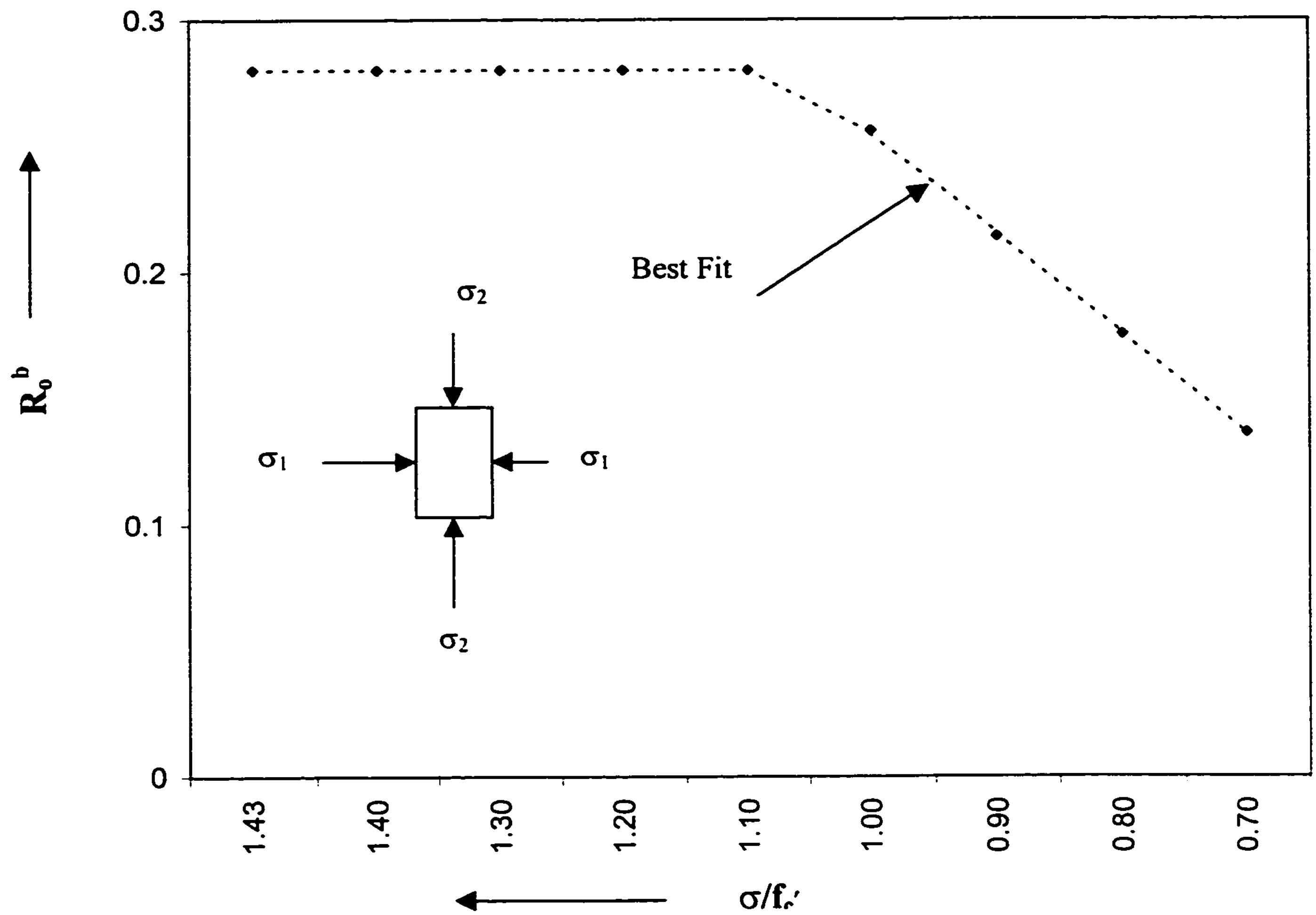

Figure 3.7 Relation Between σ/f_c' and R_o^b for $\sigma_1/\sigma_2 = 1.0$

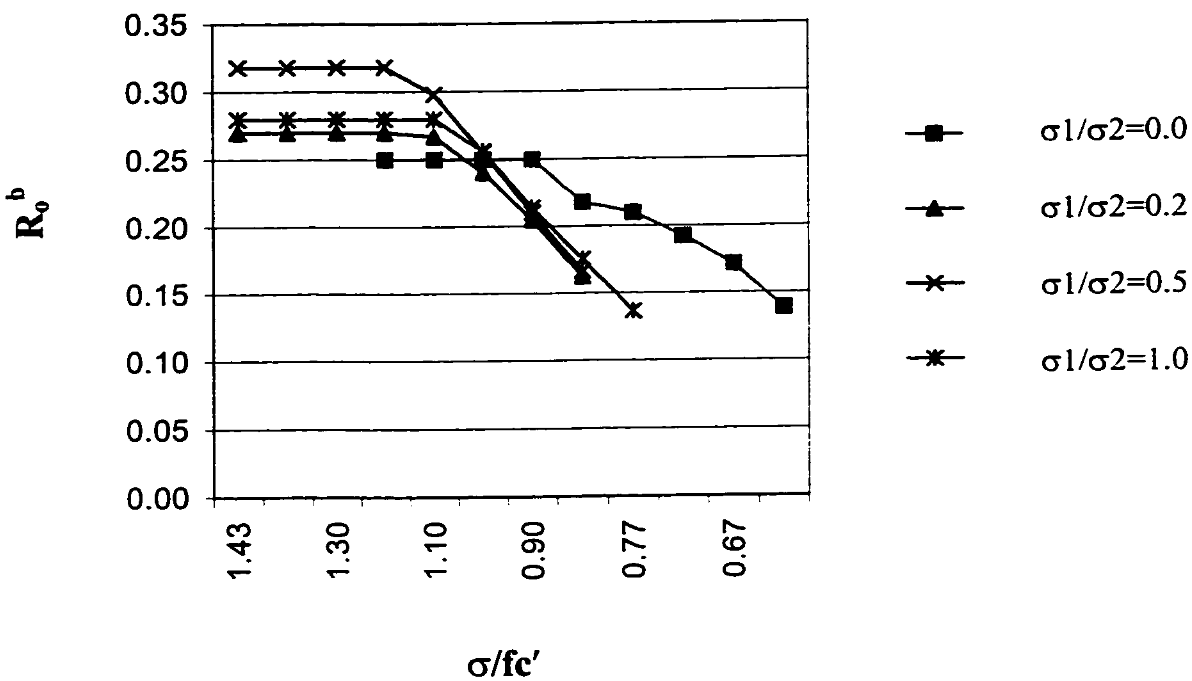

Figure 3.8 Relation Between σ/f_c' and R_o^b

The movement of the limit fracture surface is noted to a function of σ/f_c'. The maximum size of the elastic core as R_o^b is constrained so as to decrease with decreasing amplitude of peak stress, resulting in a corresponding increase in the damage growth zone that allows for greater accumulation of damage. This transition allows for failure to occur at a number of cycles commensurate with experimental findings when cycling at lower σ/f_c'.

The elastic core evolution also implies that the system is rendered more elastic or flexible when cycled at higher σ/f_c'.

3.3 UNIAXIAL RESIDUAL STRENGTH PROGRAM

3.3.1 TYPE OF THE PROGRAM

This program is a stress control program. Stress control program is driven by an input of increment of the stress followed by computation of damage and equivalent strain. In this specific program, the residual strength of an element is determined by applying a monotonic load to failure to the component which has inherent damage induced by previous application of a finite number of stress cycles, less than the fatigue life, at a given stress level, σ/f_c'.

3.3.2 SOLUTION ALGORITHM

The flow chart in Fig 3.9 consists of two steps

1. Three main do-loops: the innermost loop accounts for computation of damage increment based on the values of damage and strain energy release rate of previous increment as an initial guess. The iterative procedure is implemented until

convergence is attained in terms of a constant set $\omega_i - R_i$. The intermediate loop is related to the incrementation of stress starting from zero until the prescribed stress level is reached followed by unloading to the origin; the outermost do-loop monitors the movement of the loading surface as it approaches the bounding surface for a finite number of cycles which is less than the fatigue life. Since the number of cycles is less than the fatigue life, the loading surface will not reach the bounding surface.

2. Changing the main model parameters from cyclic load to monotonic load parameters followed by loading to failure. The residual strength is the maximum stress level attained under the monotonic load.

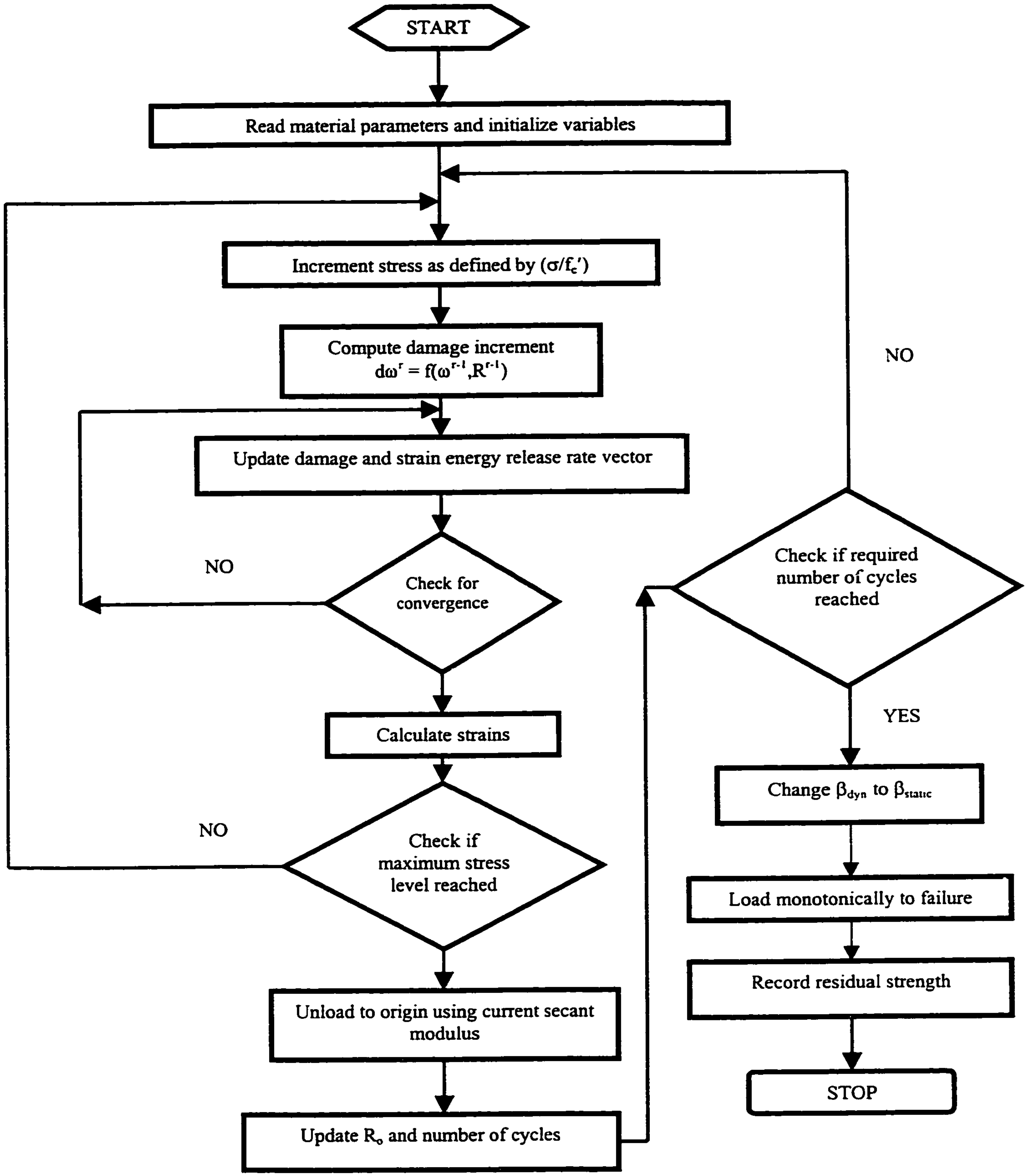

Figure 3.9 Flow Chart for Residual Strength

CHAPTER 4

RESULTS

In order to verify the based continuum damage mechanics model for prediction of response of brittle materials under monotonic and cyclic loading, results in terms of stress-strain response and number of cycles to failure for a specified σ/f_c' are compared with the reported data in the literature. The analytical results obtained from the model are based on the Fortran programs described earlier in chapter 3. These programs are:

1- STCUNMN is a static uniaxial monotonic loading program for uniaxial compression case. It is employed to predict the monotonic behavior of concrete under compressive loading to obtain the stress-strain response for a specified concrete strength f_c'.

2- DYNUNMN is a dynamic uniaxial monotonic loading case program to calibrate the model parameter β_{dyn} for uniaxial compression. It is utilized to predict the response of concrete that fails due to one cycle of cyclic loading.

3- UNICYC and BIACYC are uniaxial cyclic and biaxial cyclic compression loading programs, respectively. These programs have been used to predict the S-N curves for concrete subjected to uniaxial and biaxial compressive cyclic loading.

4- RSDSTRN is a residual strength determination program that is utilized to predict the residual capacity of concrete in uniaxial compression after being subjected to a number of stress cycles less than the fatigue life.

4.1 MONOTONIC LOAD

"STCUNMN" PROGRAM

The main feature of the STCUNMN program is the prediction of stress-strain response of concrete. Different concrete grades have been used. The different parameters calibrated inside the program are listed in Table 4.1. Experimental stress-strain response was available for uniaxial compressive loading from Ref. [22]. Stress-strain curves presented in Fig. 4.1 show the stress-strain behavior for different grades of concrete strength as reported from experimental data and as predicted from the model. The model parameter D has been calibrated in order to predict the strain softening part, where β_{st} & β_{dyn} control primarily the peak strength value. R_o and R_c control the size of the initial limit fracture surface and the bounding surface, respectively.

"DYNUNMN" PROGRAM

In order to analytically predict the difference between the observed peak strength of concrete subjected to static uniaxial compressive test and under cyclic load of one cycle to failure [1], this program has been developed. From this program, a value of β_{dyn} has been calculated for different concrete grades. Table 4.1 and Fig. 4.2 show the difference between β_{st} and β_{dyn}.

It may be observed that β_{dyn} is always less than β_{st} in order for the peak strength of the "one cycle to failure test" be greater than the peak strength as observed from a static monotonic load to failure

f_c' (Psi)	D	β_{st}	β_{dyn}	R_o (in-lb/in^3)	R_c (in-lb/in^3)
2200	0.02	0.0756	0.0605	0.04	1.29
4000	0.03	0.04745	0.03767	0.08	1.29
6100	0.08	0.03855	0.03053	0.08	1.29
7000	0.12	0.03665	0.02895	0.08	1.29
8000	0.22	0.03812	0.02975	0.17	1.29
9700	0.42	0.034475	0.0267	0.17	1.29

Table 4.1 Parameters Calibrated in STCUNMN and DYNUNMN

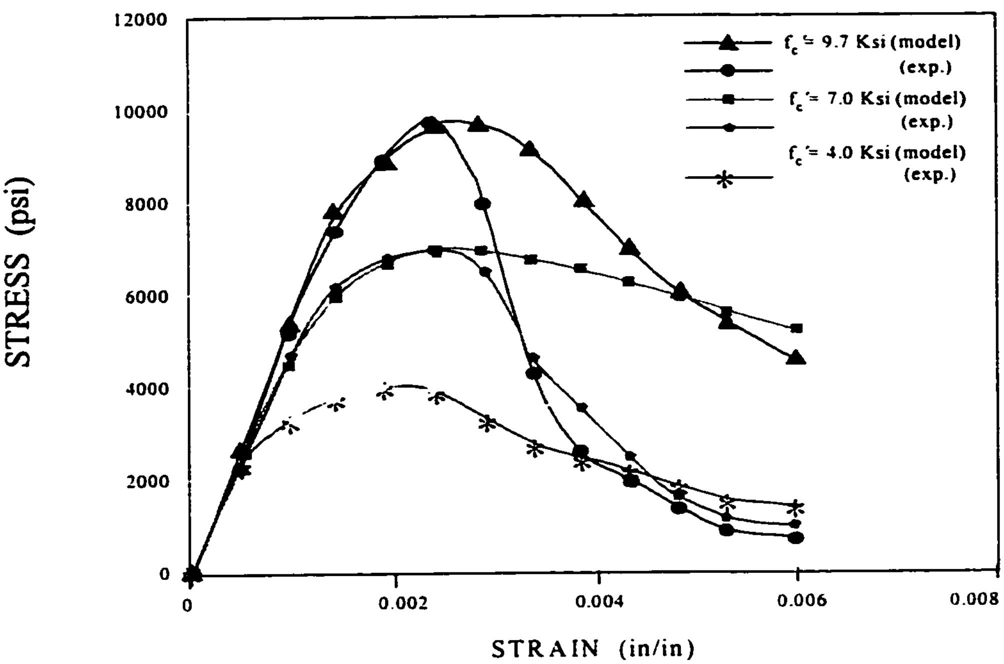

Figure 4.1 Model Response in Monotonic, Uniaxial Compression

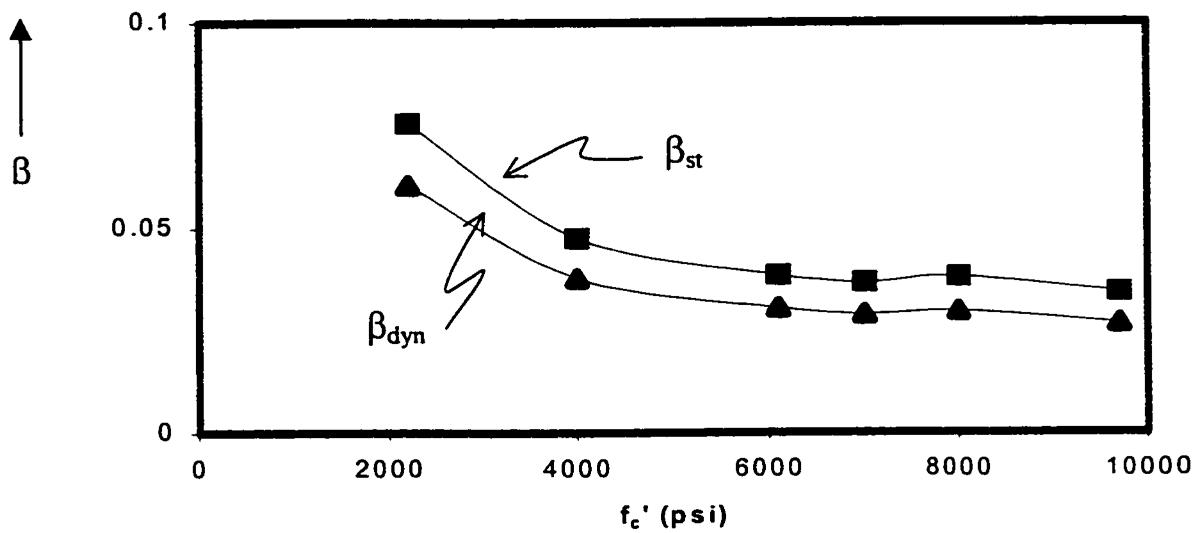

Figure 4.2 Relation Between f_c' and β_{st} & β_{dyn}

4.2 CYCLIC LOAD

"UNICYC" PROGRAMS

The S-N curves for cyclic loading of concrete under various conditions of biaxiality as defined by the ratio σ_1/σ_2 have been determined experimentally Ref. [1]. From these curves, it has been noted that strength of concrete is 17 percent greater than for the case of "one cycle reach to failure test" in comparison to the fatigue test. In order to simulate this phenomenon in the analytical model, a dynamic β_{dyn} in contrast to β_{st} has been introduced to modify the peak of the stress-strain curve.

Table 4.2 and Fig. 4.3 show a comparison of experimental and the analytical results for concrete of compressive strength of 6100 psi under cyclic loading.

"BIXCYC" PROGRAMS

Tables 4.3 to 4.5 and Figs. 5.4 to 5.6 show the difference between experimental and analytical results for a concrete with f_c' 6100 psi under biaxial load with σ_1/σ_2 equal to 0.2, 0.5 and 1.0, respectively. It is noted that the results obtained from the analytical model approximate the experimental results with reasonable accuracy. It is interested to note that for all degree of biaxiality, there is a significant increase in the fatigue load comparing to the "one cycle to failure test" due to the effect of confinement.

4.3 RESIDUAL STRENGTH

The residual strength is the residual capacity of concrete in uniaxial compression after being subjected to a number of stress cycles less than the fatigue life. The uniaxial cyclic load program has been modified to calculate the residual strength of concrete with fc' of 6100 psi using cyclic with inherent damage resulting from varying number of stress cycles at fixed σ/f_c'. The residual strength has been calculated for concrete with damage sustained in increments of 10 cycles for the case of $\sigma/f_c' = 0.93$ and 0.82. Variable increment was used for $\sigma/f_c' = 0.72$. The results are shown in Table 4.6 and Fig. 4.7.

The result indicate that there is a sharp decline in the residual of strength of concrete with inherent damage comparing to a number of stress cycles that is in the neighborhood of the fatigue life for the corresponding magnitude of σ/f_c'.

		Experimental Results		Analytical Results	
σ (Psi)	σ/fc'	n	Log(n)	n	Log(n)
7100	1.16	1	0.000	1	0.000
7000	1.15	2	0.300	1	0.000
6800	1.11	2	0.357	1	0.000
6500	1.07	5	0.678	5	0.700
6200	1.02	6	0.789	10	1.000
6100	1.00	13	1.100	12	1.080
5900	0.97	19	1.286	19	1.280
5600	0.92	40	1.600	41	1.610
5300	0.87	72	1.857	71	1.850
5000	0.82	141	2.150	147	2.170
4700	0.77	372	2.570	380	2.580
4400	0.72	1259	3.100	1216	3.080
4100	0.67	5012	3.700	5029	3.700
3660	0.60	100000	5.000	99158	5.000

Table 4.2 Comparison Between Experimental and Analytical Results for Uniaxial Cyclic Loading (f_c' = 6100 Psi)

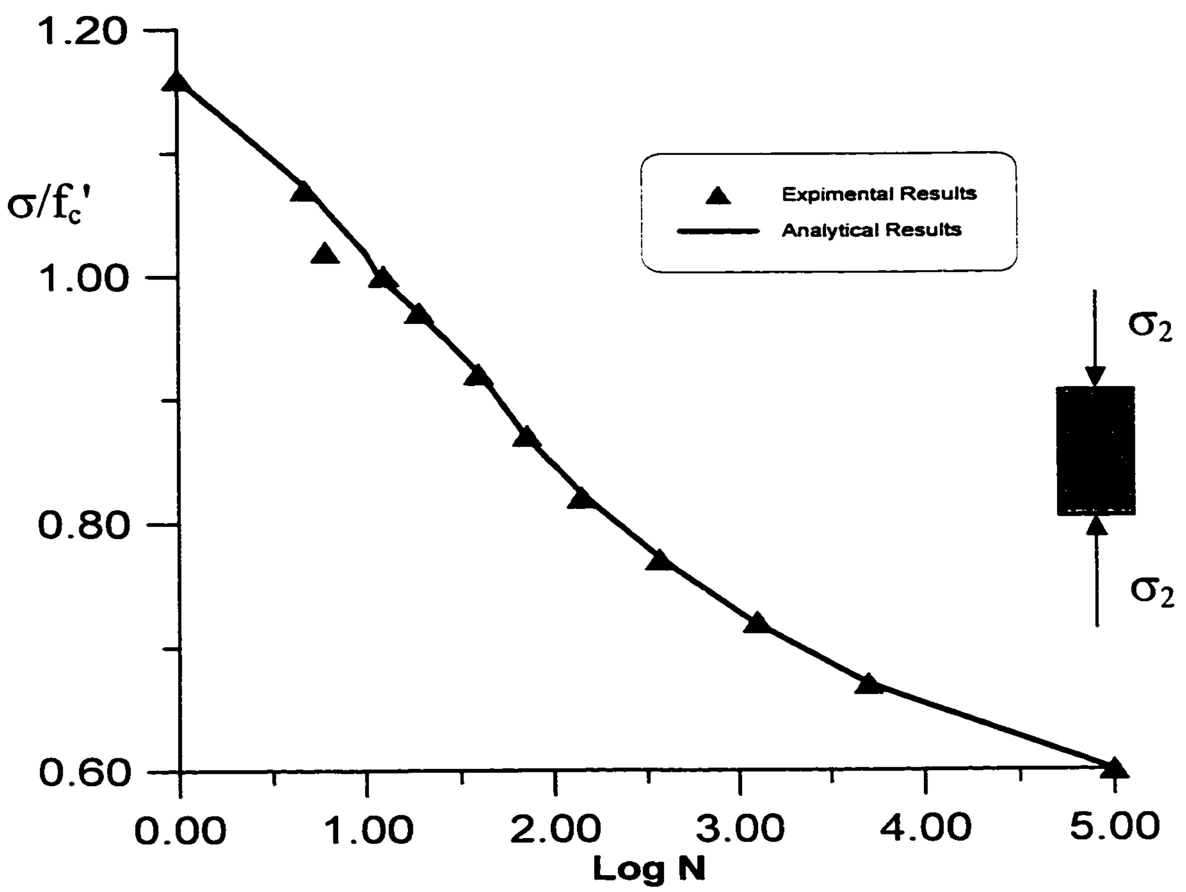

Figure 4.3 S-N Space Response of Concrete under Uniaxial Compression

σ (Psi)	σ/fc'	Experimental Results n	Experimental Results Log(n) (exp.)	Analytical Results n	Analytical Results Log(n) (model)
8967	1.47	1	0.00	1	0.00
8540	1.40	3	0.40	1	0.00
7930	1.30	10	1.00	11	1.04
7320	1.20	32	1.50	33	1.52
6710	1.10	141	2.15	140	2.15
6100	1.00	631	2.80	644	2.81
5490	0.90	15849	4.20	15903	4.20
4880	0.80	100000	5.00	99525	5.00

Table 4.3 Comparison Between Experimental and Analytical Results for Biaxial Cyclic Loading (f_c' = 6100 Psi) with $\sigma_2/\sigma_3 = 0.2$

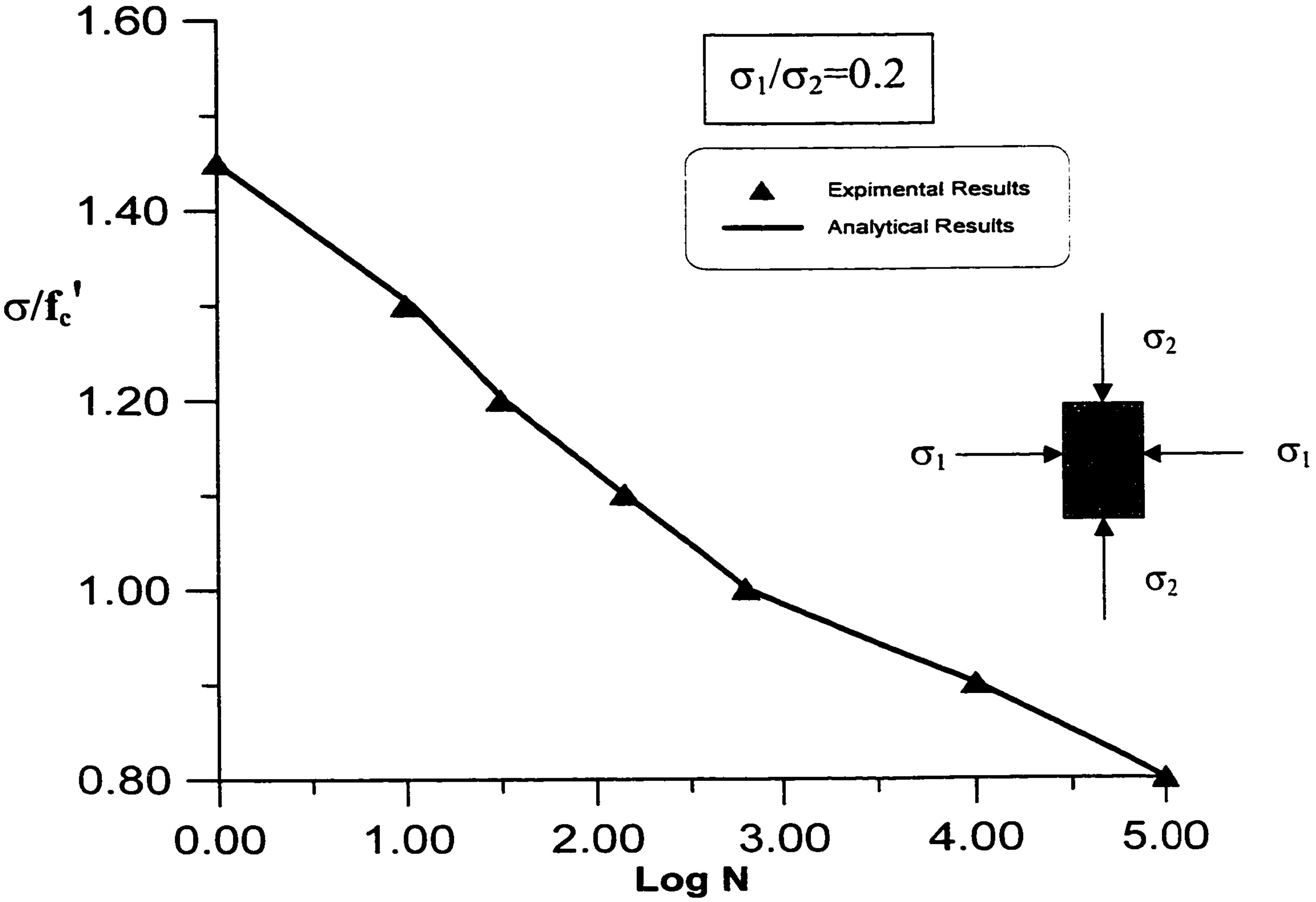

Figure 4.4 S-N Space Response of Concrete under Biaxial Compression With σ_1/σ_2= 0.2

		Experimental Results		Analytical Results	
σ (Psi)	σ/fc'	n	Log(n) (exp.)	n	Log(n) (model)
8845	1.45	1	0.00	1	0.00
8540	1.40	2	0.30	1	0.00
7930	1.30	10	1.00	14	1.10
7320	1.20	63	1.80	63	1.80
6710	1.10	398	2.60	394	2.60
6100	1.00	1995	3.30	2032	3.30
5490	0.90	25119	4.40	24736	4.40
4880	0.80	100000	5.00	98255	5.00

Table 4.4 Comparison Between Experimental and Analytical Results for Biaxial Cyclic Loading ($f_c' = 6100$ Psi) with $\sigma_2/\sigma_3 = 0.5$

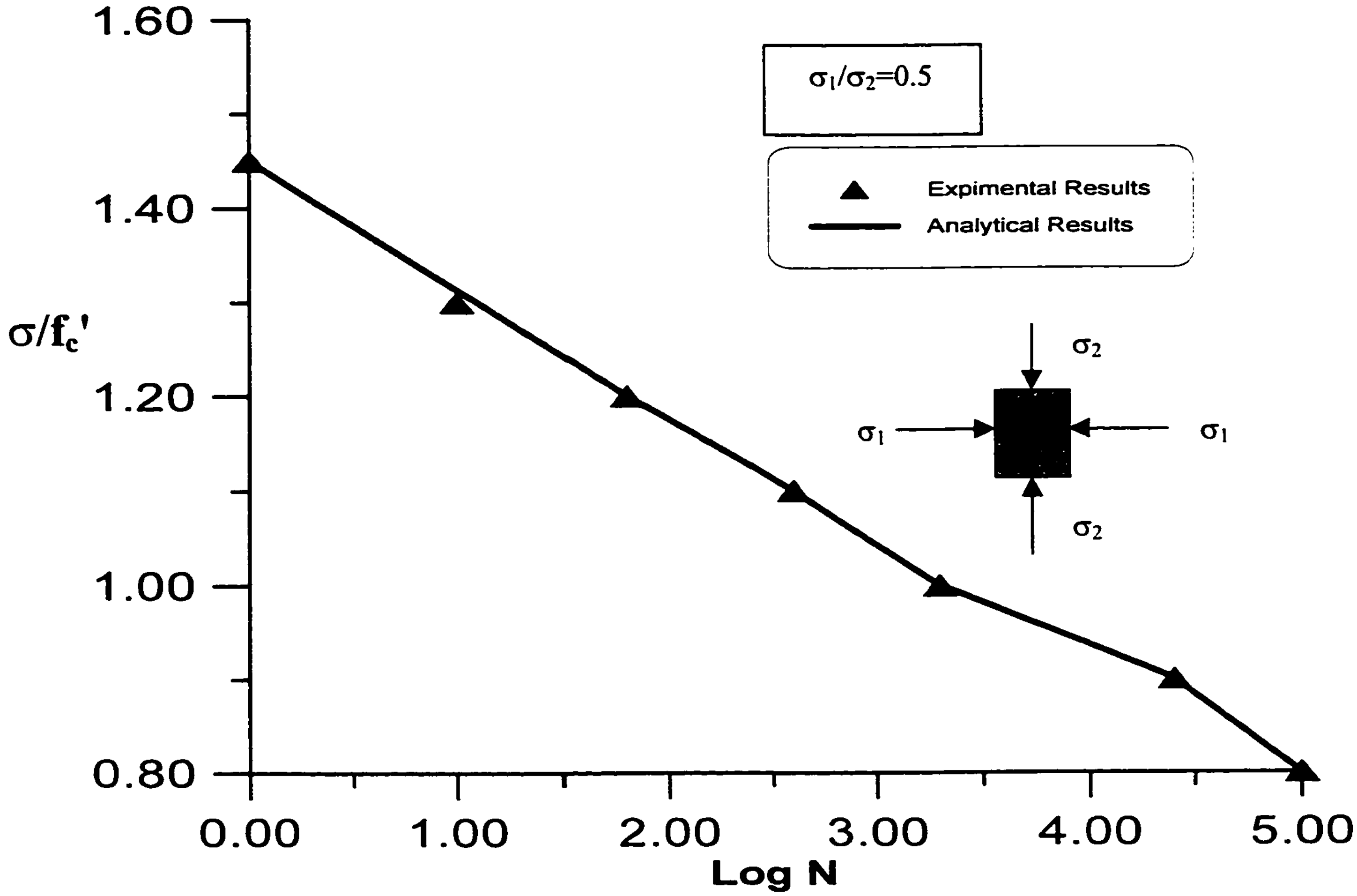

Figure 4.5 S-N Space Response of Concrete
under Biaxial Compression
With $\sigma_1/\sigma_2 = 0.5$

		Experimental Results		Analytical Results	
σ (Psi)	σ/fc'	n	Log(n) (exp.)	n	Log(n) (model)
8723	1.43	1	0.00	1	0.00
8540	1.40	1	0.174	1	0.00
7930	1.30	7	0.87	9	0.95
7320	1.20	27	1.43	27	1.43
6710	1.10	100	2.00	101	2.00
6100	1.00	398	2.60	406	2.61
5490	0.90	1122	3.05	1114	3.05
4880	0.80	6310	3.80	6323	3.80
4270	0.70	100000	5.00	97440	4.90

Table 4.5 Comparison Between Experimental and Analytical Results for Biaxial Cyclic Loading (f_c' = 6100 Psi) with σ_2/σ_3 = 1.0

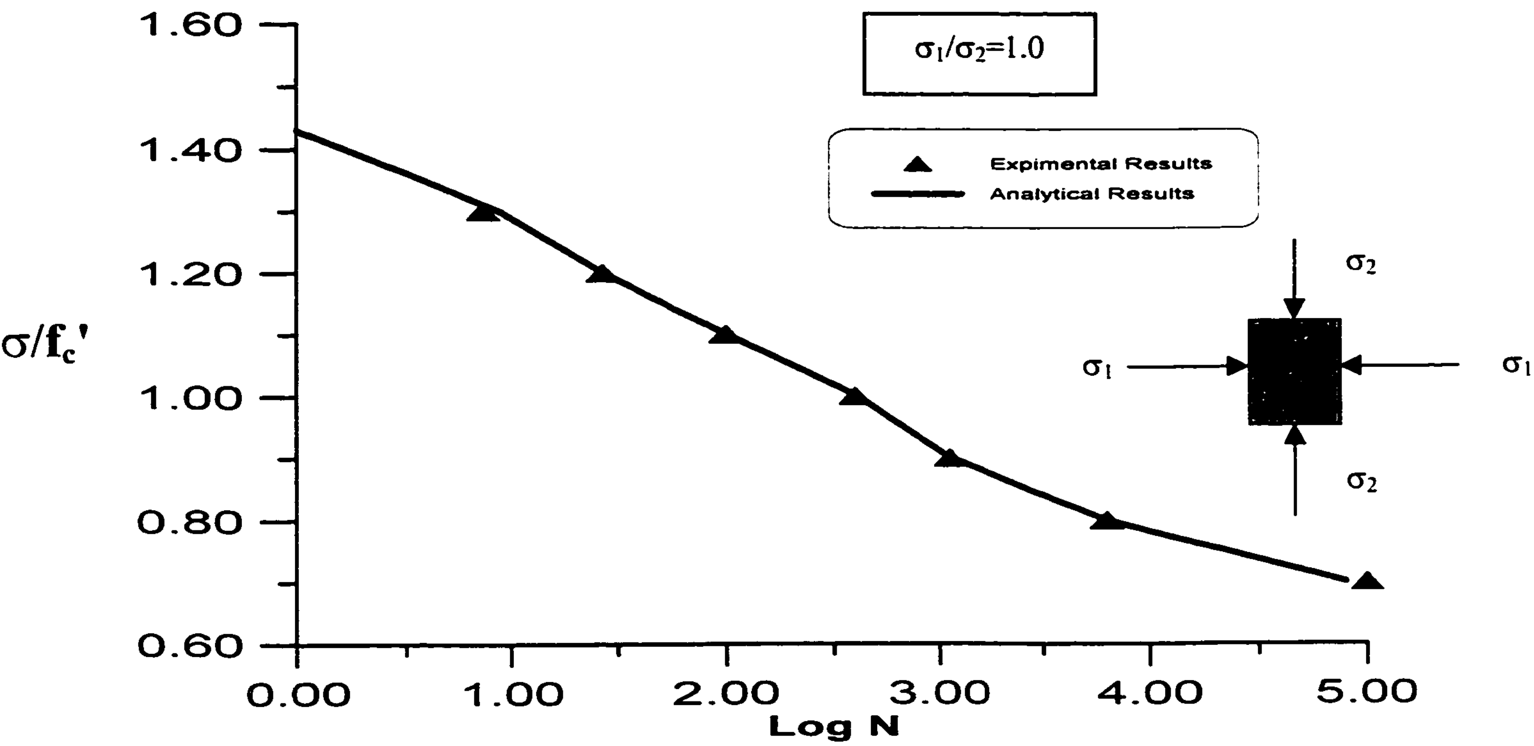

Figure 4.6 S-N Space Response of Concrete under Biaxial Compression With σ_1/σ_2= 1.0

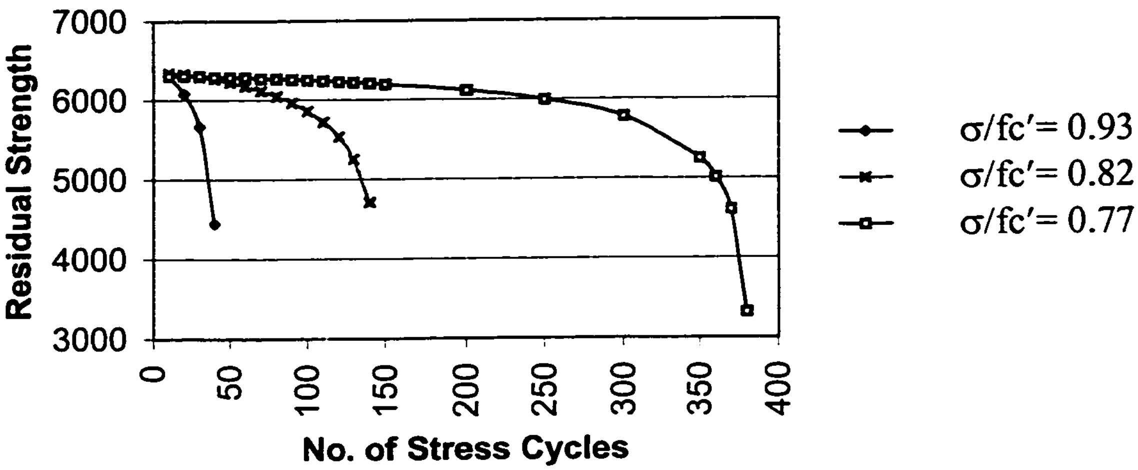

Figure 4.7 Relation between Residual Strength and Number of Stress Cycles ($f_c' = 6100$ Psi)

Number of cycles	Residual Strength		
	σ/fc′=0.77	σ/fc′=0.82	σ/fc′=0.93
10	6305	6345	6320
20	6310	6330	6080
30	6305	6300	5665
40	6295	6270	4445
50	6290	6225	
60	6285	6175	
70	6275	6115	
80	6270	6045	
90	6260	5965	
100	6250	5860	
110	6240	5720	
120	6230	5535	
130	6220	5250	
140	6210	4710	
150	6195		
200	6115		
250	5995		
300	5780		
350	5250		
360	5010		
370	4605		
380	3320		

Table 4.6 Relation between Residual Strength and Number of Stress Cycles (f_c' = 6100 Psi)

CHAPTER 5

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 SUMMARY

An incremental constitutive relation for stress and strain control has been developed along with a study on the damage evolution law for concrete with applicability for uniaxial and biaxial compression loading. The model has been investigated for monotonic and cyclic loading condition of concrete subjected to uniaxial and biaxial compression. Finally the newly developed concept of residual strength of concrete under uniaxial compressive cyclic loading using the concept of damage mechanics has been presented.

5.2 CONCLUSIONS

- The incremental elasto-damage constitutive equations have been developed for stress control as well as strain control.

- The multi-surface continuum damage model is shown to simulate stress-strain curve for monotonic loading for a wide range of compressive strength of concrete through

calibration of its various parameters D, β and R_o^b which have been found to be function of concrete strength f_c'.

- Introduction of a moving limit fracture surface whose size R_o is governed by an elliptical dependence on cumulative damage $\bar{\omega}$ and appending it to the multi-surface continuum damage model is shown to simulate the fatigue response of concrete subjected to cyclic loading under compression. The size of the limit fracture surface is noted to have a bound which is governed by the amplitude of the cyclic stress σ/f_c'.

- The model parameter β_{dyn} for predicting the response of concrete that fails due to one cycle of cyclic loading has been calibrated.

- The model was calibrated to predict the S-N curves depicting the fatigue response of concrete under uniaxial and biaxial loading for both low and high cycle fatigues.

- The innovative concept of residual strength prediction based on continuum damage formulation has been introduced.

5.3 RECOMMENDATIONS

1. The model parameters for cyclic loading under uniaxial compressive loading have been calibrated for a specific concrete strength $f_c' = 6100$ psi. Additional experimental data is necessary for investigation of fatigue life of concrete of strength different from $f_c' = 6100$ psi.

2. Experimental data for calibration of model parameters for monotonic and cyclic loading under biaxial compression for concrete of strength other than $f_c' = 6100$ psi is required.

3. The residual strain should be incorporated in the constitutive model.

4. The damage framework presented in the current work is recommended to be generalized to predict the S-N curves depicting the fatigue response of concrete under general triaxial loading.

5. The innovative residual strength model introduced can be used for a variety of concrete phenomena influenced by degradation of material parameters. This would require appropriate laboratory testing.

REFERENCES

[1] Su, C. M. and Hsu T.C., "Biaxial Compression Fatigue and Discontinuity of Concrete," *ACI Material Journal*, Vol. 85, No. 3, 1988, pp. 178-188.

[2] Krajcinovic, D. and Fonseka, U., "Continuous Damage Theory for Brittle Material," *Journal of Applied Mechanics*, Vol. 48, No. 4, 1981, pp. 809-824.

[3] Mazars, J. And Pijaudier-Cabot, G., "Continuum Damage Theory - Application to Concrete," *Journal of Engineering Mechanics*, ASCE, Vol. 115, No. 2, 1989, pp. 345-465.

[4] Suaris, W., Ouyang, C., and Fernando, V., " Damage Model for Cyclic Loading of Concrete", *Journal Engineering Mechanics, ASCE*, Vol. 116, No.5, May 1990, pp. 1020-1035.

[5] Lee, H., Peng, K., and Wang, J., "An Anisotropic Damage Criterion for Deformation Instability and its Application to Forming Limit Analysis of Metal Plates," *Engineering Fracture Mechanics*, Vol. 21, No. 5, 1985, pp. 1031 -1054.

[6] Chow, C.L., and Wang, J., "An Anistropic Theory of Elasticity for Continuum Damage Mechanics," *International Journal of Fracture,* Vol. 33, 1987, pp. 3-16.

[7] Chow, C.L., and Wang, J., "A Finite Element Analysis of Continuum Damage Mechanics for Ductile Fracture", *International Journal of Fracture,* Vol. 38, 1988, 83-102.

[8] Chow, C.L. and Wang. J. "Crack Propagation in Mixed-Mode Ductile Fracture with Continuum Damage Mechanics," *Proceedings of the Institution of mechanical Engineers*, Vol. 203, 1988, pp. 189-199.

[9] Ghrib, F. and Tinawi, R., "Nonlinear Behavior of Concrete Dams using Damage Mechanics," *Journal of Engineering Mechanics, ASCE*, Vol. 121, No. 4, 1995, pp. 513-527.

[10] Dafalias, Y.F. and Popov, E.P., "Cyclic Loading for Material with a Vanishing Elastic Region," *Nuclear Engineering Design*, Vol. 41, 1977, pp. 283-302.

[11] Al-Gadhib, A.H., Rahman, A. K. and Baluch, M.H., "CDM Based Finite Element Code for Concrete In 3-D," *Journal of Computer and Structures*, Vol. 67, 1998, pp. 451-462.

[12] Kufter, H. and Gerstle K.D., "Behavior of Concrete Under Biaxial Stresses," *Journal of Engineering Mechanics, ASCE*, Vol. 99, No. 4, 1973, pp. 853-866.

[13] Ortiz, M., "A Constitutive Theory for the Inelastic Behavior of Concrete," *Mechanics of materials*, Vol. 4, 1985, pp. 67-93.

[14] Haynurst D.R. and Leckie F. A., *Journal of Mechanics and Physics of Solids*, Vol. 21, 1973, pp. 431-446.

[15] Sidoroff, F., "Description of Anisotropic Damage Application to Elasticity," *IUTAM Colloquium, Physical Nonlinearities in Structural Analysis*, 198 1, pp. 237-244.

[16] Lemaitre, J. and Chabocha, J.L., *Journal de Mecanique Applique*, Vol. 2, 1978, pp. 317-365.

[17] Rahman A.K., Al-Gadhib, A.H. and Baluch, M.H., "An Elasto-Damage Constitutive Model for High Strength Concrete," proceedings of EURO-C

1998 Conference on Computational Modeling of concrete structures at Austria, March 1998, pp. 133-142.

[18] Coleman, B.D. and Gurtin, M.E. "Thermodynamics with Internal State Variables," *Journal of Chem. Phys.*, Vol. 47, 1967, pp. 597-613.

[19] Leckie, F.A. and Onat, F.T., "Tensional Nature of Damage Measuring Internal Variables, " *IUTAM Colloquium on Physical Nonlinearities in Structural Analysis*, 1981, pp. 140-155.

[20] Al-Gadhib, A.H. and Baluch, M.H., "Continuum Damage Mechanics Based Model for Fatigue in Concrete," *Proceedings of the 2nd Regional Conference, American Society of Civil Engineers - Saudi Arabia Section (ASCE-SAS)*, pp. 221-232, Beirut, Lebanon, Nov. 16-18, 1995.

[21] Ravindra, G., Antonio A., and Marcel, O., "Damage in High-Strength Concrete Due to Monotonic and Cyclic Compression - A Study Based on Splitting Tensile Strength," *ACI Materials Journal*, Vol. 93, No. 6, 1996, pp. 519-523.

[22] Chen, W.F. and Saleeb, A.F., "Constitutive Equations for Engineering Materials," Vol. 1, John Wiley & Son Inc., 1982.

[23] Erik, L. N., Ramon, L. C. and David, W. F., "Behavior and Failure of High-Strength Concrete Subjected to Biaxial-Cyclic Compression Loading," *ACI Material Journal*, Vol. 85, No. 3, pp. 248-253,July-August 1988.

[24] Kupfer, H., and Gerstle, K.H., " Behavior of Concrete under Biaxial Stresses, " *JEM Division, ASCE,* Vol. 99, No. EM4, Proc. paper 9917, Aug. 1973, pp. 853-866.

[25] Kachanov,L.M., "On the Creep Fracture Time", *Izv. An SSSR, Otd. Tekhn, Nauk,* No. 8, 1958, pp. 26-31. [In Russian]

[26] Suaris,W. and Shah,S.P., "Rate-Senstive Damage Theory for Brittle Solids", *Engineering Mechanics*, Vol. 110, No. 6, June, 1984.

[27] Lemaitre,J., " A Continuous Damage Mechanics Model for Ductile Fracture", *Journal of Engineering Materials and Technology*, January, 1985, vol. 107/83.

[28] Abbasi, M.S. , " Nonlinear Finite Element Modeling of Reinforced Concrete Slabs," *Thesis presented to King Fahd University of Petroleum and Minerals at Dhahran, Saudi Arabia*, in 1991, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

[29] Abdel Rahman, H., "Computational Model for the Nonlinear Analysis of Reinforced Concrete Flexural Slab System, " *Thesis presented to the University College of Swansea at Swansea, Britain, in 1982, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.*

[30] Breder, B. and Pister, K.S., " Strength of Concrete under Combined Stresses, " *Journal of American Concrete Institute*, Sep. 1958, pp. 321-345.

[31] Cedolin, L., Crutzen, Y.R.J. and Dei Poli, S., " Triaxial Stress-Strain Relationship for Concrete, " *Journal Engineering Mechanics* Division, ASCE, VOL. 103, NO. EM3, Proc. paper 12969, June 1977, pp. 525-439.

[32] Coon , M.D ., and Evans , R.J ., " Incremental Constitutive Laws and their Associated Failure Criteria with Application to Plain Concrete , " *Int. . J . of Solid and Struct*. , Vol . 8 , 1972 , pp . 1169-1183 .

[33] Dafalios , Y.F. and Popov , E.P . , " A Model of Nonlinear Hardening Materials for Complex Loading, " *Acta Mechanica* , Vol . 21,(1975) , PP . 173-92 .

[34] Desai , C.S , " A General Basis for Yield , Failure and Potential Functions in Plasticity , " *Int. J . for Numerical and Analytical Methods in Geomechanics* , Vol . 4 , 1980 , PP . 361-375.

[35] Dhir , R.H. and Sangha , M., " Development and Propagation of Microcracks in Plain Concrete , " *Materials and Structures* , RILEM , Vol. . 37, (1974) , PP . 17-23 .

[36] Drucker , D.C . and Prager , W. , "Soil Mechanics and Plastic Analysis or Limit Design , " *Quarterly of Applied Mathematics* , VO . X, NO. 1952. PP. 157-165.

[37] Ottosen , N. , " A Failure Criterion for Concretes , " *Journal of Engineering Mechanics Division , ASCE* , Vol. . 103,No. EM4, 1977, pp. 527-535.

[38] Slate , F.O . and Oleski , S., "X-Rays for Study of Internal Structure and Microcracking of Concrete , " *J . ACI, Vol.* 60, (1963), pp. 575-588.

[39] Terrien , M., " Emission Accoustique et Comprtement Mechanique Post-Critique , " *Bull des Ponts et Chaus* . Vol. 105, (1980), pp. 65-72.

[40] Willam , K.J ., and Warnke , E.P . , " Constitutive Models for the Triaxial Behavior of Concrete Structures Subjected to Triaxial Stresses, " *Papers III-I. Bergano, Italy,* May 17-19. 1974, pp. 1-3.

[41] Maotti, Sc., " Geneal Theoy of Damage Elastoplastic Models, *"Jounal of Engineering Mechanics* (ASCE), v 123, oct. 1997, pp. 1003-1011.

[42] Yazdani, S. and Schreyer, H., " Combined Placticity and Damage Mechanics Model for Plain Concrete, "*Journal of Engineering Mechanics* (ASCE), v 116, July, 1990, pp. 1435-1450.

[43] Simo, J., and Ju, J., "Strain and Stress-Based Continuum Damage Models - I. Formulation, " *International Journal of solids and structures*, 1987, pp 821-840.

[44] Lemaitre, J. and Chaboche, J. "Mechanics of Solid Materials," *Cambridge University Press*, Cambridge, 1990.

[45] Carol, I. and Part, P. "A Multicrack Model Based on the Theory of Multisurface Plasticity and Two Fracture Energies. In Owen, D., Onate, E. and Hinton, E., editors, Computational Plasticity," *Proceeding of 4th int. conf.*, 1995, pp. 1583-1595.

[46] Carol, I., Rizzi, E. and Willam, K. "Towards a General Formulation of Elastic Degradation and Damage Based on Loading Surface, " *Computer Modelling of Concrete Structures*, 1994, pp 199-208.

[47] Govindjee, S., Kay, G. and Simo, J, "Anisotropic Modeling and Numerical Simulation of Brittle Damage in Concrete," *International Journal for Numerical Methods in Engineering*, 1995, pp. 3611-3634.

[48] Loureco, P., De Borst, R. and Rots, J. "A Plane Stress Softening Plasticity Model for Orthotropic Materials," *International Journal for Numerical Methods in Engineering*, 1997, pp. 4033-4057.

[49] Meschke, G., Lackner, R. and Mang, H. " An Anisotopic Elastoplastic-Damage Model for Plain Concrete, "*International Journal for Numerical Methods in Engineering*, 1998, in print.

[50] Simo, J., Oliver, J. and Armero, F. "An Analysis of Strong Discontinuities Induced by Strain-Softening in Rate-Independent Inelastic Solids, " *Computational Mechanics*, 1993, pp. 277-296.

[51] Ignacio, C., Egidio, R. and Kaspar, W. " On the Formulation of Isotropic Damage, " *Computational Modelling of Concrete Structures*, 1998, pp. 183-192.

[52] Kuhl, E. and Ramm, E. " Anisotropic Gradient Damage with the Microplane Model," *Computational Modelling of Concrete Structures*, 1998, pp. 103-112.

[53] Bazant, P. and Part, P " Microplane Model for Brittle Plastic Material, " I. Theory and II. Verification, *Journal of Engineering Mechanics*, Vol. 114, 1988, pp. 1672-1702.

[54] Carol, I. and Bazant, P. "Damage and Plasticity in Microplane Theory," *International Journal of Solids and Structures*, Vol. 34, 1997, pp. 3807-3835.

APPENDICES

APPENDIX A

COMPONENTS OF VECTOR $\frac{\partial f}{\partial R_i}$ AND TENSORS $\frac{\partial R_i}{\partial \omega_j}, \frac{\partial R_i}{\partial \sigma_j}, \frac{\partial R_i}{\partial \varepsilon_j}$ TO BE USED IN COMPUTATION OF INCREMENT OF DAMAGE

$$f = (R_i R_i)^{\frac{1}{2}} - \frac{R_c}{b} = 0.0$$

$$f = (R_1 R_1 + R_2 R_2 + R_3 R_3)^{\frac{1}{2}} - \frac{R_c}{b} = 0 \tag{A1.1}$$

By differentiating equation A1.1 with respect to R_i yields

$$\left[\frac{\partial f}{\partial R_i}\right]^T = \left[\frac{\partial f}{\partial R_1} \quad \frac{\partial f}{\partial R_2} \quad \frac{\partial f}{\partial R_3}\right] \tag{A1.2}$$

Where

$$\frac{\partial f}{\partial R_1} = \frac{1}{2}(R_1 R_1 + R_2 R_2 + R_3 R_3)^{-\frac{1}{2}}(2R_1) = \frac{R_1}{(R_i R_i)^{1/2}} \tag{A1.3a}$$

$$\frac{\partial f}{\partial R_2} = \frac{1}{2}(R_1 R_1 + R_2 R_2 + R_3 R_3)^{-\frac{1}{2}}(2R_2) = \frac{R_2}{(R_{ui} R_i)^{1/2}} \tag{A1.3b}$$

$$\frac{\partial f}{\partial R_3} = \frac{1}{2}(R_1 R_1 + R_2 R_2 + R_3 R_3)^{-\frac{1}{2}}(2R_3) = \frac{R_3}{(R_i R_i)^{1/2}} \tag{A1.3c}$$

By differentiating equation 2.55 with respect to ω_i yields

$$\frac{\partial R_i}{\partial \omega_j} = \begin{bmatrix} \frac{\partial R_1}{\partial \omega_1} & \frac{\partial R_2}{\partial \omega_1} & \frac{\partial R_3}{\partial \omega_1} \\ \frac{\partial R_1}{\partial \omega_2} & \frac{\partial R_2}{\partial \omega_2} & \frac{\partial R_3}{\partial \omega_2} \\ \frac{\partial R_1}{\partial \omega_3} & \frac{\partial R_2}{\partial \omega_3} & \frac{\partial R_3}{\partial \omega_3} \end{bmatrix} \tag{A1.4}$$

Where

$$\frac{\partial R_1}{\partial \omega_1} = \frac{1}{2E_o}\left[\frac{2\sigma_1^{\,2}(\beta^2)}{(1-\beta\omega_2)^2(1-\beta\omega_3)^2} + \frac{6\beta^2\sigma_2^{\,2}(1-\beta\omega_2)^2}{(1-\beta\omega_3)^2(1-\beta\omega_1)^4}\right] \tag{A1.4a}$$

$$\frac{\partial R_2}{\partial \omega_1} = \frac{1}{2E_o}\left[\frac{-4\beta^2(1-\beta\omega_1)\sigma_1^{\,2}}{(1-\beta\omega_2)^3(1-\beta\omega_3)^2} - \frac{4\beta^2\sigma_2^{\,2}(1-\beta\omega_2)}{(1-\beta\omega_3)^2(1-\beta\omega_1)^3}\right] \tag{A1.4b}$$

$$\frac{\partial R_3}{\partial \omega_1} = \frac{1}{2E_o}\left[\frac{-4\beta^2(1-\beta\omega_1)\sigma_1^{\,2}}{(1-\beta\omega_2)^2(1-\beta\omega_3)^3} + \frac{4\beta^2\sigma_2^{\,2}(1-\beta\omega_2)^2}{(1-\beta\omega_3)^3(1-\beta\omega_{1)})^3}\right] \tag{A1.4c}$$

$$\frac{\partial R_1}{\partial \omega_2} = \frac{1}{2E_o}\left[\frac{-4(1-\beta\omega_1)\ \sigma_1^{\,2}(\beta^2)}{(1-\beta\omega_2)^3(1-\beta\omega_3)^2} - \frac{4\beta^2\sigma_2^{\,2}(1-\beta\omega_2)}{(1-\beta\omega_3)^2(1-\beta\omega_1)^3}\right] \tag{A1.4d}$$

$$\frac{\partial R_2}{\partial \omega_2} = \frac{1}{2E_o}\left[\frac{6\beta^2(1-\beta\omega_1)^2\sigma_1^{\,2}}{(1-\beta\omega_2)^4(1-\beta\omega_3)^2} + \frac{2\beta^2\sigma_2^{\,2}}{(1-\beta\omega_3)^2(1-\beta\omega_1)^2}\right] \tag{A1.4e}$$

$$\frac{\partial R_3}{\partial \omega_2} = \frac{1}{2E_o}\left[\frac{4\beta^2(1-\beta\omega_1)^2\sigma_1^{\,2}}{(1-\beta\omega_2)^3(1-\beta\omega_3)^3} - \frac{4\beta^2\sigma_2^{\,2}(1-\beta\omega_2)}{(1-\beta\omega_3)^3(1-\beta\omega_1)^2}\right] \tag{A1.4f}$$

$$\frac{\partial R_1}{\partial \omega_3}=\frac{1}{2E_o}\left[\frac{-4(1-\beta\omega_1)\sigma_1^{\,2}(\beta^2)}{(1-\beta\omega_2)^2(1-\beta\omega_3)^3}+\frac{4\beta^2\sigma_2^{\,2}(1-\beta\omega_2)^2}{(1-\beta\omega_3)^3(1-\beta\omega_1)^3}\right] \tag{A1.4g}$$

$$\frac{\partial R_2}{\partial \omega_3}=\frac{1}{2E_o}\left[\frac{4\beta^2(1-\beta\omega_1)^2\sigma_1^{\,2}}{(1-\beta\omega_2)^3(1-\beta\omega_3)^3}-\frac{4\beta^2\sigma_2^{\,2}(1-\beta\omega_2)}{(1-\beta\omega_3)^3(1-\beta\omega_1)^2}\right] \tag{A1.4h}$$

$$\frac{\partial R_3}{\partial \omega_3}=\frac{1}{2E_o}\left[\frac{6\beta^2(1-\beta\omega_1)^2\sigma_1^{\,2}}{(1-\beta\omega_2)^2(1-\beta\omega_{3)})^4}-\frac{12\beta^2\nu\sigma_1\sigma_2}{(1-\beta\omega_3)^4}+\frac{6\beta^2\sigma_2^{\,2}(1-\beta\omega_2)^2}{(1-\beta\omega_3)^4(1-\beta\omega_1)^2}\right] \tag{A1.4i}$$

By differentiating equation 2.55 with respect to σ_i yields

$$\frac{\partial R_i}{\partial \sigma_j}=\begin{bmatrix}\frac{\partial R_1}{\partial \sigma_1} & \frac{\partial R_2}{\partial \sigma_1} & \frac{\partial R_3}{\partial \sigma_1}\\ \frac{\partial R_1}{\partial \sigma_2} & \frac{\partial R_2}{\partial \sigma_2} & \frac{\partial R_3}{\partial \sigma_2}\\ \frac{\partial R_1}{\partial \sigma_3} & \frac{\partial R_2}{\partial \sigma_3} & \frac{\partial R_3}{\partial \sigma_3}\end{bmatrix} \tag{A1.5}$$

Where

$$\frac{\partial R_1}{\partial \sigma_1}=\frac{1}{E_o}\left[\frac{-2(1-\beta\omega_1)\sigma_1(\beta)}{(1-\beta\omega_2)^2(1-\beta\omega_3)^2}\right] \tag{A1.5a}$$

$$\frac{\partial R_2}{\partial \sigma_1}=\frac{1}{E_o}\left[\frac{2\beta(1-\beta\omega_1)^2\sigma_1}{(1-\beta\omega_2)^3(1-\beta\omega_3)^2}\right] \tag{A1.5b}$$

$$\frac{\partial R_3}{\partial \sigma_1}=\frac{1}{E_o}\left[\frac{2\beta(1-\beta\omega_1)^2\sigma_1}{(1-\beta\omega_2)^2(1-\beta\omega_3)^3}-\frac{2\beta\nu\sigma_2}{(1-\beta\omega_3)^3}\right] \tag{A1.5c}$$

$$\frac{\partial R_1}{\partial \sigma_2} = \frac{1}{E_o}\left[\frac{2\beta\sigma_2(1-\beta\omega_2)^2}{(1-\beta\omega_3)^2(1-\beta\omega_1)^3}\right] \tag{A1.5d}$$

$$\frac{\partial R_2}{\partial \sigma_2} = \frac{1}{E_o}\left[-\frac{2\beta\sigma_2(1-\beta\omega_2)}{(1-\beta\omega_3)^2(1-\beta\omega_{1)})^2}\right] \tag{A1.5e}$$

$$\frac{\partial R_3}{\partial \sigma_2} = \frac{1}{E_o}\left[-\frac{2\beta\nu\sigma_1}{(1-\beta\omega_{3)})^3} + \frac{2\beta\sigma_2(1-\beta\omega_2)^2}{(1-\beta\omega_3)^3(1-\beta\omega_1)^2}\right] \tag{A1.5f}$$

$$\frac{\partial R_1}{\partial \sigma_3} = 0.0 \tag{A1.5g}$$

$$\frac{\partial R_2}{\partial \sigma_3} = 0.0 \tag{A1.5h}$$

$$\frac{\partial R_3}{\partial \sigma_3} = 0.0 \tag{A1.5i}$$

By differentiating equation 2.64 with respect to ε_i yields

$$\frac{\partial R_i}{\partial \varepsilon_j} = \begin{bmatrix} \frac{\partial R_1}{\partial \varepsilon_1} & \frac{\partial R_2}{\partial \varepsilon_1} & \frac{\partial R_3}{\partial \varepsilon_1} \\ \frac{\partial R_1}{\partial \varepsilon_2} & \frac{\partial R_2}{\partial \varepsilon_2} & \frac{\partial R_3}{\partial \varepsilon_2} \\ \frac{\partial R_1}{\partial \varepsilon_3} & \frac{\partial R_2}{\partial \varepsilon_3} & \frac{\partial R_3}{\partial \varepsilon_3} \end{bmatrix} \tag{A1.6}$$

Where

$$\frac{\partial R_1}{\partial \varepsilon_1} = E_o\left[\frac{-(1-\beta\omega_1)(\beta)}{(1-\beta\omega_2)^2(1-\beta\omega_3)^2}\right]\left[\frac{2(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}+\varepsilon_2\nu\right)\right]$$

$$+E_o\left[\frac{\beta(1-\beta\omega_2)^2}{(1-\beta\omega_3)^2(1-\beta\omega_1)^3}\right]\left[2\nu\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right] \tag{A1.6a}$$

$$\frac{\partial R_1}{\partial \varepsilon_2} = E_o\left[\frac{-(1-\beta\omega_1)(\beta)}{(1-\beta\omega_2)^2(1-\beta\omega_3)^2}\right]\left[2\nu\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}+\varepsilon_2\nu\right)\right]$$

$$+E_o\left[\frac{\beta(1-\beta\omega_2)^2}{(1-\beta\omega_3)^2(1-\beta\omega_1)^3}\right]\left[\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right] \tag{A1.6b}$$

$$\left[\frac{\partial R_1}{\partial \varepsilon_3}\right] = 0 \tag{A1.6c}$$

$$\frac{\partial R_2}{\partial \varepsilon_1} = E_o\left[\frac{\beta(1-\beta\omega_1)^2}{(1-\beta\omega_2)^3(1-\beta\omega_3)^2}\right]\left[\frac{2(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}+\varepsilon_2\nu\right)\right]$$

$$-E_o\left[\frac{\beta(1-\beta\omega_2)}{(1-\beta\omega_3)^2(1-\beta\omega_1)^2}\right]\left[2\nu\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right]$$

(A1.6d)

$$\frac{\partial R_2}{\partial \varepsilon_2} = E_o\left[\frac{\beta(1-\beta\omega_1)^2}{(1-\beta\omega_2)^3(1-\beta\omega_3)^2}\right]\left[2\nu\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}+\varepsilon_2\nu\right)\right]$$

$$-E_o\left[\frac{\beta(1-\beta\omega_2)}{(1-\beta\omega_3)^2(1-\beta\omega_1)^2}\right]\left[\frac{2(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right] \tag{A1.6e}$$

$$\left[\frac{\partial R_2}{\partial\varepsilon_3}\right]=0 \tag{A1.6f}$$

$$\frac{\partial R_3}{\partial\varepsilon_1}=E_o\left[\frac{\beta(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2(1-\beta\omega_3)^3}\right]\left[\frac{2(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}+\varepsilon_2\nu\right)\right]$$

$$-E_o\left[\frac{2\beta\nu}{(1-\beta\omega_3)^3}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}+\varepsilon_2\nu\right)\right]\left[\nu\ \frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right]$$

$$-E_o\left[\frac{2\beta\nu}{(1-\beta\omega_{3)})^3}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right]$$

$$+E_o\left[\frac{\beta(1-\beta\omega_2)^2}{(1-\beta\omega_3)^3(1-\beta\omega_{1)})^2}\right]\left[2\nu\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right] \tag{A1.6g}$$

$$\frac{\partial R_3}{\partial\varepsilon_2}=E_o\left[\frac{\beta(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2(1-\beta\omega_3)^3}\right]\left[2\nu\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}+\varepsilon_2\nu\right)\right]$$

$$-E_o\left[\frac{2\beta\nu}{(1-\beta\omega_3)^3}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\frac{(1-\beta\omega_2)^2}{(1-\beta\omega_1)^2}+\varepsilon_2\nu\right)\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right]$$

$$-E_o\left[\frac{2\beta\nu}{(1-\beta\omega_3)^3}\right]\left[\frac{\gamma(1-\beta\omega_3)^2}{(1-\nu^2)}\right]\left[\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right]$$

$$+ E_o\left[\frac{\beta(1-\beta\omega_2)^2}{(1-\beta\omega_3)^3(1-\beta\omega_{1)})^2}\right]\left[\frac{2(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\left(\frac{(1-\beta\omega_3)^2}{(1-\nu^2)}\right)^2\left(\varepsilon_1\nu+\varepsilon_2\frac{(1-\beta\omega_1)^2}{(1-\beta\omega_2)^2}\right)\right] \tag{A1.6h}$$

$$\left[\frac{\partial R_3}{\partial \varepsilon_3}\right]=0 \tag{A1.6i}$$

APPENDIX B

LIST OF PROGRAMS

B.1. MONOTONIC LOAD

B.1.1 Static Uniaxial Monotonic Loading Program

Program name: ***STCUNMN***

```
C     ====================================================
C     THIS IS A PROGRAM FOR STATIC MONOTONIC LOADING FOR UNIAXIAL
C     COMPRESSION UNDER STRAIN CONTROL
C     ====================================================
C     ====================================================
C     THIS program is applicable for a concrete strength of 2200,
C     4000,6100,7000,8000 and 9700 psi only
C     ====================================================

      program STCUNMN

      implicit double precision (a-h,o-z)

      write(*,*)'the value of fc = ? '
      read(*,*) fc

      if (fc .eq. 2200) go to 2005
      if (fc .eq. 4000) go to 2010
      if (fc .eq. 6100) go to 2020
      if (fc .eq. 7000) go to 2025
      if (fc .eq. 8000) go to 2030
      if (fc .eq. 9700) go to 2035
c       fc = 2200
 2005     d     = 0.02
        beta  = 0.0756
        r0 = .04

      go to 3000
c====================================
c       fc = 4000
 2010     d     = 0.03
        beta  = 0.04745
      r0 = .08

      go to 3000
c====================================
```

```
c       fc = 6100
 2020     d      = 0.08
        beta  = 0.03855
       r0 = .08

      go to 3000
c====================================
c       fc = 7000
 2025     d      = 0.12
        beta  = 0.03665
       r0 = .08

      go to 3000
c====================================
c       fc = 8000
 2030     d      = 0.22
        beta  = 0.03812
       r0 = .17

      go to 3000
c====================================
c       fc = 9700
 2035     d      = 0.42
        beta  = 0.034475
       r0 = .17

      go to 3000
c====================================
 3000     continue
      WRITE(*,903) d,beta
 903  format(1x,f10.4,1x,f10.4,1x,f10.4,1x,f10.4)
       rc    =1.29

      E0    =57000*sqrt(fc)
      stresf=fc
      stranf=.0023
      bw=(stresf/(E0*stranf))**(0.25)
c      rc1 =beta*(sqrt(2.0)*stresf**2/(E0*bw**5))
C       r0 =beta*((0.4*fc)**2/(E0*(1-beta*.05)**5))
c      R0 = 0.9
c      r0 = .13 * rc

      deltin=1-r0/rc
      domega=0.0
      rrm1  =0.0
      wrm1  =0.0
      wp=0.0
      wbar  =sqrt(2.0)*wrm1
      epsed =0.0
      fac=1.0
      delta=1.0
      depsed=3.0e-06

      open (1,file = '8000.ou1',status='unknown')
```

```
      open (2,file = '4900.ou2',status='unknown')
      open (3,file = '8000.ou3',status='unknown')
      do while (epsed.lt.0.006)

        epsed=epsed+depsed



          f0=r0
        deltin=(rc-f0)/rc
        delr=1.0
        do while (delr.gt.0.000001)
        wr=wrm1
        r=rrm1

        delta  = 1.0-(r*sqrt(2.0)/rc)

        if ((deltin-delta).le.0.0) then
         h=1.0e12
        else
         h      =(d*delta)/(deltin-delta)
        end if

        f=r*sqrt(2.0)

          if (f.le.f0) then
            domega=0.0
          else
           domega =   (2*beta*E0*epsed*depsed*(1-(beta*wr))**3)
     &              /(h+fac*(3*E0*beta**2*epsed**2*(1.0-(beta*wr))**2))


          end if


        wr=wp+domega
        rr=E0*beta*((1.0-(beta*wr))**3)*epsed*epsed
        delr=abs(rr-r)
        ff=sqrt(2)*rr
        rrm1=rr
        wrm1=wr
        end do
        wp=wr
        sig = ((1.0-(beta*wr))**4.0)*E0*epsed
        if ( sig .lt. psig) stop
        psig = sig
        epse=sig/E0
        epsd=epsed-epse
        epsp=1.5*epsd
        epst=epse+epsd+epsp
         wbar  =sqrt(2.0)*wrm1
c        write(1,920) delta,epsd,epsed,epsp,epst,sig
        write(*,920) sqrt(2.0)*rr,epsd,epsed,epsp,epst,sig
        write(3,930) wr
```

```
c        write(1,940) epsed
         write(1,940) wbar
        write(2,930) sig
c        write(3,950) delta , ff
c       write(3,950) epst

      end do
        write (*,*) r0

 920  format(e12.6,4x,f8.6,2x,f8.6,3x,f8.6,4x,f8.6,2x,f10.3)
 930  format(f15.6,4x,f15.3)
 940  format(f20.10)
 950  format(f10.6,10x,f20.10)
      end

c=-=================================================================
```

B.1.2 Dynamic Uniaxial Monotonic Loading Program

Program name: ***DYNCUNMN***

```
c     ====================================================
c     THIS IS MONOTONIC LOADING CASE TO GENERATE βdyn FOR UNIAXIAL
c     COMPRESSION IN ORDER TO ENSURE A ULTIMATE STRESS LEVEL OF 1.17 fc'
c     UNDER CYCLIC LOADING FOR CASE N = 1
c     ====================================================

c     ====================================================
c     THIS program is applicable for a concrete strength of 2200,
c     4000,6100,7000,8000 and 9700 psi only
c     ====================================================

       program DYNCUNMN

      implicit double precision (a-h,o-z)

      write(*,*)'the value of fc = ? '
      read(*,*) fc

      if (fc .eq. 2200) go to 2005
      if (fc .eq. 4000) go to 2010
      if (fc .eq. 6100) go to 2020
      if (fc .eq. 7000) go to 2025
      if (fc .eq. 8000) go to 2030
      if (fc .eq. 9700) go to 2035
c       fc = 2200
 2005     d     = 0.02
        beta  = 0.0605
        r0 = .04
      go to 3000
c====================================
c       fc = 4000
```

```
 2010     d      = 0.03
        beta   = 0.03767
      r0 = .08
      go to 3000
c=====================================

c       fc = 6100
 2020     d      = 0.08
        beta   = 0.03053
       r0 = .08
      go to 3000
c=====================================
c       fc = 7000
 2025     d      = 0.12
        beta   = 0.02895
       r0 = .08
      go to 3000
c=====================================
c       fc = 8000
 2030     d      = 0.22
        beta   = 0.02975
       r0 = .17
      go to 3000
c=====================================
c       fc = 9700
 2035     d      = 0.42
        beta   = 0.0267
       r0 = .17
      go to 3000
c=====================================
 3000     continue
      WRITE(*,903) d,beta
 903  format(1x,f10.4,1x,f10.4,1x,f10.4,1x,f10.4)
       rc     =1.29

      E0    =57000*sqrt(fc)
      stresf=fc
      stranf=.0023
      bw=(stresf/(E0*stranf))**(0.25)
c      rc1 =beta*(sqrt(2.0)*stresf**2/(E0*bw**5))
C        r0 =beta*((0.4*fc)**2/(E0*(1-beta*.05)**5))
c      R0 = 0.9
c      r0 = .13 * rc

      deltin=1-r0/rc
      domega=0.0
      rrml  =0.0
      wrml  =0.0
      wp=0.0
      wbar  =sqrt(2.0)*wrml
      epsed =0.0
      fac=1.0
      delta=1.0
      depsed=3.0e-06
```

```
      open (1,file = '8000.ou1',status='unknown')
      open (2,file = '4900.ou2',status='unknown')
      open (3,file = '8000.ou3',status='unknown')
      do while (epsed.lt.0.006)

        epsed=epsed+depsed



          f0=r0
        deltin=(rc-f0)/rc
        delr=1.0
        do while (delr.gt.0.000001)
        wr=wrm1
        r=rrm1

        delta  = 1.0-(r*sqrt(2.0)/rc)

        if ((deltin-delta).le.0.0) then
         h=1.0e12
        else
         h      =(d*delta)/(deltin-delta)
        end if

        f=r*sqrt(2.0)

          if (f.le.f0) then
            domega=0.0
          else
           domega =  (2*beta*E0*epsed*depsed*(1-(beta*wr))**3)
     &             /(h+fac*(3*E0*beta**2*epsed**2*(1.0-(beta*wr))**2))


          end if


        wr=wp+domega
        rr=E0*beta*((1.0-(beta*wr))**3)*epsed*epsed
        delr=abs(rr-r)
        ff=sqrt(2)*rr
        rrm1=rr
        wrm1=wr
        end do
        wp=wr
        sig = ((1.0-(beta*wr))**4.0)*E0*epsed
        if ( sig .lt. psig) stop
        psig = sig
        epse=sig/E0
        epsd=epsed-epse
        epsp=1.5*epsd
        epst=epse+epsd+epsp

c        write(1,920) delta,epsd,epsed,epsp,epst,sig
```

```
        write(*,920) sqrt(2.0)*rr,epsd,epsed,epsp,epst,sig
c        write(2,930) wr,sig
        write(1,940) epsed
        write(2,940) sig
c        write(3,950) delta , ff
       write(3,950) epst

      end do
        write (*,*) r0

 920  format(e12.6,4x,f8.6,2x,f8.6,3x,f8.6,4x,f8.6,2x,f10.3)
 930  format(f10.6,4x,f10.3)
 940  format(f20.10)
 950  format(f10.6,10x,f20.10)
      end

c=-=================================================================
```

B.2. CYCLIC LOAD

B.2.1 Uniaxial Cyclic Compressive Loading Program

Program name: *UNICYC*

```
C=====================================================
C     THIS IS CODING FOR UNIAXIAL COMPRESSION UNDER CYCLIC LOADING, STRESS
C     CONTROL.
C     =====================================================
C     =====================================================
C     THIS program is applicable for a concrete strength 6100 psi only
C     =====================================================

       program UNICYC

        implicit double precision (a-h,o-z)

c      this is a stress control test for cyclic loading where convergence
c      is included.

      write(*,*)'the value of fc = ? '
      read(*,*) fc
      iii=0
      jjj=0
c     fc=6100
c     if (fc .eq. 2200) go to 2005
c     if (fc .eq. 4000) go to 2010

      if (fc .eq. 6100) go to 2020
c     if (fc .eq. 7000) go to 2025
c     if (fc .eq. 8000) go to 2030
c     if (fc .eq. 9700) go to 2035
c       fc = 2200

2005  delta1 = 0.2
c     delta1 = 0.8
        d     = 0.02
        beta  = 0.0605
      r0 = .04
      WRITE(*,*) 'VALUE OF SIGM'
      read(*,*) sigm

c     f0m1 IS ACTUALLY THE Ro^B VALUE
      if (sigm .ge. 1320) f0m1  = 0.06277
      if (sigm .ge. 1478) f0m1  = 0.07815
      if (sigm .ge. 1586) f0m1  = 0.0882
      if (sigm .ge. 1695) f0m1  = 0.096
      if (sigm .ge. 1803) f0m1  = 0.1
```

```
      if (sigm .ge. 1911) f0m1  = 0.11
      if (sigm .ge. 2019) f0m1  = 0.12
      go to 3000
c=======================================
c       fc = 4000


c       delta1 = 0.69
2010  delta1 = 0.2
        d     = 0.03
        beta  = 0.03767
      r0 = .08
      WRITE(*,*) 'VALUE OF SIGM'
      read(*,*) sigm
      if (sigm .ge. 2400) f0m1  = 0.0904
      if (sigm .ge. 2689) f0m1  = 0.11362
      if (sigm .ge. 2885) f0m1  = 0.128
      if (sigm .ge. 3082) f0m1  = 0.14
      if (sigm .ge. 3279) f0m1  = 0.15
      if (sigm .ge. 3475) f0m1  = 0.16
      go to 3000
c=======================================

c       fc = 6100

c       delta1 = 0.53
2020  delta1 = 0.2
        d     = 0.08
        beta  = 0.03053
      r0 = .08
      WRITE(*,*) 'VALUE OF SIGM'
      read(*,*) sigm
      if (sigm .ge. 3660) f0m1  = 0.13872
      if (sigm .ge. 4100) f0m1  = 0.1715
      if (sigm .ge. 4400) f0m1  = 0.1925
      if (sigm .ge. 4700) f0m1  = 0.21
      if (sigm .ge. 5000) f0m1  = 0.223
      if (sigm .ge. 5300) f0m1  = 0.235
      if (sigm .ge. 5600) f0m1  = 0.25
      go to 3000
c=======================================
c       fc = 7000

c       delta1 = 0.47
2025  delta1 = 0.2
        d     = 0.12
        beta  = 0.02895
         r0 = .08
       WRITE(*,*) 'VALUE OF SIGM'
      read(*,*) sigm

      if (sigm .ge. 4200) f0m1  = 0.161315
      if (sigm .ge. 4704) f0m1  = 0.1991
```

```
      if (sigm .ge. 5049) f0m1  = 0.223
      if (sigm .ge. 5393) f0m1  = 0.242
      if (sigm .ge. 5737) f0m1  = 0.255
      if (sigm .ge. 6081) f0m1  = 0.27
      if (sigm .ge. 6426) f0m1  = 0.28
      go to 3000
c=======================================
c       fc = 8000

c       delta1 = 0.33
2030  delta1 = 0.20
        d     = 0.22
        beta  = 0.02975
      r0 = .17
      WRITE(*,*) 'VALUE OF SIGM'
      read(*,*) sigm
      if (sigm .ge. 4800) f0m1  = 0.20068
      if (sigm .ge. 5377) f0m1  = 0.2488
      if (sigm .ge. 5770) f0m1  = 0.278
      if (sigm .ge. 6163) f0m1  = 0.3
      if (sigm .ge. 6557) f0m1  = 0.315
      if (sigm .ge. 6950) f0m1  = 0.34
      go to 3000
c=======================================
c       fc = 9700

c       delta1 = 0.22
2035   delta1 = 0.2
        d     = 0.42
        beta  = 0.0267
      r0 = .17
      WRITE(*,*) 'VALUE OF SIGM'
      read(*,*) sigm
      if (sigm .ge. 5820) f0m1  = 0.24047
      if (sigm .ge. 6519) f0m1  = 0.2945
      if (sigm .ge. 6996) f0m1  = 0.3263
      if (sigm .ge. 7473) f0m1  = 0.34
      go to 3000
c=======================================

 3000     continue
      WRITE(*,903) d,beta,f0m1,delta1,fc,sigm
 903  format(1x,f10.4,1x,f10.4,1x,f10.4,1x,f10.4)
c             WRITE(*,*) 'VALUE OF SIGM'
         read(*,*) xxx
      ratio = sigm/fc
      write(*,*) ratio

      rc    =1.29
c       r0 = 0.13*rc
      deltin=1-r0/rc
      e0    =57000*sqrt(fc)
      domega=0.0
      wbar  =0.0707
```

```
      wp    =0.05
      wrml  =0.05
      f0p   =0.0
      rrml  =0.0
      sig   =0.0
      fac=1.0
      delta=deltin

      open (1,file = 'cyclic.ou1',status='unknown')
      open (2,file = 'cyclic.ou2',status='unknown')
      write(1,950)
 950   format(4x,'iii',6x,'sig',11x,'f',10x,'wr')
      i=0
c       delta = .4924 (is based on caliberating the loading surface to
c                         reach the peak surface)
        do while (delta .gt. delta1)
         dsig=5.
         i=i+1

          do while(sig.lt.sigm)
          iii=iii+1
           sig=sig+dsig
           wmbar=.7
             f0m=f0m1
             x=r0
            f1=((f0m-x)**2)*(1.0-(((wbar-wmbar)**2)/((0.0707-wmbar)
     &      **2)))
              if(f1.lt.0.0) then
               f0=x
              else
                f0=x+sqrt(f1)

              end if
              if ((f0-f0p).lt.0.0) then
              f0=f0m
              end if
c**********************************************************************
           deltin=(rc-f0)/rc
           delr=1.0
            do while (delr.gt.0.001)
            wr=wrml
            r=rrml
            delta  = 1.0-(r*sqrt(2.0)/rc)
            if ((deltin-delta).le.0.0) then
            h=1.0e20
            else
            h=(d*delta)/(deltin-delta)
              end if
             f=r*sqrt(2.0)
c####################################################################
                if (f.le.f0) then
c####################################################################
                  domega=0.0
                else
```

```
                  jjj=jjj+1
c                 if(jjj .eq. 2)stop
c##################################################
                    aa=  2.0*r*dsig/ sig
                    bb= 5.0*beta*beta*sig*sig
                    cc=e0*(1.0-beta*wr)**6.0
                    domega = aa/(h-(bb/cc))
                 end if
                 wr=wp+domega
                    rr=(beta*sig*sig)/(e0*(1.0-(beta*wr))**5)
                    delr=abs(rr-r)
               rrm1=rr
               wrm1=wr
                    end do
             wp=wr
             epsed = sig/(((1.0-(beta*wr))**4)*e0)
             epse=sig/e0
             epsd=epsed-epse
             epsp=1.5*epsd
             epst=epse+epsd+epsp
              write(1,920)  iii,sig,f,wr
           end do
       wbar=(sqrt(2.0))*wp
         sig=0.0
         wpp=wp
         f0p=f0
         write (*,921) i,wbar,delta ,f0
        end do
920     format(i6,f12.2,4x,f8.6,4x,f8.6)
 921     format('unloadiing cycle',i5,5x,f15.6 ,5x,f15.6,5x,f15.6
     $ ,5x,f15.6,5x,f15.6 )
             end
```

B.2.2 Biaxial Cyclic Compressive Loading Program

Program name: ***BIACYC***

```
C      ====================================================
C      THIS IS A STRESS CONTROL CODE FOR BIAXIAL CYCLIC LOADING
C      ====================================================
C      ====================================================
C      THIS program is applicable for a concrete strength 6100 psi only
C      ====================================================

        program BIACYC


implicit double precision (a-h,o-z)
       dimension epsed(2),sig(3),epse(2),epsd(2),epsp(2),epst(2),rrf(3)
     &             ,btomg(3),domg(3),dsig(3)
```

```
c        ,ifdr(3)

      iii=0
      jjj=0
c      this is a stress control test for cyclic loading where convergence
c      is included.

      call data  (f0m1,fc,d,sigm,beta,rc,r0,anu,deltin,e0,domega,
     &     factor, wbar,wp,wrm1,f0p,rrm1,sig,fac,delta,btomg,dsig)



      open (1,file = 'cycl.ouy',status='unknown')
      open (2,file = 'cycl.ouz',status='unknown')
       write(1,950)
c 950   format(4x,'iii',6x,'sig',11x,'f',10x,'wr')
  950   format(4x,'iii',6x,'wbar',11x,'delta',10x,'f0')
      i=0
c        delta = .4924 (is based on caliberating the loading surface to
c                        reach the peak surface)
c         do while (delta .gt.  0.6)
        do while (delta .gt.  0.61)

         i=i+1
c         if(i .eq. 2)stop
          do while(sig(1).lt.sigm)
c      write(*,*)'ali2'
c           sig(1)= sig(1)+ dsig
      iii=iii+1
          sig(1)= sig(1)+ dsig(1)
          sig(2)=sig(2)+ dsig(2)
          sig(3)=0.0
c       write(*,*)sig
          wmbar=.7
c            f0m=0.450
            f0m=f0m1
            x=r0
           f1=((f0m-x)**2)*(1.0-(((wbar-wmbar)**2)/((0.0707-wmbar)
     &     **2)))
             if(f1.lt.0.0) then
              f0=x
             else
               f0=x+sqrt(f1)
c                f0=0.16
             end if

c            f1=((f0m-x1)**2)*(1.0-(((wbar-wmbar)**2)/((0.0707-wmbar)
c     &      **2)))
c             if(f1.lt.0.0) then
c              f0=x1
c             else
c              f0=x1+sqrt(f1)
```

```
c             end if


              if ((f0-f0p).lt.0.0) then

               f0=f0m

              end if
c*******************************************************************
           deltin=(rc-f0)/rc
           delr=1.0

c           do while (delr.gt.0.01)
             do while (delr.gt.0.001)
             wr=wrm1
c====================================
             if (sig(1) .lt. 20) go to 2050
c            if (r .gt. rrm1) stop
c====================================
 2050          r=rrm1
c\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
             delta  = 1.0-(rr*1/rc)
c              delta  = 1.0-(r*sqrt(2.0)/rc)
c\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
c       write (1,920) iii,delta,r,rc
              if ((deltin-delta).le.0.0) then
c              write (*,*) '1'
               h=1.0e20
              else
c              write (*,*) '2'
               h=(d*delta)/(deltin-delta)
              end if


             f=rr
c17      continue

                if (f.le.f0) then

                  domega=0.0

                else
                jjj=jjj+1
c               if(jjj .eq. 300)stop
c###############################################
      call domeg (h,dlamda,E0,beta,anu,domega,btomg,sig,rrf,domg,dsig)
c    &   ,ifdr)

c        write(1,920)jjj,domega

c###############################################


c       write(*,*)'ali3',domega
```

```
              end if

                 wr=wp+domega
c                  rr=2*(beta*sig*sig)/(2.0*e0*(1.0-(beta*wr))**3)
c                  rr=2*beta*sig*sig*(1-nu)/(e0*(1.0-(beta*wr))**3)
*=====================================
       call rrsub(E0,beta,anu,btomg, rrf,rr,sig )

*=====================================
                 delr=abs(rr-r)

            rrm1=rr
            wrm1=wr

          end do

           wp=wr
c           epsed(1) = sig(1)*(1-anu)/(((1.0-(beta*wr))**2)*e0)
            epsed(1) = sig(1)*(1-anu)/(((1.0-(beta*wr))**4)*e0)
           epse(1)=sig(1)/e0
           epsd(1)=epsed(1)-epse(1)
           epsp(1)=1.5*epsd(1)
           epst(1)=epse(1)+epsd(1)+epsp(1)
           ratio=float(i/13.0)
c      write(*,*)'ali4',sig
c                write (*,*) 'dsig',sig,wr,delta
c          if (sig . gt. 6000) read(*,*)asdf
c==========================================
c              write(1,920)  iii,sig(1),f,wr,f0
c               write(1,920)  iii,sig(1),rrf(1),rrf(2),rrf(3),rr,f
c==========================================
      write(1,920)iii,wp
          end do
c      write(*,*)'ali5'

c       unloading
c         write(1,*) sig,beta,E0
c        write(1,920)  i,wp,rr

c                if (i.eq.5) stop
       wbar=wp
c         wbar=(sqrt(2.0))*wp
c          write (*,*) 'unloading cycle' ,i
c           write (*,*) 'unloading cycle',i,delta
          write (*,921) i,wbar,delta ,f0
          write (1,921) i,wbar,delta ,f0
c         read(*,*)asd

       sig(1)=0.0
       sig(2)=0.0
       sig(3)=0.0
       wpp=wp
       f0p=f0
```

```
c        iii=0
      end do
        write (*,*) f0ml,'f0ml'
 920    format(i6,f20.10,4x,f8.6,4x,f8.6,4x,f8.6,4x,f8.6,4x,f8.6)
 921    format('unloadiing cycle',i7,5x,f15.12 ,5x,f15.6,5x,f15.6
     $ ,5x,f15.6,5x,f15.6 )
      end

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      subroutine data (f0ml,fc,d,sigm,beta,rc,r0,anu,deltin,e0,domega,
     &          factor, wbar,wp,wrml,f0p,rrml,sig,fac,delta,btomg,dsig)

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c          write(*,*)'value of f0ml = ?'
c          read(*,*) f0ml
      implicit double precision (a-h,o-z)
      dimension sig(3),btomg(3),dsig(3)
        f0ml = .13873
c       WRITE(*,*) 'VALUE OF factor'
c      read(*,*) factor
       factor=0.0
      fc = 6100
C        sigm= 5500
        sigm= 6100
         dsig(1)=5.0
         dsig(2)=factor*dsig(1)
         dsig(3)=0.0
c       WRITE(*,*) 'VALUE OF SIGM'C
c        read(*,*) sigm
      d     =.08
      beta =   0.03053
c      beta =  0.0176
      rc    =1.29
       r0 = 0.08
c       anu =0.2
       anu =0.1
      deltin=1-r0/rc
c      deltin=1-1*r0/rc
          e0    =57000*sqrt(fc)
c         e0    =5.2e6
      domega=0.0
      wbar  =0.0707
      wp    =0.05
      wrml  =0.05
      f0p   =0.0
      rrml  =0.0
      do 15 i=1,3
      btomg(i)= 1.0
 15     sig(i)   =0.0

      fac=1.0
      delta=deltin

      end
```

```
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      subroutine domeg (h,dlamda,E0,beta,anu,domega,btomg,sig,rrf,domg,
     &                     dsig)
c     &  ,ifdr)
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      implicit double precision (a-h,o-z)
      dimension dfr(3),drds(3,3),sig(3),btomg(3),omga(3)
     &         ,dromg(3,3),domg(3),xf(3), xf1(3),rrf(3),dsig(3)
c    &         ,ifdr(3)

      do 5 i = 1,3
  5   dfr(i)= rrf(i)/sqrt(rrf(1)**2+rrf(2)**2+rrf(3)**2)
c       write(*,900)(dfr(i),i=1,3)
c       read(*,*) x

      btomg(1) = 1-beta*omga(1)
      btomg(2) = 1-beta*omga(2)
      btomg(3) = 1-beta*omga(3)
c       write(*,*)beta,(omga(i),i=1,3)
c       write(*,900)(btomg(i),i=1,3)
c      read(*,*) x
c
      do 150 i = 1,3
      do 150 j = 1,3
      drds(i,j) = 0.0
 150  continue
      if (rrf(1) .eq.0.0) go to 155
      drds(1,1)=-2*btomg(1)*sig(1)*beta/(btomg(2)**2*btomg(3)**2)/E0
      drds(1,2)=2*btomg(2)**2*sig(2)*beta/(btomg(3)**2*btomg(1)**3)/E0
      drds(1,3)= 0.0



  155 continue
       if (rrf(2) .eq.0.0) go to 160
c       if (sig(1) .eq.0.0) go to 160
      drds(2,1)=2*btomg(1)**2*sig(1)*beta/(btomg(2)**3*btomg(3)**2)/E0
c       if (sig(2) .eq.0.0) go to 160
      drds(2,2)= -2*btomg(2)*sig(2)*beta/(btomg(3)**2*btomg(1)**2)/E0
      drds(2,3)=0.0

  160 continue
       if (rrf(3) .eq.0.0) go to 165
      drds(3,1)=2*btomg(1)**2*sig(1)*beta/(btomg(2)**2*btomg(3)**3)/E0
     &             -2*beta*anu*sig(2)/btomg(3)**3/E0

      drds(3,2)=2*btomg(2)**2*sig(2)*beta/(btomg(3)**3*btomg(1)**2)/E0
     &             -2*beta*anu*sig(1)/btomg(3)**3/E0

       drds(3,3)= 0.0
  165 continue
       do 200 k = 1,3
       if(dsig(k).eq.0.0) go to 205
       go to 200
```

```
 205   drds(1,k)=0.0
       drds(2,k)=0.0
       drds(3,k)=0.0
 200    continue

c         write(*,*) beta,sig(1),sig(2),e0
c        write(*,*) 'drds'
c        write(*,900)(drds(1,i),i=1,3)
c         write(*,900)(drds(2,i),i=1,3)
c        write(*,900)(drds(3,i),i=1,3)
c        read(*,*)x
c        do 105 j = 1.3
c        write(*,*) 'j' ,j
c        write(*,*) ifdr(j),'ifdr(j)',j
c        if(ifdr(j) .lt. 1) go to 106
c        do 110 k = 1,3
c       drds(j,k)=0.0
c 110   continue
c 106   continue
c 105   continue

c         write(*,900)(drds(1,i),i=1,3)
c         write(*,900)(drds(2,i),i=1,3)
c         write(*,900)(drds(3,i),i=1,3)
c         read(*,*)x
c=====================================
       do 10 i = 1,3
  10     xf(i)=0.0
       do 20 i = 1,3
       do 15 j = 1,3
  15   xf(i) = xf(i)+ dfr(j)* drds(j,i)
  20   continue

c       write(*,900)(xf(i),i=1,3)
       a2 = 0.0

       do 25 i = 1,3

  25   a2 = a2+ xf(i)* dsig(i)
c       write(*,*)'a2',a2
c=====================================
c       write(*,*)E0,anu,epsed(1),epsed(2)

       fact = sig(2)/sig(1)
c        sig(2) = 0.0
c       write(*,900)(sig(i),i=1,2),fact
c        read(*,*)x
       do 170 i = 1,3
       do 170 j = 1,3
       dromg(i,j) = 0.0
 170   continue

       if (rrf(1) .eq.0.0) go to 175
       dromg(1,1)=2*beta**2*sig(1)**2/(btomg(2)**2*btomg(3)**2)/(2*E0)+
```

```
     & 6* beta**2*sig(2)**2*btomg(2)**2/(btomg(3)**2*btomg(1)**4)/(2*E0)

      dromg(1,2)=-4*beta**2*sig(1)**2*btomg(1)/
     &            (btomg(2)**3*btomg(3)**2)/(2*E0)-
     &    4* beta**2*sig(2)**2*btomg(2)/(btomg(3)**2*btomg(1)**3)/(2*E0)

      dromg(1,3)=-4*beta**2*sig(1)**2*btomg(1)/
     &            (btomg(2)**2*btomg(3)**3)/(2*E0)+
     & 4* beta**2*sig(2)**2*btomg(2)**2/(btomg(3)**3*btomg(1)**3)/(2*E0)
 175  continue
       if (rrf(2) .eq.0.0) go to 180
      dromg(2,1)=dromg(1,2)
c      write(*,*)beta, sig(1),btomg(2),E0 ,'ali'
c      read(*,*) xd
      dromg(2,2)= 6* beta**2*sig(1)**2*btomg(1)**2/(btomg(2)**4*btomg(3)
     &             **2)/(2*E0)+
     &             2*beta**2*sig(2)**2/(btomg(3)**2*btomg(1)**2)/(2*E0)

      dromg(2,3)=4*beta**2*sig(1)**2*btomg(1)**2/(btomg(2)**3*btomg(3)
     &           **3)  /(2*E0)-
     &    4* beta**2*sig(2)**2*btomg(2)/(btomg(3)**3*btomg(1)**2)/(2*E0)
 180  continue
       if (rrf(3) .eq.0.0) go to 185

      dromg(3,1)=dromg(1,3)


      dromg(3,2)=dromg(2,3)


      dromg(3,3)=6* beta**2*sig(1)**2*btomg(1)**2/(btomg(2)**2*btomg(3)
     &             **4)/(2*E0)-
     &             12*beta**2*anu*sig(1)*sig(2)/btomg(3)**4/(2*E0)+
     &6* beta**2*sig(2)**2*btomg(2)**2/(btomg(3)**4*btomg(1)**2)  /(2*E0)
 185  continue

c      do 210 k = 1,3
c       if(domg(k).eq.0.0) go to 215
c       go to 200
c 215  dromg(1,k)=0.0
c      dromg(2,k)=0.0
c      dromg(3,k)=0.0
c 210   continue
c       do 115 j = 1.3
c       if(ifdr(j) .lt. 1) go to 105
c       do 120 k = 1,3
c 120   dromg(j,k)=0.0
c 115  continue
c      write(*,*) beta,sig(1),sig(2),e0
c       write(*,900)(dromg(1,i),i=1,3)
c       write(*,900)(dromg(2,i),i=1,3)
c       write(*,900)(dromg(3,i),i=1,3)
c      read(*,*)x
  900  format(3(2x,f15.10))
```

```
c====================================
      do 30 i = 1,3
 30     xf1(i)=0.0
      do 40 i = 1,3
      do 35 j = 1,3

 35   xf1(i) = xf1(i)+ dfr(j)* dromg(j,i)
 40   continue
      a3 = 0.0

      do 45 i = 1,3

 45   a3 = a3+ xf1(i)* dfr(i)

c====================================
      dlamda = a2/(h-a3)
c       write(*,*)'dlamda',dlamda ,(dfr(i),i=1,3)
c       read(*,*)x
c====================================
      do 50 i=1,3
c       write(*,*)'dlamda',dlamda ,dfr(i),i
      domg(i)= dlamda* dfr(i)
c       write(*,*)'dlamda',dlamda ,dfr(i),i
c      write(*,*)domg(i)
c 50        read(*,*)x
c        do 55 i=1.3
      omga(i)=omga(i)+domg(i)

 50   continue
c          write(*,900) ('domg')
c          write(1,900) (domg(i),i=1,3)
c          write(*,900) (omga(i),i=1,3)
c          read(*,*)x
c      iii=iii+1
      domega = sqrt(domg(1)**2+domg(2)**2+domg(3)**2)
c         domega = sqrt(domg(1)**2+domg(2)**2+domg(3)**2)/sqrt(2)
c             write(*,901) domega
c      read (*,*)r4

 901  format(2x,f10.8,2x,i10)
      end
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


      subroutine rrsub (E0,beta,anu,btomg, rrf,rr,sig )
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      implicit double precision (a-h,o-z)
            dimension rrf(3),btomg(3),sig(3)
c       write(*,*)'asd'
c          (page 6,7 in  biax11.doc)
c       write(*,*)E0 ,beta,anu,epsed(1),epsed(2)
c        iiii = iiii+1
c       write(*,*)'iiii',iiii
c       write(*,900) (btomg(i),i=1,3)
```

```
c       write(*,900) (rrf(i),i=1,3 )
      do 5 i = 1,3
c      ifdr(i) = 0
 5     continue
c      iii = iii+1
c       write(*,*)(btomg(i),i=1,3)
      rrf(1)=-btomg(1)*sig(1)**2*beta/(btomg(2)**2*btomg(3)**2*E0)
     &     +btomg(2)**2*sig(2)**2*beta/(btomg(3)**2*btomg(1)**3*E0)


c       write(*,*)(btomg(i),i=1,3)
      rrf(2)=btomg(1)**2*sig(1)**2*beta/(btomg(2)**3*btomg(3)**2*E0)
     &     -btomg(2)*sig(2)**2*beta/(btomg(3)**2*btomg(1)**2*E0)

      rrf(3)=btomg(1)**2*sig(1)**2*beta/(btomg(2)**2*btomg(3)**3*E0)
     &     -2*beta*anu*sig(1)*sig(2)/(btomg(3)**3*E0)
     &     +btomg(2)**2*sig(2)**2*beta/(btomg(3)**3*btomg(1)**2*E0)



c       write(*,*)(btomg(i),i=1,3)
c       write(*,900) rrf(1),rrf(2),rrf(3)
c      read (*,*)r4
      do 10 i = 1,3
      if (rrf(i) .gt. 0.0)  go to 10
       rrf(i) = 0.0
c       ifdr(i) = 1
 10   continue
c      rr = sqrt(rrf(1)**2+rrf(2)**2+rrf(3)**2)/sqrt(2)
       rr = sqrt(rrf(1)**2+rrf(2)**2+rrf(3)**2)
c       write(*,900)(btomg(i),i=1,3)
c         write(*,*) sig(1),beta,E0
c        write(*,900) (rrf(i),i=1,3  )
c      write(1,901) rr,iii
c      read (*,*)r4
  900  format(3(2x,f10.8))
  901  format(2x,f10.8,2x,i10)
c      write(*,*)'khlase'
      end
```

B.3. RESIDUAL STRENGTH OF UNIAXIAL COMPRESSION

B.3.1 Residual Strength Program

Program name: *RSDSTRN*

```
      ==========================================================
c     THIS IS A CODE FOR PREDICTION OF RESIDUAL CAPACITY OF A MEMBER IN
c     UNIAXIAL COMPRESSION AFTER BEING SUBJECTED TO A NUMBER OF STRESS
c     CYCLES < FATIGUE LIFE.
c     ==========================================================

c     ==========================================================
c     THIS program is applicable for a concrete strength 6100 psi only
c     ==========================================================

      program RSDSTRN

c       implicit real (a-h,o-z)
        implicit double precision (a-h,o-z)

c      this is a stress control test for cyclic loading where convergence
c      is included.


c           write(*,*)'value of f0m1 = ?'
c           read(*,*) f0m1
        iii=0
        jjj=0
        f0m1 = .21
      fc = 6100
        sigm= 4700
        sigm1 = 9000
c       WRITE(*,*) 'VALUE OF SIGM'C
c        read(*,*) sigm
      d     =.08
      beta =  0.03053
      beta1=.03855
      rc    =1.29
       r0 = 0.08
      deltin=1-r0/rc
c      deltin=1-sqrt(2)*r0/rc
          e0    =57000*sqrt(fc)
c         e0    =5.2e6
      domega=0.0
      wbar  =0.0707
      wp    =0.05
      wrm1  =0.05
      f0p   =0.0
      rrm1  =0.0
```

```
      sig   =0.0
      fac=1.0
      delta=deltin

      open (1,file = 'cycl.oul',status='unknown')
      open (2,file = 'cycl.ou2',status='unknown')
      write(*,*)'write the value of residual cycles'
      read(*,*)nres
c      nres=50
      i=0
c       delta = .4924 (is based on caliberating the loading surface to
c                       reach the peak surface)
c        do while (delta .gt. 0.61)
       do while (delta .gt. 0.1)
       iii=iii+1
        dsig=5.
        i=i+1
        if ( i.gt. nres) sigm = sigml
         if(i .gt.nres) beta=betal

         do while(sig.lt.sigm)
c          if(sigm .gt.sigml) beta=betal
          if ( i.gt. nres) write(*,*) sig
c      write(*,*)'ali2'
          sig=sig+dsig

c        write(*,*)sig
          wmbar=.7
c           f0m=0.450
            f0m=f0ml
            x=r0
           f1=((f0m-x)**2)*(1.0-(((wbar-wmbar)**2)/((0.0707-wmbar)
     &     **2)))
             if(f1.lt.0.0) then
              f0=x
             else
               f0=x+sqrt(f1)
c                f0=0.16
             end if

c          f1=((f0m-x1)**2)*(1.0-(((wbar-wmbar)**2)/((0.0707-wmbar)
c     &     **2)))
c            if(f1.lt.0.0) then
c             f0=x1
c            else
c             f0=x1+sqrt(f1)
c            end if


             if ((f0-f0p).lt.0.0) then

              f0=f0m

             end if
```

```
c***********************************************************************
           deltin=(rc-f0)/rc
           delr=1.0

c           do while (delr.gt.0.01)
             do while (delr.gt.0.001)
             wr=wrml

             r=rrml
             delta  = 1.0-(r*sqrt(2.0)/rc)
c             delta  = 1.0-(r*1/rc)
              if ((deltin-delta).le.0.0) then
c              write (*,*) '1'
               h=1.0e20
              else
c              write (*,*) '2'
               h=(d*delta)/(deltin-delta)
              end if

             f=r *sqrt(2.0)
c             if(sig .eq. 5400) go to 14
c             go to 17
c14             write(*,*) f,f0,'azx'
c       read(*,*)asdf
17       continue
c#####################################################################
                if (f.le.f0) then
c#####################################################################
                  domega=0.0
c                   write(*,*)'aabbcc'
c              write (*,*) '1'
                  else
c                  if (jjj .eq. 20) stop
                  jjj=jjj+1
c                 write (*,*) '123'
c                    domega =  (2.0*r*dsig/(sig))/
c     &              (h-(3.0*beta*beta*sig*sig)/
c     &              (2.0*e0*((1.0-beta*wr)**4.0)))
                    aa=  2.0*r*dsig/ sig
                    bb= 5.0*beta*beta*sig*sig
                    cc=e0*(1.0-beta*wr)**6.0
c                    iii=iii+1
                    domega = aa/(h-(bb/cc))
c       write(1,1050) domega,iii
1050   format(2x,f10.8,2x,i10)



c       write(*,*)'ali3',domega
                end if


                   wr=wp+domega
c                   rr=2*(beta*sig*sig)/(2.0*e0*(1.0-(beta*wr))**3)
```

```
                 rr=(beta*sig*sig)/(e0*(1.0-(beta*wr))**5)
                 delr=abs(rr-r)
c          write(1,1050) rr,iii
c                  read(*,*)as
            rrml=rr
            wrml=wr

          end do

           wp=wr
           epsed = sig/(((1.0-(beta*wr))**4)*e0)
           epse=sig/e0
           epsd=epsed-epse
           epsp=1.5*epsd
           epst=epse+epsd+epsp
           ratio=float(i/13.0)
c      write(*,*)'ali4',sig
c                write (*,*) 'dsig',sig,wr,delta
c          if (sig . gt. 6000) read(*,*)asdf
c         write(1,920)  iii,sig,f,wr
        write(1,920)  iii,sig,rr ,f
         end do
c      write(*,*)'ali5'

c       unloading
c      write(1,*) sig,beta,E0
       wbar=(sqrt(2.0))*wp
c       wbar=wp
c         write (*,*) 'unloading cycle' ,i
c          write (*,*) 'unloading cycle',i,delta
         write (*,921) i,wbar,delta ,f0

c          write (*,*) '1111111111111111111111'
c           if (i.gt.5) stop
c         read(*,*)asd
c       write(*,*)sig
       sig=0.0
       wpp=wp
       f0p=f0

      end do
c       write (*,*) f0m1,'f0m1'
 920   format(i6,f12.2,4x,f8.6,4x,f8.6)
 921   format('unloadiing cycle',i7,5x,f18.13 ,5x,f15.6,5x,f15.6
     $ ,5x,f15.6,5x,f15.6 )
      end
```

VITA

Ali MOHAMED Kamel Shaalan was born in Egypt, in 1963. He finished his high school in 1980. He jointed Faculty of Engineering, Cairo University in October 1980 and completed his B.Sc. in Civil Engineering in 1985.

He joined King Fahd University of Petroleum and Minerals as a part time graduate student in the Department of the Civil Engineering in 1992, where he obtained his M.Sc. in the field of structures.

He hopes that the work of his thesis will be of some benefit in the field of continuum damage simulation for compressive fatigue of high strength concrete.

IMAGE EVALUATION TEST TARGET (QA-3)

150mm

6"

APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved