PB2002-100099

Center for
Transportation Research
The University of Texas at Austin

RESEARCH REPORT 0-1831-2

CRCP-9: IMPROVED COMPUTER PROGRAM FOR MECHANISTIC ANALYSIS OF CONTINUOUSLY REINFORCED CONCRETE PAVEMENTS

Seong-Min Kim, Moon C. Won, and B. Frank McCullough

CENTER FOR TRANSPORTATION RESEARCH
BUREAU OF ENGINEERING RESEARCH
THE UNIVERSITY OF TEXAS AT AUSTIN

FEBRUARY 2001

Technical Report Documentation Page

1. Report No. 0-1831-2	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle CRCP-9: IMPROVED COMPUTER PROGRAM FOR MECHANISTIC ANALYSIS OF CONTINUOUSLY REINFORCED CONCRETE PAVEMENTS		5. Report Date February 2001
7. Author(s) Seong-Min Kim, Moon C. Won, and B. Frank McCullough		6. Performing Organization Code 8. Performing Organization Report No. Research Report 1831-2
9. Performing Organization Name and Address Center for Transportation Research The University of Texas at Austin 3208 Red River, Suite 200 Austin, TX 78705-2650		10. Work Unit No. (TRAIS) 11. Contract or Grant No. Project 0-1831
12. Sponsoring Agency Name and Address Texas Department of Transportation Research and Technology Transfer Section/Construction Division P.O. Box 5080 Austin, TX 78763-5080		13. Type of Report and Period Covered Research Report (9/99-8/00) 14. Sponsoring Agency Code

15. Supplementary Notes
Project conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration, and the Texas Department of Transportation.

16. Abstract

A new version of the CRCP computer program, CRCP-9, has been developed in this study. The numerical model of the CRC pavements was developed using finite element theories, the crack spacing prediction model was developed using the Monte Carlo method, and the failure prediction model was developed using probability theories. CRCP-9 uses two-dimensional numerical methods to calculate stresses and strains, which is a totally different approach compared with the previous computer program CRCP-8 that uses one-dimensional analytical methods to calculate them. The major characteristics of CRCP-9 that differ from CRCP-8 include consideration of nonlinear variations in temperature and drying shrinkage through the depth of the concrete slab, nonlinear bond-slip relationship between concrete and steel bars, viscoelastic effect of concrete, curling and warping effects, and the ability of changing locations of the longitudinal steel bars. Details of the theoretical background of mechanistic modeling have been presented in this report. The descriptions of the computer program and input and output procedures have also been explained. Sample analysis results from CRCP-9 have been shown and comparisons of the results between CRCP-9 and CRCP-8 have been made.

17. Key Words Continuously reinforced concrete pavement, finite element, crack spacing, punchout, bond-slip		18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.	
19. Security Classif. (of report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 116	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

CRCP-9: IMPROVED COMPUTER PROGRAM FOR MECHANISTIC ANALYSIS OF CONTINUOUSLY REINFORCED CONCRETE PAVEMENTS

by

Seong-Min Kim
Moon C. Won
B. Frank McCullough

Research Report No. 1831-2

Research Project 0-1831
Project Title: *Improvement of Concrete Pavement Performance Model*

Conducted for the

TEXAS DEPARTMENT OF TRANSPORTATION

in cooperation with the

U.S. DEPARTMENT OF TRANSPORTATION
Federal Highway Administration

by the

CENTER FOR TRANSPORTATION RESEARCH
Bureau of Engineering Research
THE UNIVERSITY OF TEXAS AT AUSTIN

February 2001

DISCLAIMERS

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Federal Highway Administration or the Texas Department of Transportation. This report does not constitute a standard, specification, or regulation.

There was no invention or discovery conceived or first actually reduced to practice in the course of or under this contract, including any art, method, process, machine, manufacture, design or composition of matter, or any new and useful improvement thereof, or any variety of plant, which is or may be patentable under the patent laws of the United States of America or any foreign country.

NOT INTENDED FOR CONSTRUCTION,
BIDDING, OR PERMIT PURPOSES

B. Frank McCullough, P.E. (Texas No. 19914)
Research Supervisor

ACKNOWLEDGMENTS

The researchers would like to acknowledge the expert assistance and guidance provided by the TxDOT project monitoring committee, which included G. Lantz (CSTC), G. Graham (CST-PAV), G. Lankes (CSTM), J. Kosel (CSTC), and Moon C. Won (CSTM).

Research performed in cooperation with the Texas Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

1.1 BACKGROUND AND OBJECTIVE

The first mechanistic model of continuously reinforced concrete pavement (CRCP), called CRCP-1, was developed in the mid-1970s under a study sponsored by the National Cooperative Highway Research Program (NCHRP) (Ref 1). This computer program evaluated the effect of such design variables as layer thicknesses and properties, concrete strength, and steel reinforcement, together with environmental and traffic loads, such as temperature variation, drying shrinkage, wheel loads, tire pressures, etc. In 1991, Won et al. developed an improvement to the CRCP program, CRCP-5, that simulates material variance to concrete tensile strength and includes fatigue failure models (Ref 2). The normalized curing curves were determined for different coarse aggregates commonly used in Texas pavements (Ref 3), and these curves and the calibrated failure prediction model were included in CRCP-7 (Ref 4). In 1995, previous versions of the CRCP programs were integrated into one program, CRCP-8, with simplification of the user input process (Ref 5). Although CRCP-8 has permitted pavement engineers to develop designs of the CRC pavements, there are some limitations owing to the simplified assumptions of the one-dimensional analysis. In 1996, Texas Department of Transportation (TxDOT) Project 0-1758 was conducted to expand the ability of the mechanistic model by incorporating the variations in temperature and moisture changes through the depth of concrete slab. As a result of the project, a two-dimensional finite element model was developed (Refs 6, 7). In 1998, TxDOT decided to extend the project to complete the development of a new mechanistic model, CRCP-9. CRCP-9 uses two-dimensional finite element theories to reduce the cost of computation, but in order to increase the accuracy of the 2-D model, three-dimensional analyses were performed first using a commercial finite element analysis program, ABAQUS (Ref 8), and the differences between 2-D and 3-D analysis results were investigated (Refs 9, 10).

The first version of the CRCP-9 computer program has recently been developed. The analysis engine of this mechanistic model was developed using finite element theories, the crack spacing prediction model using the Monte Carlo method, and the failure prediction model using probability theories. The previous computer program, CRCP-8, uses one-dimensional analytical methods to calculate stresses and strains, but CRCP-9 uses totally different two-dimensional numerical methods to calculate them. The major characteristics of CRCP-9 that differ from CRCP-8 include consideration of nonlinear variations in temperature and drying shrinkage through the depth of the concrete slab, nonlinear bond-slip relationship between concrete and steel bars, viscoelastic effect of concrete, curling and warping effects, and the ability to investigate the effect of placing the longitudinal steel bars at different depths. The objective of this report is to describe the details of the CRCP-9 development.

1.2 ORGANIZATION

This report consists of six chapters and four appendices. The background and objective of this study are presented in Chapter 1. The finite element model of CRC pavements and the material characteristics are explained in Chapter 2. In Chapter 3, crack spacing and fatigue failure prediction models are presented. Chapter 4 describes the computer program; input and output guides are also included. Sample analysis results obtained by CRCP-9 are presented in Chapter 5, while Chapter 6 includes summary, conclusions, and recommendations. Sample input and output files and the source code of the CRCP-9 computer program are listed in the appendices.

CHAPTER 2. NUMERICAL MODELING OF CRCP

2.1 FINITE ELEMENT MODEL OF CRCP

The configuration of the continuously reinforced concrete (CRC) pavement system is shown in Figure 2.1. There are longitudinal and transverse steel bars in the system. The major purpose of the use of longitudinal steel bars is to hold cracks tightly to provide structural continuities at the cracks. These steel bars are normally placed at the mid-depth of the concrete slab. The transverse steel bars are used mainly to keep the longitudinal steel bars in the proper places and to keep the longitudinal cracks in case they occur.

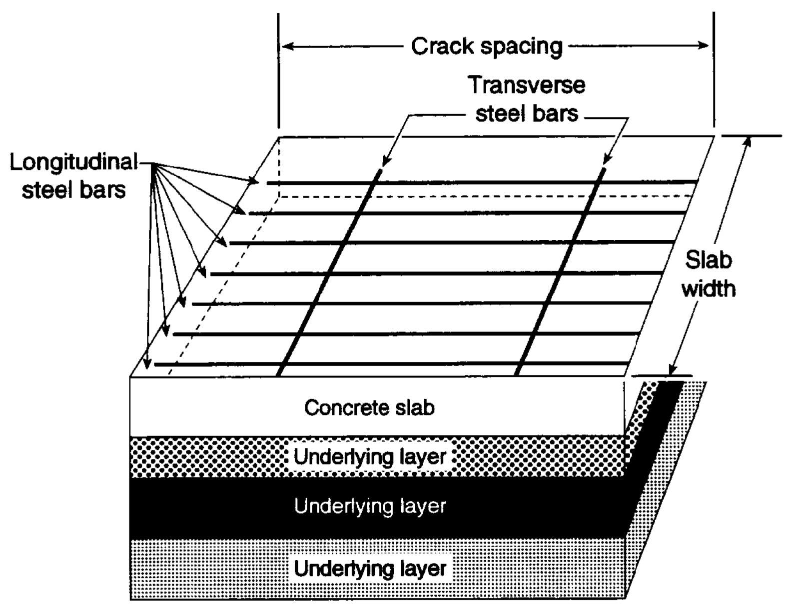


Figure 2.1. Configuration of CRC pavement

The finite element method has been used to model the CRC pavement systems. The concrete slab has been discretized using two-dimensional plane elements with the thickness of the element as the longitudinal steel bar spacing that is the distance between two longitudinal steel bars. The shape functions for a four-noded bilinear rectangular element, which has ξ and η coordinates of (-1, 1), (-1, -1), (1, 1), and (1, -1) at nodes 1, 2, 3, and 4, respectively, can be written as

$$f_i = \frac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i) \tag{2.1}$$

where ξ_i and η_i are the coordinates at node i. The displacements u and v in the two principal directions can be written as

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 & 0 \\ 0 & f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} = \mathbf{N}^{\mathbf{T}}\mathbf{U} \qquad (2.2)$$

For a rectangular element with the side lengths of a and b in the ξ and η directions, respectively, strains can be determined by

$$\begin{Bmatrix} \varepsilon_\xi \\ \varepsilon_\eta \\ \gamma_{\xi\eta} \end{Bmatrix} = \begin{bmatrix} \frac{2}{a}\frac{\partial f_1}{\partial \xi} & 0 & \frac{2}{a}\frac{\partial f_2}{\partial \xi} & 0 & \frac{2}{a}\frac{\partial f_3}{\partial \xi} & 0 & \frac{2}{a}\frac{\partial f_4}{\partial \xi} & 0 \\ 0 & \frac{2}{b}\frac{\partial f_1}{\partial \eta} & 0 & \frac{2}{b}\frac{\partial f_2}{\partial \eta} & 0 & \frac{2}{b}\frac{\partial f_3}{\partial \eta} & 0 & \frac{2}{b}\frac{\partial f_4}{\partial \eta} \\ \frac{2}{b}\frac{\partial f_1}{\partial \eta} & \frac{2}{a}\frac{\partial f_1}{\partial \xi} & \frac{2}{b}\frac{\partial f_2}{\partial \eta} & \frac{2}{a}\frac{\partial f_2}{\partial \xi} & \frac{2}{b}\frac{\partial f_3}{\partial \eta} & \frac{2}{a}\frac{\partial f_3}{\partial \xi} & \frac{2}{b}\frac{\partial f_4}{\partial \eta} & \frac{2}{a}\frac{\partial f_4}{\partial \xi} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} = \mathbf{BU} \qquad (2.3)$$

The element stiffness matrix can be obtained by

$$\mathbf{K} = \frac{ab}{4} t_w \int_{-1}^{1} \int_{-1}^{1} \mathbf{B}^{\mathbf{T}}\mathbf{D}\mathbf{B} \; d\xi d\eta \qquad (2.4)$$

where t_w is the thickness of the plane element, which is the distance between longitudinal steel bars (longitudinal steel bar spacing), and $\mathbf{D}$ is a material property matrix. For an isotropic material, if the problem is plane strain, $\mathbf{D}$ can be written as

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \qquad (2.5)$$

where E is Young's modulus of elasticity and ν is Poisson's ratio. For a plane stress problem, **D** can be obtained by

$$\mathbf{D} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{2.6}$$

The longitudinal steel bar has been discretized using frame elements. Assume that an element with a length of a has a constant cross-sectional area (A), moment of inertia (I), and modulus of elasticity (E) within the element. Node 1 has degrees of freedom of u_1 in the principal direction, w_1 in the direction perpendicular to u_1, θ_1 in the counterclockwise direction, and node 2 has u_2, w_2, θ_2 in the same directions as u_1, w_1, θ_1, respectively, then the shape functions can be written as

$$\begin{aligned} f_1 &= 1 - \frac{\xi}{a} \\ f_2 &= 1 - \frac{3\xi^2}{a^2} + \frac{2\xi^3}{a^3} \\ f_3 &= \xi - \frac{2\xi^2}{a} + \frac{\xi^3}{a^2} \\ f_4 &= \frac{\xi}{a} \\ f_5 &= \frac{3\xi^2}{a^2} - \frac{2\xi^3}{a^3} \\ f_6 &= -\frac{\xi^2}{a} + \frac{\xi^3}{a^2} \end{aligned} \tag{2.7}$$

The displacements can be written as

$$\{u\} = \begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \end{bmatrix}\begin{Bmatrix} u_1 \\ w_1 \\ \theta_1 \\ u_2 \\ w_2 \\ \theta_2 \end{Bmatrix} = \mathbf{N}^{\mathbf{T}}\mathbf{U} \tag{2.8}$$

The strain and the curvature can then be obtained by

$$\begin{Bmatrix} \varepsilon_\xi \\ \kappa \end{Bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \xi} & 0 & 0 & \frac{\partial f_4}{\partial \xi} & 0 & 0 \\ 0 & \frac{\partial^2 f_2}{\partial \xi^2} & \frac{\partial^2 f_3}{\partial \xi^2} & 0 & \frac{\partial^2 f_5}{\partial \xi^2} & \frac{\partial^2 f_6}{\partial \xi^2} \end{bmatrix} \begin{Bmatrix} u_1 \\ w_1 \\ \theta_1 \\ u_2 \\ w_2 \\ \theta_2 \end{Bmatrix} = \mathbf{BU} \tag{2.9}$$

The element stiffness matrix can be obtained by

$$\mathbf{K} = \int_0^a \mathbf{B^T D B}\, d\xi = \frac{E}{a^3} \begin{bmatrix} Aa^2 & 0 & 0 & -Aa^2 & 0 & 0 \\ 0 & 12I & 6aI & 0 & -12I & 6aI \\ 0 & 6aI & 4a^2 I & 0 & -6aI & 2a^2 I \\ -Aa^2 & 0 & 0 & Aa^2 & 0 & 0 \\ 0 & -12I & -6aI & 0 & 12I & -6aI \\ 0 & 6aI & 2a^2 I & 0 & -6aI & 4a^2 I \end{bmatrix} \tag{2.10}$$

The element stiffness matrices should be assembled to create a total stiffness matrix of the system.

When there are initial strains without any other loads, the equilibrium equation can be written as

$$\int_V \mathbf{B^T D B}\, dV\ \mathbf{U} = \int_V \mathbf{B^T D} \varepsilon_0\, dV \tag{2.11}$$

where

V = volume of element,

$\boldsymbol{\varepsilon}_0$ = initial concrete strain vector due to temperature and drying shrinkage changes.

The right-side term of the above equation is the load vector that makes the displacements. The initial strain vector resulting from temperature change and drying shrinkage for the plane strain problem can be written as

$$\begin{Bmatrix} \varepsilon_{x0} \\ \varepsilon_{y0} \\ \gamma_{xy0} \end{Bmatrix} = (1+\nu) \begin{Bmatrix} \alpha\, \Delta T + \varepsilon_{sh} \\ \alpha\, \Delta T + \varepsilon_{sh} \\ 0 \end{Bmatrix} \tag{2.12}$$

where α is the coefficient of thermal expansion, ΔT is the change in temperature from the reference temperature, and ε_{sh} is the drying shrinkage strain (negative sign for shrinkage). For the plane stress problem, the initial strain vector can be obtained by

$$\begin{Bmatrix} \varepsilon_{x0} \\ \varepsilon_{y0} \\ \gamma_{xy0} \end{Bmatrix} = \begin{Bmatrix} \alpha \, \Delta T + \varepsilon_{sh} \\ \alpha \, \Delta T + \varepsilon_{sh} \\ 0 \end{Bmatrix} \tag{2.13}$$

Finally, the stresses can be obtained by

$$\sigma = \mathbf{D}(\mathbf{BU} - \varepsilon_{\mathbf{0}}) \tag{2.14}$$

The stresses are calculated at the integration points and the average values are used for each element.

The bond-slip between concrete and longitudinal steel bar has been modeled using spring elements between the nodes at contact positions with the active degree of freedom in the longitudinal direction. The stiffness of the spring element can be obtained by

$$K = k_{bond} A_{contact} \tag{2.15}$$

where k_{bond} is the bond stiffness per unit area, which is a slope of the curve representing the relationship between bond stress and slip. The contact area, $A_{contact}$, between concrete and steel is defined by

$$A_{contact} = \pi \, d \, (\frac{a_L}{2} + \frac{a_R}{2}) \tag{2.16}$$

where d is a diameter of steel bar, and a_L and a_R are horizontal lengths of the elements on the left and right sides of a contact node, respectively. In this study, the bond-slip relationship can be defined as several different approaches such as linear, bilinear, linear with ultimate slip, and bilinear with ultimate slip. As shown in Figure 2.2 in the linear bond-slip model, it is assumed that the bond stress increases with increasing the relative movement. In the linear bond-slip model with an ultimate slip, there will be no connection between concrete and steel bar after the ultimate slip. In the bilinear bond-slip model, the stiffness of bond stress decreases after a certain value of relative movement, which is a yield slip. The bilinear bond-slip model with an ultimate slip model is also considered.

The frictional bond-slip that occurs at the interface between concrete and base layer has also been modeled using spring elements. The spring stiffness can also be obtained by Eq. 2.15 with the contact area as

$$A_{contact} = t_w \, (\frac{a_L}{2} + \frac{a_R}{2}) \tag{2.17}$$

where t_w is the thickness of the plane element. The relationship between bond stress and slip is assumed to be linear. The effect of the nonlinear bond-slip model was investigated in previous research (Ref 10) and it was found that the nonlinear modeling of this frictional bond-slip affected the analysis results very little. Therefore, the frictional bond-slip between concrete and base has been assumed to be linear in this study. The frictional bond-slip between concrete and base (or subbase if there is no base) depends on base layer types. Five different subbase types of flexible, asphalt-stabilized, cement-stabilized, lime-treated clay, and untreated clay subbases are used in the computer program, and the typical values of bond-slip stiffness have been defined as listed in Table 2.1 (Ref 5).

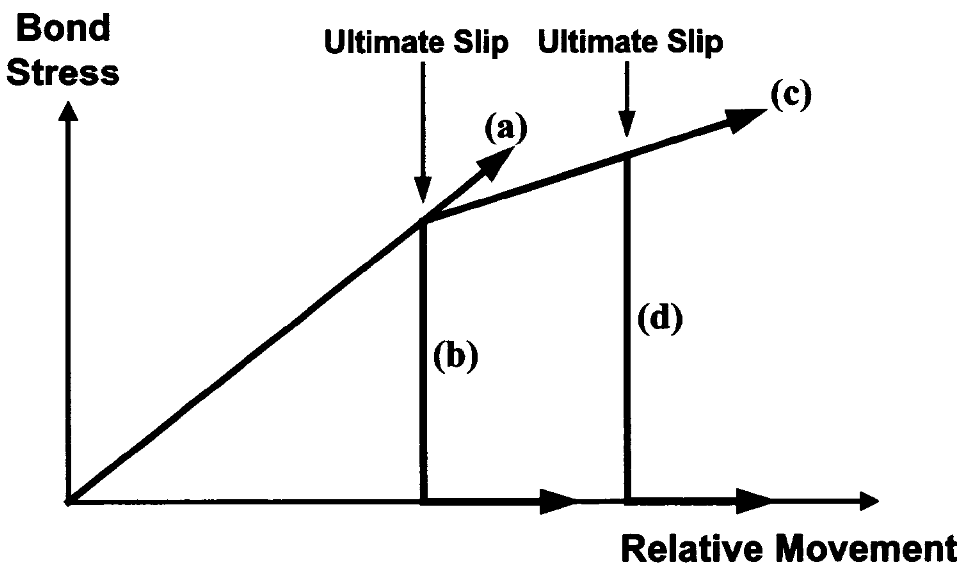


Figure 2.2. Bond-slip relationships: (a) linear; (b) linear with ultimate slip; (c) bilinear; (d) bilinear with ultimate slip

Table 2.1. Bond-slip stiffness for various subbase types

Subbase Types	Bond-Slip Stiffness / Unit Area (pci)
Flexible	145.5
Asphalt-Stabilized	55.9
Cement-Stabilized	15400
Lime-Treated Clay	154.5
Untreated Clay	22

The underlying layers have been modeled using vertical spring elements. If any side of the concrete slab curls up, there can be a gap between the slab and the base layer. To properly model this curling effect, tensionless vertical springs that can sustain only the compressive forces have been developed. In the computer program, all vertical springs are first assumed to have connections with a concrete slab and the pavement system is analyzed. After the analysis, the stresses in the vertical springs are investigated. If there is any vertical spring in tension, that spring is removed from the system and another analysis is performed with a new system. This procedure continues until there is no vertical spring in tension. It is noted that the curling effect in the CRC pavement is significant when the transverse crack spacing is long and the temperature variation between the top and bottom of the concrete slab is large (Ref 10). The stiffness of the vertical spring under the bottom of the concrete slab can be obtained by multiplying the vertical stiffness of underlying layers per unit area by the contact area defined in Eq. 2.17.

The boundary conditions of the finite element model should be correctly defined to obtain viable results. In this model, it is assumed that at cracks, there are no restraints for concrete and no longitudinal and rotational displacements for longitudinal steel bars. At the center of the two transverse cracks, it is assumed that vertical degrees of freedom exist and the longitudinal and rotational displacements are restrained. Figure 1 shows the finite element model developed in this research. To find the optimal size of the finite element, the convergence test has been performed and the element size has been selected as 38.1 mm (1.5 in.) for each side. Figure 2.2 shows the finite element model developed in this study.

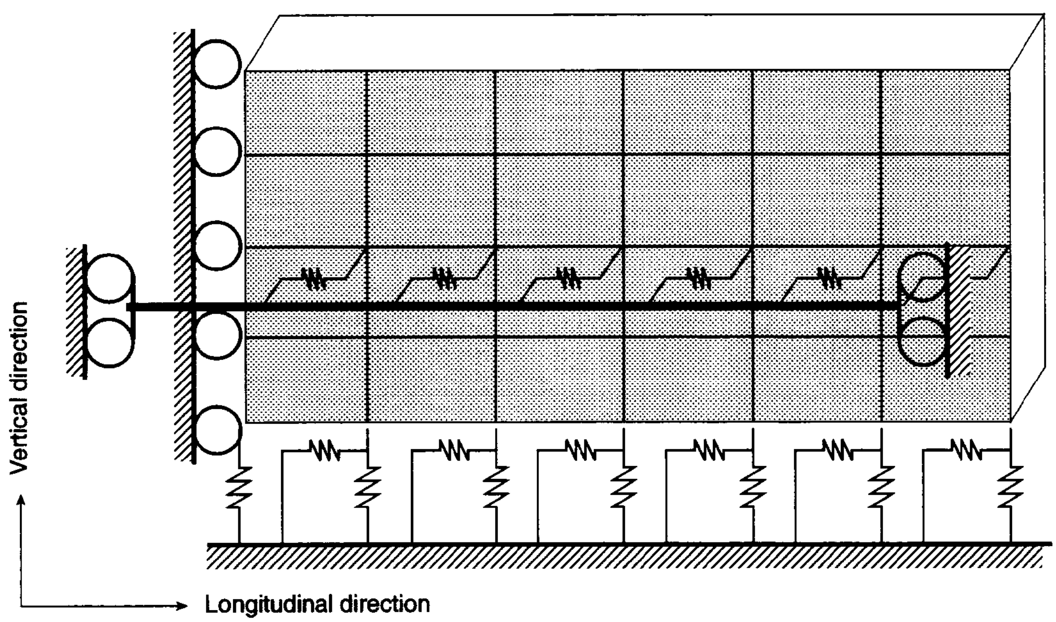


Figure 2.3. Finite element model of CRCP

The accuracy of the two-dimensional finite element model has been investigated by comparison with the three-dimensional model (Ref 10). It is noted that the stresses are slightly overestimated if the plane strain elements are used and are slightly underestimated with the plane stress elements compared with the 3-D analysis results. The differences between the stresses from 2-D and 3-D models are negligible with the plane stress elements. In the 3-D models, however, the stresses tend to increase near the edge or longitudinal joint on the transverse steel bar locations within about 10 inches (25 cm). This phenomenon becomes clearer when the bond between concrete and transverse steel bars is strong. Except in those regions, the 2-D models have predicted the results from 3-D models quite well (Ref 11).

2.2 WHEEL LOAD STRESS CALCULATION

If the external load is applied to any finite element node, this load will act as a line load because the finite element model developed in this study is two-dimensional. Therefore, the stresses due to the wheel load applications have been obtained using Westergaard equations in the interior of a slab (Ref 12) as

$$\sigma = \frac{3(1+\nu)P}{2\pi h^2}(\ln\frac{l}{b} + 0.6159) \tag{2.18}$$

where h is the thickness of the slab (inches), ν is Poisson's ratio, and P is the magnitude of the load (lbs). The radius of relative stiffness l is defined by

$$l = \left[\frac{Eh^3}{12(1-\nu^2)k}\right]^{0.25} \tag{2.19}$$

where E is the modulus of elasticity and k is the modulus of subgrade reaction. In Eq. 2.18, b is defined by

$$b = a \qquad \text{when } a \geq 1.724h \tag{2.20}$$

$$b = \sqrt{1.6a^2 + h^2} - 0.675h \qquad \text{when } a < 1.724h \tag{2.21}$$

where a is the radius of the circular loaded area.

The stresses caused by the environmental loads are calculated by using the finite element model, the stresses caused by the external wheel loads are calculated by the above Westergaard equations, and those stresses are added to obtain the concrete stresses due to the combined effects in the slab. Figure 2.4 shows concrete stresses through the depth of the concrete slab for various temperature conditions. As shown in the figure, the maximum stress can occur at the bottom, not always at the surface, and can occur when the temperature increases, not always when the temperature drops. In this case, a new crack will be initiated from the bottom. In order to include this phenomenon, the daily maximum temperature information is also needed in addition to the daily minimum temperature information. It is noted that further studies are being conducted to obtain more realistic wheel load stresses including effects of discontinuities at cracks and multiple wheel loads.

2.3 MATERIAL PROPERTIES

The changes in temperature and drying shrinkage occur slowly. Because of these slow changes in environmental loads, the creep of strain and the relaxation of stress in concrete may occur. There are a number of concepts used to analyze the creep effect for concrete (Ref 13). In this study, the creep effect has been considered using the effective modulus method (Ref 14). In this method, the creep of concrete is accounted for by reducing the elastic modulus of concrete (effective modulus or reduced modulus) as

$$E_{eff}(t) = \frac{E(t_0)}{1+\phi(t,t_0)} \tag{2.22}$$

where

t = time of consideration,

t_0 = time at first application of load,

$E(t_0)$ = modulus of elasticity at time t_0, and

$\phi(t,t_0)$ = creep coefficient at time t for concrete loaded at time t_0, and defined in this study by

$$\phi(t,t_0) = \phi_{max}[1-(1-\phi_x)^{\frac{t}{t_x}}] \tag{2.23}$$

where

ϕ_{max} = maximum creep ratio to instantaneous elastic strain,

ϕ_x = ratio to ϕ_{max} to define a point on creep curve, and

t_x = time corresponding to ϕ_x.

The effects of variables in Eq. 2.23 on the analysis results have been investigated in previous research (Ref 6).

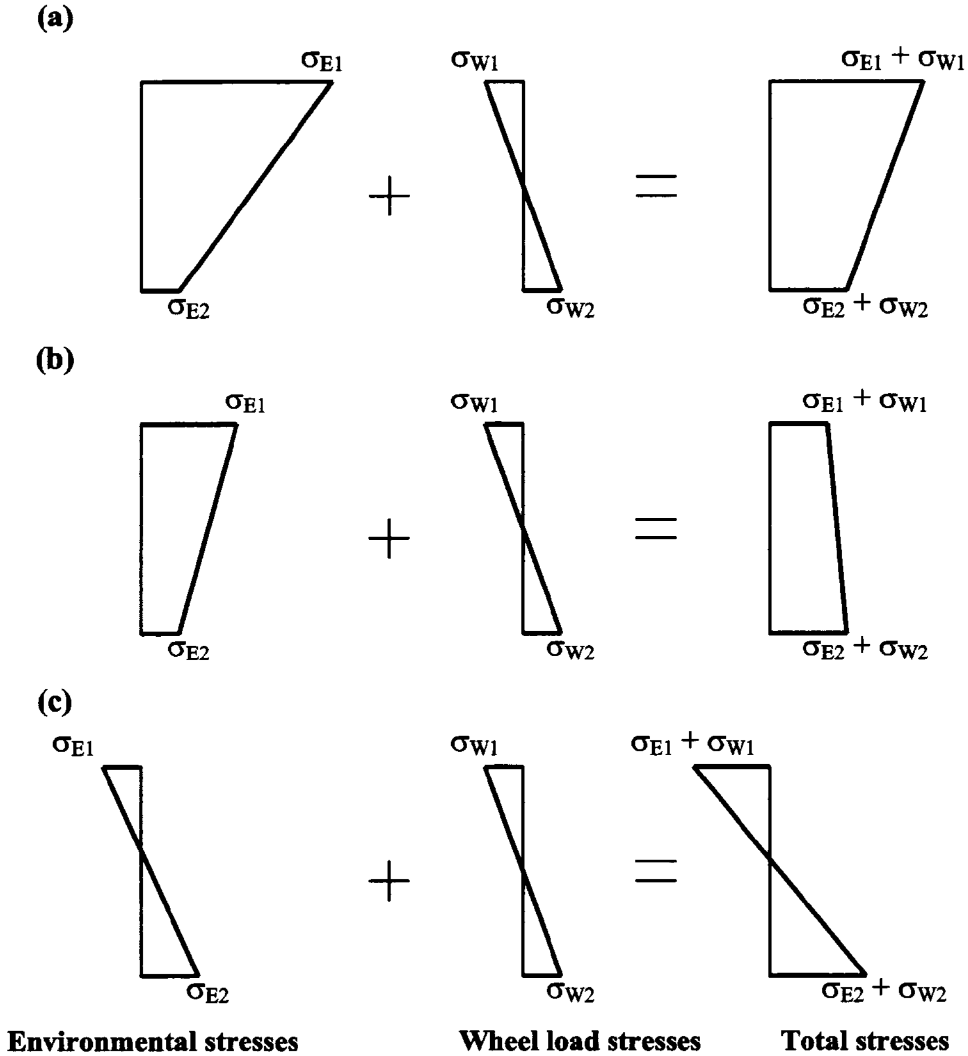


Figure 2.4. Stress distribution through depth: (a) when temperature drops and maximum total stress occurs at surface; (b) when temperature drops and maximum total stress occurs at bottom; (c) when temperature increases and maximum total stress occurs at bottom

Concrete material properties such as elastic modulus, tensile strength, compressive strength, and drying shrinkage change with time. The equations for variations of these material properties with time were developed for concrete with various coarse aggregate types in previous research (Ref 3) and are written by

$$F(t) = A(2 - e^{-Bt} - e^{-Ct}) \quad (2.24)$$

where

t = time of curing (days),
e = base of natural log,
$F(t)$ = concrete properties such as elastic modulus, tensile and compressive strengths, and drying shrinkage at time t, and
A, B, and C = coefficients of curvature specific to a given aggregate.

The above equation is used to determine absolute values for material properties. In order to estimate the material properties at a curing time t relative to a chosen final curing time t_f, a normalized model is required. Using a final curing time of 28 days for elastic modulus, tensile strength, and compressive strength, and a final curing time of 256 days for drying shrinkage, normalized models could be developed from the above equation by dividing the A coefficient by the value of the respective material properties at a final curing time t_f.

$$F_N(t) = \mathrm{N}_{28}(2 - e^{-\mathrm{B}t} - e^{-\mathrm{C}t}) \tag{2.25}$$

$$Z_N(t) = \mathrm{N}_{256}(2 - e^{-\mathrm{B}t} - e^{-\mathrm{C}t}) \tag{2.26}$$

where

$F_N(t)$ = normalized concrete properties of elastic modulus, tensile strength, and compressive strength at time t,
$Z_N(t)$ = normalized drying shrinkage at time t,
N_{28}, N_{256}, B, and C = coefficients of curvature specific to each aggregate.

In the computer program, eight different coarse aggregate types of limestone, siliceous river gravel (SRG), granite, dolomite, Vega, Bridgeport, western tascosa, and Ferris are used. The coefficients of A, B, C, N_{28}, and N_{256}, and the thermal expansion coefficients for concrete with different coarse aggregates are listed in Table 2.2. The coefficient of thermal expansion of concrete depends on coarse aggregate types and the typical value is used for each aggregate type as listed in Table 2.2.

Table 2.2. Coefficients for Equations 2.25 and 2.26

		Aggregate Types							
		Granite	**Dolomite**	**Vega**	**Bridgeport / Tin Top**	**Western Tascosa**	**Ferris**	**Limestone**	**Siliceous River Gravel**
Coefficient of thermal expansion (/F)		5.74E-6	5.9E-6	6.5E-6	4.84E-6	6.15E-6	5.44E-6	6.29E-6	8.18E-6
Elastic modulus (psi)	A	1678000	2324000	1882000	1992000	1803000	1979000	1802000	2282000
	B	0.78	0.485	0.301	0.688	0.405	0.738	0.535	0.574
	C	1.65E14	3.537	1.574	2.00	97.056	2.67E12	110.46	61755.1
	N_{28}	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Tensile strength (psi)	A	266.46	247.06	221.08	221.85	216.01	241.94	217.83	231.07
	B	0.15	0.261	0.302	0.332	0.198	0.137	0.177	0.267
	C	1.05	1.094	0.3014	0.723	2.505	2.479	1.068	0.468
	N_{28}	0.504	0.5	0.5	0.5	0.501	0.505	0.502	0.5
Compres. strength (psi)	A	2570.8	2236.7	1995.3	2038.2	2068.5	2000.1	2550.57	2445.25
	B	0.096	0.231	0.367	0.582	0.214	0.206	0.115	0.182
	C	0.623	0.562	0.367	0.220	0.647	0.801	0.490	0.473
	N_{28}	0.5176	0.5009	0.4978	0.4980	0.4998	0.5014	0.5102	0.5020
Drying shrinkage	A	3.2123 E-4	2.5206 E-4	2.3519 E-4	3.4362 E-4	3.58456 E-4	3.2723 E-4	2.291 E-4	1.9839 E-4
	B	0.0851	0.04062	0.03948	0.0328	0.03109	0.0745	0.0398	0.0619
	C	0.001	0.00155	0.01255	0.00069	0.00072	0.00119	0.00754	0.005
	Z_{256}	0.8112	0.7569	0.5146	0.8582	0.8600	0.7828	0.5403	0.5636

CHAPTER 3. CRACK SPACING AND PUNCHOUT PREDICTION MODELS

3.1 CRACK SPACING PREDICTION

A crack will occur when and where the concrete stress exceeds the tensile strength of concrete. If the concrete slab is assumed to be homogeneous, the new crack will occur at the center of the two previously formed transverse cracks because the maximum concrete stress occurs at the center. However, because the tensile strength of concrete is governed by the weakest element in it, there exists variation in concrete tensile strength from location to location. The concrete tensile strength depends largely on the bond characteristics between cement paste and the aggregate surface, which depends on local conditions, such as aggregate surface shape, texture, and the existence of voids between cement paste and the aggregate surface.

To include the effect of the variation of the tensile strength along the pavement length in the model, the concrete tensile strength at each finite element is selected randomly using a normal distribution because the concrete tensile strength distribution along the pavement length is reported to be sufficiently close to the normal distribution (Refs 2, 15). Once the tensile strength at each finite element is determined and the stresses are calculated from the model, the difference between the tensile strength and the concrete stress is obtained at each finite element. Where the concrete stress exceeds the tensile strength and the difference between them is the largest, a new crack will occur. With the obtained new crack spacings, the program calculates again the concrete tensile stresses and compares these with the tensile strength. If the concrete stresses exceed the concrete tensile strength, then another new crack will occur at the point where the largest difference between the stress and the strength exists. The program repeats the same procedure until the concrete stresses are smaller than the concrete tensile strength. This methodology is known as the Monte Carlo method (Ref 2) and Figure 3.1 shows this approach. As for initial lengths of the slab for the analysis, several different spacings were evaluated for their effects on the analysis results.

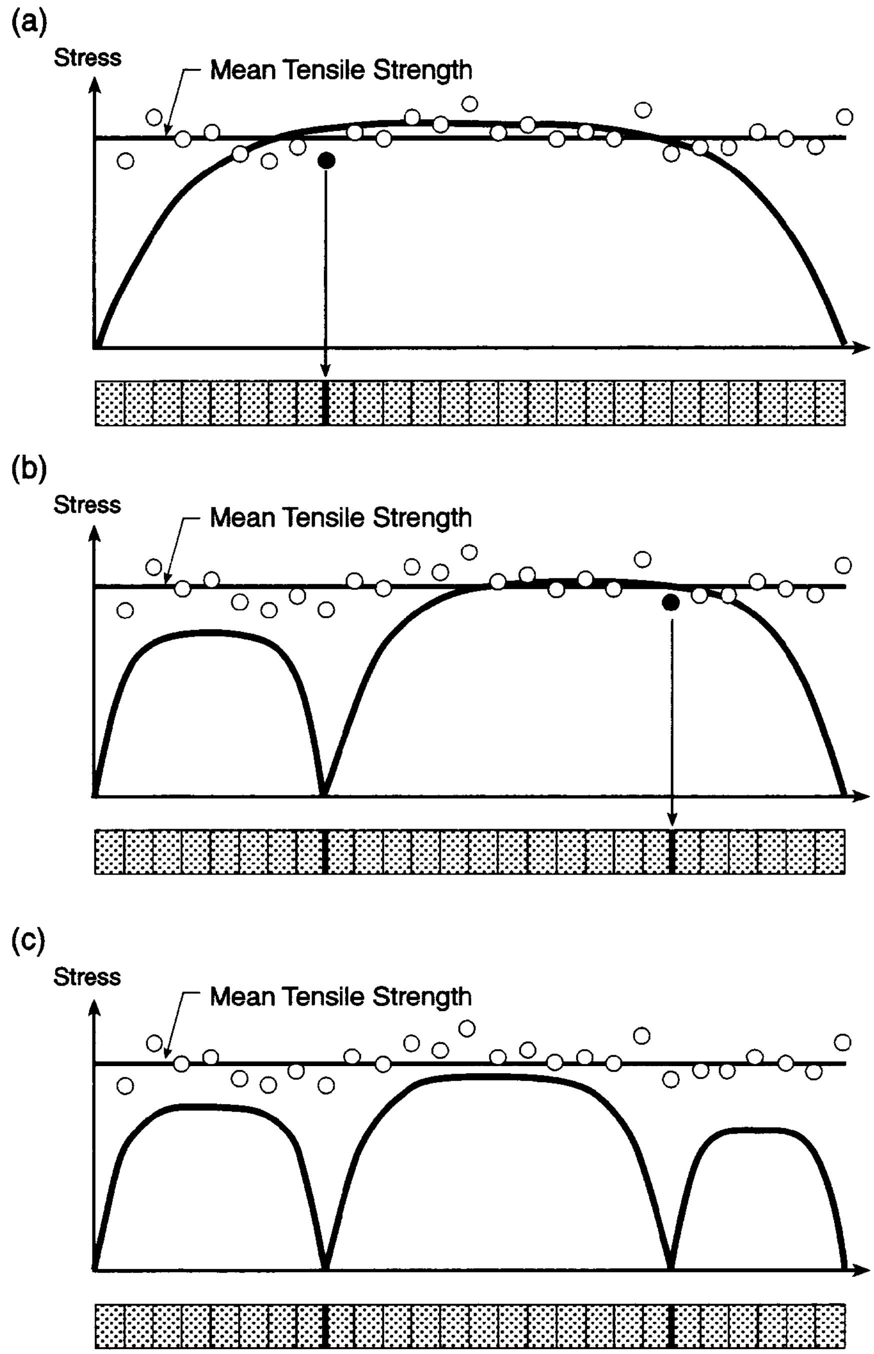


Figure 3.1. Methodology of crack spacing prediction

3.2 PUNCHOUT PREDICTION

There are structural failures and functional failures in the pavement systems (Ref 16). The structural failures lead to functional failures in continuously reinforced concrete (CRC) pavements and the major failure manifestation is the punchout (Ref 17). The punchout is a structural failure in which a small segment of pavement is loosened from the main body and displaced downward under traffic. As shown in Figure 3.2, the punchout occurs when a new longitudinal crack forms near a previous longitudinal crack or near the edge of the pavement so that the segment bounded by two closely spaced transverse cracks, a longitudinal crack, and the pavement edge or a previously formed longitudinal crack is loosened from the main body of the pavement. Even though the punchout development mechanism is complicated, it is assumed that the longitudinal crack is the most significant contributing factor in the punchout development. Therefore, it is assumed that longitudinal cracks result in punchouts. Once the crack spacings along the pavement length are obtained from the analysis, the transverse stresses are calculated for each crack spacing (Ref 2) by

$$s = e^{9.8474} D^{-1.8143} X^{-0.4477} \tag{3.1}$$

where

s = stress in transverse direction for 9000 lb single wheel load (psi),
e = base of natural log,
D = slab thickness (in.), and
X = crack spacing (ft).

The most widely used form of the fatigue failure equation is

$$N = C_1 \left(\frac{f}{s}\right)^{C_2} \tag{3.2}$$

where

N = number of load applications,
f = flexural strength,
s = flexural stress from Eq. 3.1, and
C_1 and C_2 = coefficients

Using the above equation, the number of load applications corresponding to various probabilities of fatigue failure is calculated. After the relationship between the fatigue failure (or punchouts) and load applications has been obtained for each crack spacing, the final fatigue failure versus load application curve can be obtained by adding each curve from each crack spacing. The appropriate coefficients of C_1 and C_2 in Eq. 3.2 for CRC pavements have been found by Suh et al. (Ref 4). The widely used value of C_2 is 4. These values change with the percentage of reliability and the swelling conditions of subgrade soil as listed in Table 3.1.

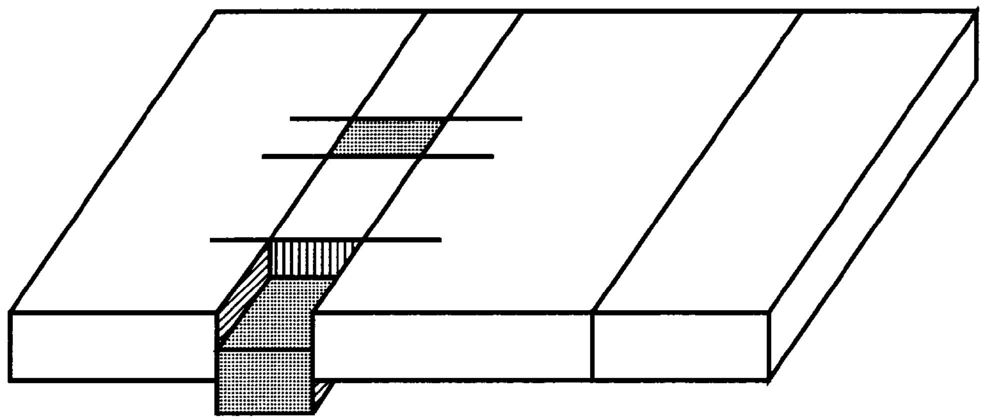


Figure 3.2. Configuration of punchouts

Table 3.1. Coefficient C_1 values in Eq. 3.2. (Unit: Millions, CV: 30 %)

	Coarse Agg. Type	SRG		Limestone	
	Swelling Condition	Yes	No	Yes	No
Reliability	95 %	1.4	2.0	1.8	2.6
	75 %	2.4	3.1	2.5	3.7
	50 %	3.1	4.2	3.1	4.8

CHAPTER 4. DESCRIPTION OF COMPUTER PROGRAM

4.1 MAIN PROGRAM

The CRCP-9 computer program has been written in Fortran language. CRCP-9 consists of a main program and subroutine programs. The main program is divided into several distinct parts. The dimensions of arrays are defined first, input values are read, and basic variables are defined. The output format of crcpfor.sum, which is a summary file of the analysis, for input values is defined next. Then, the analysis is performed with calling subroutine programs. A number of output files are created during the analysis to communicate with a Windows-based, user-friendly interface program and to inform users of the progress of the analysis. When the analysis is completed, these output files are deleted automatically by the user-friendly interface program. Finally, analysis results are written in output files of crcpfor.out and crcpfor.sum. The descriptions of subroutine programs and input and output guides are explained in the following sections.

The dimensions of arrays should be larger than the required dimensions. However, using very large dimensions can lead to hardware memory problems. The subroutine CHMEMO checks the dimensions of arrays, and if more dimensions are needed for any array, a notice is given and the program stops. The required dimensions of the arrays used in the program are as follows.

ALOAD(ITNOD1)
SK(ITNOD1,MBAND)
TEMLD(ITNOD1)
TLOAD(NUMELY,8)
SS(KS)
IMAXC(NUMELX)
TEMPR(29,NUMELY+1)
DRYSH(29,NUMELY+1)
STRSSF(NUMELX*2)
DISTRB(NUMELX*2)
DISP(ITNOD1)
BNSL(NUMELX)
STRINI(NUMELY,4)
INOD(INODDIM,8)
IMAX(NUMELX)
TEMPR1(NUMELY+1)
TEMPRA(29,NUMELY+1)
STRSS(NUMELX)
TSTARR(NUMELX*2)

where

ITNOD1 = 2*(NUMELX+1)*(NUMELY+2)
MBAND = (NUMELY+3)*2+4
KS = ITNOD1*MBAND
INODDIM = (NUMELY+1)*NUMELX

and, NUMELX and NUMELY are the numbers of plane elements in the longitudinal and vertical directions, respectively. In the program, NUMELY is selected to be 10, and the maximum NUMELX is 200 because half of the first primary crack spacing is selected to be

300 inches and the length of a finite element in the longitudinal direction is 1.5 inches. Users can change the number of finite elements in the vertical direction by modifying NUMELY, and in the longitudinal direction, by modifying A (the length of a finite element in the longitudinal direction) and CRSPC(IPCS,1) (primary crack spacing). The maximum NUMELX is calculated by dividing half of CRSPC(1,1) into A.

4.2 SUBROUTINE PROGRAMS

There are a number of subroutine programs used in the CRCP-9 computer code and the descriptions are as follows:

STRDIST
This subroutine is used to impose random coefficients to each finite element in the longitudinal direction of the pavement. These coefficients are multiplied by the mean concrete tensile strength to define variations of concrete tensile strengths. The distribution of the concrete tensile strength is assumed to be a normal distribution. The random values are distributed within the coefficient of variation in concrete tensile strength.

CHMEMO
This subroutine is used to check the required dimensions of the arrays. If a required dimension of an array is larger than the current dimension of the array, a notice that a larger dimension of the array is needed is given and the program stops. Users must revise the dimensions of the arrays that need more dimensions. The arrays that need to be revised are informed by the program.

CONCPRO
The concrete material properties such as elastic modulus, tensile strength, compressive strength, and drying shrinkage at each given day are calculated in this subroutine. Eq. 2.24 and Table 2.2 are used to define these values.

CREEP
To account for creep of strain in concrete, the reduced elastic modulus of concrete is calculated using the effective modulus method. The creep coefficient curve is defined in this subroutine.

LINREC4
This is used to find the element stiffness matrix using four-noded bilinear isoparametric or rectangular elements for the concrete slab. The material property matrix **D** is defined here according to the type of plane element (plane stress or plane strain).

MATPRO
This subroutine is used to multiply matrices.

FRAME
The longitudinal steel bar is discretized using frame elements. The element stiffness matrix of the frame element, which includes horizontal, vertical, and rotational degrees of freedom, is defined in this subroutine.

SHRINK

The equivalent temperature variations owing to drying shrinkage strains are calculated here. A positive value of drying shrinkage makes a negative value of the equivalent temperature.

TEMPRT

This subroutine calculates the initial stresses and equivalent nodal loads due to changes in temperature and drying shrinkage. We can use an average temperature from the nodal temperatures for an entire element or interpolated temperatures from the nodal temperatures at integration points. To use an average temperature in the subroutine, select INDEX=0, and to use interpolated temperatures, select INDEX=1. The differences in the results are negligible when there are a sufficient number of elements (Ref 6).

ASSEM

The element stiffness matrices and nodal load vectors are assembled to create the global stiffness matrix and load vector. The half-bandwidth is used for the global stiffness matrix.

BOUND

The boundary conditions of the finite element model are defined in this subroutine. To model the restraints, spring elements with a very large stiffness value are used.

BNSLIP

The bond-slip relationship between concrete and steel bar is defined here. The linear, linear with an ultimate slip, bilinear, and bilinear with an ultimate slip cases can be considered by controlling input values. For example, if we have the same bond-slip stiffness values for the primary and secondary stiffness values, we can model the linear bond-slip relationship. Also, if we have a very large yield slip value so that the relative movement between concrete and steel cannot reach the value, we can model the linear bond-slip relationship.

FRSLIP

The frictional bond-slip relationship at the interface between the bottom of the concrete slab and the base layer is defined here. The linear bond-slip relationship is used for this frictional bond-slip.

CURL

The underlying layers are modeled using a Winkler type elastic foundation. To account for the curling and warping effects, the spring elements for the elastic foundation sustain only the compressive forces. In other words, the tensionless springs are used for the foundation.

DWEIGHT

The body forces (dead weights) of concrete steel are calculated in this subroutine and added to the load vector.

SOLVE

To solve the equilibrium equations developed in the program, the one-dimensional compacted array of banded stiffness matrix and Gaussian elimination are used (Ref 18).

CHSLIP

This subroutine is used to determine if the relative movement between concrete and steel exceeds the limit values such as yield slip and ultimate slip.

CHCURL

This is used to check if any vertical spring for the underlying layers is in tension. If there is any spring in tension, this spring will be removed from the system and a new analysis will be performed with a new system.

COMPST

This subroutine compares the stresses when the temperature is high and when the temperature is low at a given day to find the maximum tensile stress. Because the wheel load stresses are added to obtain the final stresses in the system, the maximum tensile stress does not always occur when the temperature is low. An explanation about this fact is presented in Chapter 2.

TSTRTH

The tensile strength distribution along the pavement is defined here. It is assumed that the tensile strength is constant along the depth of the concrete slab and the variation exists only in the longitudinal direction.

STRESS

The stress at each integration point is calculated in this subroutine. For any finite element, an average value of the stresses at integration points is used for the stress at that finite element.

WHEEL

The stresses in the concrete slab due to the wheel load applications are calculated here by using the Westergaard equation.

PUNCHOUT

The relationships between the number of punchouts and the wheel load applications are predicted in this subroutine.

4.3 INPUT GUIDE

The required inputs in this program are explained. The format for input values is free form, such that only a space is needed between input values. Sample input is shown in Appendix A.

LINE 1

TITLECRCP: This is used in conjunction with a Windows-based, user-friendly interface program.

LINE 2

TITLEINP: Title of analysis

LINE 3

TOTD: Thickness of concrete slab (unit: in.)

IELTYPE: Index for plane element type (1: plane stress, other: plane strain)

NUMPCS: Number of different primary crack spacings (Maximum = 10, cannot be 3, 6, 7, 9)
If more than 10 primary crack spacings are considered, the dimensions of the arrays of CRSPC, CRWID, STLSTR, ICRLOC, and INDCR should be changed according to the maximum number of different primary crack spacings.

LINE 4

POIS1: Poisson's ratio of concrete

ALPC: Coefficient of thermal expansion of concrete (unit: /F)

WEIGHT: Specific weight of concrete (unit: pcf)

CV: Coefficient of variation in concrete tensile strength (unit: %)

STRAT: Percent reinforcement (steel ratio) (unit: %)

LINE 5

IAGG: Index for coarse aggregate type
(1: Granite, 2: Dolomite, 3: Vega, 4: Bridgeport/Tin Top,
5: Western Tascosa, 6: Ferris, 7: Limestone,
8: Siliceous river gravel)

IDEFAULT Index whether or not using default values (0: use default values)

LINE 6

CSTR28: 28-day concrete compressive strength (put 0 to use a default value)

TSTR28: 28-day concrete tensile strength (put 0 to use a default value)

ECONC28: 28-day concrete elastic modulus (put 0 to use a default value)

ZSH256: 256-day concrete drying shrinkage (put 0 to use a default value)

LINE 7

STLOC: Steel bar location from surface (unit: in.)

ESTL: Young's modulus of elasticity of steel

DIASTL: Diameter of longitudinal steel bar

ALPS: Coefficient of thermal expansion of steel (unit: /F)

STWEI: Specific weight of steel (unit: pcf)

LINE 8

IBONDCS: Index for bond-slip relationship between concrete and steel (It is used in conjunction with a Windows-based user interface; 0 to 12; 12: user-defined input values)

AKBUA: Primary bond-slip stiffness per unit area between concrete and steel (unit: pci)

AKBUA2: Secondary bond-slip stiffness per unit area between concrete and steel (unit: pci)

YLSLIP: Yield slip between concrete and steel (unit: in.)

ULSLIP:	Ultimate slip between concrete and steel (unit: in.)

LINE 9

BOTKV:	Vertical stiffness per unit area of underlying layers (unit: pci)
INDSPR:	Index for tensionless springs for underlying layers (1: no tensionless)

LINE 10

INDFR:	Index for subbase type for friction (1: Flexible (145.5 pci), 2: Asphalt-stabilized (55.9 pci), 3: Cement-stabilized (15400 pci), 4: Lime-treated clay (154.5 pci), 5: Untreated clay (22 pci))
AKFRUA:	Bond-slip stiffness per unit area between concrete and base (unit: pci)

LINE 11

WHLOAD:	Wheel load (unit: lbs)
WHRAD:	Radius of loaded area (unit: in.)
WHDAY:	Days before wheel load applied

LINE 12

ICREEP:	Index for creep analysis (1: creep analysis)
PHIMAX:	Maximum creep ratio to instantaneous elastic strain
PERCEN:	Consideration percent to maximum creep (unit: %)
DAYPC:	Corresponding day to PERCEN
TIME:	Loading duration (unit: hour)

LINE 13

IFATPRO:	Index for reliability of punchout prediction (1: 95%, 2: 75%, 3: 50%)
ISWELL:	Index for swelling condition (1: swelling = yes)
ITEMIND:	Index for temperature variation through concrete depth (0: uniform, 1: linear, 2: nonlinear)
IDRYIND:	Index for drying shrinkage variation through concrete depth (0: uniform, 1: linear, 2: nonlinear)

LINE 14

TEMF:	Reference (curing) temperature

LINE 15

IDAY2:	Second day for temperature and drying shrinkage input
IDAY3:	Third day for temperature and drying shrinkage input
IDAY4:	Fourth day for temperature and drying shrinkage input
IFDAY:	Final day for temperature and drying shrinkage input

LINE 16 –LINE 334

DO I=1, 29: From day 1 to 28, and another day for the minimum yearly temperature

DAYCON(I): Consideration day after concrete placement

DO J=1, 11: At each finite element node, temperature and drying shrinkage are defined. Because the number of elements in the vertical direction is 10, the number of finite element nodes is 11.

TEMPR(I,J): Minimum daily temperatures at each finite element node from surface on day I (unit: F)

TEMPRA(I,J): Maximum daily temperatures at each finite element node from surface on day I (unit: F)

DRYSH(I,J): Daily drying shrinkage ratio at each finite element node from surface on day I (range from 0 to 1)

ENDDO

ENDDO

4.4 OUTPUT GUIDE

Two output files are created after performing the analysis. One is named crcpfor.out, and the other, crcpfor.sum. The file crcpfor.out is used to provide output values to a Windows-based, user-friendly interface program developed using MS Visual Basic. The file crcpfor.sum summarizes input values and analysis results. Sample output files of crcpfor.out and crcpfor.sum are shown in Appendices B and C. The output format can be modified in the main program.

CHAPTER 5. SAMPLE ANALYSIS RESULTS

The analysis results from the CRCP-9 computer program include time histories of mean crack spacing, mean crack width, and mean steel stress until 28 days after placement of concrete and at the final day of the analysis, which is generally selected to be the day when the minimum yearly temperature occurs. The crack spacing distribution and cumulative values are also obtained at the final of the analysis. For example, to obtain the crack spacing distribution at 60 days after placement, the final day of the analysis has to be selected to be sixty in the input process. The standard deviation of the crack spacing at each day is also provided. If the standard deviations of the crack width and the steel stress are needed, users can modify the program. The number of punchouts per mile is predicted for a given wheel load application. In this program, the number of punchouts can be estimated up to 100 million ESALs.

5.1 CRACK SPACING

First, the variations in mean crack spacings for the first 28 days are investigated for different longitudinal steel ratios as shown in Figure 5.1. The large decrease in the crack spacing is observed within the first several days. As the amount of steel increases, the smaller crack spacings are obtained.

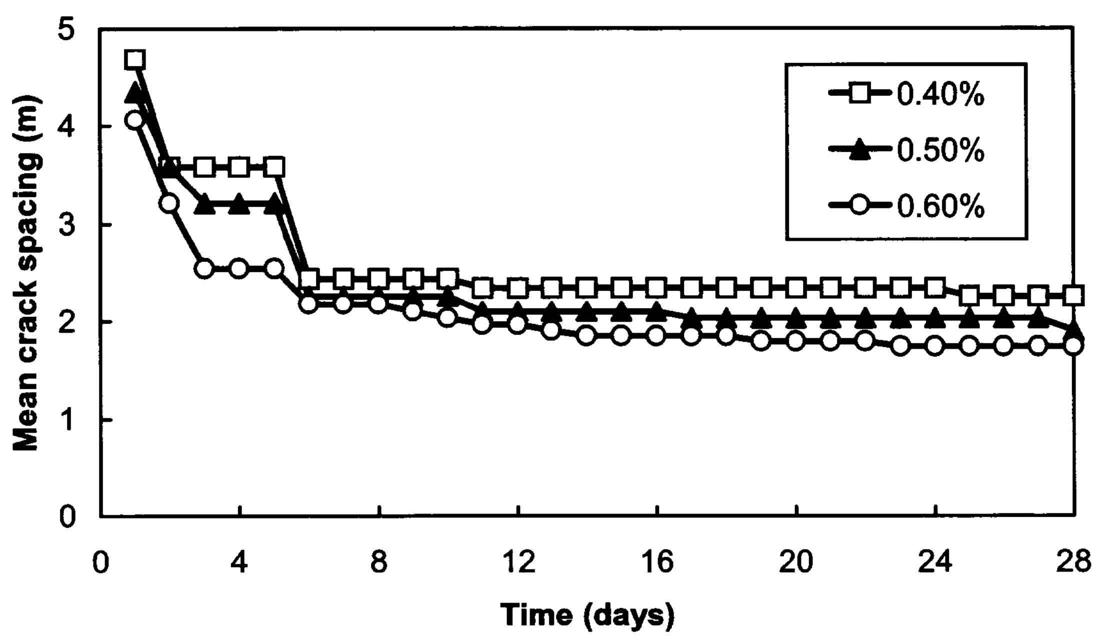


Figure 5.1. Time histories of mean crack spacing for various steel ratios (1 m = 3.28 ft)

The bond-slip relationship between concrete and steel is reported to have significant effects on the behavior of the continuously reinforced concrete (CRC) pavements (Refs 9, 10). To examine this bond-slip relationship, the time histories of mean crack spacing have been compared between linear and nonlinear models (in this study, we use a bilinear with an ultimate slip model for the nonlinear bond-slip model) for three different coarse aggregates of granite, limestone, and siliceous river gravel (SRG) as shown in Figure 5.2. If we have the same linear or nonlinear bond-slip model for different coarse aggregates, the mean crack spacings are the largest with the granite pavement and the smallest with the SRG pavement at a given day. However, the limestone pavement with a nonlinear bond-slip model shows larger mean crack spacings than the granite pavement with a linear bond-slip model after 14 days of placement. The same results can be observed between the SRG pavement with a nonlinear bond-slip model and the limestone pavement with a linear bond-slip model. This implies that more reasonable bond-slip input values should be obtained by experiments to improve the accuracy of the analysis results.

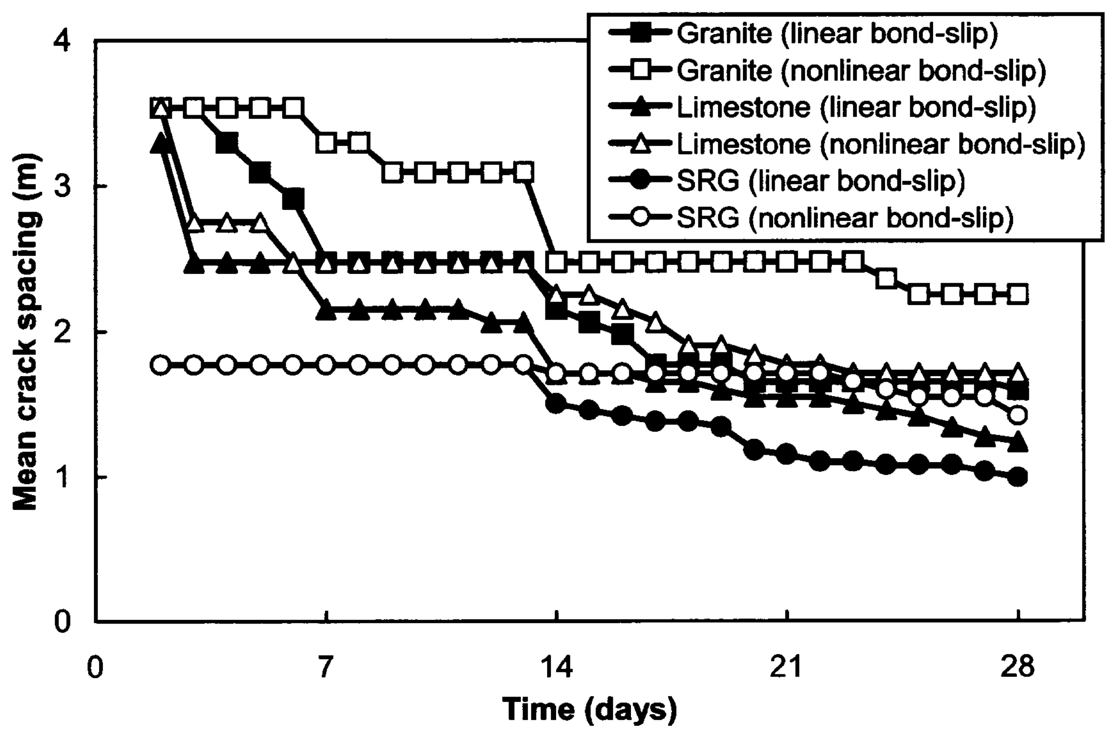


Figure 5.2. Effect of bond-slip types on mean crack spacing (1 m = 3.28 ft)

Figure 5.3 shows the time histories of mean crack spacing for pavements with three different aggregate types of granite, limestone, and SRG. Figures 5.3(*a*) and (*b*) show the results with a linear and a nonlinear bond-slip relationship, respectively. The pavement with granite has the largest crack spacings and the pavement with SRG has the smallest. The mean crack spacings are larger with a nonlinear bond-slip model because the predicted concrete stresses are smaller with the nonlinear bond-slip model.

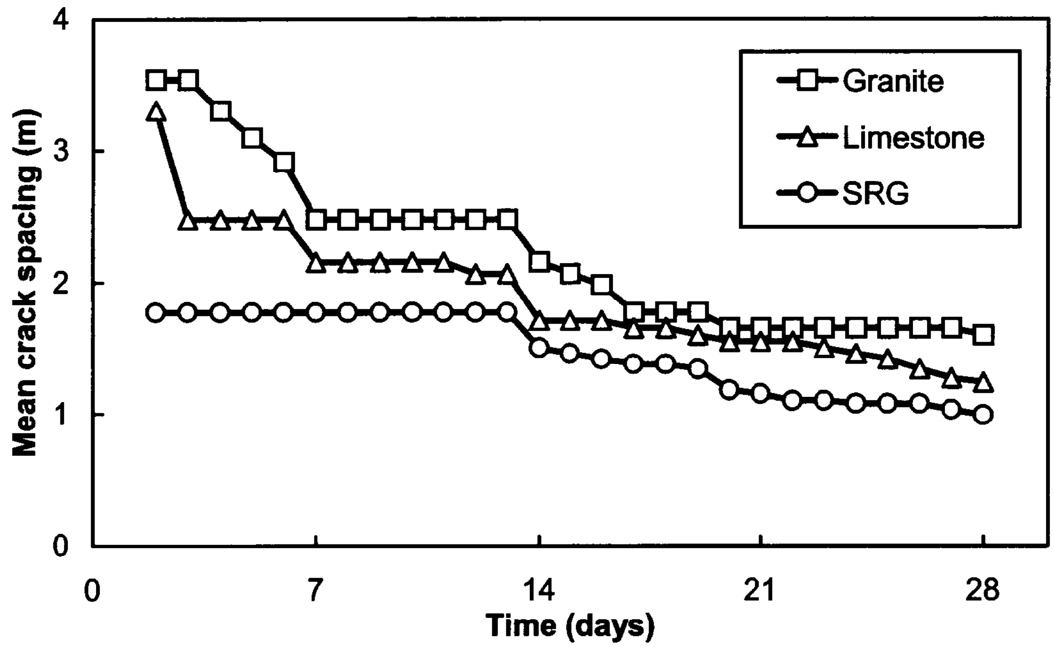
Figure 5.4(*a*) shows the crack spacing distributions at 28 days after concrete placement for three different aggregate types. The pavement with SRG shows high frequencies of crack spacings at smaller values and the pavement with granite has high frequencies at larger values. Figure 5.4(*b*) shows the cumulative crack spacing distributions from the results shown in Figure 5.4(*a*).

Figure 5.5 shows the cumulative crack spacing distributions for different values of the coefficient of variation (CV) in concrete tensile strength. The crack spacings tend to decrease when CV values increase. This means that the material variability should be well controlled to prevent very narrow crack spacings.

5.2 CRACK WIDTH AND STEEL STRESS

The time histories of mean crack width are investigated as shown in Figure 5.6. The pavement with granite tends to have larger crack widths and the pavement with limestone has smaller crack widths. The time histories of mean steel stress at crack are also investigated as shown in Figure 5.7. The results are very similar to the time histories of mean crack width.

(a)



(b)

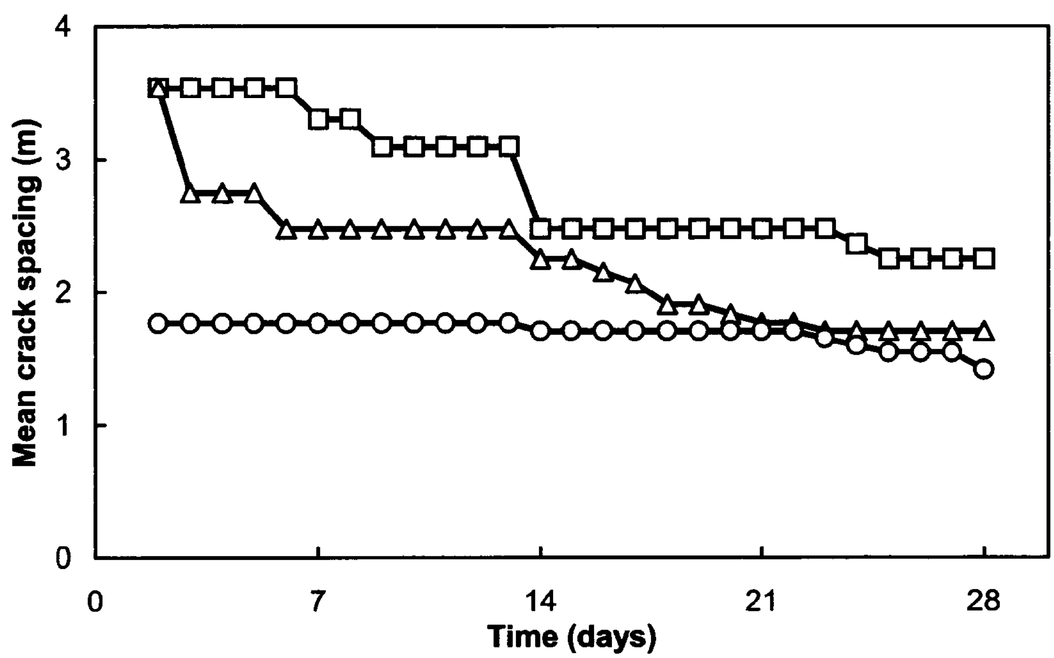
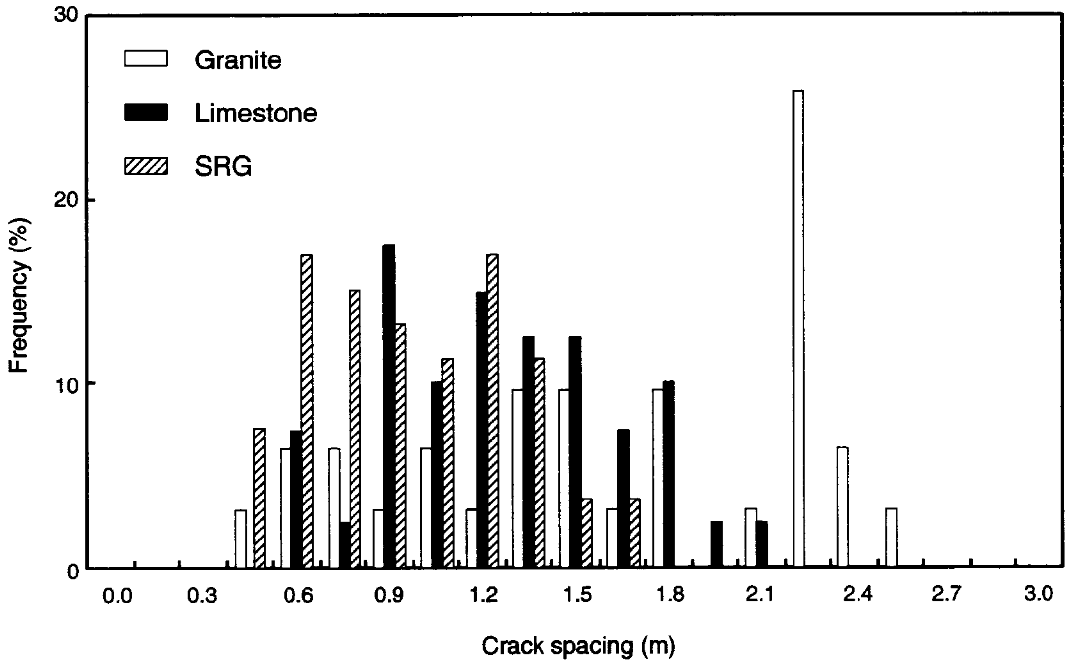


Figure 5.3. Time histories of mean crack spacing for various coarse aggregate types: (a) with linear bond-slip model; (b) with nonlinear bond-slip model (1 m = 3.28 ft)

(a)



(b)

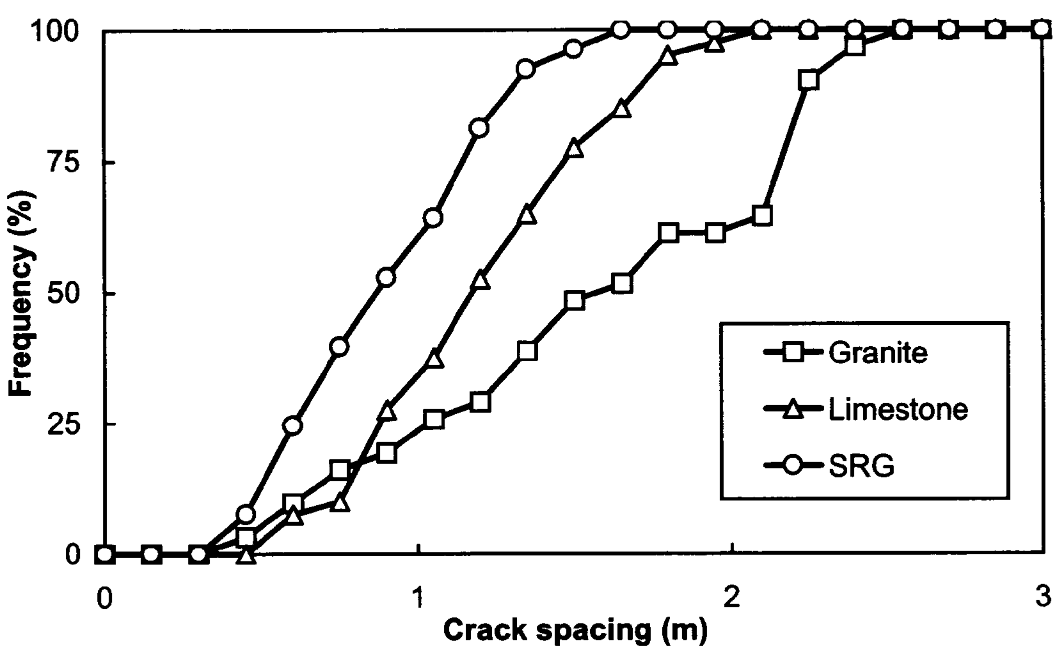


Figure 5.4 Crack spacing distribution at 28 days after concrete placement: (a) crack spacing distribution, (b) cumulative crack spacing distribution (1 m = 3.28 ft)

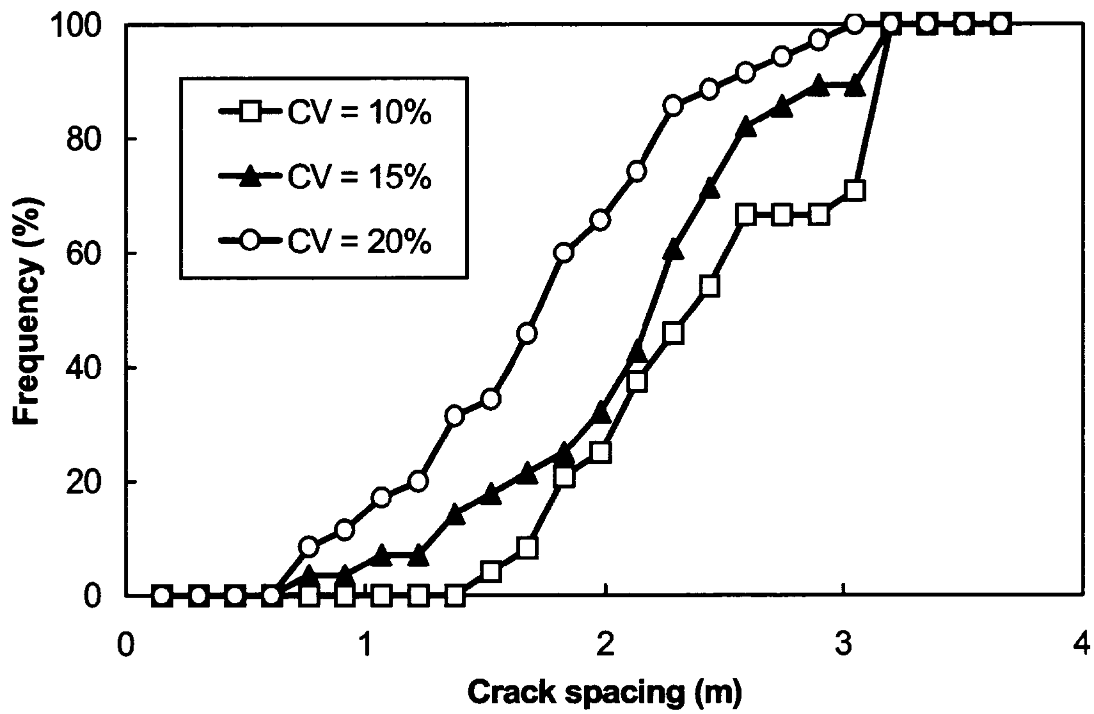
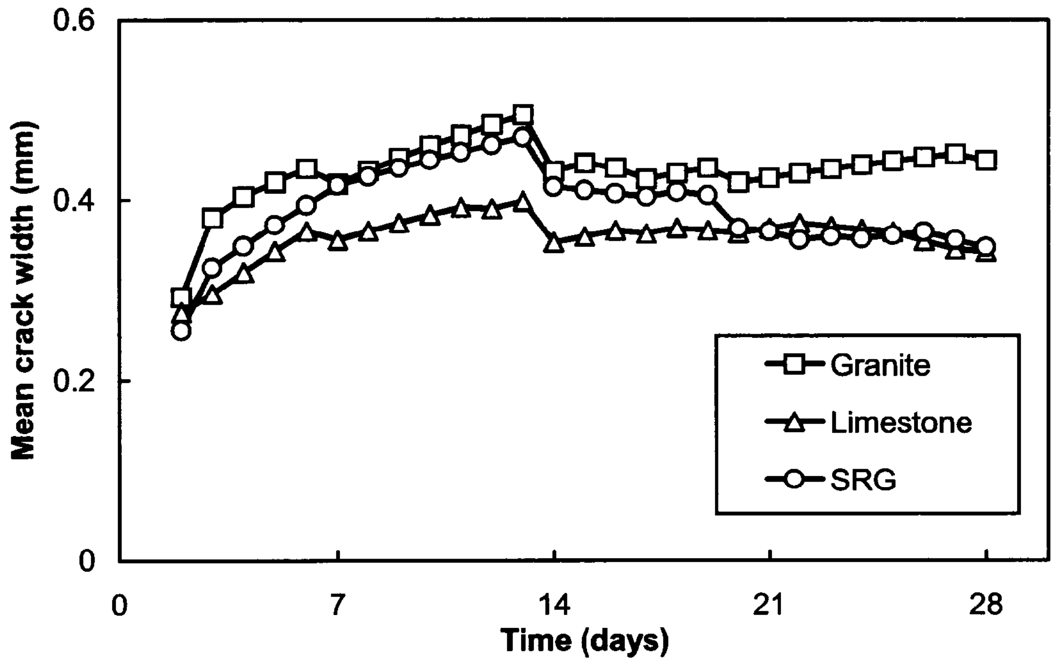


Figure 5.5. Cumulative crack spacing distribution for various CV values (1 m = 3.28 ft)

(a)



(b)

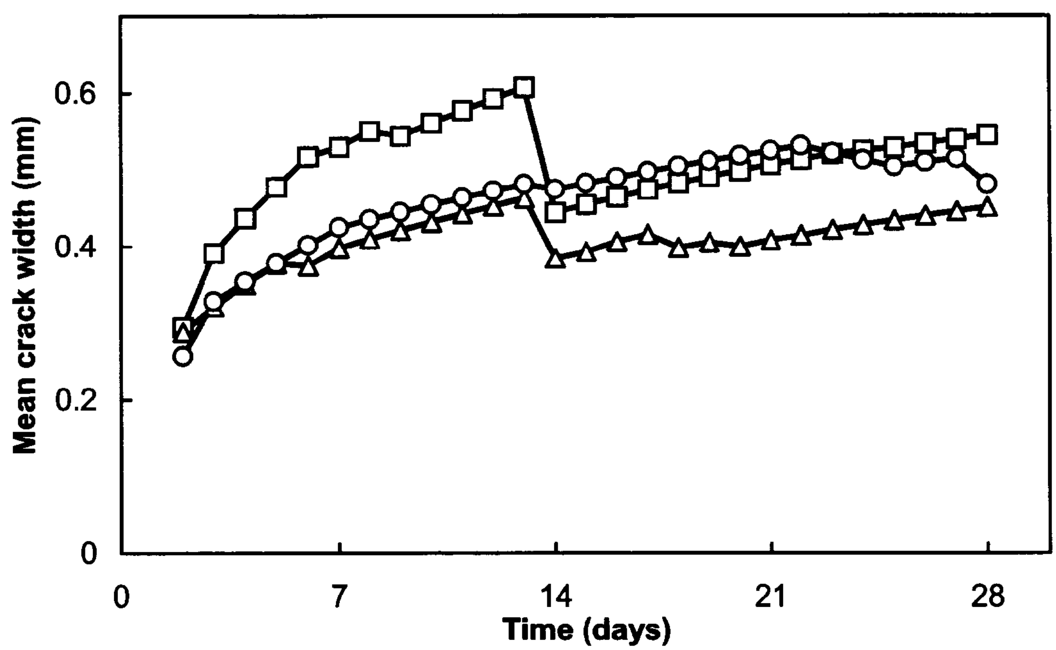


Figure 5.6. Time histories of mean crack width: (a) with linear bond-slip model; (b) with nonlinear bond-slip model (1 mm = 0.0394 in.)

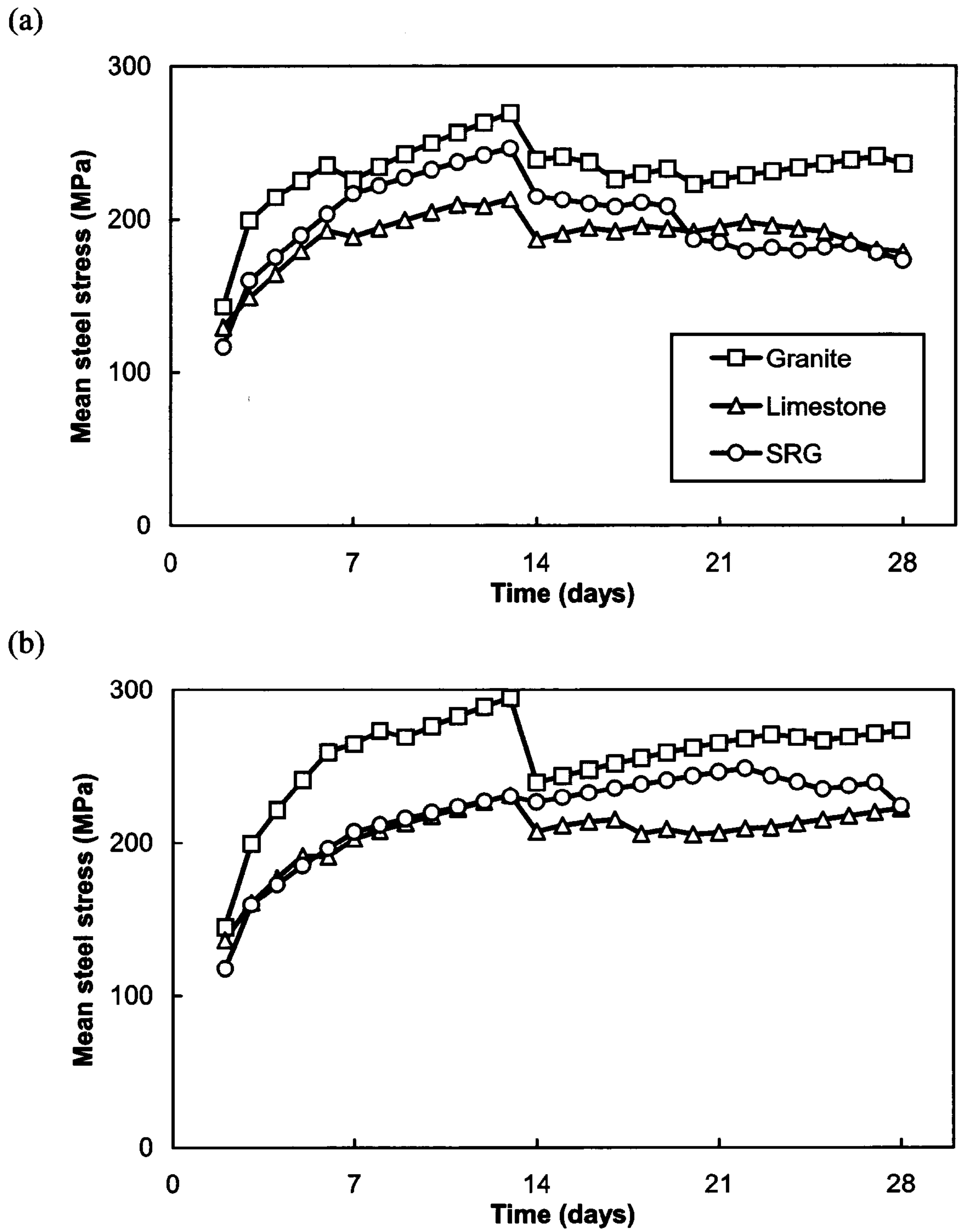


Figure 5.7. Time histories of mean steel stress at crack: (a) with linear bond-slip model; (b) with nonlinear bond-slip model (1MPa = 145.14 psi)

5.3 PUNCHOUTS

Figure 5.8 shows the number of punchouts per kilometer according to the wheel load applications for various coefficients of variation (CV) values in the concrete tensile strength. As the CV value increases, the frequency of punchouts increases. This is partly because of the shorter crack spacings with larger CV values, as already investigated in Figure 5.5. The punchout failures are closely related to the transverse crack spacings partly because the transverse stresses that cause longitudinal cracks depend on the transverse crack spacing among other factors such as load transfer and base conditions. The shorter the transverse crack spacings, the higher the transverse stresses obtained.

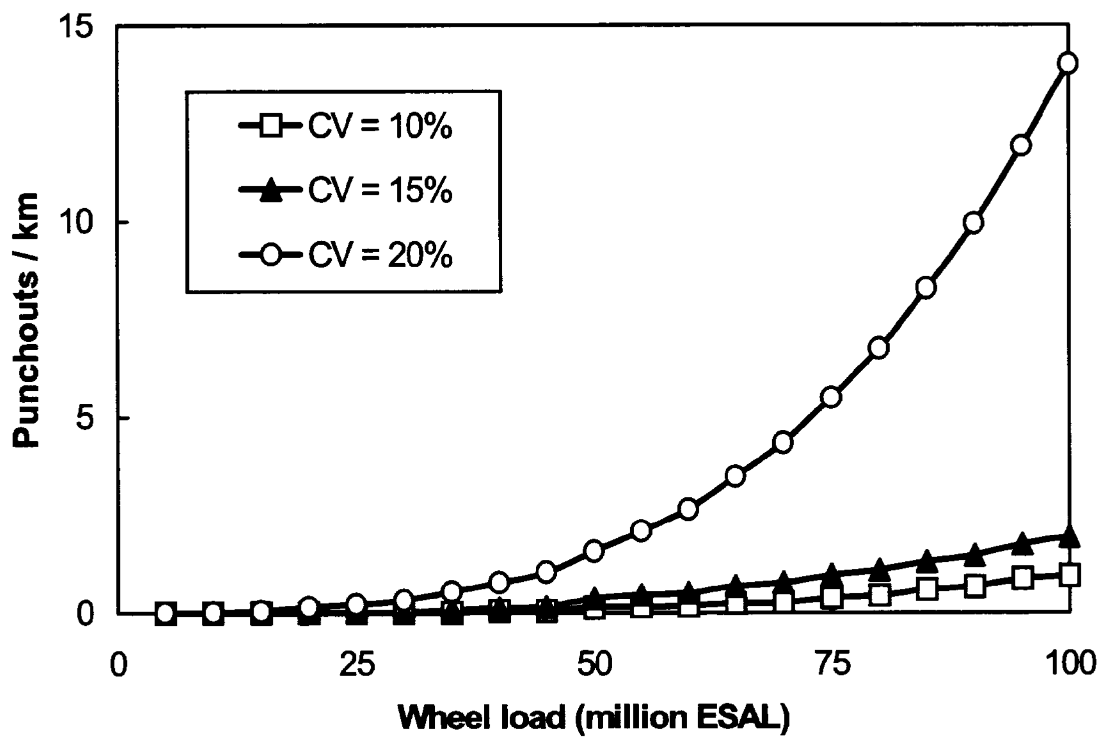
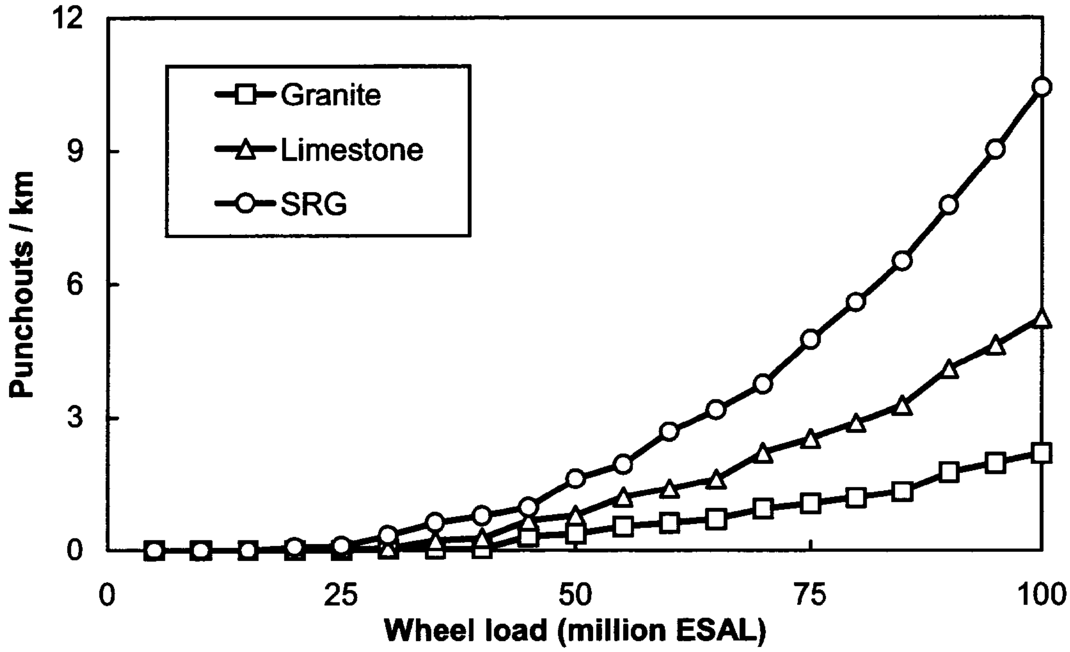


Figure 5.8. Punchout frequency curve for various CV values (1 km = 0.63 miles)

Figure 5.9 shows punchout frequencies for three different coarse aggregate types under the identical base conditions and load transfer characteristics except for concrete strength variability. The pavement with SRG shows higher frequencies of punchouts and the pavement with granite shows lower frequencies of punchouts. Those are because of shorter crack spacings for the pavement with SRG and larger crack spacings for the pavement with granite, as already investigated in Figures 5.3 and 5.4.

(a)



(b)

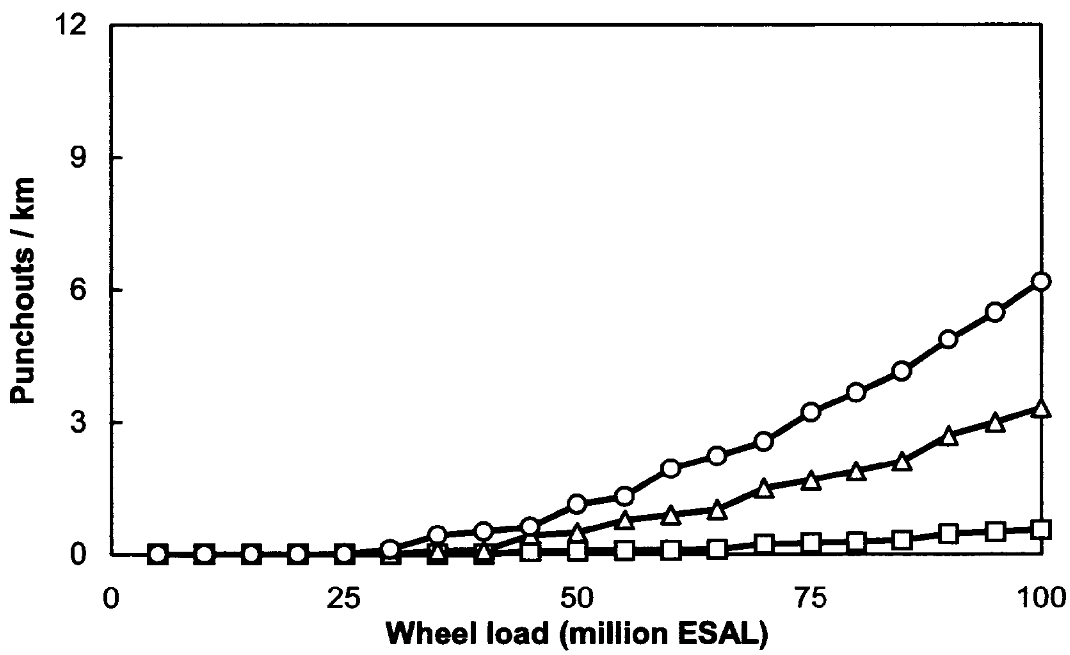


Figure 5.9. Punchout frequency curves: (a) with linear bond-slip model; (b) with nonlinear bond-slip model (1 km = 0.63 miles)

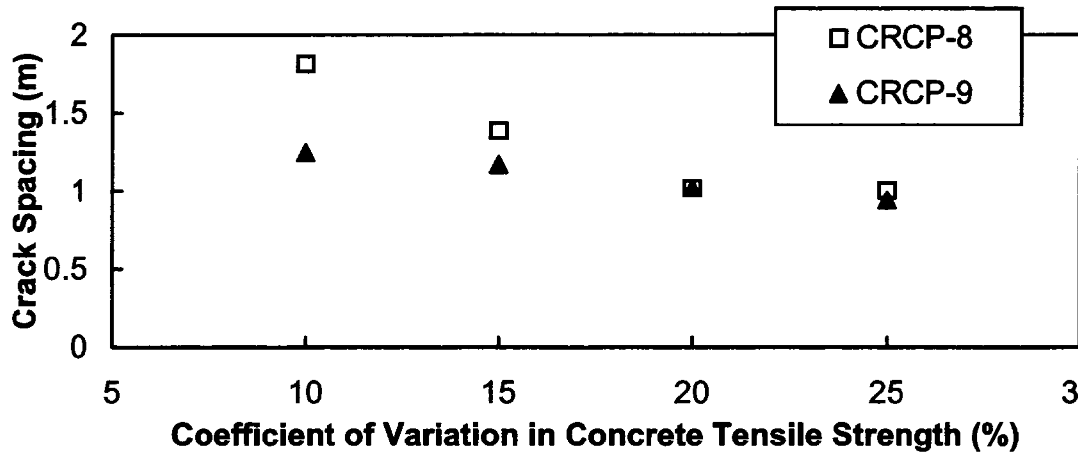
5.4 COMPARISON WITH CRCP-8

The results from CRCP-8 and CRCP-9 have been compared as shown in Figure 5.10 to investigate how much difference exists in the results between the two programs. Because the input parameters in those two computer programs are not the same, efforts have been made to have close input values for the two programs. The crack spacing, crack width, and steel stress are compared for different values of the coefficient of variation (CV) in concrete tensile strength. The crack spacings from CRCP-8 are larger than those from CRCP-9 when CV values are small and are very close with CV values of 20 and larger as shown in Figure 5.10(*a*). The crack widths are larger and the steel stresses are smaller in the results from CRCP-9 compared with CRCP-8 (Figures 5.10[*b*] and [*c*]). It should be noted that the analysis results from CRCP-9 are very sensitive to the bond-slip relationship between concrete and steel bars. Because CRCP-8 and CRCP-9 use different algorithms for the stress transfer from steel to concrete, further experimental studies should be conducted to have more reasonable input values for the bond-slip relationship.

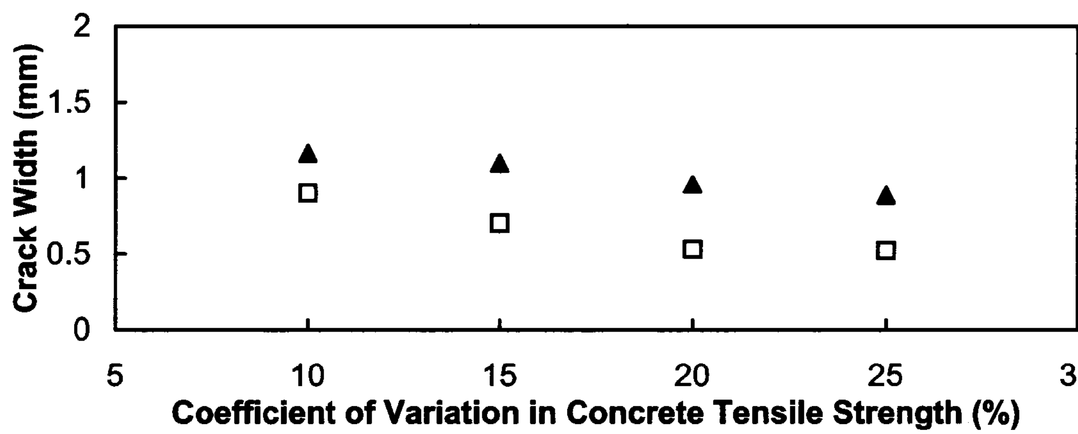
CRCP-9 includes all the input parameters required in CRCP-8. There are additional input parameters in CRCP-9 as listed below.

- The variations of temperature and drying shrinkage through the depth of the concrete slab can be considered.
- The location of the longitudinal steel bar can be changed.
- The specific weights of concrete and steel are included for the calculation of the curling and warping effects.
- The stiffness of the bond-slip between concrete and steel is included.
- The parameters for the creep effects are included.
- The type of the plane element can be selected (plane stress or plane strain).

(a)



(b)



(c)

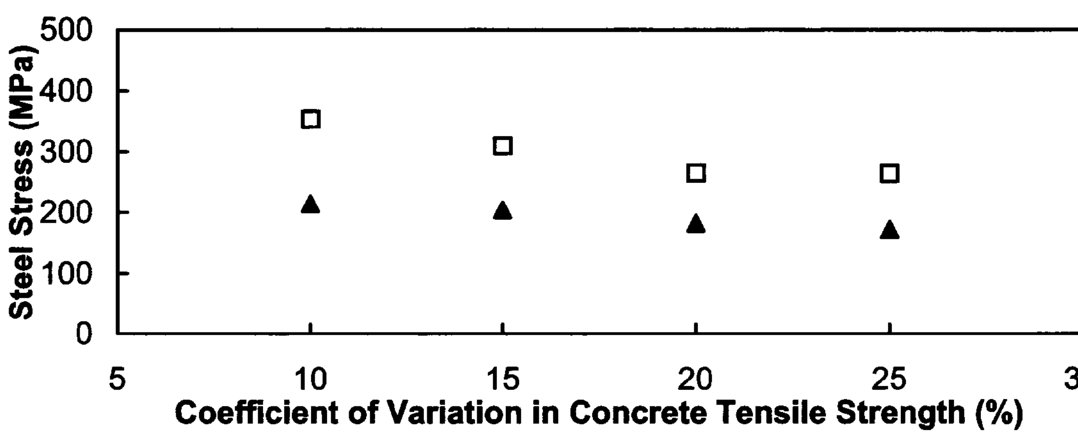


Figure 5.10. Results from CRCP-8 and CRCP-9: (a) mean crack spacing; (b) crack width; (c) steel stress at crack (1MPa = 145.14 psi, 1 mm = 0.0394 in., 1 m = 3.28 ft)

CHAPTER 6. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

6.1 SUMMARY

A new version of the CRCP computer program, CRCP-9, has been developed in this study. The numerical model of the continuously reinforced concrete (CRC) pavements was developed using finite element theories, the crack spacing prediction model using the Monte Carlo method, and the failure prediction model using probability theories. CRCP-9 uses two-dimensional numerical methods to calculate stresses and strains, which is a totally different approach compared with the previous version of the program, which used one-dimensional analytical methods to calculate them. The major characteristics of CRCP-9 that differ from CRCP-8 include consideration of nonlinear variations in temperature and drying shrinkage through the depth of the concrete slab, nonlinear bond-slip relationship between concrete and steel bars, viscoelastic effect of concrete, curling and warping effects, and the ability of changing locations of the longitudinal steel bars. Details of the theoretical background of mechanistic modeling have been presented. The descriptions of the computer program and input and output procedures have also been explained. Sample analysis results from CRCP-9 and comparisons of the results between CRCP-9 and CRCP-8 are presented.

6.2 CONCLUSIONS

This study to develop a new mechanistic model of the CRC pavements, CRCP-9, points to the following conclusions:

1. CRCP-9 considers the variations of temperature and drying shrinkage through the depth of the concrete slab, which cause stresses due to curling and warping of the concrete slab.
2. The bond-slip relationships between concrete and steel have been considered with various models. The various bond-slip models and their input values significantly affect the analysis results.
3. The time histories of mean crack spacing, crack width, and steel stress at crack can be predicted by using CRCP-9.
4. CRCP-9 provides the crack spacing distribution and its cumulative values at a given day.
5. CRCP-9 also estimates the number of punchouts per mile according to the wheel load applications from the predicted crack spacing distribution
6. CRCP-9 can be used for various purposes. It can be used, for instance, to determine the optimum longitudinal steel design, such as amount of steel, bar diameter, and location of the steel bar for given pavement structure, material properties, and environmental conditions.

6.3 RECOMMENDATIONS

Efforts have been made to develop a more realistic mechanistic model that predicts behavior and performance of the CRC pavement systems. To improve the accuracy of the program, it is needed that the calibration of the program with field data should be further performed to obtain more reasonable ranges of input values and analysis results. As investigated, the bond-slip relationship between concrete and steel significantly affects the analysis results. Further investigations with experimental studies to identify reasonable input values for the bond-slip relationship should be undertaken. The wheel load stresses in CRCP-9 are obtained simply by using Westergaard equations. A study needs to be performed to find more realistic wheel load stresses with considerations of discontinuities at cracks and edges of the pavement and effects of multiple wheel loads.

REFERENCES

1. McCullough, B. F., A. A. Ayyash, W. R. Hudson, and J. P. Randall. *Design of Continuously Reinforced Concrete Pavements for Highways*. NCHRP 1-15. Center for Transportation Research, The University of Texas at Austin, 1975.

2. Won, M. C., K. Hankins, and B. F. McCullough. *Mechanistic Analysis of Continuously Reinforced Concrete Pavements Considering Material Characteristics, Variability, and Fatigue*. Report 1169-2. Center for Transportation Research, The University of Texas at Austin, 1991.

3. Dossey, T. and B. F. McCullough. *Characterization of Concrete Properties with Age*. Report 1244-2. Center for Transportation Research, The University of Texas at Austin, 1991.

4. Suh, Y. C., K. Hankins, and B. F. McCullough. *Early-Age Behavior of Continuously Reinforced Concrete Pavement and Calibration of the Failure Prediction Model in the CRCP-7 Program*. Report 1244-3. Center for Transportation Research, The University of Texas at Austin, 1992.

5. Won, M. C., T. Dossey, S. Easley, and J. Speer. *CRCP-8 Program User's Guide*. Center for Transportation Research, The University of Texas at Austin, 1995.

6. Kim, S. M., M. Won, and B. F. McCullough. *Development of a Finite Element Program for Continuously Reinforced Concrete Pavements*. Report 1758-S. Center for Transportation Research, The University of Texas at Austin, 1997.

7. Kim, S. M., M. Won, and B. F. McCullough. "Numerical Modeling of Continuously Reinforced Concrete Pavement Subjected to Environmental Loads." *Transportation Research Record 1629*, TRB, National Research Council, Washington, D.C., 1998, pp. 76-89.

8. *ABAQUS, User's Manual Version 5.8*, Hibbit, Karlsson & Sorensen, Inc., Pawtucket, R.I., 1998.

9. Kim, S. M., M. C. Won, and B. F. McCullough. "Three-Dimensional Analysis of Continuously Reinforced Concrete Pavements." *Transportation Research Record 1730*, TRB, National Research Council, Washington, D.C., 2000, pp. 43-52.

10. Kim, S. M., M. C. Won, and B. F. McCullough. *Three-Dimensional Nonlinear Finite Element Analysis of Continuously Reinforced Concrete Pavements*. Report 1831-1. Center for Transportation Research, The University of Texas at Austin, 2000.

11. Kim, S. M., M. C. Won, and B. F. McCullough. "Effect of Transverse Reinforcement on CRC Pavement Response to Thermal Loads." *Proceedings of the Second*

International Symposium on 3D Finite Element for Pavement Analysis, Design, and Research. Charleston, West Virginia, 2000, pp. 43-64.

12. Westergaard, H. M. "Stresses in Concrete Pavements Computed by Theoretical Analysis." *Public Roads,* Vol. 7, 1925, pp. 25-35.

13. Neville, A. M., W. H. Dilger, and J. J. Brooks. *Creep of Plain and Structural Concrete.* Construction Press, London, 1983.

14. Faber, O. "Plastic Yield, Shrinkage and Other Problems of Concrete and Their Effect on Design." *Minutes of Proc. ICE,* Vol. 225, Part 1, London, 1927, pp. 27-73.

15. Hankins, K., and M. C. Won. *Condition Survey on US 290.* Technical Memorandum 422-25. Center for Transportation Research, The University of Texas at Austin, 1988.

16. Yoder, E. J., and M. W. Witczak. *Principles of Pavement Design.* 2nd Edition, John Wiley & Sons, Inc., New York, 1975.

17. McCullough, B. F., J. C. M. Ma, and C. S. Noble. *Limiting Criteria for the Design of CRCP.* Report 177-17, Center for Highway Research, The University of Texas at Austin, 1979.

18. Cook, R. D. *Concepts and Applications of Finite Element Analysis.* 2nd Edition, John Wiley & Sons, Inc., New York, 1981.

APPENDIX A:

SAMPLE INPUT FILE OF CRCPFOR.INP

```
This is an input file of CRCP-9
crcpfor
 12           1            2
 0.15         0            145          10           0.6
 7            0
 0            0            0            0
 6            2.9E+07      0.75         0.000005     490
 12           250000       250000       100.01       100.05
 400          0
 1            145.5
 9000         6            14
 1            2            99           30           12
 3            0            1            0
 110
 2            3            7            180
 1
 100          110          0.5
 101          109          0.5
 102          108          0.5
 103          107          0.5
 104          106          0.5
 105          105          0.5
 106          104          0.5
 107          103          0.5
 108          102          0.5
 109          101          0.5
 110          100          0.5
 2
 90           110          0.5
 91           109          0.5
 92           108          0.5
 93           107          0.5
 94           106          0.5
 95           105          0.5
 96           104          0.5
 97           103          0.5
 98           102          0.5
 99           101          0.5
 100          100          0.5
 3
 85           105          0.5
 86           104          0.5
 87           103          0.5
 88           102          0.5
 89           101          0.5
 90           100          0.5
 91           99           0.5
 92           98           0.5
 93           97           0.5
 94           96           0.5
 95           95           0.5
 4
 83.75        103.75       0.5
 84.75        102.75       0.5
 85.75        101.75       0.5
 86.75        100.75       0.5
 87.75        99.75        0.5
 88.75        98.75        0.5
 89.75        97.75        0.5
 90.75        96.75        0.5
 91.75        95.75        0.5
 92.75        94.75        0.5
```

93.75	93.75	0.5
5		
82.5	102.5	0.5
83.5	101.5	0.5
84.5	100.5	0.5
85.5	99.5	0.5
86.5	98.5	0.5
87.5	97.5	0.5
88.5	96.5	0.5
89.5	95.5	0.5
90.5	94.5	0.5
91.5	93.5	0.5
92.5	92.5	0.5
6		
81.25	101.25	0.5
82.25	100.25	0.5
83.25	99.25	0.5
84.25	98.25	0.5
85.25	97.25	0.5
86.25	96.25	0.5
87.25	95.25	0.5
88.25	94.25	0.5
89.25	93.25	0.5
90.25	92.25	0.5
91.25	91.25	0.5
7		
80	100	0.5
81	99	0.5
82	98	0.5
83	97	0.5
84	96	0.5
85	95	0.5
86	94	0.5
87	93	0.5
88	92	0.5
89	91	0.5
90	90	0.5
8		
79.72254	99.78035	0.5
80.72832	98.78035	0.5
81.7341	97.78035	0.5
82.73988	96.78035	0.5
83.74567	95.78035	0.5
84.75145	94.78035	0.5
85.75723	93.78035	0.5
86.76301	92.78035	0.5
87.76879	91.78035	0.5
88.77457	90.78035	0.5
89.78035	89.78035	0.5
9		
79.44508	99.56069	0.5
80.45664	98.56069	0.5
81.46821	97.56069	0.5
82.47977	96.56069	0.5
83.49133	95.56069	0.5
84.50288	94.56069	0.5
85.51445	93.56069	0.5
86.52601	92.56069	0.5
87.53757	91.56069	0.5
88.54913	90.56069	0.5
89.56069	89.56069	0.5
10		
79.16763	99.34104	0.5

80.18497	98.34104	0.5
81.20232	97.34104	0.5
82.21966	96.34104	0.5
83.237	95.34104	0.5
84.25433	94.34104	0.5
85.27168	93.34104	0.5
86.28902	92.34104	0.5
87.30636	91.34104	0.5
88.3237	90.34104	0.5
89.34104	89.34104	0.5
11		
78.89017	99.12138	0.5
79.9133	98.12138	0.5
80.93642	97.12138	0.5
81.95953	96.12138	0.5
82.98266	95.12138	0.5
84.00578	94.12138	0.5
85.0289	93.12138	0.5
86.05202	92.12138	0.5
87.07514	91.12138	0.5
88.09827	90.12138	0.5
89.12138	89.12138	0.5
12		
78.61272	98.90173	0.5
79.64162	97.90173	0.5
80.67052	96.90173	0.5
81.69942	95.90173	0.5
82.72832	94.90173	0.5
83.75723	93.90173	0.5
84.78613	92.90173	0.5
85.81503	91.90173	0.5
86.84393	90.90173	0.5
87.87283	89.90173	0.5
88.90173	88.90173	0.5
13		
78.33526	98.68208	0.5
79.36994	97.68208	0.5
80.40462	96.68208	0.5
81.43931	95.68208	0.5
82.47399	94.68208	0.5
83.50867	93.68208	0.5
84.54335	92.68208	0.5
85.57803	91.68208	0.5
86.61272	90.68208	0.5
87.6474	89.68208	0.5
88.68208	88.68208	0.5
14		
78.0578	98.46243	0.5
79.09827	97.46243	0.5
80.13873	96.46243	0.5
81.17918	95.46243	0.5
82.21965	94.46243	0.5
83.26012	93.46243	0.5
84.30058	92.46243	0.5
85.34103	91.46243	0.5
86.3815	90.46243	0.5
87.42197	89.46243	0.5
88.46243	88.46243	0.5
15		
77.78035	98.24277	0.5
78.82659	97.24277	0.5
79.87283	96.24277	0.5
80.91908	95.24277	0.5

81.96532	94.24277	0.5
83.01157	93.24277	0.5
84.05781	92.24277	0.5
85.10405	91.24277	0.5
86.15029	90.24277	0.5
87.19653	89.24277	0.5
88.24277	88.24277	0.5
16		
77.50289	98.02312	0.5
78.55492	97.02312	0.5
79.60694	96.02312	0.5
80.65896	95.02312	0.5
81.71098	94.02312	0.5
82.76301	93.02312	0.5
83.81503	92.02312	0.5
84.86706	91.02312	0.5
85.91908	90.02312	0.5
86.9711	89.02312	0.5
88.02312	88.02312	0.5
17		
77.22543	97.80347	0.5
78.28323	96.80347	0.5
79.34104	95.80347	0.5
80.39884	94.80347	0.5
81.45665	93.80347	0.5
82.51445	92.80347	0.5
83.57225	91.80347	0.5
84.63006	90.80347	0.5
85.68786	89.80347	0.5
86.74567	88.80347	0.5
87.80347	87.80347	0.5
18		
76.94798	97.58382	0.5
78.01156	96.58382	0.5
79.07514	95.58382	0.5
80.13873	94.58382	0.5
81.20231	93.58382	0.5
82.2659	92.58382	0.5
83.32948	91.58382	0.5
84.39307	90.58382	0.5
85.45665	89.58382	0.5
86.52023	88.58382	0.5
87.58382	87.58382	0.5
19		
76.67052	97.36416	0.5
77.73988	96.36416	0.5
78.80924	95.36416	0.5
79.87861	94.36416	0.5
80.94798	93.36416	0.5
82.01733	92.36416	0.5
83.0867	91.36416	0.5
84.15607	90.36416	0.5
85.22543	89.36416	0.5
86.29479	88.36416	0.5
87.36416	87.36416	0.5
20		
76.39307	97.14451	0.5
77.46821	96.14451	0.5
78.54336	95.14451	0.5
79.6185	94.14451	0.5
80.69364	93.14451	0.5
81.76878	92.14451	0.5
82.84393	91.14451	0.5

83.91908	90.14451	0.5
84.99422	89.14451	0.5
86.06937	88.14451	0.5
87.14451	87.14451	0.5
21		
76.11561	96.92486	0.5
77.19653	95.92486	0.5
78.27746	94.92486	0.5
79.35838	93.92486	0.5
80.43931	92.92486	0.5
81.52023	91.92486	0.5
82.60116	90.92486	0.5
83.68208	89.92486	0.5
84.76301	88.92486	0.5
85.84393	87.92486	0.5
86.92486	86.92486	0.5
22		
75.83815	96.7052	0.5
76.92486	95.7052	0.5
78.01156	94.7052	0.5
79.09827	93.7052	0.5
80.18497	92.7052	0.5
81.27168	91.7052	0.5
82.35838	90.7052	0.5
83.44508	89.7052	0.5
84.53179	88.7052	0.5
85.61849	87.7052	0.5
86.7052	86.7052	0.5
23		
75.56069	96.48555	0.5
76.65318	95.48555	0.5
77.74567	94.48555	0.5
78.83815	93.48555	0.5
79.93063	92.48555	0.5
81.02312	91.48555	0.5
82.11561	90.48555	0.5
83.20809	89.48555	0.5
84.30058	88.48555	0.5
85.39307	87.48555	0.5
86.48555	86.48555	0.5
24		
75.28323	96.2659	0.5
76.3815	95.2659	0.5
77.47977	94.2659	0.5
78.57803	93.2659	0.5
79.6763	92.2659	0.5
80.77457	91.2659	0.5
81.87283	90.2659	0.5
82.9711	89.2659	0.5
84.06937	88.2659	0.5
85.16763	87.2659	0.5
86.2659	86.2659	0.5
25		
75.00578	96.04624	0.5
76.10983	95.04624	0.5
77.21387	94.04624	0.5
78.31792	93.04624	0.5
79.42197	92.04624	0.5
80.52602	91.04624	0.5
81.63006	90.04624	0.5
82.7341	89.04624	0.5
83.83815	88.04624	0.5
84.9422	87.04624	0.5

86.04624	86.04624	0.5
26		
74.72832	95.82659	0.5
75.83815	94.82659	0.5
76.94798	93.82659	0.5
78.05781	92.82659	0.5
79.16763	91.82659	0.5
80.27746	90.82659	0.5
81.38728	89.82659	0.5
82.49711	88.82659	0.5
83.60694	87.82659	0.5
84.71677	86.82659	0.5
85.82659	85.82659	0.5
27		
74.45087	95.60693	0.5
75.56647	94.60693	0.5
76.68208	93.60693	0.5
77.79768	92.60693	0.5
78.91329	91.60693	0.5
80.0289	90.60693	0.5
81.14451	89.60693	0.5
82.26012	88.60693	0.5
83.37572	87.60693	0.5
84.49133	86.60693	0.5
85.60693	85.60693	0.5
28		
74.17341	95.38728	0.5
75.2948	94.38728	0.5
76.41618	93.38728	0.5
77.53757	92.38728	0.5
78.65896	91.38728	0.5
79.78035	90.38728	0.5
80.90173	89.38728	0.5
82.02312	88.38728	0.5
83.14451	87.38728	0.5
84.2659	86.38728	0.5
85.38728	85.38728	0.5
180		
32	62	0.5
34	61	0.5
36	60	0.5
38	59	0.5
40	58	0.5
42	57	0.5
44	56	0.5
46	55	0.5
48	54	0.5
50	53	0.5
52	52	0.5

APPENDIX B:

SAMPLE OUTPUT FILE OF CRCPFOR.OUT

```
This is an output file of CRCP-9
   1.00000000000000
   350.000000000000          184.108663565841
  9.173568787602710E-004
   8317.91391175345
   2.00000000000000
   150.000000000000          10.2678416148951
  1.167077814600686E-002
   20425.4129763802
   3.00000000000000
   95.4545454545455          36.9160315910277
  1.134121552494068E-002
   21112.3148521433
   4.00000000000000
   95.4545454545455          36.9160315910277
  1.228242724288305E-002
   23297.1572909952
   5.00000000000000
   95.4545454545455          36.9160315910277
  1.316733538713401E-002
   25345.6321460105
   6.00000000000000
   95.4545454545455          36.9160315910277
  1.401419105064969E-002
   27303.8289697395
   7.00000000000000
   80.7692307692308          26.4288707117915
  1.346063846300188E-002
   26328.2810886234
   8.00000000000000
   80.7692307692308          26.4288707117915
  1.382684797485888E-002
   27089.7615590943
   9.00000000000000
   80.7692307692308          26.4288707117915
  1.417949902580445E-002
   27819.7574763242
   10.0000000000000
   80.7692307692308          26.4288707117915
  1.452038371574588E-002
   28525.3890353521
   11.0000000000000
   80.7692307692308          26.4288707117915
  1.485247384102198E-002
   29209.8075832916
   12.0000000000000
   80.7692307692308          26.4288707117915
  1.517614075732510E-002
   29876.2543598551
   13.0000000000000
   80.7692307692308          26.4288707117915
  1.549166129568539E-002
   30526.9290479329
   14.0000000000000
   70.0000000000000          26.3969695230343
  1.416315452452465E-002
```

27730.2406346326	
15.0000000000000	
70.0000000000000	26.3969695230343
1.443331017666517E-002	
28278.7753974233	
16.0000000000000	
70.0000000000000	26.3969695230343
1.469831841800895E-002	
28815.5291340378	
17.0000000000000	
65.6250000000000	26.6830353408303
1.420776467140416E-002	
27769.8555372361	
18.0000000000000	
65.6250000000000	26.6830353408303
1.444957980321882E-002	
28255.5849882840	
19.0000000000000	
58.3333333333333	23.2856655954306
1.349902243096111E-002	
26238.7677836583	
20.0000000000000	
55.2631578947368	22.0307013177978
1.317075882281815E-002	
25519.6896929872	
21.0000000000000	
55.2631578947368	22.0307013177978
1.337223197074354E-002	
25916.8489342125	
22.0000000000000	
55.2631578947368	22.0307013177978
1.357024456247735E-002	
26306.7406833351	
23.0000000000000	
52.5000000000000	20.2027225887998
1.326216301606916E-002	
25624.7372746929	
24.0000000000000	
52.5000000000000	20.2027225887998
1.344622140958038E-002	
25985.1871916674	
25.0000000000000	
52.5000000000000	20.2027225887998
1.362704571186657E-002	
26339.6452422606	
26.0000000000000	
52.5000000000000	20.2027225887998
1.380445745711014E-002	
26688.5673180314	
27.0000000000000	
47.7272727272727	17.0405447474327
1.302059168987639E-002	
24989.9745971601	
28.0000000000000	
47.7272727272727	17.0405447474327
1.318078925935605E-002	
25300.6001175648	

```
 180.000000000000
 27.6315789473684         8.26349312092391
1.787270481960061E-002
 33234.2351221087
0.000000000000000E+000
0.000000000000000E+000
0.000000000000000E+000
0.000000000000000E+000
0.000000000000000E+000
0.000000000000000E+000
0.196558969086464
0.237943642110965
0.637187190281326
0.747056310474214
 1.18136419621568
 1.36471713626507
 1.55655795966973
 2.32897808248276
 2.62605759476493
 2.96245294884088
 3.32955709849632
 4.23406724022063
 4.73079726777734
 5.26837431431935
0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000
0.000000000000000E+000  0.000000000000000E+000
 15.7894736842105         15.7894736842105
 28.9473684210526         44.7368421052632
 23.6842105263158         68.4210526315789
 23.6842105263158         92.1052631578947
 2.63157894736842         94.7368421052632
 2.63157894736842         97.3684210526316
 2.63157894736842         100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
0.000000000000000E+000    100.000000000000
```

0.000000000000000E+000	100.000000000000
0.000000000000000E+000	100.000000000000
0.000000000000000E+000	100.000000000000
0.000000000000000E+000	100.000000000000
0.000000000000000E+000	100.000000000000
0.000000000000000E+000	100.000000000000
0.000000000000000E+000	100.000000000000
0.000000000000000E+000	100.000000000000

APPENDIX C:

SAMPLE OUTPUT FILE OF CRCPFOR.SUM

```
CRCP-9
=================================================
     crcpfor
=================================================

*****INPUT VALUES*****

CONCRETE PROPERTIES
===================
Pavement Thickness (in.) =    12.0000000000000
Poisson Ratio of Concrete =   0.150000000000000
Specific Weight of Concrete (pcf) =    145.000000000000
Coefficient of Variation for Concrete Tensile Strength (%) =
  10.0000000000000
Coarse Aggregate Type = Limestone
Thermal Coefficient (/F) =   6.289999873843044E-006
Elastic Modulus at 28 days (psi) = 3604000
Tensile Strength at 28 days (psi) = 434
Compressive Strength at 28 days (psi) = 5000
Drying Shrinkage at 256 days = 0.000425

STEEL PROPERTIES
================
Elastic Modulus of Steel Bar (psi) =    29000000.0000000
Steel Bar Diameter (in.) =   0.750000000000000
Thermal Coefficient of Steel (/F) =   5.000000000000000E-006
Specific Weight of Steel (pcf) =    490.000000000000
Percent Reinforcement (Steel Ratio) (%) =   0.600000000000000
Steel Location from Surface (in.) =    6.00000000000000

BOND-SLIP RELATIONSHIP
======================
1. Bond Between Concrete and Steel Reinforcement
Type of Bond-Slip = User Defined
Bond-Slip Stiffness / Unit Area (psi/in) =    250000.000000000
Secondary B-S Stiffness / Unit Area (psi/in) =    250000.000000000
Yield Slip (in.) =    100.010000000000
Ultimate Slip (in.) =    100.050000000000
2. Bond Between Concrete and Base Layers
Vertical Stiffness of Subgrade (psi/in) =    400.000000000000
Subbase Type = Flexible
Bond-Slip Stiffness / Unit Area (psi/in) =    145.500000000000

LOADS
=====
1. External Load
Wheel Load (lbs) =    9000.00000000000
Wheel Base Radius (in.) =    6.00000000000000
Days After Concrete Sets Before Wheel Load Applied = 14.0000000000000
2. Environmental Load
Curing Temperature (F) =    110.000000000000
Days Before Minimum Temperature =          180
Type of Temperature Variation through Depth = Linear
Type of Drying Shrinkage Variation through Depth = Uniform

TEMPERATURE AND DRYING SHRINKAGE
================================
Depth  Min. Temp.  Max. Temp.  Drying Shrinkage Ratio
( in )   ( F )       ( F )
DAY =            1
 0.00   100.000     110.000          0.500
 1.20   101.000     109.000          0.500
```

```
 2.40   102.000     108.000          0.500
 3.60   103.000     107.000          0.500
 4.80   104.000     106.000          0.500
 6.00   105.000     105.000          0.500
 7.20   106.000     104.000          0.500
 8.40   107.000     103.000          0.500
 9.60   108.000     102.000          0.500
10.80   109.000     101.000          0.500
12.00   110.000     100.000          0.500
DAY =             2
 0.00    90.000     110.000          0.500
 1.20    91.000     109.000          0.500
 2.40    92.000     108.000          0.500
 3.60    93.000     107.000          0.500
 4.80    94.000     106.000          0.500
 6.00    95.000     105.000          0.500
 7.20    96.000     104.000          0.500
 8.40    97.000     103.000          0.500
 9.60    98.000     102.000          0.500
10.80    99.000     101.000          0.500
12.00   100.000     100.000          0.500
DAY =             3
 0.00    85.000     105.000          0.500
 1.20    86.000     104.000          0.500
 2.40    87.000     103.000          0.500
 3.60    88.000     102.000          0.500
 4.80    89.000     101.000          0.500
 6.00    90.000     100.000          0.500
 7.20    91.000      99.000          0.500
 8.40    92.000      98.000          0.500
 9.60    93.000      97.000          0.500
10.80    94.000      96.000          0.500
12.00    95.000      95.000          0.500
DAY =             7
 0.00    80.000     100.000          0.500
 1.20    81.000      99.000          0.500
 2.40    82.000      98.000          0.500
 3.60    83.000      97.000          0.500
 4.80    84.000      96.000          0.500
 6.00    85.000      95.000          0.500
 7.20    86.000      94.000          0.500
 8.40    87.000      93.000          0.500
 9.60    88.000      92.000          0.500
10.80    89.000      91.000          0.500
12.00    90.000      90.000          0.500
DAY =           180
 0.00    32.000      62.000          0.500
 1.20    34.000      61.000          0.500
 2.40    36.000      60.000          0.500
 3.60    38.000      59.000          0.500
 4.80    40.000      58.000          0.500
 6.00    42.000      57.000          0.500
 7.20    44.000      56.000          0.500
 8.40    46.000      55.000          0.500
 9.60    48.000      54.000          0.500
10.80    50.000      53.000          0.500
12.00    52.000      52.000          0.500

ADVANCED INPUTS
===============
Number of Different Primary Crack Spacings =              2
Finite Element Type = Plane Stress
Consideration of Viscoelasticity = Yes
```

Max. Creep Ratio to Instantaneous Strain = 2.00000000000000
Percent Ratio to Max. Creep = 99.0000000000000
Corresponding Day to Above Ratio = 30.0000000000000
Duration of Temperature Drop = 12.0000000000000
Use of Tensionless Springs for Curling Effect = Yes
Punchout Prediction Parameters
Swelling Condition = No
Reliability = 50 %

*****ANALYSIS RESULTS*****

Day	Crack Sp. (ft)	Std. Dev. (ft)	Crack Width (in)	Steel Stress (psi)
1.	29.167	15.342	0.000917	8317.91
2.	12.500	0.856	0.011671	20425.41
3.	7.955	3.076	0.011341	21112.31
4.	7.955	3.076	0.012282	23297.16
5.	7.955	3.076	0.013167	25345.63
6.	7.955	3.076	0.014014	27303.83
7.	6.731	2.202	0.013461	26328.28
8.	6.731	2.202	0.013827	27089.76
9.	6.731	2.202	0.014179	27819.76
10.	6.731	2.202	0.014520	28525.39
11.	6.731	2.202	0.014852	29209.81
12.	6.731	2.202	0.015176	29876.25
13.	6.731	2.202	0.015492	30526.93
14.	5.833	2.200	0.014163	27730.24
15.	5.833	2.200	0.014433	28278.78
16.	5.833	2.200	0.014698	28815.53
17.	5.469	2.224	0.014208	27769.86
18.	5.469	2.224	0.014450	28255.58
19.	4.861	1.940	0.013499	26238.77
20.	4.605	1.836	0.013171	25519.69
21.	4.605	1.836	0.013372	25916.85
22.	4.605	1.836	0.013570	26306.74
23.	4.375	1.684	0.013262	25624.74
24.	4.375	1.684	0.013446	25985.19
25.	4.375	1.684	0.013627	26339.65
26.	4.375	1.684	0.013804	26688.57
27.	3.977	1.420	0.013021	24989.97
28.	3.977	1.420	0.013181	25300.60
180.	2.303	0.689	0.017873	33234.24

PREDICTED PUNCHOUTS / MILE
==========================

Million ESALS	Number of Punchouts / Mile
5	0.000
10	0.000
15	0.000
20	0.000
25	0.000
30	0.000
35	0.197
40	0.238
45	0.637
50	0.747
55	1.181
60	1.365
65	1.557
70	2.329
75	2.626
80	2.962

85	3.330
90	4.234
95	4.731
100	5.268

CRACK SPACING DISTRIBUTION

Range of Crack Sp. (+ 0.25 ft)	Crack Sp. Dist. (%)	Cumulative Value (%)
0.00	0.000	0.000
0.50	0.000	0.000
1.00	0.000	0.000
1.50	15.789	15.789
2.00	28.947	44.737
2.50	23.684	68.421
3.00	23.684	92.105
3.50	2.632	94.737
4.00	2.632	97.368
4.50	2.632	100.000
5.00	0.000	100.000
5.50	0.000	100.000
6.00	0.000	100.000
6.50	0.000	100.000
7.00	0.000	100.000
7.50	0.000	100.000
8.00	0.000	100.000
8.50	0.000	100.000
9.00	0.000	100.000
9.50	0.000	100.000
10.00	0.000	100.000
10.50	0.000	100.000
11.00	0.000	100.000
11.50	0.000	100.000
12.00	0.000	100.000
12.50	0.000	100.000
13.00	0.000	100.000
13.50	0.000	100.000
14.00	0.000	100.000
14.50	0.000	100.000
15.00	0.000	100.000
15.50	0.000	100.000
16.00	0.000	100.000
16.50	0.000	100.000
17.00	0.000	100.000
17.50	0.000	100.000
18.00	0.000	100.000
18.50	0.000	100.000
19.00	0.000	100.000
19.50	0.000	100.000
20.00	0.000	100.000

APPENDIX D:

LIST OF PROGRAM CRCP-9

```
PROGRAM CRCP9
C
C       Version 1.0
C       Computer program to analyze the behavior of CRC pavements
C
C       Developed by Seong-Min Kim, Moon C. Won, and B. Frank McCullough
C
C       Center for Transportation Research, University of Texas at Austin
C       &
C       Texas Department of Transportation
C
C       August 2000
C
        IMPLICIT REAL*8 (A-H,O-Z)
        CHARACTER*40 TITLECRCP
        CHARACTER*60 TITLEINP
        DIMENSION ALOAD(8844),DISP(8844),AKS(2,2),AKSB(4,4),DRYSH(29,21)
        DIMENSION SK(8844,50),AK(8,8),D(3,3),BNSL(200),TEMPR(29,21)
        DIMENSION TEMLD(8844),STRINI(20,4),TLOAD(20,8),TEMPR1(21)
        DIMENSION INOD(4200,8),SS(442200),IMAX(200),IMAXC(200)
        DIMENSION TEMPRA(29,21),DAYCON(29),CRWTH(29),STLSS(29)
        DIMENSION CRSPC(10,4095),CRWID(10,4095),STLSTR(10,4095)
        DIMENSION STRSS(200), STRSSF(400),TSTARR(400),DISTRB(400)
        DIMENSION ICRLOC(10,29),INDCR(10,4095),CRSPAVG(29),CRWDAVG(29)
        DIMENSION CRSPSDD(29),SSTRAVG(29),CRSPDIST(41)
        COMMON NUMELX,NUMELY,A,B,E1,POIS1,THICK,ITNOD1,MBAND
C
C       minimum memory requirements
C
C         ALOAD(ITNOD1),DISP(ITNOD1),SK(ITNOD1,MBAND),BNSL(NUMELX)
C         TEMLD(ITNOD1),STRINI(NUMELY,4),TLOAD(NUMELY,8)
C         INOD(INODDIM,8),SS(KS),IMAX(NUMELX),IMAXC(NUMELX)
C         TEMPR1(NUMELY+1),TEMPR(29,NUMELY+1),TEMPRA(29,NUMELY+1)
C         DRYSH(29,NUMELY+1),STRSS(NUMELX),STRSSF(NUMELX*2)
C         TSTARR(NUMELX*2),DISTRB(NUMELX*2)
C
C      open input and output files
C
        OPEN (5,FILE='crcpfor.inp',STATUS='OLD')
        OPEN (6,FILE='crcpfor.out',STATUS='UNKNOWN')
        OPEN (9,FILE='crcpfor.sum',STATUS='UNKNOWN')
C
C      read input data
C
        READ(5,*) TITLECRCP
        READ(5,*) TITLEINP
        READ(5,*) TOTD,IELTYPE,NUMPCS
        READ(5,*) POIS1,ALPC,WEIGHT,CV,STRAT
        READ(5,*) IAGG,IDEFAULT
        READ(5,*) CSTR28,TSTR28,ECONC28,ZSH256
        READ(5,*) STLOC,ESTL,DIASTL,ALPS,STWEI
        READ(5,*) IBONDCS,AKBUA,AKBUA2,YLSLIP,ULSLIP
        READ(5,*) BOTKV,INDSPR
        READ(5,*) INDFR,AKFRUA
        READ(5,*) WHLOAD,WHRAD,WHDAY
        READ(5,*) ICREEP,PHIMAX,PERCEN,DAYPC,TIME
        READ(5,*) IFATPRO,ISWELL, ITEMIND, IDRYIND
        READ(5,*) TEMF
        READ(5,*) IDAY2, IDAY3, IDAY4, IFDAY
        DO I=1,29
           READ(5,*) DAYCON(I)
           DO J=1,11
```

```
                  READ(5,*) TEMPR(I,J),TEMPRA(I,J),DRYSH(I,J)
        ENDDO
      ENDDO
C
C     WHERE,
C       TOTD = thickness of concrete slab (pavement thickness)
C       IELTYPE = index for element type (1:plane stress, other:plane strain)
C       NUMPCS = no. of different primary crack spacing
C                (max. = 10,  cannot be 3,6,7,9)
C       POIS1 = Poisson's ratio of concrete
C       WEIGHT = specific weight of concrete (pcf)
C       ALPC,ALPS = coefficient of thermal expansion for concrete and steel
C       CV = coefficient of variation for concrete tensile strength (%)
C       STRAT = percent reinforcement (%), steel ratio
C       IAGG = index for aggregate type
C            1 = Granite
C            2 = Dolomite
C            3 = Vega
C            4 = Bridgeport/Tin Top
C            5 = Western Tascosa
C            6 = Ferris
C            7 = Limestone
C            8 = Siliceous River Gravel
C       IDEFAULT = Index for using default values (0=default values)
C       CSTR28 = 28 day concrete compressive strength (put 0 for default)
C       TSTR28 = 28 day concrete tensile strength (put 0 for default)
C       ECONC28 = 28 day concrete elastic modulus (put 0 for default)
C       ZSH256 = 256 day concrete drying shrinkage (put 0 for default)
C       DAYCON(I) = consideration day after placement of concrete
C       STLOC = steel location from surface
C       ESTL,DIASTL = modulus and diameter of longitudinal steel
C       STWEI = specific weight of steel bar (pcf)
C       IBONDCS = Index for B-S (0 to 12) (12: user defined)
C       AKBUA,AKBUA2 = primary and secondary bond-slip stiffness per unit area
C       YLSLIP,ULSLIP = yield and ultimate slips
C       BOTKV = vertical stiffness per unit area of the base layer
C       INDSPR = index for tensionless sprs for foundation (1=no tensionless)
C       INDFR = index for subbase friction (subbase type for friction)
C              1 = Flexible (145.5 pci)
C              2 = Asphalt-Stabilized (55.9)
C              3 = Cement-Stabilized (15400)
C              4 = Lime-Treated Clay (154.5)
C              5 = Untreated Clay (22)
C       AKFRUA = bond-slip stiffness per unit area between concrete and base
C       WHLOAD = wheel load (lbs)
C       WHRAD = radius of loaded area (in.)
C       WHDAY = days before wheel load applied
C       ICREEP = index for creep analysis (1=creep analysis)
C       PHIMAX = maximum creep ratio to instantaneous strain
C       PERCEN = consideration percent to maximum creep
C       DAYPC = corresponding day to PERCEN
C       TIME = loading duration (hr)
C       IFATPRO = Index for reliability of punchout prediction
C                (1:95%, 2:75%, 3:50%)
C       ISWELL = Index for swelling condition (1:swelling=yes)
C       ITEMIND = Index for temp. variation (0:uniform,1:linear,2:nonlinear)
C       IDRYIND = Index for drying shrinkage(" ")
C       IDAY2, IDAY3, IDAY4, IFDAY = Days for temp. and drying shrinkage input
C       TEMF = reference temperature
C       TEMPR(I,J) = minimum temperatures at each node from surface on I day
C       TEMPRA(I,J) = maximum temperatures at each node from surface on I day
C       DRYSH(I,J) = drying shrinkage ratio at each node from the surface
C
```

```
C     define basic variables
C
      WRITE(6,*) 'This is an output file of CRCP-9'
C
      NUMELY=10
      FULLDAY=28
      IHIST=FULLDAY+1
C
      DO I=1,IHIST
             CRWTH(I)=0.
             STLSS(I)=0.
      ENDDO
C
      A=1.5
      B=TOTD/FLOAT(NUMELY)
      ASTL=(DIASTL/2.)*(DIASTL/2.)*3.141592654
      THICK=ASTL*100./(TOTD*STRAT)
      LSTLOC=STLOC/B+1
      IF(LSTLOC.EQ.1) THEN
         WRITE(*,*) 'Steel location from surface must be larger than 0'
         GO TO 1000
      ENDIF
      IF(ALPC.EQ.0.) THEN
             IF(IAGG.EQ.1) ALPC=0.00000574
             IF(IAGG.EQ.2) ALPC=0.0000059
             IF(IAGG.EQ.3) ALPC=0.0000065
             IF(IAGG.EQ.4) ALPC=0.00000484
             IF(IAGG.EQ.5) ALPC=0.00000615
             IF(IAGG.EQ.6) ALPC=0.00000544
             IF(IAGG.EQ.7) ALPC=0.00000629
             IF(IAGG.EQ.8) ALPC=0.00000818
      ENDIF
      AKSTL=ESTL*ASTL/A
      BNAREA=A*3.141592654*DIASTL
      AKBOND=AKBUA*BNAREA
      AKBOND2=AKBUA2*BNAREA
      BNFORCE=YLSLIP*BNAREA*(AKBUA-AKBUA2)
      IF(INDFR.EQ.1) AKFRUA=145.5             ! Flexible
      IF(INDFR.EQ.2) AKFRUA=55.9       ! Asphalt-Stabilized
      IF(INDFR.EQ.3) AKFRUA=15400             ! Cement-Stabilized
      IF(INDFR.EQ.4) AKFRUA=154.5             ! Lime-Treated Clay
      IF(INDFR.EQ.5) AKFRUA=22.0       ! Untreated Clay
      FRAREA=A*THICK
      AKFR=AKFRUA*FRAREA
      BOTK=BOTKV*A*THICK
      MBAND=(NUMELY+3)*2+4
      ELWEI=WEIGHT*A*B*THICK/1728.
      ELSTWEI=STWEI*A*3.141592*DIASTL**2/(4.*1728.)
      SEED=10000001.
C     IF(NUMPCS.EQ.(3.OR.6.OR.7.OR.9)) NUMPCS=NUMPCS+1
      DELPCS=600./FLOAT(2*NUMPCS)
C
      WRITE(9,*) 'CRCP-9'
      WRITE(9,*) '=================================================='
      WRITE(9,*) '     ',TITLEINP
      WRITE(9,*) '=================================================='
      WRITE(9,*) ' '
      WRITE(9,*) '*****INPUT VALUES*****'
      WRITE(9,*) ' '
      WRITE(9,*) 'CONCRETE PROPERTIES'
      WRITE(9,*) '==================='
      WRITE(9,*) 'Pavement Thickness (in.) = ',TOTD
      WRITE(9,*) 'Poisson Ratio of Concrete = ',POIS1
```

```
      WRITE(9,*) 'Specific Weight of Concrete (pcf) = ',WEIGHT
      WRITE(9,*) 'Coefficient of Variation for Concrete Tensile ',
     1          'Strength (%) = ',CV
      IF(IAGG.EQ.1) WRITE(9,*) 'Coarse Aggregate Type = Granite'
      IF(IAGG.EQ.2) WRITE(9,*) 'Coarse Aggregate Type = Dolomite'
      IF(IAGG.EQ.3) WRITE(9,*) 'Coarse Aggregate Type = Vega'
      IF(IAGG.EQ.4) WRITE(9,*) 'Coarse Aggregate Type = ',
     1                         'Bridgeport/Tin Top'
      IF(IAGG.EQ.5) WRITE(9,*) 'Coarse Aggregate Type = Western Tascosa'
      IF(IAGG.EQ.6) WRITE(9,*) 'Coarse Aggregate Type = Ferris'
      IF(IAGG.EQ.7) WRITE(9,*) 'Coarse Aggregate Type = Limestone'
      IF(IAGG.EQ.8) WRITE(9,*) 'Coarse Aggregate Type = ',
     1                         'Siliceous River Gravel'
      WRITE(9,*) 'Thermal Coefficient (/F) = ',ALPC
      IF(IDEFAULT.EQ.1) THEN
        WRITE(9,*) 'Elastic Modulus at 28 days (psi) = ',ECONC28
        WRITE(9,*) 'Tensile Strength at 28 days (psi) = ',TSTR28
        WRITE(9,*) 'Compressive Strength at 28 days (psi) = ',CSTR28
        WRITE(9,*) 'Drying Shrinkage at 256 days = ',ZSH256
      ELSE
        IF(IAGG.EQ.1) THEN
          WRITE(9,*) 'Elastic Modulus at 28 days (psi) = 3356000'
          WRITE(9,*) 'Tensile Strength at 28 days (psi) = 529'
          WRITE(9,*) 'Compressive Strength at 28 days (psi) = 4967'
          WRITE(9,*) 'Drying Shrinkage at 256 days = 0.000394'
        ENDIF
        IF(IAGG.EQ.2) THEN
          WRITE(9,*) 'Elastic Modulus at 28 days (psi) = 4648000'
          WRITE(9,*) 'Tensile Strength at 28 days (psi) = 494'
          WRITE(9,*) 'Compressive Strength at 28 days (psi) = 4470'
          WRITE(9,*) 'Drying Shrinkage at 256 days = 0.000335'
        ENDIF
        IF(IAGG.EQ.3) THEN
          WRITE(9,*) 'Elastic Modulus at 28 days (psi) = 3764000'
          WRITE(9,*) 'Tensile Strength at 28 days (psi) = 442'
          WRITE(9,*) 'Compressive Strength at 28 days (psi) = 3990'
          WRITE(9,*) 'Drying Shrinkage at 256 days = 0.000461'
        ENDIF
        IF(IAGG.EQ.4) THEN
          WRITE(9,*) 'Elastic Modulus at 28 days (psi) = 3984000'
          WRITE(9,*) 'Tensile Strength at 28 days (psi) = 444'
          WRITE(9,*) 'Compressive Strength at 28 days (psi) = 4072'
          WRITE(9,*) 'Drying Shrinkage at 256 days = 0.000399'
        ENDIF
        IF(IAGG.EQ.5) THEN
          WRITE(9,*) 'Elastic Modulus at 28 days (psi) = 3606000'
          WRITE(9,*) 'Tensile Strength at 28 days (psi) = 431'
          WRITE(9,*) 'Compressive Strength at 28 days (psi) = 4132'
          WRITE(9,*) 'Drying Shrinkage at 256 days = 0.000418'
        ENDIF
        IF(IAGG.EQ.6) THEN
          WRITE(9,*) 'Elastic Modulus at 28 days (psi) = 3958000'
          WRITE(9,*) 'Tensile Strength at 28 days (psi) = 479'
          WRITE(9,*) 'Compressive Strength at 28 days (psi) = 3994'
          WRITE(9,*) 'Drying Shrinkage at 256 days = 0.000413'
        ENDIF
        IF(IAGG.EQ.7) THEN
          WRITE(9,*) 'Elastic Modulus at 28 days (psi) = 3604000'
          WRITE(9,*) 'Tensile Strength at 28 days (psi) = 434'
          WRITE(9,*) 'Compressive Strength at 28 days (psi) = 5000'
          WRITE(9,*) 'Drying Shrinkage at 256 days = 0.000425'
        ENDIF
        IF(IAGG.EQ.8) THEN
```

```
      WRITE(9,*) 'Elastic Modulus at 28 days (psi) = 4564000'
      WRITE(9,*) 'Tensile Strength at 28 days (psi) = 462'
      WRITE(9,*) 'Compressive Strength at 28 days (psi) = 4876'
      WRITE(9,*) 'Drying Shrinkage at 256 days = 0.000342'
    ENDIF
  ENDIF
  WRITE(9,*) ' '
  WRITE(9,*) 'STEEL PROPERTIES'
  WRITE(9,*) '================'
  WRITE(9,*) 'Elastic Modulus of Steel Bar (psi) = ',ESTL
  WRITE(9,*) 'Steel Bar Diameter (in.) = ',DIASTL
  WRITE(9,*) 'Thermal Coefficient of Steel (/F) = ',ALPS
  WRITE(9,*) 'Specific Weight of Steel (pcf) = ',STWEI
  WRITE(9,*) 'Percent Reinforcement (Steel Ratio) (%) = ',STRAT
  WRITE(9,*) 'Steel Location from Surface (in.) = ',STLOC
  WRITE(9,*) ' '
  WRITE(9,*) 'BOND-SLIP RELATIONSHIP'
  WRITE(9,*) '======================'
  WRITE(9,*) '1. Bond Between Concrete and Steel Reinforcement'
  IF(IBONDCS.EQ.0) THEN
    WRITE(9,*) 'Type of Bond-Slip = Strong Linear'
    WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
  ENDIF
  IF(IBONDCS.EQ.1) THEN
    WRITE(9,*) 'Type of Bond-Slip = Moderate Linear'
    WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
  ENDIF
  IF(IBONDCS.EQ.2) THEN
    WRITE(9,*) 'Type of Bond-Slip = Weak Linear'
    WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
  ENDIF
  IF(IBONDCS.EQ.3) THEN
    WRITE(9,*) 'Type of Bond-Slip = ',
1               'Strong Linear with Ultimate Slip'
    WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
    WRITE(9,*) 'Ultimate Slip (in.) = ',ULSLIP
  ENDIF
  IF(IBONDCS.EQ.4) THEN
    WRITE(9,*) 'Type of Bond-Slip = ',
1               'Moderate Linear with Ultimate Slip'
    WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
    WRITE(9,*) 'Ultimate Slip (in.) = ',ULSLIP
  ENDIF
  IF(IBONDCS.EQ.5) THEN
    WRITE(9,*) 'Type of Bond-Slip = Weak Linear with Ultimate Slip'
    WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
    WRITE(9,*) 'Ultimate Slip (in.) = ',ULSLIP
  ENDIF
  IF(IBONDCS.EQ.6) THEN
    WRITE(9,*) 'Type of Bond-Slip = Strong Bilinear'
    WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
    WRITE(9,*) 'Secondary B-S Stiffness / Unit Area (psi/in) = ',
1               AKBUA2
    WRITE(9,*) 'Yield Slip (in.) = ',YLSLIP
  ENDIF
  IF(IBONDCS.EQ.7) THEN
    WRITE(9,*) 'Type of Bond-Slip = Moderate Bilinear'
    WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
    WRITE(9,*) 'Secondary B-S Stiffness / Unit Area (psi/in) = ',
1               AKBUA2
    WRITE(9,*) 'Yield Slip (in.) = ',YLSLIP
  ENDIF
  IF(IBONDCS.EQ.8) THEN
```

```
      WRITE(9,*) 'Type of Bond-Slip = Weak Bilinear'
      WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
      WRITE(9,*) 'Secondary B-S Stiffness / Unit Area (psi/in) = ',
1                AKBUA2
      WRITE(9,*) 'Yield Slip (in.) = ',YLSLIP
    ENDIF
    IF(IBONDCS.EQ.9) THEN
      WRITE(9,*) 'Type of Bond-Slip = ',
1                'Strong Bilinear with Ultimate Slip'
      WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
      WRITE(9,*) 'Secondary B-S Stiffness / Unit Area (psi/in) = ',
1                AKBUA2
      WRITE(9,*) 'Yield Slip (in.) = ',YLSLIP
      WRITE(9,*) 'Ultimate Slip (in.) = ',ULSLIP
    ENDIF
    IF(IBONDCS.EQ.10) THEN
      WRITE(9,*) 'Type of Bond-Slip = ',
1                'Moderate Bilinear with Ultimate Slip'
      WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
      WRITE(9,*) 'Secondary B-S Stiffness / Unit Area (psi/in) = ',
1                AKBUA2
      WRITE(9,*) 'Yield Slip (in.) = ',YLSLIP
      WRITE(9,*) 'Ultimate Slip (in.) = ',ULSLIP
    ENDIF
    IF(IBONDCS.EQ.11) THEN
      WRITE(9,*) 'Type of Bond-Slip = ',
1                'Weak Bilinear with Ultimate Slip'
      WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
      WRITE(9,*) 'Secondary B-S Stiffness / Unit Area (psi/in) = ',
1                AKBUA2
      WRITE(9,*) 'Yield Slip (in.) = ',YLSLIP
      WRITE(9,*) 'Ultimate Slip (in.) = ',ULSLIP
    ENDIF
    IF(IBONDCS.EQ.12) THEN
      WRITE(9,*) 'Type of Bond-Slip = User Defined'
      WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKBUA
      WRITE(9,*) 'Secondary B-S Stiffness / Unit Area (psi/in) = ',
1                AKBUA2
      WRITE(9,*) 'Yield Slip (in.) = ',YLSLIP
      WRITE(9,*) 'Ultimate Slip (in.) = ',ULSLIP
    ENDIF
    WRITE(9,*) '2. Bond Between Concrete and Base Layers'
    WRITE(9,*) 'Vertical Stiffness of Subgrade (psi/in) = ',BOTKV
    IF(INDFR.EQ.1) WRITE(9,*) 'Subbase Type = Flexible'
    IF(INDFR.EQ.2) WRITE(9,*) 'Subbase Type = Asphalt-Stabilized'
    IF(INDFR.EQ.3) WRITE(9,*) 'Subbase Type = Cement-Stabilized'
    IF(INDFR.EQ.4) WRITE(9,*) 'Subbase Type = Lime-Treated Clay'
    IF(INDFR.EQ.5) WRITE(9,*) 'Subbase Type = Untreated Clay'
    IF(INDFR.EQ.0) WRITE(9,*) 'Subbase Type = User Defined'
    WRITE(9,*) 'Bond-Slip Stiffness / Unit Area (psi/in) = ',AKFRUA
    WRITE(9,*) ' '
    WRITE(9,*) 'LOADS'
    WRITE(9,*) '====='
    WRITE(9,*) '1. External Load'
    WRITE(9,*) 'Wheel Load (lbs) = ',WHLOAD
    WRITE(9,*) 'Wheel Base Radius (in.) = ',WHRAD
    WRITE(9,*) 'Days After Concrete Sets Before Wheel Load Applied',
1              ' = ',WHDAY
    WRITE(9,*) '2. Environmental Load'
    WRITE(9,*) 'Curing Temperature (F) = ',TEMF
    WRITE(9,*) 'Days Before Minimum Temperature = ',IFDAY
    IF(ITEMIND.EQ.0) WRITE(9,*) 'Type of Temperature Variation ',
1                   'through Depth = Uniform'
```

```
      IF(ITEMIND.EQ.1) WRITE(9,*) 'Type of Temperature Variation ',
     1                 'through Depth = Linear'
      IF(ITEMIND.EQ.2) WRITE(9,*) 'Type of Temperature Variation ',
     1                 'through Depth = Nonlinear'
      IF(IDRYIND.EQ.0) WRITE(9,*) 'Type of Drying Shrinkage Variation ',
     1                 'through Depth = Uniform'
      IF(IDRYIND.EQ.1) WRITE(9,*) 'Type of Drying Shrinkage Variation ',
     1                 'through Depth = Linear'
      IF(IDRYIND.EQ.2) WRITE(9,*) 'Type of Drying Shrinkage Variation ',
     1                 'through Depth = Nonlinear'
      WRITE(9,*) ' '
      WRITE(9,*) 'TEMPERATURE AND DRYING SHRINKAGE'
      WRITE(9,*) '================================'
      DO I=1,29
       IF(I.EQ.1) THEN
          WRITE(9,*) 'Depth',' Min. Temp.',' Max. Temp.',
     1             ' Drying Shrinkage Ratio'
          WRITE(9,*) '( in )','   ( F )','       ( F )'
          WRITE(9,*) 'DAY = ',I
          DO J=1,11
            DEPPR=FLOAT(J-1)*TOTD/10.
               WRITE(9,9010) DEPPR,TEMPR(I,J),TEMPRA(I,J),DRYSH(I,J)
          ENDDO
       ENDIF
       IF(I.EQ.IDAY2) THEN
          WRITE(9,*) 'DAY = ',I
          DO J=1,11
            DEPPR=FLOAT(J-1)*TOTD/10.
               WRITE(9,9010) DEPPR,TEMPR(I,J),TEMPRA(I,J),DRYSH(I,J)
          ENDDO
       ENDIF
       IF(I.EQ.IDAY3) THEN
          WRITE(9,*) 'DAY = ',I
          DO J=1,11
            DEPPR=FLOAT(J-1)*TOTD/10.
               WRITE(9,9010) DEPPR,TEMPR(I,J),TEMPRA(I,J),DRYSH(I,J)
          ENDDO
       ENDIF
       IF(I.EQ.IDAY4) THEN
          WRITE(9,*) 'DAY = ',I
          DO J=1,11
            DEPPR=FLOAT(J-1)*TOTD/10.
               WRITE(9,9010) DEPPR,TEMPR(I,J),TEMPRA(I,J),DRYSH(I,J)
          ENDDO
       ENDIF
       IF(I.EQ.29) THEN
          WRITE(9,*) 'DAY = ',IFDAY
          DO J=1,11
            DEPPR=FLOAT(J-1)*TOTD/10.
               WRITE(9,9010) DEPPR,TEMPR(I,J),TEMPRA(I,J),DRYSH(I,J)
          ENDDO
       ENDIF
      ENDDO
 9010 FORMAT(F6.2,3x,F7.3,5x,F7.3,8x,F7.3)
      WRITE(9,*) ' '
      WRITE(9,*) 'ADVANCED INPUTS'
      WRITE(9,*) '==============='
      WRITE(9,*) 'Number of Different Primary Crack Spacings = ',NUMPCS
      IF(IELTYPE.EQ.1) THEN
        WRITE(9,*) 'Finite Element Type = Plane Stress'
      ELSE
        WRITE(9,*) 'Finite Element Type = Plane Strain'
      ENDIF
```

```
      IF(ICREEP.EQ.1) THEN
        WRITE(9,*) 'Consideration of Viscoelasticity = Yes'
        WRITE(9,*) 'Max. Creep Ratio to Instantaneous Strain = ',PHIMAX
        WRITE(9,*) 'Percent Ratio to Max. Creep = ',PERCEN
        WRITE(9,*) 'Corresponding Day to Above Ratio = ',DAYPC
        WRITE(9,*) 'Duration of Temperature Drop = ',TIME
      ELSE
        WRITE(9,*) 'Consideration of Viscoelasticity = No'
      ENDIF
      IF(INDSPR.EQ.0) THEN
        WRITE(9,*) 'Use of Tensionless Springs for Curling Effect = Yes'
      ELSE
        WRITE(9,*) 'Use of Tensionless Springs for Curling Effect = No'
      ENDIF
      WRITE(9,*) 'Punchout Prediction Parameters'
      IF(ISWELL.EQ.1) THEN
        WRITE(9,*) 'Swelling Condition = Yes'
      ELSE
        WRITE(9,*) 'Swelling Condition = No'
      ENDIF
      IF(IFATPRO.EQ.1) WRITE(9,*) 'Reliability = 95 %'
      IF(IFATPRO.EQ.2) WRITE(9,*) 'Reliability = 75 %'
      IF(IFATPRO.EQ.3) WRITE(9,*) 'Reliability = 50 %'
C
      WRITE(9,*) ' '
      WRITE(9,*) ' '
      WRITE(9,*) '*****ANALYSIS RESULTS*****'
      WRITE(9,*) ' '
      WRITE(9,*) 'Day    Crack Sp. Std. Dev.  Crack Width  Steel Stress'
      WRITE(9,*) '        ( ft )    ( ft )      ( in )       ( psi )'
C
C
      DO 3333 IPCS=1,NUMPCS
C
      CRSPC(IPCS,1)=600.-FLOAT(IPCS-1)*DELPCS
      INDCR(IPCS,1)=1
      DO I=2,4095
             INDCR(IPCS,I)=0
      ENDDO
C
      SEED=SEED+3000.
      CALL STRDIST(SEED,DISTRB,CV,CRSPC,IPCS,A)
C
      ICRACK=1
C
      DO 2222 IDAY=1,IHIST
C
      ECONDAY=DAYCON(IDAY)
C
  150 CONTINUE
C      WRITE(*,*) 'CRACK =',ICRACK
C
      JCRACK1=2**(ICRACK-1)
      JCRACK2=2**ICRACK-1
C
      DO 4444 JCRACK=JCRACK1,JCRACK2
C
      IF(CRSPC(IPCS,JCRACK).EQ.0.) THEN
             II=JCRACK*2
             CRSPC(IPCS,II)=0.
             CRSPC(IPCS,II+1)=0.
             GO TO 333
      ENDIF
```

```
      IF(INDCR(IPCS,JCRACK).EQ.0) THEN
            II=JCRACK*2
            CRSPC(IPCS,II)=CRSPC(IPCS,JCRACK)
            CRSPC(IPCS,II+1)=0.
            CRWID(IPCS,JCRACK)=CRWID(IPCS,JCRACK/2)
            STLSTR(IPCS,JCRACK)=STLSTR(IPCS,JCRACK/2)
            GO TO 333
      ENDIF
      NA=1
      JOO=1
C
      TOTL=CRSPC(IPCS,JCRACK)/2.
      NUMELX=TOTL/A
      NUM2=NUMELX*2
C
  180 CONTINUE
  190 CONTINUE
C
      DO I=1,NUMELX
         IMAXC(I)=0
      ENDDO
      MIN=1
  170  CONTINUE
C
      ITNOD=(NUMELX+1)*(NUMELY+1)
      ITNOD1=ITNOD*2+(NUMELX+1)*2
      KS=ITNOD1*MBAND
      INODDIM=(NUMELY+1)*NUMELX
      IF(JOO.EQ.1) THEN
         DELT=TEMPR(IDAY,LSTLOC)-TEMF
         DO I=1,NUMELY+1
            TEMPR1(I)=TEMPR(IDAY,I)
         ENDDO
      ELSE
         DELT=TEMPRA(IDAY,LSTLOC)-TEMF
         DO I=1,NUMELY+1
            TEMPR1(I)=TEMPRA(IDAY,I)
         ENDDO
      ENDIF
C
      CALL CHMEMO(KS,INODDIM,ICHMEMO)
C     This program can take up to 200 by 20 elements
C
      IF(ICHMEMO.EQ.1) GO TO 1000
C
      CALL CONCPRO(IAGG,CSTR28,TSTR28,ECONC28,ZSH256,
     +                 ECONDAY,CSTR,TSTRNG,ECONC,ZSH)
      IF(ECONDAY.LE.28.) ACSTR28=CSTR
C
      IF(ICREEP.EQ.1) THEN
        CALL CREEP(TIME,ECONC,E1,PHIMAX,PERCEN,DAYPC)
      ELSE
        E1=ECONC
      ENDIF
C
      CALL LINREC4(AK,D,IELTYPE)
C
      CALL FRAME(A,ESTL,DIASTL,AKSTL,AKS,AKSB)
C
      CALL SHRINK(DRYSH,TEMPR1,ALPC,IDAY,ZSH)
C
      CALL TEMPRT(TEMPR1,TEMF,ALPC,TLOAD,STRINI,D,IELTYPE)
C
```

```
      KIM=1
      IULSLP=0
      DO I=1,NUMELX
        IMAX(I)=0
      ENDDO
C
  100 CONTINUE
C
      DO I=1,ITNOD1
        ALOAD(I)=0.
      ENDDO
C
      CALL ASSEM(AK,SK,INOD,LSTLOC,AKS,TLOAD,TEMLD,AKSB)
C
      CALL BOUND(SK,LSTLOC)
C
      CALL BNSLIP(SK,AKBOND,AKBOND2,LSTLOC,KIM,ALOAD,IMAX,BNFORCE,
     +             BNSLMAXS,BNSL,ULSLIP)
C
      CALL FRSLIP(SK,AKFR,MIN,IMAXC)
C
      CALL CURL(SK,BOTK,MIN,IMAXC)
C
      DO 101 I=1,ITNOD1
      DO 102 J=1,MBAND
        K=(J-1)*ITNOD1+I
        SS(K)=SK(I,J)
  102 CONTINUE
        ALOAD(I)=ALOAD(I)+TEMLD(I)
  101 CONTINUE
C
      CALL DWEIGHT(ALOAD,ELWEI,ELSTWEI,LSTLOC)
C
      CALL SOLVE(ITNOD1,MBAND,ALOAD,SS,KS)
C
      DO I=1,ITNOD1
        DISP(I)=ALOAD(I)
      ENDDO
C
      CALL CHSLIP(DISP,IMAX,KIM,BNSL,BNSLMAX,BNSLMAXS,LSTLOC,YLSLIP,
     +             IULSL,ULSLIP)
C
      IF(IMAX(KIM).EQ.1) GO TO 555
C
      IF(BNSLMAX.GT.YLSLIP) THEN
         KIM=KIM+1
         IULSLP=IULSL
         GO TO 100
      ENDIF
C
      IF(IULSL.NE.IULSLP) THEN
C        write(*,*) 'IULSLP and IULSL',IULSLP,IULSL
         IULSLP=IULSL
         GO TO 100
      ENDIF
C
  555 continue
C
      IF(INDSPR.EQ.1) GO TO 105
C
      CALL CHCURL(DISP,MIN,IMAXC,INDCURL)
C
      IF(INDCURL.EQ.1) GO TO 170
```

```
C
  105  CONTINUE
C
       INDST=0
       IF(NA.EQ.1) THEN
         CALL COMPST(STRINI,INOD,LSTLOC,ESTL,ALPS,JOO,NA,INDMA,
     +                DELT,DISP,D,CHSTRES,INDST,CSTR1,SSTR1,WHDAY,
     +                  TOTD,WHRAD,WHLOAD,ECONC,BOTKV,ECONDAY)
          IF(INDMA.EQ.1) GO TO 190
          IF(INDMA.EQ.2) GO TO 180
       ENDIF
C
       IWIDTH=ITNOD1-2*(NUMELY+2)+1
       CRWID(IPCS,JCRACK)=DABS(2.*DISP(IWIDTH))
C
       CALL TSTRTH(CRSPC,IPCS,JCRACK,JCRACK1,DISTRB,TSTRNG,A,
     +             NUM2,TSTARR)
C
       DO III=1,NUMELX
              STRSS(III)=0.
        DO 400 JSTR=1,NUMELY+1
          IF(JSTR.EQ.LSTLOC) GO TO 400
          IF(JSTR.EQ.LSTLOC-1) GO TO 400
          IF(JSTR.EQ.LSTLOC+1) GO TO 400
           IELINP=(III-1)*(NUMELY+1)+JSTR
          CALL STRESS(IELINP,STRINI,INOD,LSTLOC,ESTL,ALPS,DELT,DISP,
     +               D,CHSTRES,INDST,CSTR1,SSTR1)
           IF(ECONDAY.GE.WHDAY) THEN
                CALL WHEEL(TOTD,WHRAD,WHLOAD,ECONC,BOTKV,WHSTR)
             IF(JSTR.LT.LSTLOC) THEN
                     CHWHSTR=WHSTR*(FLOAT(2*JSTR-1)/FLOAT(NUMELY)-1.)
             ELSE
                CHWHSTR=WHSTR*(FLOAT(2*JSTR-3)/FLOAT(NUMELY)-1.)
                ENDIF
                CHSTRES=CHSTRES+CHWHSTR
           ENDIF
           IF(CHSTRES.GT.STRSS(III)) STRSS(III)=CHSTRES
  400   CONTINUE
       ENDDO
C
       DO I=1,NUMELX
              J=NUMELX+I
              K=NUMELX+1-I
              STRSSF(J)=STRSS(I)
              STRSSF(K)=STRSS(I)
c      write(6,*) i,'  ',strss(i)
       ENDDO
C
       DIFMAX=0.
       DO I=1,NUM2
              DIFE=STRSSF(I)-TSTARR(I)
              IF(DIFE.GT.DIFMAX) THEN
                     DIFMAX=DIFE
                     IDIFLOC=I
              ENDIF
       ENDDO
       II=JCRACK*2
       IF(DIFMAX.EQ.0.) THEN
              CRSPC(IPCS,II)=CRSPC(IPCS,JCRACK)
              CRSPC(IPCS,II+1)=0.
       ELSE
              J=IDIFLOC/2
              IDIFLOC=J*2
```

```
            IF(IDIFLOC.EQ.0.) THEN
                   CRSPC(IPCS,II)=CRSPC(IPCS,JCRACK)
                   CRSPC(IPCS,II+1)=0.
            ELSE
                   CRSPC(IPCS,II)=A*FLOAT(IDIFLOC)
                   CRSPC(IPCS,II+1)=FLOAT(NUM2)*A-CRSPC(IPCS,II)
                   INDCR(IPCS,II)=1
                   INDCR(IPCS,II+1)=1
            ENDIF
      ENDIF
C
      INDST=2
      CALL STRESS(IELINP,STRINI,INOD,LSTLOC,ESTL,ALPS,DELT,DISP,D,
     +            CHSTRES,INDST,CSTR1,SSTR1)
      STLSTR(IPCS,JCRACK)=SSTR1
C
  333 CONTINUE
C
      IF(JCRACK.EQ.JCRACK2) THEN
C     do n=jcrack1,jcrack2
C     write(6,*) n,'  ',crspc(ipcs,n),'  ',CRWID(IPCS,N),' ',
C    +           INDCR(IPCS,N)
C     enddo
            DO M=JCRACK1,JCRACK2
                   MM=M*2+1
                   IF(CRSPC(IPCS,MM).NE.0.) THEN
                          IF(ICRACK.EQ.11) THEN
C                                WRITE(6,*) ECONDAY,' ',IPCS,' ',ICRACK
                                 GO TO 4444
                          ELSE
                                 ICRACK=ICRACK+1
                                 GO TO 150
                          ENDIF
                   ENDIF
            ENDDO
      ENDIF
C
 4444 CONTINUE
C
      ICRLOC(IPCS,IDAY)=ICRACK
      DO I=JCRACK1,JCRACK2
            INDCR(IPCS,I)=1
            CRWTH(IDAY)=CRWTH(IDAY)+CRWID(IPCS,I)
            STLSS(IDAY)=STLSS(IDAY)+STLSTR(IPCS,I)
      ENDDO
C
      IF(IDAY.EQ.4) THEN
        IF(IPCS.EQ.1) THEN
            OPEN (8,FILE='crcp01.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.2) THEN
            OPEN (8,FILE='crcp07.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.3) THEN
            OPEN (8,FILE='crcp13.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.4) THEN
            OPEN (8,FILE='crcp19.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
```

```
      IF(IPCS.EQ.5) THEN
           OPEN (8,FILE='crcp25.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.6) THEN
           OPEN (8,FILE='crcp31.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.7) THEN
           OPEN (8,FILE='crcp37.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.8) THEN
           OPEN (8,FILE='crcp43.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
    ENDIF
C
    IF(IDAY.EQ.9) THEN
      IF(IPCS.EQ.1) THEN
           OPEN (8,FILE='crcp02.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.2) THEN
           OPEN (8,FILE='crcp08.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.3) THEN
           OPEN (8,FILE='crcp14.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.4) THEN
           OPEN (8,FILE='crcp20.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.5) THEN
           OPEN (8,FILE='crcp26.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.6) THEN
           OPEN (8,FILE='crcp32.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.7) THEN
           OPEN (8,FILE='crcp38.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.8) THEN
           OPEN (8,FILE='crcp44.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
    ENDIF
C
    IF(IDAY.EQ.14) THEN
      IF(IPCS.EQ.1) THEN
           OPEN (8,FILE='crcp03.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.2) THEN
           OPEN (8,FILE='crcp09.out',STATUS='UNKNOWN')
           CLOSE(8)
      ENDIF
      IF(IPCS.EQ.3) THEN
```

```
            OPEN (8,FILE='crcp15.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.4) THEN
            OPEN (8,FILE='crcp21.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.5) THEN
            OPEN (8,FILE='crcp27.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.6) THEN
            OPEN (8,FILE='crcp33.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.7) THEN
            OPEN (8,FILE='crcp39.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.8) THEN
            OPEN (8,FILE='crcp45.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
      ENDIF
C
      IF(IDAY.EQ.19) THEN
        IF(IPCS.EQ.1) THEN
            OPEN (8,FILE='crcp04.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.2) THEN
            OPEN (8,FILE='crcp10.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.3) THEN
            OPEN (8,FILE='crcp16.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.4) THEN
            OPEN (8,FILE='crcp22.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.5) THEN
            OPEN (8,FILE='crcp28.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.6) THEN
            OPEN (8,FILE='crcp34.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.7) THEN
            OPEN (8,FILE='crcp40.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
        IF(IPCS.EQ.8) THEN
            OPEN (8,FILE='crcp46.out',STATUS='UNKNOWN')
            CLOSE(8)
        ENDIF
      ENDIF
C
      IF(IDAY.EQ.24) THEN
        IF(IPCS.EQ.1) THEN
            OPEN (8,FILE='crcp05.out',STATUS='UNKNOWN')
```

```
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.2) THEN
          OPEN (8,FILE='crcp11.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.3) THEN
          OPEN (8,FILE='crcp17.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.4) THEN
          OPEN (8,FILE='crcp23.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.5) THEN
          OPEN (8,FILE='crcp29.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.6) THEN
          OPEN (8,FILE='crcp35.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.7) THEN
          OPEN (8,FILE='crcp41.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.8) THEN
          OPEN (8,FILE='crcp47.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
      ENDIF
C
      IF(IDAY.EQ.IHIST) THEN
        IF(IPCS.EQ.1) THEN
          OPEN (8,FILE='crcp06.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.2) THEN
          OPEN (8,FILE='crcp12.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.3) THEN
          OPEN (8,FILE='crcp18.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.4) THEN
          OPEN (8,FILE='crcp24.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.5) THEN
          OPEN (8,FILE='crcp30.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.6) THEN
          OPEN (8,FILE='crcp36.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.7) THEN
          OPEN (8,FILE='crcp42.out',STATUS='UNKNOWN')
          CLOSE(8)
        ENDIF
        IF(IPCS.EQ.8) THEN
          OPEN (8,FILE='crcp48.out',STATUS='UNKNOWN')
```

```
            CLOSE(8)
         ENDIF
       ENDIF
C
 2222  CONTINUE
C
 3333  CONTINUE
C
       DO I=1,IHIST
             ISUM=0
             CRSPSUM=0.
             CRSPSUM2=0.
             DO J=1,NUMPCS
                   K1=2**(ICRLOC(J,I)-1)
                   K2=2**ICRLOC(J,I)-1
                   DO K=K1,K2
                         IF(CRSPC(J,K).NE.0.) THEN
                               ISUM=ISUM+1
                               CRSPSUM=CRSPSUM+CRSPC(J,K)
C      if(i.eq.ihist) write(6,*) crspc(j,k)
                               CRSPSUM2=CRSPSUM2+CRSPC(J,K)*CRSPC(J,K)
                         ENDIF
                   ENDDO
             ENDDO
C      if(i.eq.ihist) write(6,*) 'ISUM',isum
C
             CRSPAVG(I)=CRSPSUM/FLOAT(ISUM)
             CRWDAVG(I)=CRWTH(I)/FLOAT(ISUM)
             SSTRAVG(I)=STLSS(I)/FLOAT(ISUM)
             CRSPSDD(I)=DSQRT((CRSPSUM2-FLOAT(ISUM)*CRSPAVG(I)*
     +                 CRSPAVG(I))/FLOAT(ISUM))
             WRITE(6,*) DAYCON(I)
             WRITE(6,*) CRSPAVG(I),CRSPSDD(I)
             WRITE(6,*) CRWDAVG(I)
             WRITE(6,*) SSTRAVG(I)
             WRITE(9,9001) DAYCON(I),CRSPAVG(I)/12.,CRSPSDD(I)/12.,
     +                  CRWDAVG(I),SSTRAVG(I)
 9001     FORMAT(1x,F4.0,3x,F6.3,4x,F6.3,4x,F10.6,4x,F10.2)
       ENDDO
C
       CALL PUNCHOUT(CRSPC,TOTD,IAGG,ISWELL,ACSTR28,ICRLOC,
     +                IFATPRO,IHIST,NUMPCS)
C
       DO I=1,41
             CRSPDIST(I)=0.
       ENDDO
C
C      IF(INDDAY.EQ.1) THEN
C            I=28
C      ELSE
             I=IHIST
C      ENDIF
C
       ISUM1=0
C
       DO J=1,NUMPCS
             K1=2**(ICRLOC(J,I)-1)
             K2=2**ICRLOC(J,I)-1
             DO K=K1,K2
                   IF(CRSPC(J,K).NE.0.) THEN
                         ISUM1=ISUM1+1
                         POS=(CRSPC(J,K)/12.-0.25)/0.5+2.
                         IPOS=POS
```

```
                        IF(IPOS.GT.22) IPOS=22
                        CRSPDIST(IPOS)=CRSPDIST(IPOS)+1.
                    ENDIF
                ENDDO
            ENDDO
C
            CUM=0.
            WRITE(9,*) ' '
            WRITE(9,*) 'CRACK SPACING DISTRIBUTION'
            WRITE(9,*) '=========================='
            WRITE(9,*) 'Range of Crack Sp.   Crack Sp. Dist. Cumulative Value'
            WRITE(9,*) ' ( +- 0.25 ft )          ( % )           ( % )'
            DO I=1,41
                CRSPDIST(I)=CRSPDIST(I)/FLOAT(ISUM1)*100.
                AI=FLOAT(I)*0.5-0.5
                CUM=CUM+CRSPDIST(I)
                WRITE(6,*) CRSPDIST(I),CUM
                WRITE(9,9002) AI,CRSPDIST(I),CUM
 9002       FORMAT(4x,F6.2,14x,F7.3,10x,F7.3)
            ENDDO
C
 1000 CONTINUE
      OPEN (7,FILE='crcpend.out',STATUS='UNKNOWN')
       CLOSE(7)
      STOP
      END
C*****************************************************************
       SUBROUTINE STRDIST(SEED,DISTRB,CV,CRSPC,IPCS,A)
C
C      to impose random coefficients to each element of the pavement
C      for variation of concrete tensile strength
C
      IMPLICIT REAL*8 (A-H, O-Z)
       DIMENSION DISTRB(400),CRSPC(10,4095)
C
       J=CRSPC(IPCS,1)/A
       DO I=1,J
              DISTRB(I)=GGNQF()
       ENDDO
       DO I=1,J
              DISTRB(I)=DISTRB(I)*CV/100.+1.
              IF(DISTRB(I).LT.0.) DISTRB(I)=0.
       ENDDO
C
       RETURN
       END
C
      FUNCTION GGNQF()
C
      REAL*4 X1,T,RANVAL
C
 10   CALL RANDOM(RANVAL)
      IF (RANVAL.EQ.0) GOTO 10
C
      X1=SQRT(-2.*LOG(RANVAL))
C
 20   CALL RANDOM(RANVAL)
      IF (RANVAL.EQ.0) GOTO 20
C
      T=6.2831853072*RANVAL
C
      GGNQF = (X1*COS(T))
C
```

```
      RETURN
      END
C*************************************************************
      SUBROUTINE TSTRTH(CRSPC,IPCS,JCRACK,JCRACK1,DISTRB,TSTRNG,A,
     +                          NUM2,TSTARR)
C
C      to calculate tensile strength distribution at each element
C     of the pavement
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION DISTRB(400),CRSPC(10,4095),TSTARR(400)
C
      DO I=1,NUM2
              TSTARR(I)=0.
      ENDDO
C
      IF(JCRACK.EQ.JCRACK1) THEN
              DO I=1,NUM2
                      TSTARR(I)=DISTRB(I)*TSTRNG
              ENDDO
      ELSE
              SP1=0.
              DO I=JCRACK1,JCRACK-1
                      SP1=SP1+CRSPC(IPCS,I)
              ENDDO
              N1=SP1/A
              DO I=1,NUM2
                      TSTARR(I)=DISTRB(N1+I)*TSTRNG
              ENDDO
      ENDIF
C
      RETURN
      END
C*************************************************************
      SUBROUTINE CHMEMO(KS,INODDIM,ICHMEMO)
C
C     to check required dimensions of the arrays
C
C     minimum memory requirements
C       ALOAD(ITNOD1),DISP(ITNOD1),SK(ITNOD1,MBAND),BNSL(NUMELX)
C       TEMLD(ITNOD1),STRINI(NUMELY,4),TLOAD(NUMELY,8)
C       INOD(INODDIM,8),SS(KS),IMAX(NUMELX),IMAXF(NUMELX)
C
C     where
C       ITNOD1=(NUMELY+1)*(NUMELX+1)*2+(NUMELX+1)*2
C       MBAND=(NUMELY+3)*2+4
C       KS=ITNOD1*MBAND
C       INODDIM=(NUMELY+1)*NUMELX
C
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON NUMELX,NUMELY,SKIP(5),ITNOD1,MBAND
C
C     the following values should be changed to fit current computer memory
C
C      ITNOD1C=9664
C      MBANDC=70
C      NUMELXC=150
C      NUMELYC=30
C      KSC=676480
C      INODC=4650
C
      ITNOD1C=8844
      MBANDC=50
```

```
      NUMELXC=200
      NUMELYC=20
      KSC=442200
      INODC=4200
C
      ICHMEMO=0
      IF(ITNOD1.GT.ITNOD1C) THEN
        WRITE(*,*) 'memory fault: check dimensions of following arrays'
        WRITE(*,*) 'ALOAD, DISP, TEMLD, SK'
        ICHMEMO=1
      ENDIF
      IF(MBAND.GT.MBANDC) THEN
        WRITE(*,*) 'memory fault: check dimension of SK array'
        ICHMEMO=1
      ENDIF
      IF(NUMELX.GT.NUMELXC) THEN
        WRITE(*,*) 'memory fault: check dimensions of following arrays'
        WRITE(*,*) 'BNSL, IMAX, IMAXF'
        ICHMEMO=1
      ENDIF
      IF(NUMELY.GT.NUMELYC) THEN
        WRITE(*,*) 'memory fault: check dimensions of STRINI and TLOAD'
        ICHMEMO=1
      ENDIF
      IF(KS.GT.KSC) THEN
        WRITE(*,*) 'memory fault: check dimension of SS array'
        ICHMEMO=1
      ENDIF
      IF(INODDIM.GT.INODC) THEN
        WRITE(*,*) 'memory fault: check dimension of INOD array'
        ICHMEMO=1
      ENDIF
C
      RETURN
      END
C***************************************************************
       SUBROUTINE CONCPRO(IAGG,CSTR28,TSTR28,ECONC28,ZSH256,
     +                ECONDAY,CSTR,TSTR,ECONC,ZSH)
C
C     to calculate concrete properties at a consideration day
C
       IMPLICIT REAL*8 (A-H, O-Z)
C
       DIMENSION AC(10),BC(10),CC(10),AN28C(10)
       DIMENSION AT(10),BT(10),CT(10),AN28T(10)
       DIMENSION AE(10),BE(10),CE(10),AN28E(10)
       DIMENSION AD(10),BD(10),CD(10),AN256D(10)
C
C      IAGG
C     Granite = 1,                 Dolomite = 2
C      Vega = 3,                          Bridgeport/Tin Top = 4
C      Western Tascosa = 5,        Ferris = 6
C      Limestone = 7,                     Siliceous River Gravel = 8
C
C      Define coefficients
C
C      COMPRESSIVE STRENGTH
C      Granite
       AC(1)=2570.8
       BC(1)=0.096
       CC(1)=0.623
       AN28C(1)=0.5176
C      Dolomite
```

```
      AC(2)=2236.7
      BC(2)=0.231
      CC(2)=0.562
      AN28C(2)=0.5009
C     Vega
      AC(3)=1995.3
      BC(3)=0.367
      CC(3)=0.367
      AN28C(3)=0.4978
C     Bridgeport/Tin Top
      AC(4)=2038.2
      BC(4)=0.582
      CC(4)=0.220
      AN28C(4)=0.4980
C     Western Tascosa
      AC(5)=2068.5
      BC(5)=0.214
      CC(5)=0.647
      AN28C(5)=0.4998
C     Ferris
      AC(6)=2000.1
      BC(6)=0.206
      CC(6)=0.801
      AN28C(6)=0.5014
C     Limestone
      AC(7)=2550.57
      BC(7)=0.115
      CC(7)=0.490
      AN28C(7)=0.5102
C     Siliceous River Gravel
      AC(8)=2445.25
      BC(8)=0.182
      CC(8)=0.473
      AN28C(8)=0.5020
C
C     TENSILE STRENGTH
C     Granite
      AT(1)=266.46
      BT(1)=0.15
      CT(1)=1.05
      AN28T(1)=0.504
C     Dolomite
      AT(2)=247.06
      BT(2)=0.261
      CT(2)=1.094
      AN28T(2)=0.500
C     Vega
      AT(3)=221.08
      BT(3)=0.302
      CT(3)=0.3014
      AN28T(3)=0.500
C     Bridgeport/Tin Top
      AT(4)=221.85
      BT(4)=0.332
      CT(4)=0.723
      AN28T(4)=0.500
C     Western Tascosa
      AT(5)=216.01
      BT(5)=0.198
      CT(5)=2.505
      AN28T(5)=0.501
C     Ferris
      AT(6)=241.94
```

```
      BT(6)=0.137
      CT(6)=2.479
      AN28T(6)=0.505
C     Limestone
      AT(7)=217.83
      BT(7)=0.177
      CT(7)=1.068
      AN28T(7)=0.502
C     Siliceous River Gravel
      AT(8)=231.07
      BT(8)=0.267
      CT(8)=0.468
      AN28T(8)=0.500
C
C     ELASTIC MODULUS
C     Granite
      AE(1)=1678000.
      BE(1)=0.78
      CE(1)=1.65E14
      AN28E(1)=0.500
C     Dolomite
      AE(2)=2324000.
      BE(2)=0.485
      CE(2)=3.537
      AN28E(2)=0.500
C     Vega
      AE(3)=1882000.
      BE(3)=0.301
      CE(3)=1.574
      AN28E(3)=0.500
C     Bridgeport/Tin Top
      AE(4)=1992000
      BE(4)=0.688
      CE(4)=2.00
      AN28E(4)=0.500
C     Western Tascosa
      AE(5)=1803000.
      BE(5)=0.405
      CE(5)=97.056
      AN28E(5)=0.500
C     Ferris
      AE(6)=1979000.
      BE(6)=0.738
      CE(6)=2.668E12
      AN28E(6)=0.500
C     Limestone
      AE(7)=1802000.
      BE(7)=0.535
      CE(7)=110.46
      AN28E(7)=0.500
C     Siliceous River Gravel
      AE(8)=2282000.
      BE(8)=0.574
      CE(8)=61755.1
      AN28E(8)=0.500
C
C     DRYING SHRINKAGE
C     Granite
      AD(1)=0.00032123
      BD(1)=0.0851
      CD(1)=0.001
      AN256D(1)=0.8112
C     Dolomite
```

```
      AD(2)=0.00025206
      BD(2)=0.04062
      CD(2)=0.00155
      AN256D(2)=0.7569
C     Vega
      AD(3)=0.00023519
      BD(3)=0.03948
      CD(3)=0.01255
      AN256D(3)=0.5146
C     Bridgeport/Tin Top
      AD(4)=0.00034362
      BD(4)=0.0328
      CD(4)=0.00069
      AN256D(4)=0.8582
C     Western Tascosa
      AD(5)=0.000358456
      BD(5)=0.03109
      CD(5)=0.000715
      AN256D(5)=0.8600
C     Ferris
      AD(6)=0.00032723
      BD(6)=0.0745
      CD(6)=0.00119
      AN256D(6)=0.7828
C     Limestone
      AD(7)=0.0002291
      BD(7)=0.0398
      CD(7)=0.00754
      AN256D(7)=0.5403
C     Siliceous River Gravel
      AD(8)=0.00019839
      BD(8)=0.0619
      CD(8)=0.005
      AN256D(8)=0.5636
C
C     Define concrete properties at a consideration day
C
      ECOND1=ECONDAY
C
C     DRYING SHRINKAGE
C
      IF(ECONDAY.GE.256.) ECOND1=256.
      CONST4=2.-DEXP(-1.*BD(IAGG)*ECOND1)-DEXP(-1.*CD(IAGG)*ECOND1)
      IND4=ZSH256
      IF(IND4.EQ.0) THEN
         ZSH=AD(IAGG)*CONST4
      ELSE
         ZSH=ZSH256*AN256D(IAGG)*CONST4
      ENDIF
C
C     COMPRESSIVE STRENGTH
C
      IF(ECONDAY.GE.28.) ECOND1=28.
      CONST1=2.-DEXP(-1.*BC(IAGG)*ECOND1)-DEXP(-1.*CC(IAGG)*ECOND1)
      IND1=CSTR28
      IF(IND1.EQ.0) THEN
         CSTR=AC(IAGG)*CONST1
      ELSE
         CSTR=CSTR28*AN28C(IAGG)*CONST1
      ENDIF
C
C     TENSILE STRENGTH
C
```

```
      CONST2=2.-DEXP(-1.*BT(IAGG)*ECOND1)-DEXP(-1.*CT(IAGG)*ECOND1)
      IND2=TSTR28
      IF(IND2.EQ.0) THEN
         TSTR=AT(IAGG)*CONST2
      ELSE
         TSTR=TSTR28*AN28T(IAGG)*CONST2
      ENDIF
C
C     ELASTIC MODULUS
C
      CONST3=2.-DEXP(-1.*BE(IAGG)*ECOND1)-DEXP(-1.*CE(IAGG)*ECOND1)
      IND3=ECONC28
      IF(IND3.EQ.0) THEN
         ECONC=AE(IAGG)*CONST3
      ELSE
         ECONC=ECONC28*AN28E(IAGG)*CONST3
      ENDIF
C
C     write(*,*) CSTR
C     write(*,*) TSTR
C     write(*,*) ECONC
C     write(*,*) ZSH
C
C
      RETURN
      END
C****************************************************************
      SUBROUTINE CREEP(TIME,ECONC,E1,PHIMAX,PERCEN,DAYPC)
C
C     to find the reduced modulus using Effective Modulus Method
C
      IMPLICIT REAL*8 (A-H, O-Z)
C
      IF(PERCEN.EQ.100.) THEN
         WRITE(*,*) 'PUT LESS THAN 100 FOR PERCEN'
         WRITE(*,*) 'CREEP ANALYSIS WAS NOT PERFORMED'
         GO TO 806
      ENDIF
      CONST=(1./(1.-PERCEN/100.))**(TIME/(DAYPC*24.))
      PHI=PHIMAX*(1.-1./CONST)
      E1=ECONC/(1.+PHI)
C
  806 RETURN
      END
C****************************************************************
      SUBROUTINE LINREC4(AK,D,IELTYPE)
C
C     to find the element stiffness matrix using 4-node isoparametric
C     element (or 4-node bi-linear rectangular element)
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION AK(8,8),XI(4),ETA(4),XINT(2),YINT(2),WX(2),WY(2)
      DIMENSION D(3,3),BMAT(3,8),BT(8,3),F(4),FPXI(4),FPET(4),AUX(8,8)
      DIMENSION AJAC(2)
      COMMON NUMELX,NUMELY,A,B,E1,POIS1,THICK
      DATA XI/-1.,-1.,1.,1./,ETA/1.,-1.,1.,-1./
      DATA XINT/-0.57735027,0.57735027/,YINT/-0.57735027,0.57735027/
      DATA WX/1.,1./,WY/1.,1./
C
C     initialize matrices
C
      DO 1 I=1,3
      DO 3 J=1,3
```

```
    3 D(I,J)=0.
    1 CONTINUE
      DO 4 I=1,8
      DO 5 J=1,8
    5 AK(I,J)=0.
    4 CONTINUE
C
       IF(IELTYPE.EQ.1) THEN
C        define D matrix (plane stress)
         COND=E1/(1.-POIS1*POIS1)
         D(1,1)=COND
         D(1,2)=D(1,1)*POIS1
         D(2,1)=D(1,2)
         D(2,2)=D(1,1)
         D(3,3)=E1/(2.+2.*POIS1)
       ELSE
C        define D matrix (plane strain)
         COND=E1/((1.+POIS1)*(1.-2.*POIS1))
         D(1,1)=COND*(1.-POIS1)
         D(1,2)=COND*POIS1
         D(2,1)=D(1,2)
         D(2,2)=D(1,1)
         D(3,3)=COND*(1.-2.*POIS1)/2.
       ENDIF
C
C     number of integration points in x and y directions
C
      NPX=2
      NPY=2
C
      DO 100 IX=1,NPX
      DO 101 IY=1,NPY
C
      FACT=A*B*WX(IX)*WY(IY)*THICK/4.
      X=XINT(IX)
      Y=YINT(IY)
C
C     find derivatives of shape functions at integration points
C
      DO 10 I=1,4
         F(I)=(1.+X*XI(I))*(1.+Y*ETA(I))/4.
         FPXI(I)=XI(I)*(1.+Y*ETA(I))/4.
         FPET(I)=ETA(I)*(1.+X*XI(I))/4.
   10 CONTINUE
C
C     define B matrix and transpose of B
C
      DO I=1,3
      DO J=1,8
      BMAT(I,J)=0.
      ENDDO
      ENDDO
      AJAC(1)=A/2.
      AJAC(2)=B/2.
      DO 20 I=1,4
         I1=2*I-1
         I2=I1+1
         BMAT(1,I1)=FPXI(I)/AJAC(1)
         BMAT(2,I2)=FPET(I)/AJAC(2)
         BMAT(3,I1)=BMAT(2,I2)
         BMAT(3,I2)=BMAT(1,I1)
   20 CONTINUE
      DO 21 I=1,3
```

```
      DO 22 J=1,8
         BT(J,I)=BMAT(I,J)
   22 CONTINUE
   21 CONTINUE
C
C     define element stiffness matrix
C
      CALL MATPRO(D,BMAT,AUX,3,3,8,3,3,8)
C
      DO 23 I=1,3
      DO 24 J=1,8
         BMAT(I,J)=AUX(I,J)
   24 CONTINUE
   23 CONTINUE
C
      CALL MATPRO(BT,BMAT,AUX,8,3,8,8,3,8)
C
      DO 25 I=1,8
      DO 26 J=1,8
         AK(I,J)=AK(I,J)+FACT*AUX(I,J)
   26 CONTINUE
   25 CONTINUE
C
  101 CONTINUE
  100 CONTINUE
      RETURN
      END
C****************************************************************
      SUBROUTINE MATPRO(AA,BB,CC,NA,NB,NC,M,N,L)
C
C     to multiply matrices
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AA(NA,N),BB(NB,L),CC(NC,L)
      DO 56 I=1,M
      DO 57 J=1,L
      SUM=0.
      DO 58 K=1,N
      SUM=SUM+AA(I,K)*BB(K,J)
   58 CONTINUE
      CC(I,J)=SUM
   57 CONTINUE
   56 CONTINUE
      RETURN
      END
C****************************************************************
      SUBROUTINE FRAME(A,ESTL,DIASTL,AKSTL,AKS,AKSB)
C
C     to define element stiffness matrix for frame (truss and beam) elements
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION AKS(2,2),AKSB(4,4)
C
C     element stiffness matrix for truss element
C
      AKS(1,1)=AKSTL
      AKS(1,2)=-AKSTL
      AKS(2,2)=AKS(1,1)
C
      AMINER=3.141592654*(DIASTL**4)/64.
      CONBE=ESTL*AMINER/(A**3)
C
C     element stiffness matrix for beam element
```

```
C
      AKSB(1,1)=12.*CONBE
      AKSB(1,2)=6.*A*CONBE
      AKSB(1,3)=-12.*CONBE
      AKSB(1,4)=6.*A*CONBE
      AKSB(2,2)=4.*A*A*CONBE
      AKSB(2,3)=-6.*A*CONBE
      AKSB(2,4)=2.*A*A*CONBE
      AKSB(3,3)=12.*CONBE
      AKSB(3,4)=-6.*A*CONBE
      AKSB(4,4)=4.*A*A*CONBE
C
      RETURN
      END
C*****************************************************************
      SUBROUTINE ASSEM(AK,SK,INOD,LSTLOC,AKS,TLOAD,TEMLD,AKSB)
C
C     to assemble element stiffness matrices and temperature load
C     vectors into global systems (using half-bandwidth for stiffness)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION INOD(4200,8),SK(8844,50),AK(8,8),AKS(2,2),AKSB(4,4)
      DIMENSION TLOAD(20,8),TEMLD(8844)
      COMMON NUMELX,NUMELY,SKIP(5),ITNOD1,MBAND
C
C     initialize banded stiffness matrix and temperature load vector
C
      DO 30 I=1,ITNOD1
      DO 31 J=1,MBAND
         SK(I,J)=0.
   31 CONTINUE
         TEMLD(I)=0.
   30 CONTINUE
C
C     element numbering and connectivity with degrees of freedom
C
      DO 32 KX=1,NUMELX
      DO 33 KY=1,NUMELY
         IF(KY.LT.LSTLOC) THEN
           IEL=(KX-1)*(NUMELY+1)+KY
           INOD(IEL,1)=IEL*2+2*KX-3
           INOD(IEL,2)=INOD(IEL,1)+1
           INOD(IEL,3)=INOD(IEL,1)+2
           INOD(IEL,4)=INOD(IEL,1)+3
           INOD(IEL,5)=INOD(IEL,1)+(NUMELY+2)*2
           INOD(IEL,6)=INOD(IEL,5)+1
           INOD(IEL,7)=INOD(IEL,5)+2
           INOD(IEL,8)=INOD(IEL,5)+3
         ELSE
           IEL=(KX-1)*NUMELY+KX+KY
           IF(KY.EQ.LSTLOC) THEN
             INOD(IEL,1)=IEL*2+2*KX-5
             INOD(IEL,2)=INOD(IEL,1)+1
             INOD(IEL,3)=INOD(IEL,1)+4
             INOD(IEL,4)=INOD(IEL,1)+5
             INOD(IEL,5)=INOD(IEL,1)+(NUMELY+2)*2
             INOD(IEL,6)=INOD(IEL,5)+1
             INOD(IEL,7)=INOD(IEL,5)+4
             INOD(IEL,8)=INOD(IEL,5)+5
           ELSE
             INOD(IEL,1)=IEL*2+2*KX-3
             INOD(IEL,2)=INOD(IEL,1)+1
             INOD(IEL,3)=INOD(IEL,1)+2
```

```
            INOD(IEL,4)=INOD(IEL,1)+3
            INOD(IEL,5)=INOD(IEL,1)+(NUMELY+2)*2
            INOD(IEL,6)=INOD(IEL,5)+1
            INOD(IEL,7)=INOD(IEL,5)+2
            INOD(IEL,8)=INOD(IEL,5)+3
          ENDIF
        ENDIF
C
C     assemble element stiffness matrix and load vector into global system
C
        DO 35 I=1,8
        DO 36 J=I,8
           SK(INOD(IEL,I),INOD(IEL,J)-INOD(IEL,I)+1)=
     +           SK(INOD(IEL,I),INOD(IEL,J)-INOD(IEL,I)+1)+AK(I,J)
   36   CONTINUE
           TEMLD(INOD(IEL,I))=TEMLD(INOD(IEL,I))+TLOAD(KY,I)
   35   CONTINUE
C
   33 CONTINUE
   32 CONTINUE
C
C     element and d.o.f. numbering for frame elements
C
      DO 37 KX=1,NUMELX
         IEL=LSTLOC+(KX-1)*(NUMELY+1)
         INOD(IEL,1)=IEL*2+2*KX-2
         INOD(IEL,2)=INOD(IEL,1)+1
         INOD(IEL,3)=INOD(IEL,1)+(NUMELY+2)*2
         INOD(IEL,4)=INOD(IEL,3)+1
         INOD(IEL,5)=IEL*2+2*KX
         INOD(IEL,6)=INOD(IEL,5)+(NUMELY+2)*2
C
C     assemble element stiffness matrix for beam elements
C
      DO 41 I=1,4
      DO 42 J=I,4
            SK(INOD(IEL,I),INOD(IEL,J)-INOD(IEL,I)+1)=
     +          SK(INOD(IEL,I),INOD(IEL,J)-INOD(IEL,I)+1)+AKSB(I,J)
   42 CONTINUE
   41 CONTINUE
C
C     assemble element stiffness matrix for truss elements
C
      DO 38 I=5,6
      DO 39 J=I,6
            SK(INOD(IEL,I),INOD(IEL,J)-INOD(IEL,I)+1)=
     +          SK(INOD(IEL,I),INOD(IEL,J)-INOD(IEL,I)+1)+AKS(I-4,J-4)
   39 CONTINUE
   38 CONTINUE
   37 CONTINUE
C
      RETURN
      END
C************************************************************
      SUBROUTINE TEMPRT(TEMPR1,TEMF,ALPC,TLOAD,STRINI,D,IELTYPE)
C
C     to calculate initial stresses and applied loads due to
C     temperature variations.
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION D(3,3),XI(4),ETA(4),XINT(2),YINT(2),WX(2),WY(2)
      DIMENSION STRINI(20,4),AJAC(2),BMAT(3,8),BT(8,3),TEMPR1(21)
      DIMENSION TLOAD(20,8),F(4),FPXI(4),FPET(4),TEML(4)
```

```
C*****************************************************************
      SUBROUTINE BOUND(SK,LSTLOC)
C
C     to apply boundary conditions (using big spring method)
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION SK(8844,50)
      COMMON NUMELX,NUMELY
C
      BIGK=1.0E15
C
C     boundary condition for left side (no d.o.f. in x-direction)
C
      LSTLOC1=LSTLOC+1
      DO I=1,NUMELY+2
         IF(I.LE.LSTLOC) THEN
           J=I*2-1
         ELSE
           IF(I.EQ.LSTLOC1) THEN
             J=I*2
           ELSE
             J=I*2-1
           ENDIF
         ENDIF
         SK(J,1)=SK(J,1)+BIGK
      ENDDO
C
C     boundary condition for left side of steel (no rotational d.o.f.)
C
      J=LSTLOC1*2-1
      SK(J,1)=SK(J,1)+BIGK
C
C     bound. cond. for right side of steel(no dof in x- and rot. direc.)
C
      NSTLL=((NUMELY+2)*2)*NUMELX+LSTLOC*2+1
      SK(NSTLL,1)=SK(NSTLL,1)+BIGK
      SK(NSTLL+1,1)=SK(NSTLL+1,1)+BIGK
C
C     boundary condition for bottom
C
C      NST1=(NUMELY+2)*2
C      NST11=NST1+NST1*NUMELX
C      SK(NST1,1)=SK(NST1,1)+BOTK/2.
C      SK(NST11,1)=SK(NST11,1)+BOTK/2.
C      DO I=2,NUMELX
C         J1=NST1+NST1*(I-1)
C         SK(J1,1)=SK(J1,1)+BOTK
C      ENDDO
C
      RETURN
      END
C*****************************************************************
      SUBROUTINE BNSLIP(SK,AKBOND,AKBOND2,LSTLOC,KIM,ALOAD,IMAX,
     +                  BNFORCE,BNSLMAXS,BNSL,ULSLIP)
C
C     define bond-slip relation between concrete and steel bar
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION SK(8844,50),ALOAD(8844),IMAX(200),BNSL(200)
      COMMON NUMELX,NUMELY,SKIP(5),ITNOD1,MBAND
C
      NINC=(NUMELY+2)*2
      NST=LSTLOC*2-1+NINC
```

```
      DO I=1,NUMELX
         J1=NST+(I-1)*NINC
         J2=J1+3
         IF(I.EQ.NUMELX) THEN
           SK(J1,1)=SK(J1,1)+AKBOND/2.
           SK(J1,4)=SK(J1,4)-AKBOND/2.
           SK(J2,1)=SK(J2,1)+AKBOND/2.
         ELSE
           SK(J1,1)=SK(J1,1)+AKBOND
           SK(J1,4)=SK(J1,4)-AKBOND
           SK(J2,1)=SK(J2,1)+AKBOND
         ENDIF
      ENDDO
C
      IF(KIM.EQ.1) GO TO 889
C
      DO IKIM=1,KIM-1
         J1=NST+(IMAX(IKIM)-1)*NINC
         J2=J1+3
         IF(DABS(BNSL(IMAX(IKIM))).LT.ULSLIP) THEN
           IF(IMAX(IKIM).EQ.NUMELX) THEN
             AKB=AKBOND2/2.-AKBOND/2.
             BNFOR=BNFORCE/2.
           ELSE
             AKB=AKBOND2-AKBOND
             BNFOR=BNFORCE
           ENDIF
         ELSE
           BNFOR=0.
           IF(IMAX(IKIM).EQ.NUMELX) THEN
             AKB=-AKBOND/2.
           ELSE
             AKB=-AKBOND
           ENDIF
         ENDIF
         SK(J1,1)=SK(J1,1)+AKB
         SK(J1,4)=SK(J1,4)-AKB
         SK(J2,1)=SK(J2,1)+AKB
         IF(BNSLMAXS.LT.0.) THEN
           ALOAD(J1)=ALOAD(J1)+BNFOR
           ALOAD(J2)=ALOAD(J2)-BNFOR
         ELSE
           ALOAD(J1)=ALOAD(J1)-BNFOR
           ALOAD(J2)=ALOAD(J2)+BNFOR
         ENDIF
      ENDDO
C
  889 RETURN
      END
C***************************************************************
      SUBROUTINE FRSLIP(SK,AKFR,MIN,IMAXC)
C
C     define bond-slip relation between concrete and base
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION SK(8844,50),IMAXC(200)
      COMMON NUMELX,NUMELY,SKIP(5),ITNOD1,MBAND
C
      NINC=(NUMELY+2)*2
      NSTFR=(NUMELY+2)*2-1+NINC
      DO I=1,NUMELX
         J1=NSTFR+(I-1)*NINC
         IF(I.EQ.NUMELX) THEN
```

```
            SK(J1,1)=SK(J1,1)+AKFR/2.
          ELSE
            SK(J1,1)=SK(J1,1)+AKFR
          ENDIF
       ENDDO
C
        IF(MIN.EQ.1) GO TO 689
C
        IMIN=MIN-1
        DO J=IMAXC(IMIN),NUMELX
           J1=NINC*(J+1)-1
           IF(J.EQ.NUMELX) THEN
              AKF=AKFR/2.
           ELSE
              AKF=AKFR
           ENDIF
           SK(J1,1)=SK(J1,1)-AKF
        ENDDO
C
  689 RETURN
      END
C************************************************************
        SUBROUTINE CURL(SK,BOTK,MIN,IMAXC)
C
C       to define the elastic foundation for underlying layers
C
       IMPLICIT REAL*8 (A-H, O-Z)
        DIMENSION SK(8844,50),IMAXC(200)
        COMMON NUMELX,NUMELY
C
        NINC=(NUMELY+2)*2
        NINCE=NINC*(NUMELX+1)
        SK(NINC,1)=SK(NINC,1)+BOTK/2.
        SK(NINCE,1)=SK(NINCE,1)+BOTK/2.
        DO I=2,NUMELX
           J1=NINC*I
           SK(J1,1)=SK(J1,1)+BOTK
        ENDDO
C
        IF(MIN.EQ.1) GO TO 579
C
        IMIN=MIN-1
        DO J=IMAXC(IMIN),NUMELX
           J1=NINC*(J+1)
           IF(J.EQ.NUMELX) THEN
              AKC=BOTK/2.
           ELSE
              AKC=BOTK
           ENDIF
           SK(J1,1)=SK(J1,1)-AKC
        ENDDO
C
  579  RETURN
        END
C************************************************************
        SUBROUTINE DWEIGHT(ALOAD,ELWEI,ELSTWEI,LSTLOC)
C
C       to add the dead weight (body force) of concrete and steel
C
       IMPLICIT REAL*8 (A-H, O-Z)
       DIMENSION ALOAD(8844)
       COMMON NUMELX,NUMELY
C
```

```
      NXL=NUMELX+1
      NYL=NUMELY+1
C
      DO I=1,NYL
         IF(I.LT.LSTLOC) THEN
            IF(I.EQ.1) THEN
               DO J=1,NXL
                  K=2*I+(NUMELY+2)*2*(J-1)
                  IF((J.EQ.1).OR.(J.EQ.NXL)) THEN
                     ALOAD(K)=ALOAD(K)-ELWEI/4.
                 ELSE
                     ALOAD(K)=ALOAD(K)-ELWEI/2.
                  ENDIF
               ENDDO
            ELSE
               DO J=1,NXL
                  K=2*I+(NUMELY+2)*2*(J-1)
                  IF((J.EQ.1).OR.(J.EQ.NXL)) THEN
                      ALOAD(K)=ALOAD(K)-ELWEI/2.
                  ELSE
                      ALOAD(K)=ALOAD(K)-ELWEI
                  ENDIF
               ENDDO
            ENDIF
         ELSE
            IF(I.EQ.LSTLOC) THEN
               DO J=1,NXL
                  K=2*I+(NUMELY+2)*2*(J-1)
                  IF((J.EQ.1).OR.(J.EQ.NXL)) THEN
                      IF(I.EQ.1) THEN
                         ALOAD(K)=ALOAD(K)-ELWEI/4.-ELSTWEI/2.
                      ELSE
                         ALOAD(K)=ALOAD(K)-ELWEI/2.-ELSTWEI/2.
                      ENDIF
                  ELSE
                      IF(I.EQ.1) THEN
                         ALOAD(K)=ALOAD(K)-ELWEI/2.-ELSTWEI
                      ELSE
                         ALOAD(K)=ALOAD(K)-ELWEI-ELSTWEI
                      ENDIF
                  ENDIF
               ENDDO
            ELSE
               IF(I.EQ.NYL) THEN
                  DO J=1,NXL
                     K=2*I+(NUMELY+2)*2*(J-1)+2
                     IF((J.EQ.1).OR.(J.EQ.NXL)) THEN
                        ALOAD(K)=ALOAD(K)-ELWEI/4.
                     ELSE
                        ALOAD(K)=ALOAD(K)-ELWEI/2.
                     ENDIF
                  ENDDO
               ELSE
                  DO J=1,NXL
                     K=2*I+(NUMELY+2)*2*(J-1)+2
                     IF((J.EQ.1).OR.(J.EQ.NXL)) THEN
                        ALOAD(K)=ALOAD(K)-ELWEI/2.
                     ELSE
                        ALOAD(K)=ALOAD(K)-ELWEI
                     ENDIF
                  ENDDO
               ENDIF
            ENDIF
```

```
         ENDIF
      ENDDO
C
       RETURN
       END
C*****************************************************************
      SUBROUTINE SOLVE(NEQ,MBAND,R,SS,KS)
C
C     to solve equilibrium equations using 1-D compacted array of
C     banded stiffness matrix and Gaussian eliminations
C
C     NEQ   = number of equations
C     MBAND = half-bandwidth
C     R     = constant vector (load vector)
C     SS    = 1-D array of coefficient matrix (banded stiffness matrix)
C     KS    = NEQ*MBAND
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION R(NEQ),SS(KS)
C
C     forward reduction of 1-D array of banded coefficient matrix
C
      DO 790 N=1,NEQ
      DO 780 L=2,MBAND
      KK1=(L-1)*NEQ+N
      IF(SS(KK1).EQ.0.) GO TO 780
      I=N+L-1
      C=SS(KK1)/SS(N)
      J=0
      DO 750 K=L,MBAND
      J=J+1
  750 SS((J-1)*NEQ+I)=SS((J-1)*NEQ+I)-C*SS((K-1)*NEQ+N)
      SS(KK1)=C
  780 CONTINUE
  790 CONTINUE
C
C     forward reduction of the vector of constants
C
      DO 830 N=1,NEQ
      DO 820 L=2,MBAND
      IF(SS((L-1)*NEQ+N).EQ.0.) GO TO 820
      I=N+L-1
      R(I)=R(I)-SS((L-1)*NEQ+N)*R(N)
  820 CONTINUE
  830 R(N)=R(N)/SS(N)
C
C     solve for unknowns by back-substitution
C
      DO 860 M=2,NEQ
      N=NEQ+1-M
      DO 850 L=2,MBAND
      IF(SS((L-1)*NEQ+N).EQ.0.) GO TO 850
      K=N+L-1
      R(N)=R(N)-SS((L-1)*NEQ+N)*R(K)
  850 CONTINUE
  860 CONTINUE
      RETURN
      END
C*****************************************************************
      SUBROUTINE CHSLIP(DISP,IMAX,KIM,BNSL,BNSLMAX,BNSLMAXS,LSTLOC,
     +                  YLSLIP,IULSL,ULSLIP)
C
C     to check if the slip exceeds the limit value
```

```
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION BNSL(200),DISP(8844),IMAX(200)
      COMMON NUMELX,NUMELY
C
      DO I=1,NUMELX
         BNSL(I)=0.
      ENDDO
      NINC=(NUMELY+2)*2
      NST=LSTLOC*2-1+NINC
      DO I=1,NUMELX
         J1=NST+(I-1)*NINC
         J2=J1+3
         BNSL(I)=DISP(J1)-DISP(J2)
      ENDDO
C
      BNSLMAX=0.
      DO 881 I=1,NUMELX
         IF(KIM.EQ.1) GO TO 883
         DO K=1,KIM-1
            IF(I.EQ.IMAX(K)) GO TO 888
         ENDDO
  883    ABNSL=DABS(BNSL(I))
         IF(ABNSL.GE.BNSLMAX) THEN
           BNSLMAX=ABNSL
           IMAX(KIM)=I
           BNSLMAXS=BNSL(I)
         ENDIF
  888 CONTINUE
  881 CONTINUE
C
       IULSL=0
       DO I=1,KIM
          IF(DABS(BNSL(IMAX(I))).GT.ULSLIP) THEN
             IULSL=IULSL+1
          ENDIF
       ENDDO
C      write(*,*) 'Concrete-Steel Bond-Slip =',KIM,IMAX(KIM),
C     1             BNSL(IMAX(KIM))
C
      RETURN
      END
C***************************************************************
       SUBROUTINE CHCURL(DISP,MIN,IMAXC,INDCURL)
C
C      to check if any vertical spring is in tension
C
      IMPLICIT REAL*8 (A-H, O-Z)
       DIMENSION DISP(8844),IMAXC(200)
       COMMON NUMELX,NUMELY
C
       NINC=(NUMELY+2)*2
       DO I=1,NUMELX
          J1=NINC*(I+1)
          IF(DISP(J1).GT.0) THEN
             IMAXC(MIN)=I
             GO TO 578
          ENDIF
C      write(*,*) I,DISP(J1)
       ENDDO
       IMAXC(MIN)=NUMELX+1
  578  CONTINUE
C
```

```
      IF(MIN.EQ.1) THEN
         IF(IMAXC(MIN).GT.NUMELX) THEN
            INDCURL=0
         ELSE
            INDCURL=1
         ENDIF
      ELSE
         MIN1=MIN-1
         IF((IMAXC(MIN1)-IMAXC(MIN)).GT.0) THEN
            INDCURL=1
         ELSE
            INDCURL=0
         ENDIF
      ENDIF
C
C     write(*,*) 'Curling index (MIN) = ',MIN,IMAXC(MIN)
      MIN=MIN+1
C
      RETURN
      END
C*************************************************************
      SUBROUTINE COMPST(STRINI,INOD,LSTLOC,ESTL,ALPS,JOO,NA,INDMA,
     +                DELT,DISP,D,CHSTRES,INDST,CSTR1,SSTR1,WHDAY,
     +                      TOTD,WHRAD,WHLOAD,ECONC,BOTKV,ECONDAY)
C
C     to compare stresses at morning and afternoon
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION STRINI(20,4),INOD(4200,8),DISP(8844),D(3,3)
      COMMON NUMELX,NUMELY,A,B,E1,POIS1,THICK,ITNOD1,MBAND
C
      IF(JOO.EQ.1) THEN
         STMAX1=0.
        DO 401 I=1,NUMELY+1
           IF(I.EQ.LSTLOC) GO TO 401
           CALL STRESS(I,STRINI,INOD,LSTLOC,ESTL,ALPS,DELT,DISP,
     +                 D,CHSTRES,INDST,CSTR1,SSTR1)
            IF(ECONDAY.GE.WHDAY) THEN
              CALL WHEEL(TOTD,WHRAD,WHLOAD,ECONC,BOTKV,WHSTR)
              IF(I.LT.LSTLOC) THEN
                      CHWHSTR=WHSTR*(FLOAT(2*I-1)/FLOAT(NUMELY)-1.)
              ELSE
                 CHWHSTR=WHSTR*(FLOAT(2*I-3)/FLOAT(NUMELY)-1.)
                ENDIF
                CHSTRES=CHSTRES+CHWHSTR
            ENDIF
            IF(CHSTRES.GT.STMAX1) STMAX1=CHSTRES
  401   CONTINUE
         JOO=2
         INDMA=1
         GO TO 404
      ELSE
         STMAX2=0.
        DO 403 I=1,NUMELY+1
           IF(I.EQ.LSTLOC) GO TO 403
           CALL STRESS(I,STRINI,INOD,LSTLOC,ESTL,ALPS,DELT,DISP,
     +                 D,CHSTRES,INDST,CSTR1,SSTR1)
            IF(ECONDAY.GE.WHDAY) THEN
              CALL WHEEL(TOTD,WHRAD,WHLOAD,ECONC,BOTKV,WHSTR)
              IF(I.LT.LSTLOC) THEN
                      CHWHSTR=WHSTR*(FLOAT(2*I-1)/FLOAT(NUMELY)-1.)
              ELSE
                 CHWHSTR=WHSTR*(FLOAT(2*I-3)/FLOAT(NUMELY)-1.)
```

```
                  ENDIF
                  CHSTRES=CHSTRES+CHWHSTR
              ENDIF
              IF(CHSTRES.GT.STMAX2) STMAX2=CHSTRES
  403       CONTINUE
         ENDIF
C
         IF(STMAX1.GE.STMAX2) THEN
            JOO=1
         ELSE
            JOO=2
         ENDIF
C
         NA=2
         INDMA=2
C
  404 CONTINUE
C
         RETURN
         END
C*****************************************************************
      SUBROUTINE STRESS(IELINP,STRINI,INOD,LSTLOC,ESTL,ALPS,
     +                  DELT,DISP,D,CHSTRES,INDST,CSTR1,SSTR1)
C
C     to find stresses for concrete and steel bar
C
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION STR(4,3),AJAC(2),STRINI(20,4),INOD(4200,8)
      DIMENSION XI(4),ETA(4),XINT(2),YINT(2),DISP(8844)
      DIMENSION D(3,3),BMAT(3,8),FPXI(4),FPET(4),AUX(8,8),UL(8)
      COMMON NUMELX,NUMELY,A,B,E1,POIS1,THICK,ITNOD1,MBAND
      DATA XI/-1.,-1.,1.,1./,ETA/1.,-1.,1.,-1./
      DATA XINT/-0.57735027,0.57735027/,YINT/-0.57735027,0.57735027/
C
C     number of integration points in x and y directions
C
      NPX=2
      NPY=2
C
      ITNUMEL=NUMELX*(NUMELY+1)
      MINUS=-1*(NUMELY+1)
      DO LOC=ITNUMEL,0,MINUS
         IF(LOC.LT.IELINP) GO TO 77
      ENDDO
   77 ILOC1=IELINP-LOC
      IF(ILOC1.LT.LSTLOC) THEN
        ILOC=ILOC1
      ELSE
        ILOC=ILOC1-1
      ENDIF
C
      DO I=1,8
         J=INOD(IELINP,I)
         UL(I)=DISP(J)
      ENDDO
C
      NUMIP=1
C
      DO 200 IX=1,NPX
      DO 201 IY=1,NPY
C
      X=XINT(IX)
      Y=YINT(IY)
```

```
C
C     find derivatives of shape functions at integration points
C
      DO 20 I=1,4
         FPXI(I)=XI(I)*(1.+Y*ETA(I))/4.
         FPET(I)=ETA(I)*(1.+X*XI(I))/4.
   20 CONTINUE
C
C     define B matrix and transpose of B
C
      DO I=1,3
      DO J=1,8
      BMAT(I,J)=0.
      ENDDO
      ENDDO
      AJAC(1)=A/2.
      AJAC(2)=B/2.
      DO 21 I=1,4
         I1=2*I-1
         I2=I1+1
         BMAT(1,I1)=FPXI(I)/AJAC(1)
         BMAT(2,I2)=FPET(I)/AJAC(2)
         BMAT(3,I1)=BMAT(2,I2)
         BMAT(3,I2)=BMAT(1,I1)
   21 CONTINUE
C
C     calculate element stresses at integration points
C
      CALL MATPRO(D,BMAT,AUX,3,3,8,3,3,8)
      DO 23 I=1,3
         STRTEMP=0.
         DO 24 J=1,8
            STRTEMP=STRTEMP+AUX(I,J)*UL(J)
   24 CONTINUE
         STR(NUMIP,I)=STRTEMP
   23 CONTINUE
      NUMIP=NUMIP+1
C
  201 CONTINUE
  200 CONTINUE
C
      DO I=1,4
      DO J=1,2
         STR(I,J)=STR(I,J)+STRINI(ILOC,I)
      ENDDO
      ENDDO
C
      CHSTRES=(STR(1,1)+STR(2,1)+STR(3,1)+STR(4,1))/4.
C
      IF(INDST.EQ.1) THEN
C        WRITE(6,*) 'CONCRETE STRESS'
        STROUT1=0.
        STROUT2=0.
        STROUT3=0.
        DO I=1,4
         STROUT1=STR(I,1)+STROUT1
         STROUT2=STR(I,2)+STROUT2
         STROUT3=STR(I,3)+STROUT3
        ENDDO
        STROUT1=STROUT1/4.
        STROUT2=STROUT2/4.
        STROUT3=STROUT3/4.
C
```

```
C      to print element numbers and stresses, use below
C         WRITE(6,205) IELINP,STROUT1
C  205    FORMAT(I5,10X,F12.4)
C      to print only the stresses, use below
C         WRITE(6,205) STROUT1
C  205    FORMAT(F12.4)
       CSTR1=STROUT1
C
      ENDIF
      IF(INDST.EQ.2) THEN
C         WRITE(6,*) 'STRESS IN STEEL BAR'
        THERMAL=-ESTL*ALPS*DELT
C         WRITE(6,*) 'Initial thermal stress'
C        WRITE(6,833) THERMAL
C  833   FORMAT(F12.4)
C         WRITE(6,*) 'Maximum stress in steel bar'
C         DO I=1,NUMELX
C            J=LSTLOC+(I-1)*(NUMELY+1)
C            SIGST=ESTL*(DISP(INOD(J,6))-DISP(INOD(J,5)))/A+THERMAL
C            WRITE(6,207) SIGST
C  207       FORMAT(F15.4)
C         ENDDO
         JJ=LSTLOC+(NUMELX-1)*(NUMELY+1)
        SIGST=ESTL*(DISP(INOD(JJ,6))-DISP(INOD(JJ,5)))/A+THERMAL
C         WRITE(6,207) SIGST
C  207    FORMAT(F15.4)
       SSTR1=SIGST
      ENDIF
C
      RETURN
      END
C*************************************************************
       SUBROUTINE WHEEL(TOTD,WHRAD,WHLOAD,ECONC,BOTKV,WHSTR)
C
C      to calculate wheel load stress by Westergaard equation
C
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON NUMELX,NUMELY,A,B,E1,POIS1,THICK,ITNOD1,MBAND
C
       BBIND=1.724*TOTD
       IF(WHRAD.GE.BBIND) THEN
             BBB=WHRAD
       ELSE
             BBB=DSQRT(1.6*WHRAD*WHRAD+TOTD*TOTD)-0.675*TOTD
       ENDIF
C
C      calculate radius of relative stiffness
       ALC1=ECONC*TOTD*TOTD*TOTD/(12.*(1.-POIS1*POIS1)*BOTKV)
       AL1=DSQRT(DSQRT(ALC1))
C
C      calculate wheel load stress
       C1=3.*(1.+POIS1)*WHLOAD/(2.*3.141592*TOTD*TOTD)
       C2=DLOG(AL1/BBB)+0.6159
       WHSTR=C1*C2
C
       RETURN
       END
C*************************************************************
       SUBROUTINE PUNCHOUT(CRSPC,TOTD,IAGG,ISWELL,ACSTR28,
     +                     ICRLOC,IFATPRO,IHIST,NUMPCS)
C
C      to calculate punchout prediction
C
```

```
      IMPLICIT REAL*8 (A-H, O-Z)
       DIMENSION CRSPC(10,4095),FPM(20),ICRLOC(10,29),XNOR(30)
C
C      Define probability of standardized normal distribution
C      XNOR(1)=probability of minus infinity to 0.1
C      XNOR(25)=prob. of minus infinity to 2.5, and so on
C
       XNOR(1)=0.5398
       XNOR(2)=0.5793
       XNOR(3)=0.6179
       XNOR(4)=0.6554
       XNOR(5)=0.6915
       XNOR(6)=0.7257
       XNOR(7)=0.7580
       XNOR(8)=0.7881
       XNOR(9)=0.8159
       XNOR(10)=0.8413
       XNOR(11)=0.8643
       XNOR(12)=0.8849
       XNOR(13)=0.9032
       XNOR(14)=0.9192
       XNOR(15)=0.9332
       XNOR(16)=0.9452
       XNOR(17)=0.9554
       XNOR(18)=0.9641
       XNOR(19)=0.9713
       XNOR(20)=0.9772
       XNOR(21)=0.9821
       XNOR(22)=0.9861
       XNOR(23)=0.9893
       XNOR(24)=0.9918
       XNOR(25)=0.9938
       XNOR(26)=0.9953
       XNOR(27)=0.9965
       XNOR(28)=0.9974
       XNOR(29)=0.9981
       XNOR(30)=0.9987
C
C      Define coefficients of fatigue equation
C      Unit=million, COEFVA=0.3
C
C      A value             SRG(IAGG=8)           Limestone and others
C      swelling     Yes          No              Yes            No
C             95%         1.4           2.0            1.8           2.6
C             75%         2.4           3.1            2.5           3.7
C             50%         3.1           4.2            3.1           4.8
C
C      if ISWELL=1, swelling=yes
C      IFATPRO=1;95%, 2;75%, 3;50%
C
       IF(IAGG.EQ.8) THEN
             IF(ISWELL.EQ.1) THEN
                   IF(IFATPRO.EQ.1) FATA=1400000.
                   IF(IFATPRO.EQ.2) FATA=2400000.
                   IF(IFATPRO.EQ.3) FATA=3100000.
             ELSE
                   IF(IFATPRO.EQ.1) FATA=2000000.
                   IF(IFATPRO.EQ.2) FATA=3100000.
                   IF(IFATPRO.EQ.3) FATA=4200000.
             ENDIF
       ELSE
             IF(ISWELL.EQ.1) THEN
                   IF(IFATPRO.EQ.1) FATA=1800000.
```

```
                IF(IFATPRO.EQ.2) FATA=2500000.
                IF(IFATPRO.EQ.3) FATA=3100000.
          ELSE
                IF(IFATPRO.EQ.1) FATA=2600000.
                IF(IFATPRO.EQ.2) FATA=3700000.
                IF(IFATPRO.EQ.3) FATA=4800000.
          ENDIF
      ENDIF
C
      FATB=4.
      COEFVA=0.3
      FSTR=7.5*DSQRT(ACSTR28)
C     write(6,*) 'Flexural strength   ',fstr
      TPAVL=0.
      DO I=1,NUMPCS
            TPAVL=TPAVL+CRSPC(I,1)
      ENDDO
C     i mile = 5280ft = 63360in.
      AMUL=63360/TPAVL
C
      DO I=1,20
            FPM(I)=0.
      ENDDO
C
      DO I=1,NUMPCS
C           IF(INDDAY.EQ.1) THEN
                  ICR=ICRLOC(I,IHIST)
C           ELSE
C                 ICR=ICRLOC(I,1)
C           ENDIF
            ICR1=2**(ICR-1)
            ICR2=2**ICR-1
            DO J=ICR1,ICR2
                  CRSPA=CRSPC(I,J)
                  IF(CRSPA.EQ.0.) GO TO 765
                  TST1=EXP(9.8474)*TOTD**(-1.8143)*(CRSPA/12.)**(-0.4477)
                  ANUMLOAD=FATA*(FSTR/TST1)**FATB
                  STDD=COEFVA*ANUMLOAD
                  DO K=1,20
                        AESAL=5000000.*FLOAT(K)
                        Z=(AESAL-ANUMLOAD)/STDD
                        ZZ=DABS(Z)
                        IF(ZZ.GE.3.) THEN
                              PHYZ=1.
                              GO TO 764
                        ENDIF
                        IF(ZZ.LT.0.1) THEN
                              DEL=0.0398*ZZ*10.
                              PHYZ=0.5+DEL
                              GO TO 764
                        ENDIF
                        IZ=ZZ*10.
                        DIF=XNOR(IZ+1)-XNOR(IZ)
                        DEL=DIF*(ZZ*10.-FLOAT(IZ))
                        PHYZ=XNOR(IZ)+DEL
  764 CONTINUE
                        IF(Z.LT.0.) THEN
                              PRFTG=1.-PHYZ
                        ELSE
                              PRFTG=PHYZ
                        ENDIF
                        FPM(K)=AMUL*PRFTG+FPM(K)
                  ENDDO
```

```
  765 CONTINUE
            ENDDO
      ENDDO
      WRITE(9,*) ' '
      WRITE(9,*) 'PREDICTED PUNCHOUTS / MILE'
      WRITE(9,*) '=========================='
      WRITE(9,*) ' Million ESALS    Number of Punchouts / Mile'
      do i=1,20
      write(6,*) fpm(i)
      WRITE(9,9011) I*5,FPM(I)
 9011 FORMAT(I10,13x,F10.3)
      enddo
C
      RETURN
      END
C****************************************************************
```