Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA/TX-88/371-2F		

4. Title and Subtitle	5. Report Date
TTICRCP - A Mechanistic Model for the Prediction of Stresses, Strains, and Displacements in Continuously Reinforced Concrete Pavements.	July 1988
	6. Performing Organization Code
7. Author(s) Richard P. Palmer, Mikael P.J. Olsen and Robert L. Lytton	8. Performing Organization Report No. 371-2F
9. Performing Organization Name and Address Texas Transportation Institute The Texas A&M University System College Station, Texas 77843	10. Work Unit No. (TRAIS) 11. Contract or Grant No. HPR 2-8-85-371
12. Sponsoring Agency Name and Address Texas State Department of Highways and Public Transportation, Transportation Planning Division P.O. Box 5051 Austin, Texas 78763	13. Type of Report and Period Covered Final November 1985 August 1987 14. Sponsoring Agency Code

15. Supplementary Notes

Study conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration, Study entitled, "Environmental Effects on the Physical Properties of Concrete the First 90 Days."

16. Abstract

The accuracy of the CRCP-1, CRCP-2, and CRCP-3 models developed by the Center for Transportation Research at the University of Texas, Austin, for predicting the behavior of continuously reinforced concrete pavement (CRCP) is limited, in part, by many of the simplifying assumptions that were made for their development. Modeling the bond stress distribution between the concrete and the steel reinforcing as an average stress acting over a development length is, in particular, a gross simplification of actual behavior.

The TTICRCP model was developed by assuming that the bond stress relationship between the concrete and the steel could be approximated accurately by a bond stress-slip function. The function was used for the generation of a system of linear, 2nd order differential equations that describe the CRCP slab. The correct solution to the system of differential equations yields the displacement functions of the concrete and the steel, which, in turn, allow the stress distributions for both the steel and the concrete to be found.

Time dependency was incorporated into the model to allow for multi-day analysis of a CRCP system. Material properties, drying shrinkage, environmental conditions, and wheel loads were all assumed to be time dependent.

17. Key Words	18. Distribution Statement
Continuously Reinforced Concrete Pavement, TTICRCP, Bond Stress-Slip Relationship, Differential Equation System, Energy Balance, Equivalent Load Method	No restrictions. This document is available to the public through the National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia, 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	276	

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

TTICRCP - A MECHANISTIC MODEL FOR THE PREDICTION OF STRESSES, STRAINS AND DISPLACEMENTS IN CONTINUOUSLY REINFORCED CONCRETE PAVEMENTS

by

Richard P. Palmer
Mikael P. J. Olsen
Robert L. Lytton

Research Report 371-2F
Research Study 2-8-85-371

Sponsored by

The Texas State Department of Highways and Public Transportation

in cooperation with

The U. S. Department of Transportation
Federal Highway Administration

July, 1988

Texas Transportation Institute
The Texas A&M University System
College Station, Texas 77843

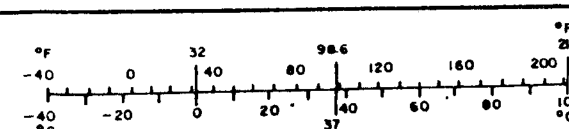
METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



SUMMARY

Rational, computer-based models that can predict the behavior of a continuously reinforced concrete pavement (CRCP) system under the influence of thermal, shrinkage, and traffic loading are available. The CRCP-1, CRCP-2, and the CRCP-3 models, developed by the Center for Transportation Research at the University of Texas, Austin, can perform time-based simulation of a CRCP system. However, the accuracy of the CRCP-1,2, and 3 models is limited by the underlying assumptions that were made to develop the models. One such assumption is that the distribution of bond stress that exists between the concrete and the steel reinforcing bar can be modeled as an average stress acting over a calculated development length.

To improve upon the CRCP-1,2, and 3 models, a method of predicting the actual bond stress distribution was developed. The local bond stress-slip relationship that exists between the concrete and the steel was approximated by a bond stress function. This function was used, along with a similar function for the friction stress between the concrete and the base course, to generate a system of linear, 2nd order differential equations that describe the displacement functions of the concrete and the steel. The correct solution to the system of differential equations is found by equating the total energy that is available to the slab with the energy that is stored or lost as potential, frictional work, and stress relief energy.

This mathematical theory is implemented in the TTICRCP computer program, which uses matrix and iterative methods to find the exact and correct solutions to the differential equation system that exists. In addition, the TTICRCP model allows for time-based simulation of a CRCP system that is subjected to thermal, shrinkage, and wheel loading. The range and capabilities of the TTICRCP model are briefly examined, with the influence of several key input parameters on CRCP behavior being presented. Also, the ability of the model to predict the performance of the CRCP system over a 40 day analysis period is shown.

IMPLEMENTATION

The TTICRCP computer program developed in this study can be used for time-based simulation of a continuously reinforced concrete pavement system. The effects of thermal, shrinkage, and wheel loads may all be seen, in addition to the influence of material properties and geometric characteristics of the pavement slab. The displacement and stress distributions of both the concrete and the steel reinforcing, the bond stress distribution between the steel and the concrete, and the friction stress distribution between the concrete and the base course are found explicitly. The program is also capable of determining the effects of one or more layers of reinforcing steel.

The program may be used by the Highway Design Division, D-8, to design CRCP by investigating the formation of transverse cracks and the development of steel stress during the first 90 days of the life of the pavement. Material properties for the bond stress - slip relationship between the steel and the concrete must be determined by laboratory measurements in order to make the best use of this program.

DISCLAIMER

The information contained herein was developed on Research Study 2-8-85-371 titled "Environmental Effects on the Physical Properties of Concrete for the First 90 Days" in a cooperative research program with the Texas State Department of Highways and Public Transportation and the U.S. Department of Transportation, Federal Highway Administration.

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. This report does not constitute a standard, specification, or regulation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Symbol	Definition
a	= Contact area of wheel load
α_c	= Concrete's thermal coefficient of contraction
α_s	= Steel's thermal coefficient of contraction
A_c	= Area of concrete
A_s	= Area of steel
b	= Spacing of steel reinforcement
C_1	= Intercept for bond stress function
C_2	= Intercept for friction stress function
$C.F.$	= Correction factor for shrinkage strain
d	= Thickness of the pavement
d_s	= Diameter of the steel bar
δ_b	= Slip at which the maximum bond stress occurs
δ_{bl}	= Slip at which the bond stress returns to zero
δ_f	= Displacement at which the maximum friction stress occurs
δ_{fl}	= Displacement at which the friction stress returns to zero
D	= Differential operator
ϵ_c	= Concrete strain
ϵ_s	= Steel strain
ϵ_{shr}	= Shrinkage strain
$(\epsilon_{shr})_{ult}$	= Ultimate shrinkage strain
ϵ_{st}	= Steel strain at the crack
E_c	= Elastic modulus of the concrete
E_s	= Elastic modulus of the steel
E_{fric}	= Frictional work energy expended by the prism
E_{pot_c}	= Potential energy stored by a unit volume of concrete
E_{pot_s}	= Potential energy stored by a unit volume of steel
E_{pot}	= Potential energy stored by the prism
E_{rel}	= Stress relief energy expended by a unit volume of concrete
E_{relief}	= Stress relief energy expended by the prism
E_{total}	= Total energy available to the prism

f'_c = Compressive strength of the concrete

f'_t = Tensile strength of the concrete

H = Relative humidity

k = Modulus of subgrade reaction

K_1 = Positive slope of the bond stress function

K_2 = Negative slope of the bond stress function

K_3 = Postive slope of the friction stress function

K_4 = Negative slope of the friction stress function

ℓ_1 = Distance from crack to point of slip =

ℓ_2 = Distance from crack to point of slip =

ℓ_3 = Distance from crack to point of displacement =

P = Wheel load

S = One half the crack spacing

σ = True stress

σ' = Displacement stress

σ_c = Concrete stress

σ_s = Steel stress

$\Delta\sigma_c$ = Change in concrete stress

$\Delta\sigma_s$ = Change in steel stress

σ'_c = Concrete displacement stress

σ'_s = Steel displacement stress

ΔT = Change in temperature

τ_b = Bond stress

τ_f = Friction stress

u_0 = Displacement of the concrete at the crack

u_c = Concrete displacement

u_s = Steel displacement

CHAPTER I

INTRODUCTION

OVERVIEW OF CONCRETE PAVEMENTS

The use of concrete as a paving material has a long history that dates back to the early 1900's in the United States. Since its introduction, considerable effort has been made to develop design methods and construction procedures that capitalize on concrete's high compressive strength, while minimizing the effect that its low tensile strength and susceptibility to volume changes have on the efficiency and durability of the pavement structure. It has been an evolutionary process, that began with the use of expansion joints to control volumetric changes. Since random cracking still occurred, the distance between expansion joints was increased and contraction joints were installed at 15 to 20 feet intervals between the expansion joints. The distance between the contraction joints increased when longitudinal steel reinforcing was introduced to hold the pavement slab together as it reacted to temperature drops and shrinkage strains. The reinforcing allowed for contraction joint spacings of up to 100 feet.(1)

Jointed Reinforced Concrete Pavements

Jointed reinforced concrete pavement design today requires that the pavement slab be reinforced both longitudinally and transversely, with contraction joints typically spaced 40 to 60 feet apart. Sophisticated load transfer systems of dowels and keys are used to insure that detrimental deflections and strains do not occur at the joints, and, quite often, the joints are skewed to reduce the impact of traffic loads. However, the joints are still a major weakness in the pavement's design for several reasons. Joints provide a pathway that allows water from the surface to seep into, and weaken, the supporting base courses. In addition, they present a maintenance problem if the seepage is to be kept minimal, since they must be resealed every few years. The joints also have an adverse effect on the riding quality of the entire pavement and can be a source of driving discomfort.(1)

Continuously Reinforced Concrete Pavements

Continuously reinforced concrete pavement (CRCP) was developed as an alternative to jointed concrete pavement that eliminates the need for contraction joints. In CRCP, the longitudinal reinforcement is continuous, in that each individual length of reinforcing bar is welded or lapped end-to-end to each of its adjoining reinforcing bars in the reinforcement grid. The presence of the steel in the pavement does not prevent the formation of cracks. Rather, it induces the concrete to develop numerous transverse, hairline cracks. Theoretically, the steel keeps the cracks tightly closed, which maintains the integrity of the aggregate interlock across the cracks. This prevents the formation of high shear stresses in the steel due to traffic loads, and blocks the passage of water from the surface to the subgrade.(1,2)

Cracking begins just a few days after the placement of the pavement slab and will continue throughout the life of the pavement, although new crack development is very slow after 3 or 4 years. Well-performing pavements have final crack spacings in the range of 3 to 10 feet (2). Pavements with crack spacings larger than 10 feet may have crack openings that are too wide to maintain the integrity of the aggregate interlock across the crack, which forces the reinforcing steel to carry the full magnitude of the shear force induced by wheel loads at the crack. This can rupture the steel reinforcing. Wide crack openings also allow contaminants, such as sand or gravel, to become lodged in the crack. The contaminants resist the expansion of the slabs, which may lead to spalling at the cracks. Crack spacings less than 3 feet may give the pavement excessive flexibility, which is also undesirable (2).

There are many variables that are known to influence the crack spacing, including the type and amount of longitudinal steel, the air temperature at the time of placement, the composition and uniformity of the concrete, the base friction, traffic loadings, and the air temperature range relative to the placing temperature. The relationships between these variables are highly complex and not easily defined, which makes accurate prediction of the final crack spacing for a specific pavement very difficult. Empirical observations of previously constructed pavements are often relied upon as the predicting tool. CRCP also does not carry the distinction of being easy and inexpensive to repair when rehabilitation is necessary. CRCP is often more difficult and more costly to repair than other pavements because of the large amount of steel reinforcing it contains and the necessity of maintaining the continuity of the reinforcement. For these reasons, the importance of developing a rational method of predicting CRCP behavior based on the mechanics of the pavement structure, the environmental conditions, and the material properties is great.(2)

PREDICTING CRCP PERFORMANCE

The CRCP-1, CRCP-2, and CRCP-3 Models

Rational methods of predicting CRCP performance do exist. Most notably, the CRCP-1, 2 and 3 computer programs developed under the guidance of B.F. McCullough at the Center for Transportation Research at the University of Texas in Austin allow for a theoretically based approach to the prediction of CRCP performance under a wide range of conditions.(3)

The theory behind the CRCP-1 model, and its modified versions, CRCP-2 and CRCP-3, is based on material properties, stress, the strain interaction between the steel, the concrete, and the subgrade, and the internal forces caused by the temperature drop and shrinkage of the slab. The geometric model of the slab used in the theory basically represents a pavement section extending from the crack face to the midpoint of the slab. Both the concrete and the steel are assumed to be anchored at the midpoint, while at the crack face, only the steel is anchored, with the concrete being free to move against the restraining bond and frictional forces provided by the steel and base material. The other half of the slab is not considered due to the slab's symmetry. (3)

An average, uniform stress distribution is assumed to occur over the cross section of the concrete, and both the concrete and steel are assumed to be linearly elastic. Warping stresses due to a temperature differential across the depth of the slab are not considered. The bond that exists between the concrete and the steel is modeled by an average bond stress acting over a computed development length, with the bond stress value being a function of the concrete strength and the bar geometry. The frictional resistance between the concrete and the base material is characterized by an elastic force displacement curve. It is also assumed that: 1) a crack will occur when the concrete stress exceeds the concrete strength, 2) the concrete stress at the crack face is equal to zero, 3) no movement occurs between the steel and the concrete in fully bonded sections of the slab, and, 4) all material properties are independent of space. In addition, time dependent material properties, such as the concrete's compressive and tensile strengths, are handled as such where appropriate.(3)

For a given temperature drop and magnitude of drying shrinkage, the model first calculates the concrete slab's movement as if there were no frictional force between the base material and the concrete. Once that initial movement is calculated, a binary search technique is used to move along the force displacement curve until the calculated movement and the movement obtained from the curve are within tolerance of each other. Once the correct movement is found, the corresponding frictional force is used to solve for the steel stress at the crack and the concrete stress in the center of the slab. If the concrete stress at mid-slab is greater than the concrete strength, the slab is cracked in half and the calculations are started over with the new slab length.

Wheel loads can also be handled by the model in two different ways. The model can calculate the stress induced by a given wheel load using Westergaard's equation for interior loading, or the user can input a stress level that was calculated by some other means. In either case, the wheel load stress is added to the concrete stress at mid-slab to determine the final stress state of the concrete.

Limitations of the CRCP Model

The CRCP model falls short of being truly representative of an actual CRCP system. Although it provides good approximations of true behavior for some conditions, the large number of simplifying assumptions that were made to develop the model limit its accuracy to a great extent. Although many of the assumptions are necessary to reduce the mathematical complexity of the model to a manageable level, some are gross simplifications of the conditions that actually exist in the pavement system. Assuming that the bond stress between the steel and the concrete acts as an average value over a specified development length is one such assumption. Research has shown that the actual bond stress distribution has a range of values that vary from zero to a peak magnitude that far exceeds the average value (4). Figure 1.1 compares the shape of an actual distribution with an average distribution.

Assuming an average bond stress distribution severely limits the validity of the CRCP model in many instances. In cases where there is high frictional resistance between the base material and the pavement slab, or there is a large temperature drop, the computed development length for the bond stress can actually exceed the length of the slab over which the bond forces have to act (3). This is a physical impossibility in an actual pavement, since the concrete and the steel have to be fully bonded at the midpoint of the slab. In addition, the stress distributions in both the steel and the concrete are governed, in part, by the bond stress distribution between them. If the bond stress distribution is invalid, then the stress distributions in the steel and the concrete will also be invalid.

Other assumptions that limit the accuracy of the CRCP model include: 1) modeling the concrete as a linear, elastic material, 2) assuming that the temperature variations and shrinkage strains are uniform throughout the slab, and 3) neglecting the effect of creep and slab warping.

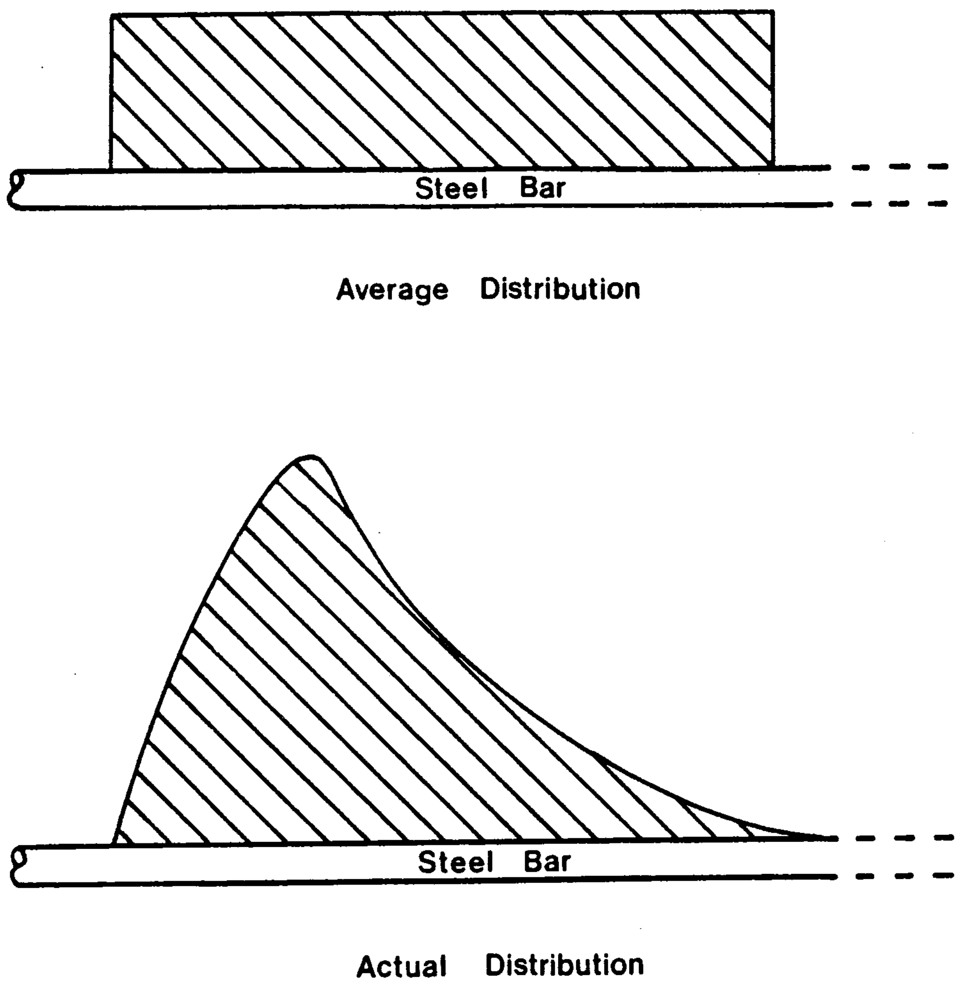


Figure 1.1. Comparison Between an Average Bond Stress Distribution and an Actual Bond Stress Distribution.

Improving the CRCP Model

To improve upon the CRCP model, it is necessary to discard one or more of the above assumptions and rework the theory using a more realistic approach. Attempting to account for non-linear behavior by the concrete or trying to model the slab in two or more dimensions are highly complex tasks that would be entirely invalid if a more accurate method of predicting the bond stress is not used. For this reason, a model has been developed that accounts for the variation of bond stress over the length of the slab by using a fundamental bond stress-slip relationship. The underlying theory and mathematical procedures that are used in this model are presented in the following chapters.

CHAPTER II

THE BOND STRESS AND FRICTION STRESS FUNCTIONS

LITERATURE SURVEY OF BOND STRESS

The initial task in the development of the new model was to find a way to characterize the bond stress distribution between the steel reinforcing and the concrete more accurately than just assuming an average value. To this end, a literature search was conducted to review the pertinent research on bond stress development.

Early Research on Bond

The first major study of bond was conducted in 1913 by Abrams, who performed a very extensive study of the bond of plain and deformed bars by using 1500 pull-out tests and 110 beam tests. He investigated the influence of concrete properties, concrete cover, embedment length, bar diameter, and many other variables. In 1919, he conducted additional pull-out tests to supplement his previous work.(5)

Lutz (5) has presented a comprehensive and concise review of the major contributions in bond research from Abrams until the mid-1960's. Much of this early work was concerned with the bond strength of deformed bars and the effect that different deformation patterns had on the ultimate bond strength. The testing technique used with pull-out and beam specimens did not allow the bond stress distribution between the reinforcing and the concrete to be determined. Only an average bond stress over the embedded length of the bar could be calculated due to the simplistic nature of the tests.

In 1948, Mylrea (6) elaborated on the careful study that Abrams had made on the variation in bond stress at a point on plain bars at increasing load levels during pull-out tests. Abrams had concluded that bond stress was a function of the relative slip that occurred between the steel bar and the concrete (see Fig. 2.1). Mylrea extended this

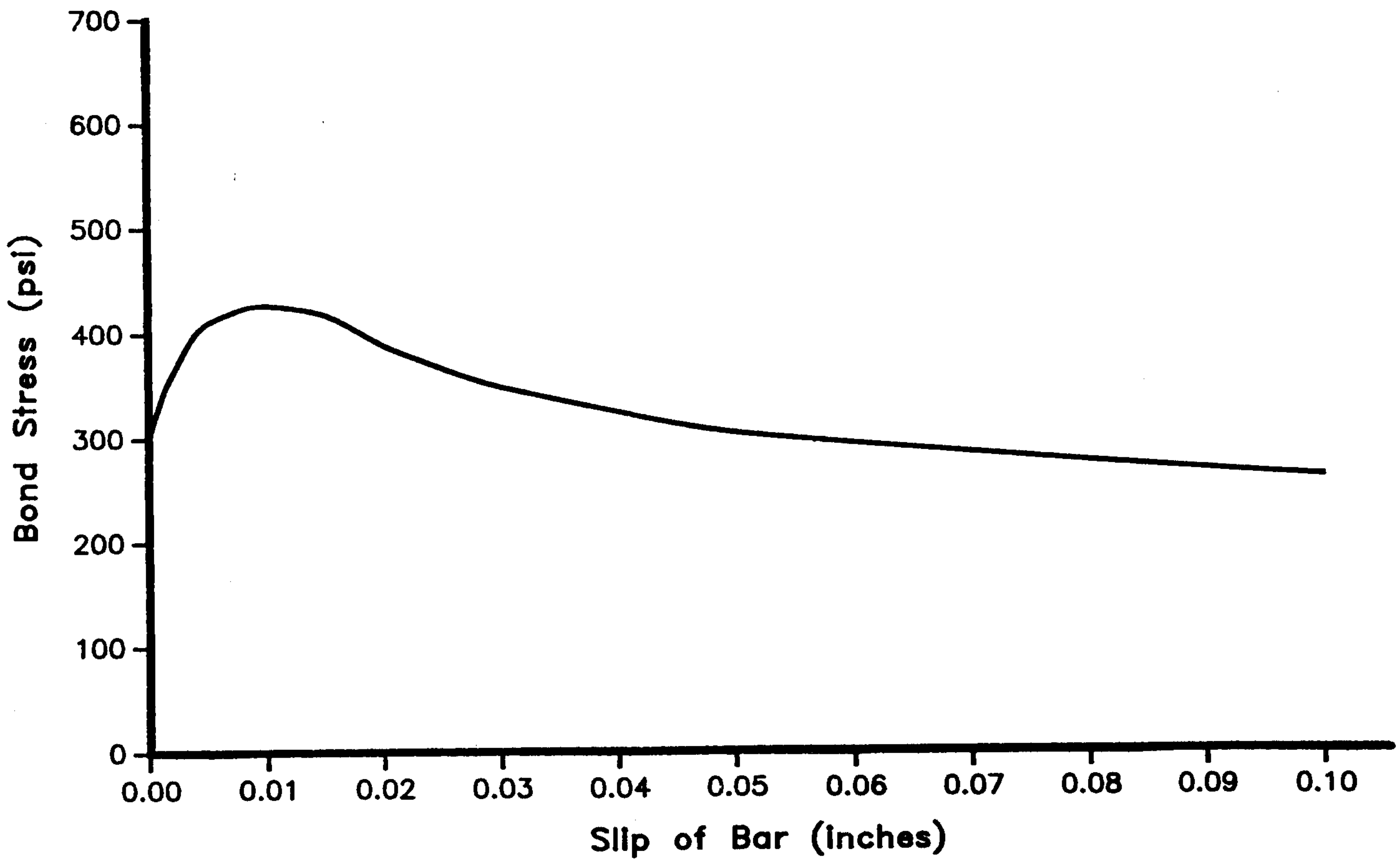


Figure 2.1. Abrams' Bond Stress-Slip Relationship (6).

premise to describe the distribution of bond stresses throughout the length of a pull-out specimen as the bar is subjected to increasing pull (see Fig. 2.2). He also acknowledged that bar deformations improve the bond strength of the reinforcing, but he could not definitively describe the effect the deformations have on the bond stress distribution.

Watstein (7,8), in two separate investigations in the 1940's, indirectly measured the bond stress distribution of several different plain and deformed bars in pull-out specimens with embedment lengths of 8, 12, and 18 inches. The tensile stress in each bar was directly measured by the displacement of the bar over several 2 inch gage lengths. Holes were formed in the concrete specimens, through which strain-transferring devices mounted with Tuckerman strain gages were placed into holes drilled in the embedded steel bars. At each increasing load level during the pull-out tests, the steel strains were recorded and used to construct plots of the steel stress distribution. By measuring the slope of the steel stress distribution, the bond stress at that point could be calculated, and from that, the bond stress distribution could be found. This technique was very prone to error, and many of the results had to be discarded due to faulty measurements. Additional error was created during the calculation process, since it was difficult to accurately determine the slope of the steel stress distribution. Even so, Watstein found that a very large variation in bond stress existed over the length of the specimens, and the variation was even more pronounced with the highly deformed bars.

Watstein also constructed local bond stress vs. slip curves for uniformly spaced points along the embedded bars. He could not directly measure the relative slip that occurred between the steel and the concrete, nor could he calculate it from only the steel strain distribution. However, by estimating the concrete compressive strain distribution and combining it with the steel strains, the slip could be approximated.

In 1951, Mains (9) utilized a new technique of measuring the tensile stresses in the embedded steel bars that furnished a more detailed picture of undisturbed steel and bond stresses than was

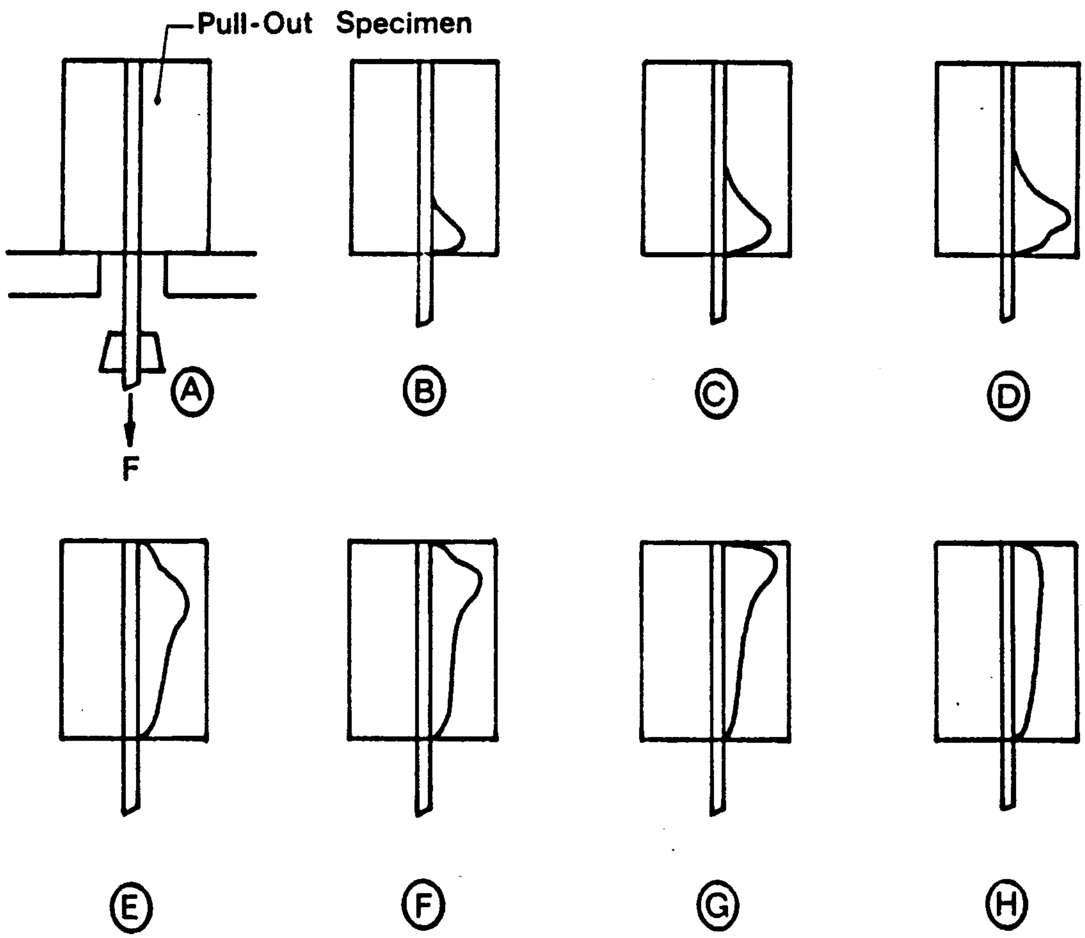


Figure 2.2. Progression of the Bond Stress During a Pull-Out Test (6).

previously available. By sawing the reinforcing bar in half and grooving the cut face, he was able to mount electrical resistance strain gauges inside the bar. After waterproofing the gages, he tack-welded the halves of the bar back together. This allowed the bond between the concrete and the steel bar to be essentially unchanged by the addition of instrumentation.

Tests were performed on both pull-out specimens and beams, and extensive tensile stress data was collected during each test. Bond stress distributions were calculated in the same manner as Watstein (7,8), and similar results were obtained. Mains was able to apply a greater load to the pull-out specimens than Watstein, since he did not have to stop the test before failure. (Watstein stopped loading his specimens well before the failure point in fear of damaging the strain measuring devices.) At higher levels of loading, Mains' pull-out tests showed that the bond stress distribution was peaked, with the maximum bond stress occurring several inches from the loaded end of the bar (see Fig. 2.3). This general shape was quite similar to the distribution that Mylrea had predicted occurred in pull-out specimens with plain reinforcing bars.

The Bond Stress-Slip Relationship

Mains (9) did not attempt to calculate approximate bond stress vs. slip curves as Watstein (8) had done previously, but many researchers since have worked to measure the bond stress-slip behavior directly. Some of the earliest work in this area was performed by Wahla (10), who conducted pull-out and tension tests on a variety of specimens that were designed to model concrete beam behavior.

Wahla used a method similar to Mains' to internally instrument the reinforcing bar, although he grooved both halves of the cut bar to allow strain gauges to be mounted in each half. He also found that epoxying the bar halves back together was superior to tack-welding. To measure the localized slip that occurred between the reinforcing and the concrete, he designed and fabricated a slip gage from a wire resistance strain gage and a short, metal bar. The gage was designed such that one

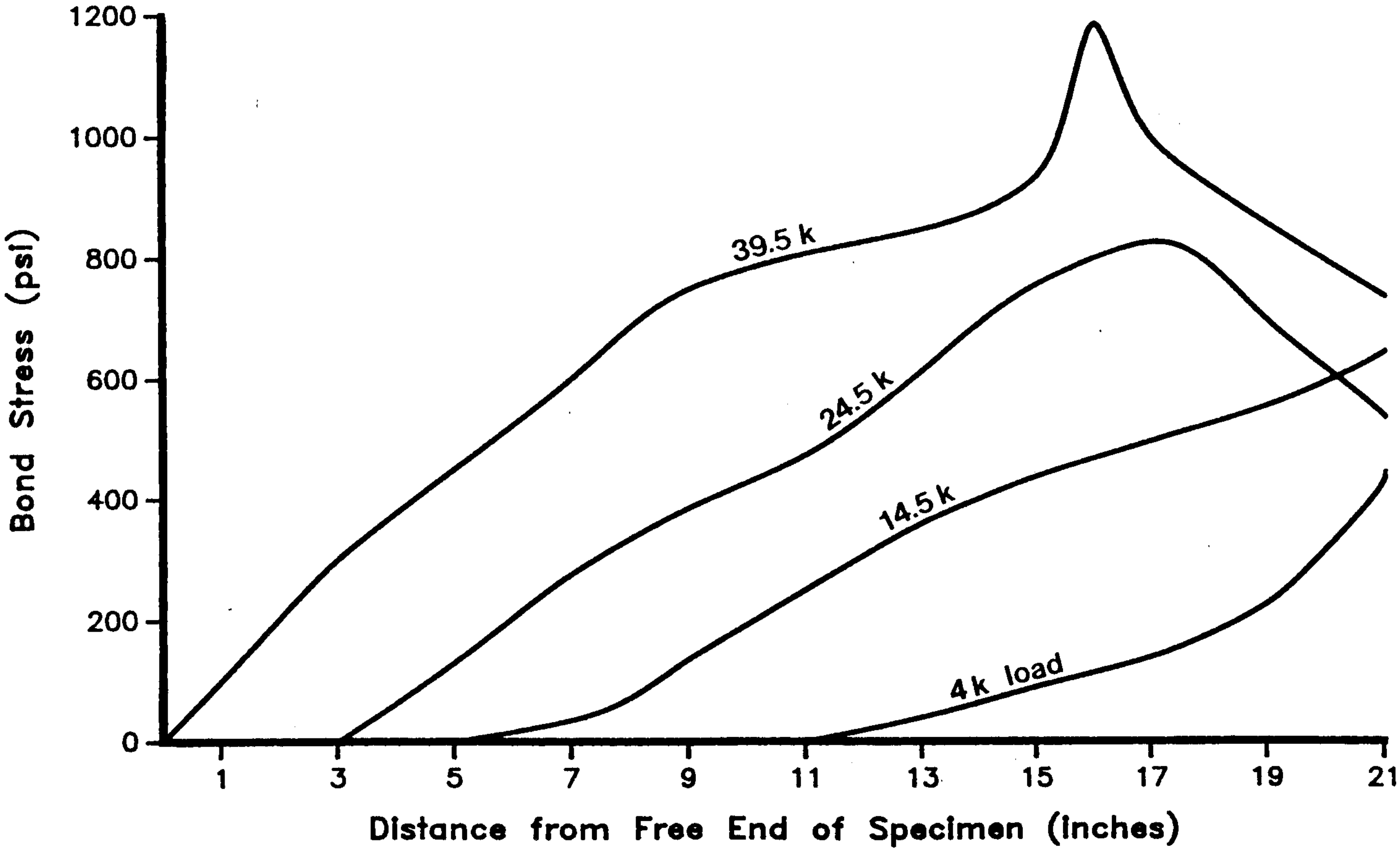


Figure 2.3. Bond Stress Distribution Along a 21" Pull-Out Specimen at Different Bar Loadings (9).

end of the gage was glued directly to the exterior of the reinforcing bar, while the other end of the gage, which was attached to the metal bar, was free to move. When the concrete was placed around the reinforcing, the metal bar became anchored in the concrete. Upon testing the specimen, the slip that occurred between the reinforcing bar and the concrete could be measured by the change in resistance of the slip gage at that point.

Wahla found that the local bond stress-slip relationship was not constant over the length of the tensile specimen. Rather, within a distance equal to twice the diameter of the reinforcing from the face of the specimen, the bond stiffness decreased substantially. Figure 2.4 shows the experimental curves of both the external elements (within 2 diameters of the specimen face) and the internal elements (more than 2 diameters away from the face) obtained by Wahla for his tensile specimens. Wahla recommended the use of linear functions to model the bond stress-slip relationships (see Fig. 2.5).

Nilson (11) approached the problem of measuring the slip between the concrete and the reinforcing bar in a different manner. Instead of mounting slip gages on the reinforcing bar, he placed concrete strain gauges along the length of the specimen approximately 0.5 in. from the surface of the bar. He used an internally instrumented bar very similar to Wahla's. By integrating each of the strain distributions, the displacements of the steel and concrete could be found. The bond stress-slip relationships obtained again showed the influence of the proximity of the specimen face. Like Wahla, Nilson developed a linear function to approximate the local bond behavior.

Edwards and Yannopoulos (12) conducted their investigation with much smaller concrete specimens than were used by either Wahla or Nilson, with the thought that the length of a basic bond specimen should be as small as possible so that the basic local bond stress-slip relationship could be determined. Using 16 mm (0.63 in.) bar cast through a 38 mm (1.5 in.) thick block of concrete, pull-out tests were performed on a large number of samples. Slips were measured by linear displacement transducers located on the top and bottom faces of the

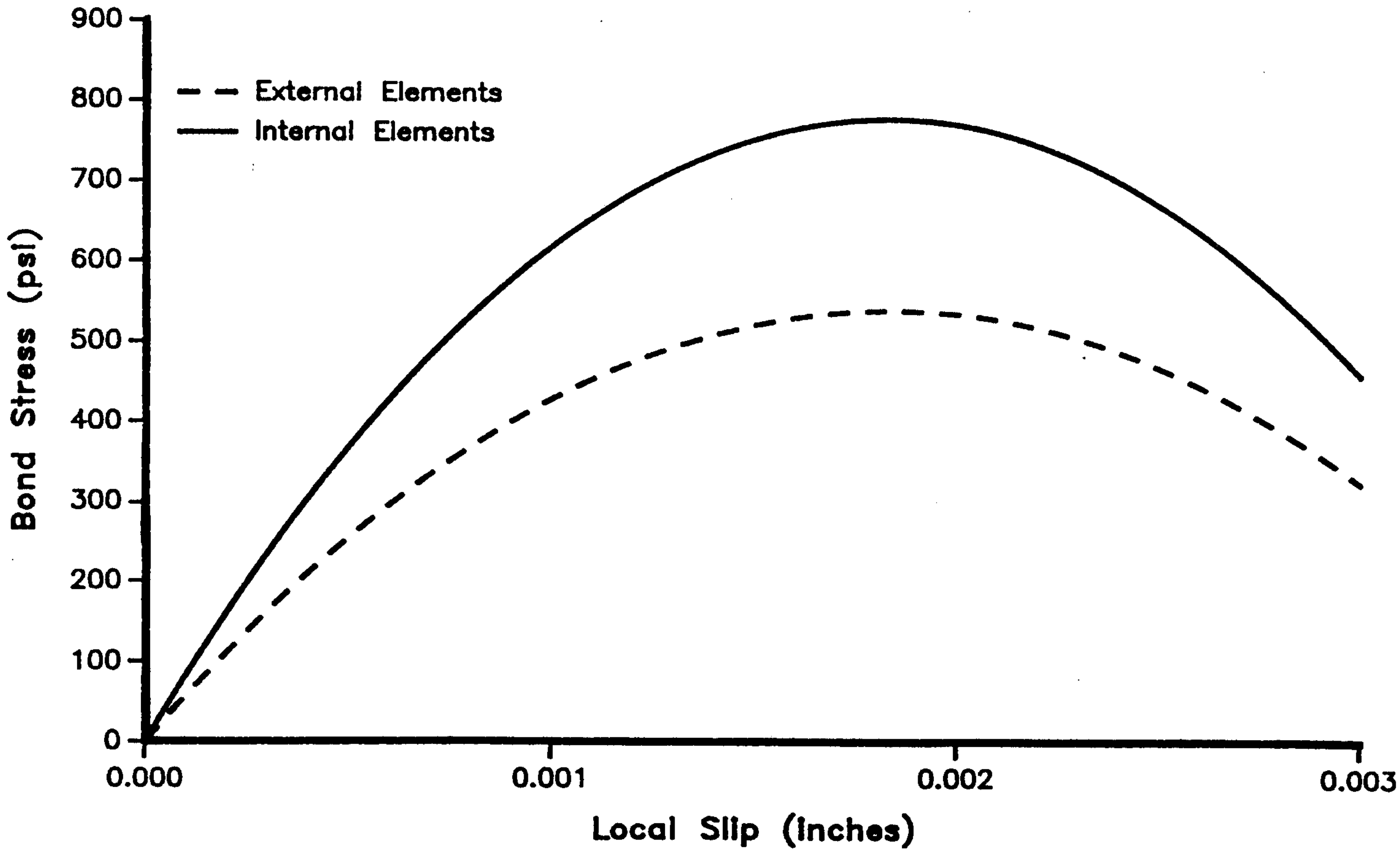


Figure 2.4. Experimental Bond Stress vs. Slip Curves for Tensile Specimens from Wahla (10).

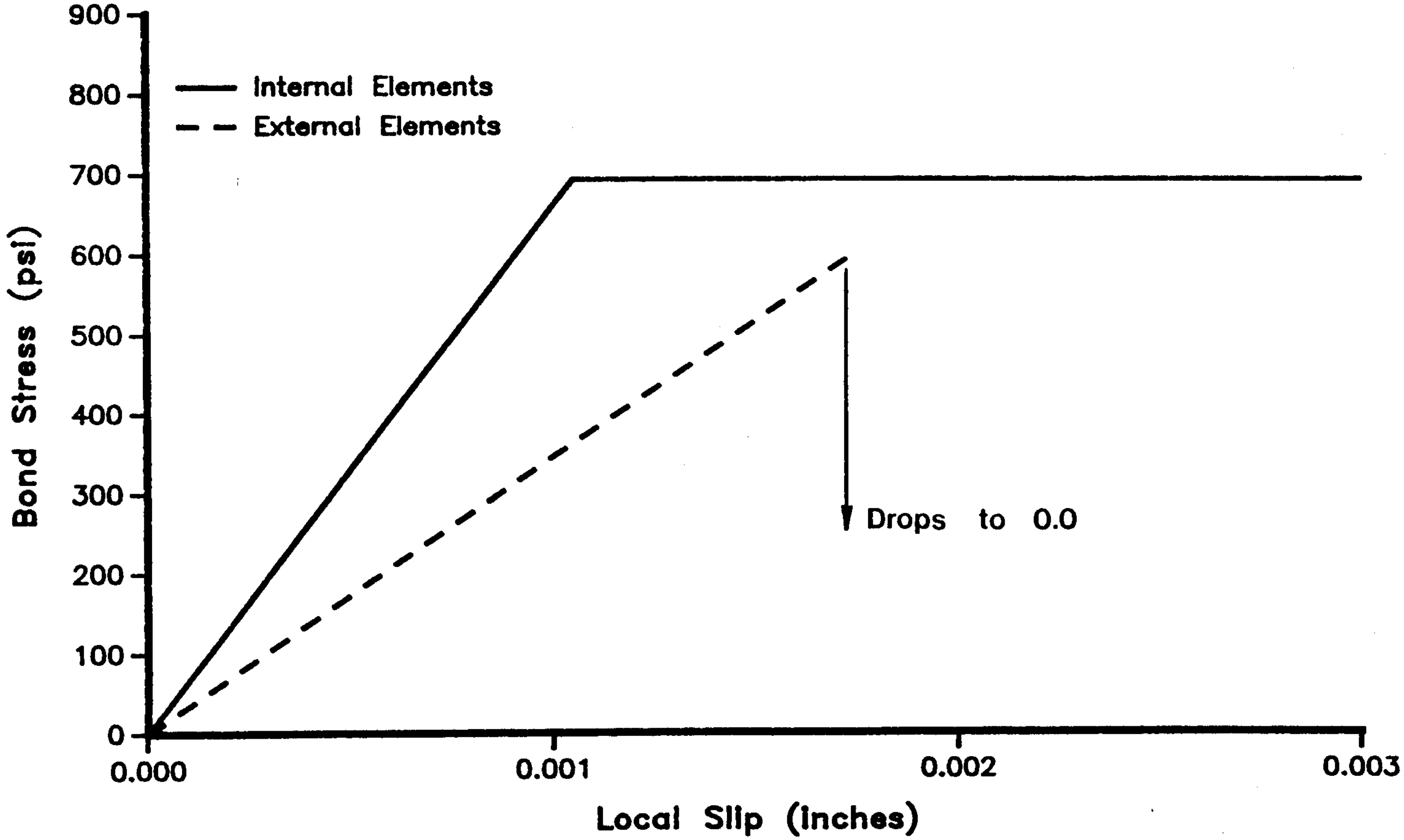


Figure 2.5. Recommended Linear Approximations for the Bond Stress-Slip Relationship for Tensile Specimens from Wahla (10).

specimen. The small size of the specimen allowed the bond stress to be computed as an average stress over the bonded area of the bar.

The bond stress-slip curves obtained from their testing differ from those obtained by Wahla and Nilson. Wahla's and Nilson's curves indicate that after a peak bond stress value is reached, the bond stress levels off or decreases with increasing slip. For Edwards and Yannopoulos, a peak bond stress was rarely reached before failure of the specimen, even though a significant slope change often occurred at low slip values. Figure 2.6 shows a comparison between typical experimental curves obtained in each of the investigations.

THE BOND STRESS FUNCTION

Based on the literature reviewed, it was decided that the bond stress distribution over the length of a CRCP slab could be realistically modeled by assuming a local bond stress-slip relationship for the development of bond stress between the reinforcing and the concrete. Determining how the relationship was to be mathematically modeled was governed by two requirements: 1) it had to be as simple as possible without compromising accuracy and 2) it must be substantiated by the current research findings available.

Looking at the work of Wahla, Nilson, and Edwards and Yannopoulos, it was felt that Wahla's and Nilson's test specimens and testing methods more closely resemble actual CRCP behavior. For this reason, their results were preferred for the formulation of the general bond stress-slip relationship. Both Wahla and Nilson recommended the use of two-part linear functions to approximate the bonding behavior, with most of their linear approximations maintaining a constant level of bond stress equal to the maximum bond stress after the maximum stress level is reached. However, the experimental data collected in their studies indicated that after the maximum bond stress value was reached, the bond stress decreased with further slip (as has been previously seen in Fig. 2.6). For this reason, it was felt that a three-part linear function would model the experimental data more accurately.

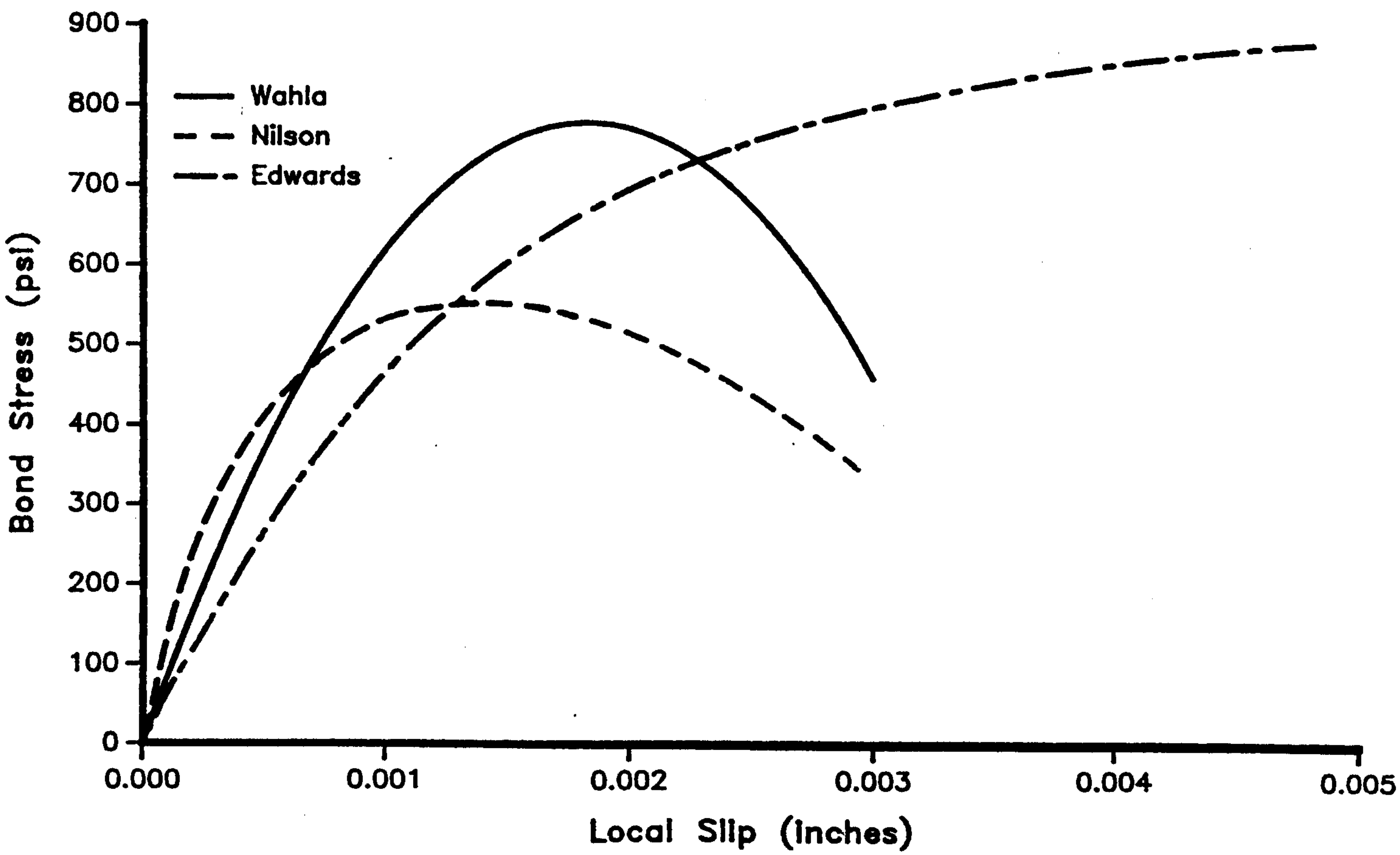


Figure 2.6. Comparison of Experimental Bond Stress vs. Slip Curves (10, 11, 12).

Figure 2.7 shows the proposed shape of the function. Bond stress is treated entirely as a function of the relative slip between the concrete and the reinforcing in this model. As the magnitude of slip increases from zero, the bond stress increases at a rate equal to K_1 until the maximum bond stress is reached at a slip of δ_b. The bond stress then decreases at a rate equal to K_2 as further slip occurs. The bond stress has decreased to zero at a slip of δ_{bl}, and is equal to zero at all greater magnitudes of slip. K_1, K_2, and δ_b are not fixed values, but rather material properties of the specific concrete and steel reinforcing used. Figure 2.8 shows the bond stress-slip function imposed on one of Nilson's experimental curves.

The effect of decreased bond stiffness near the face of the specimen, noted by both Wahla and Nilson, was disregarded in the three-part function. Wahla indicated that the effect was noticeable within a distance of 2 bar diameters from the face, while Nilson reported that bond behavior was influcenced up to 6 inches into his specimens. It was felt that neglecting the influence of this phenomenom would not severely compromise the accuracy of the model, since the slabs in CRCP are considerably longer than the specimens tested by Wahla and Nilson.

THE NATURE OF FRICTIONAL RESISTANCE

The frictional resistance between the base material and the concrete has also been a long-standing research concern, with studies being conducted as early as 1924 (13). These studies have shown that frictional force is a function of slab displacement, and that the general shape of the friction force - displacement curve is quite uniform. Initially, the friction force rises with increasing displacement. The rate of the rise is defined by the elastic properties of the base material, and the force continues to rise until it reaches a peak value. Once the peak value has been reached, the friction force may or may not drop slightly before becoming virtually constant as the displacement continues to increase. The magnitude of the constant force

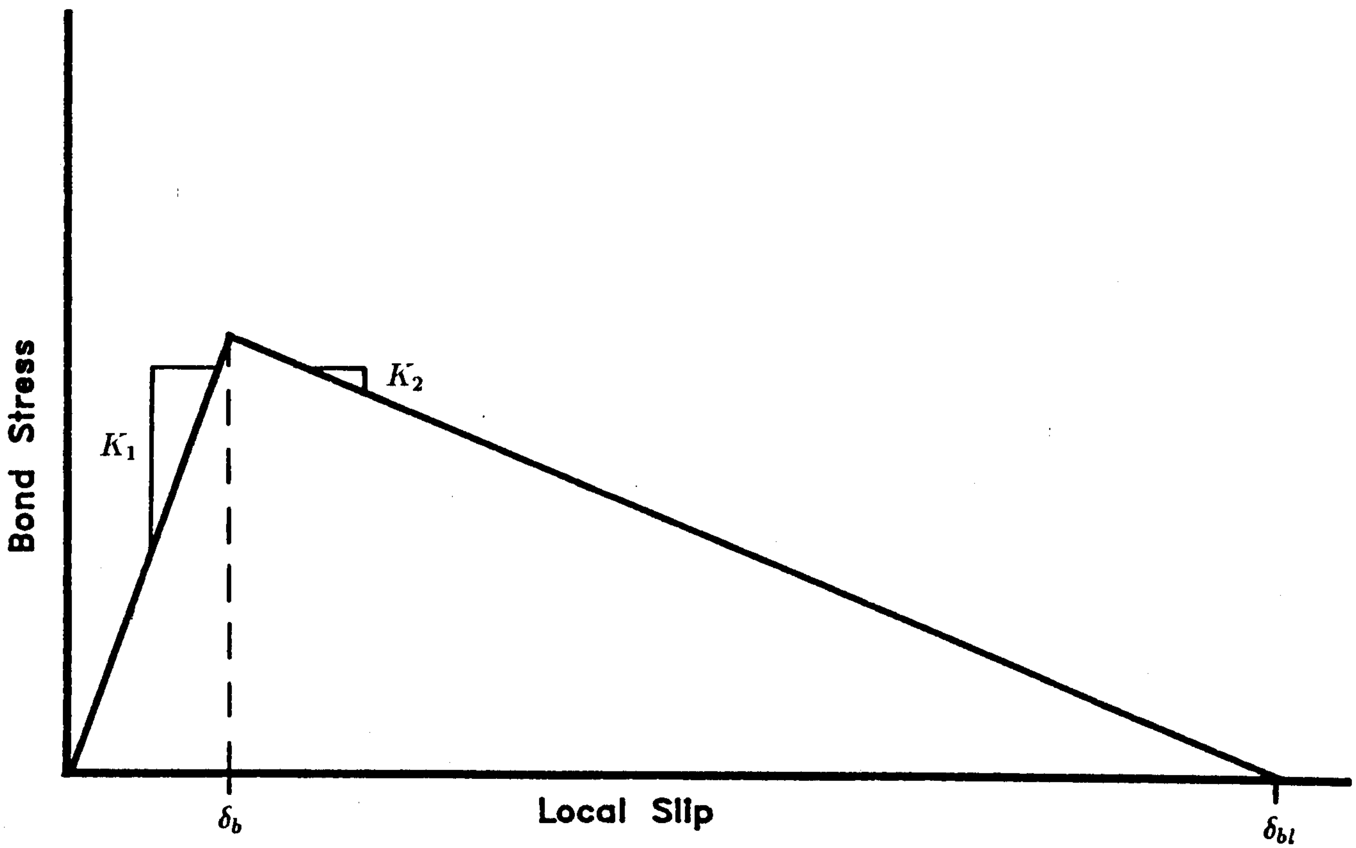


Figure 2.7. Proposed Bond Stress Function.

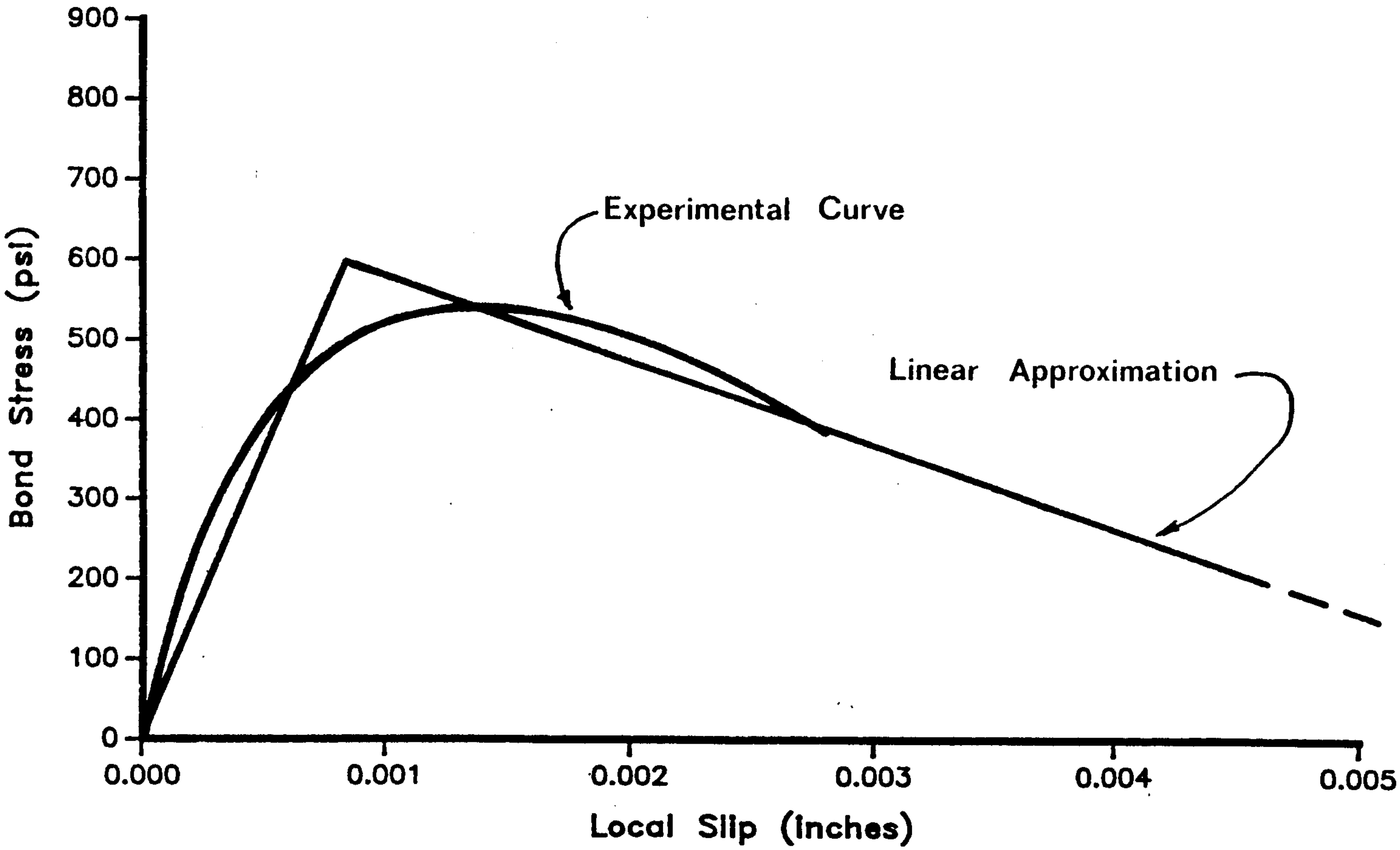


Figure 2.8. Bond Stress Function and Experimental Bond Stress vs. Slip Curve from Nilson (11).

is determined by the nature and the condition of the sliding plane. Figure 2.9 shows a typical force vs. displacement curve (16).

THE FRICTION STRESS FUNCTION

It is desirable for the bond stress function and the friction stress function to have similar forms, since it helps simplify the mathematical theory. For this reason, the friction force vs. concrete displacement curve was converted into a friction stress vs. displacement curve. This does not change the shape of the curve, as the friction stress is equal to the friction force divided by the area over which it acts. Figure 2.10 shows the friction stress function that was chosen to model the curve. By allowing the friction stress function to be composed of three linear portions, it will be able to closely approximate almost any friction stress vs. displacement curve. It would have been possible to model the curve with a two-part function (i.e. the friction stress would be held constant after the peak value is reached). However, by allowing K_4 to be negative, the acccuracy of the function is not compromised and may actually increase for select base materials.

The only differences between the bond stress function and the friction stress function are found in the magnitudes of the defining coefficients. K_1 is assumed to be much greater than K_3, and K_2 is assumed to be greater than K_4. These two assumptions are extremely important. In addition, δ_f is usually greater than δ_b.

With these functions governing the bond force and friction force development, a mathematical model can now be constructed that realistically predicts the distribution of bond stesses and friction stresses throughout the length of the CRCP slab. Chapter III describes the creation of that model.

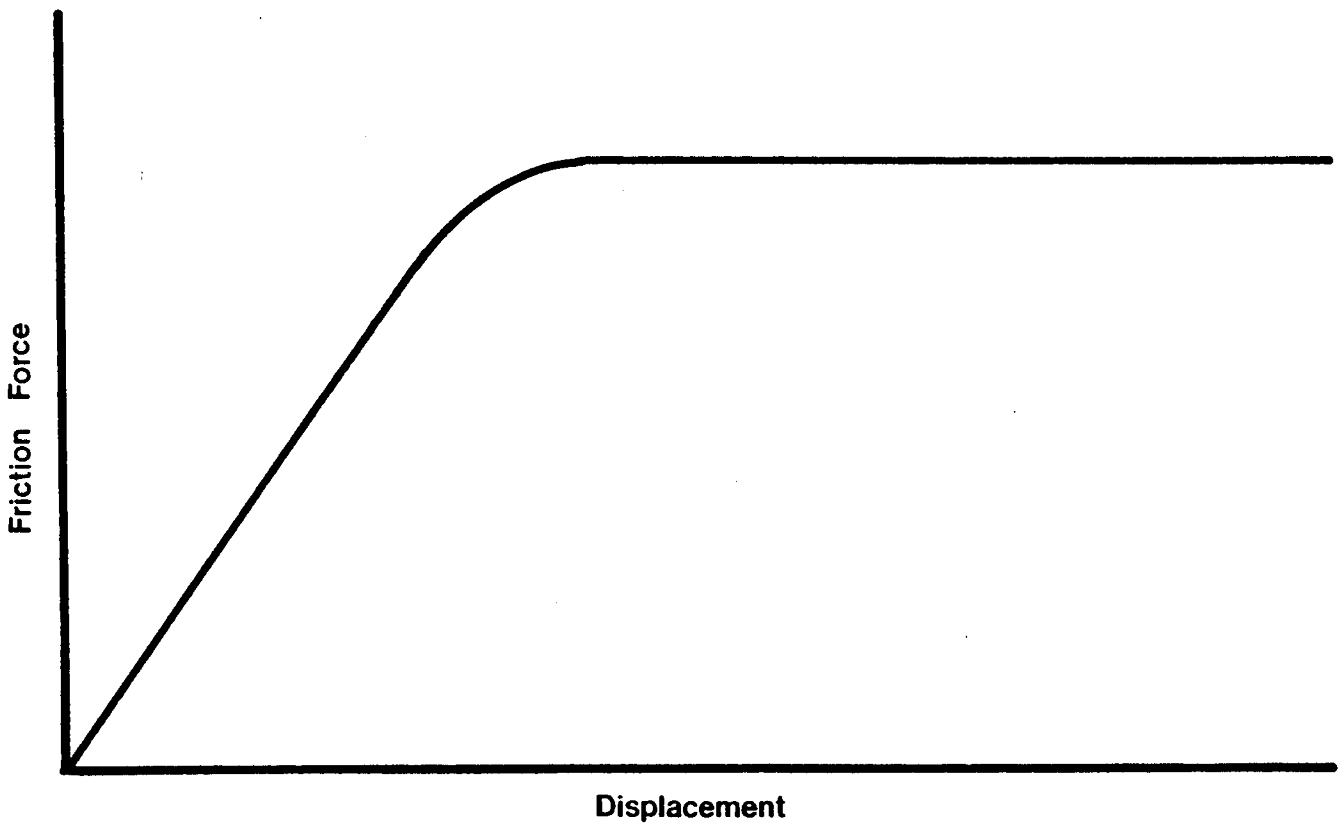


Figure 2.9. Typical Friction Force vs. Displacement Curve (16).

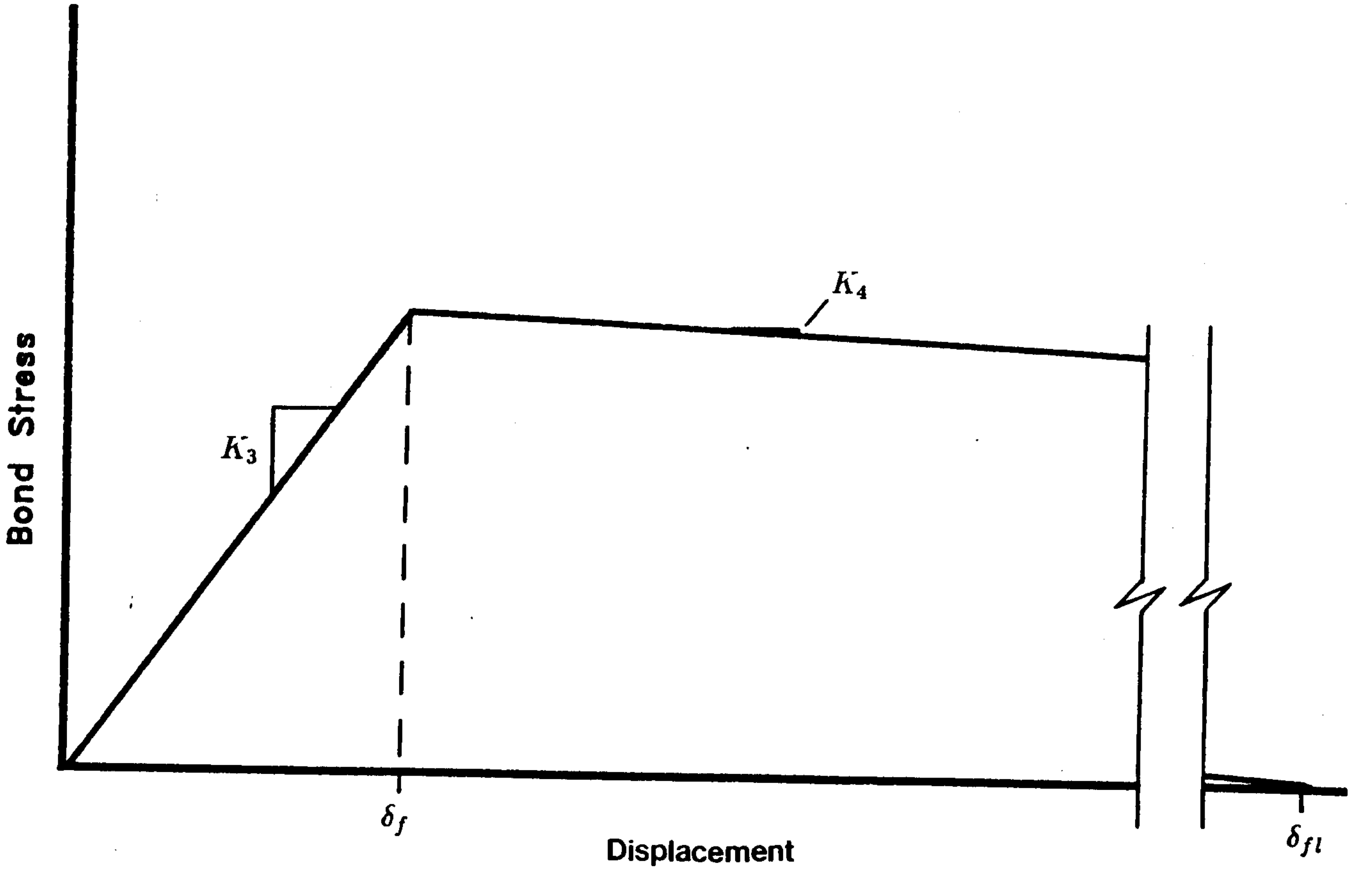


Figure 2.10. Proposed Friction Stress Function.

CHAPTER III

FORMULATION AND SOLUTION OF THE MATHEMATICAL MODEL

GENERATION OF THE MODEL

Assumptions

With the bond stress and friction stress functions providing the means necessary to realistically describe the bond stress and friction stress distributions in the CRCP slab, the formulation of the mathematical model is relatively straight forward. Many assumptions, similar to those made for the CRCP-1, 2 and 3 models (3), were required at the onset to ensure that the theory would be solvable. These assumptions are as follows:

1. The concrete and the reinforcing steel behave in a linearly elastic manner.
2. The base material underneath the concrete slab is rigid, and will not deflect under loading by the horizontal friction force.
3. All materials are homogenous.
4. All temperature and shrinkage induced strains are uniformly distributed throughout the depth and the width of the CRCP slab.
5. The effect of warpage and creep will be ignored.
6. All behavior in the slab is symmetrical about the midpoint of the slab.

By assuming that the temperature and shrinkage induced strains are uniformly distributed throughout the depth and the width of the slab, the slab can be modeled by a prism with a length equal to one-half the length of the slab. Figure 3.1 shows the prism as it is used in the model. It has a thickness d, a width b, which corresponds to the spacing of the steel reinforcement, and a length S, which equals one-half the crack spacing. There is also a cylindrical reinforcing bar

running through the length of the prism. The uniformity of the temperature and shrinkage strains across the depth and width of the prism also allow the mathematical model to be considered a one dimensional, uniaxial problem, since variations in stress and strain will occur only along the length of the prism.

Equivalent Load Method

To solve for the stresses, strains, and displacements in the prism, it is necessary to transform the temperature and shrinkage strains into a force which is designated as an equivalent load (14). The equivalent load takes the place of the thermal and shrinkage effects and appears in the equilibrium equations and boundary equations as a conventional load. For simple cases, the magnitude of the equivalent load is equal to the magnitude of a conventional load that induces a strain equal to the combined strain from the thermal and shrinkage effects. The equivalent load method will be discussed in greater detail in Chapter IV.

Derivation of the Governing Equations

The derivation of the governing equations for the model begins by taking a slice of the prism of width Δx. As Figure 3.2 indicates, a change in the magnitude of both the concrete stress, σ_c, and the steel stress, σ_s, occurs across the width of the slice, which equals $\Delta\sigma_c$ and $\Delta\sigma_s$, respectively. A bond stress, τ_b, is present between the steel reinforcing and the concrete, and a friction stress, τ_f, is present between the concrete and the base material. The forces corresponding to these stress levels are seen in Figures 3.3a and 3.3b.

Examining the concrete in Figure 3.3a, the summation of forces is equal to:

$$\sum_{+\rightarrow} F = (\sigma_c + \Delta\sigma_c)A_c - \sigma_c A_c - \pi d_s \tau_b \Delta x - b\tau_f \Delta x \qquad (3.1)$$

where: A_c = the cross section area of the concrete,

d_s = the diameter of the steel reinforcing, and the other variables are as previously defined.

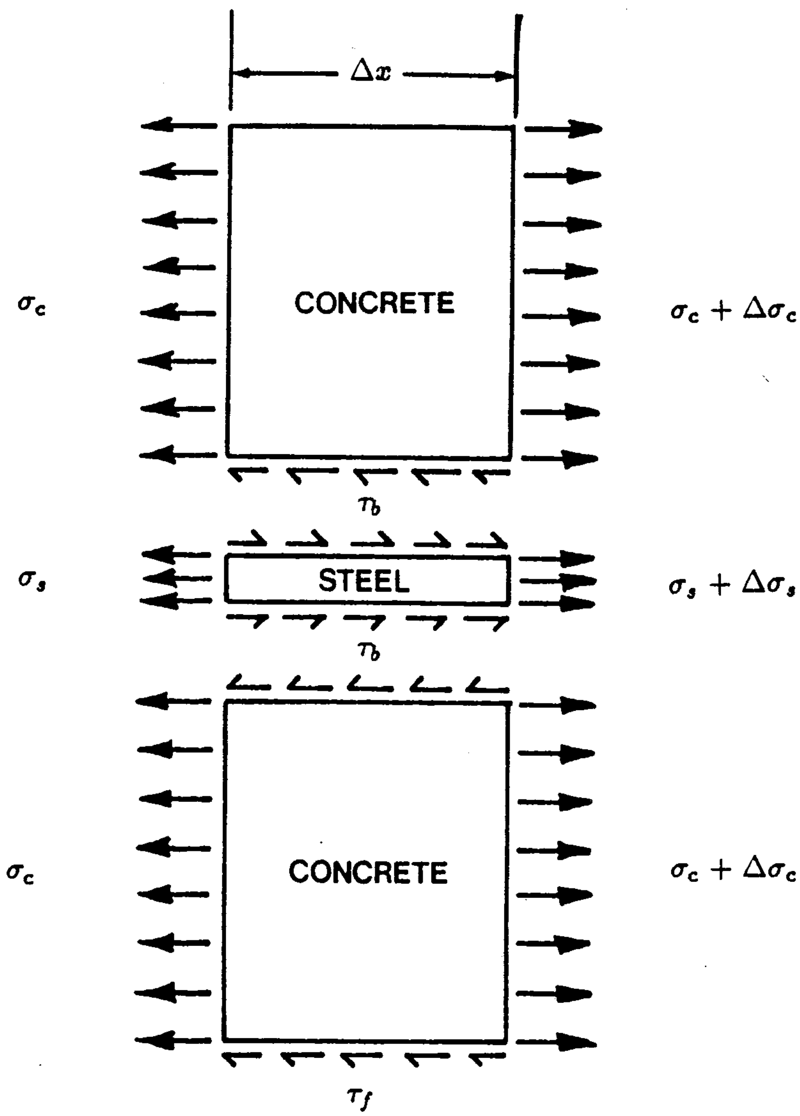


Figure 3.2. Stresses Acting Upon an Elemental Slice of the Prism.

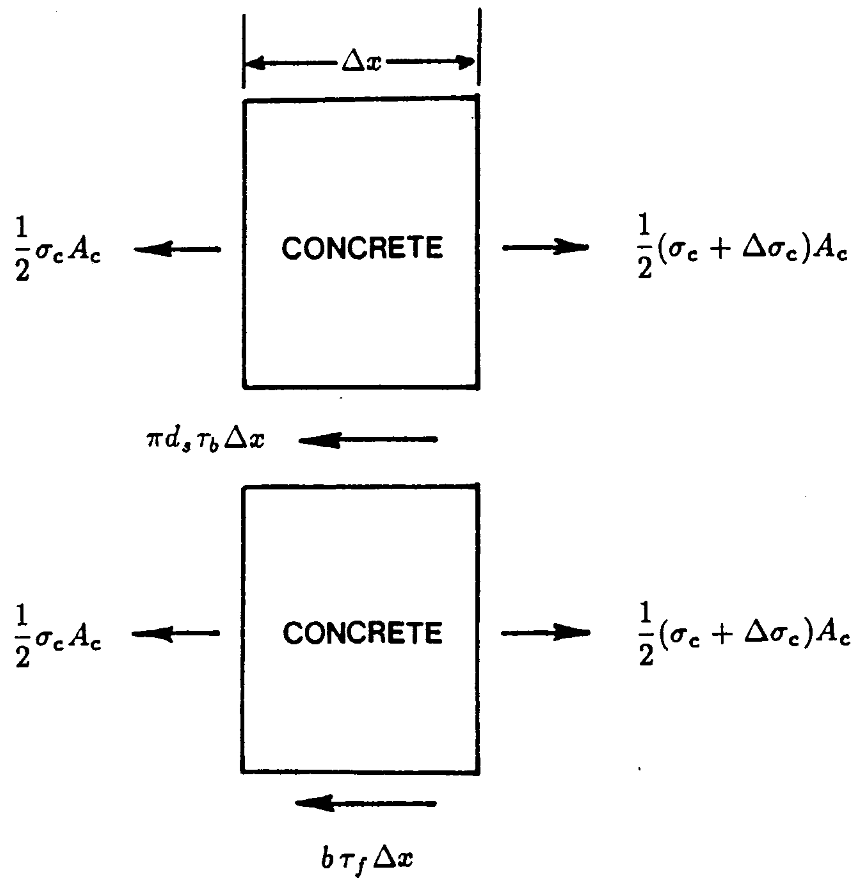


Figure 3.3a. Forces Acting on the Concrete.

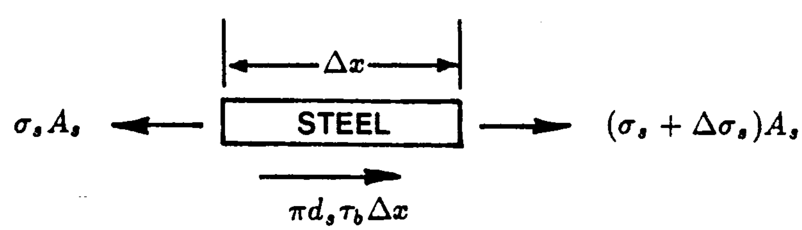


Figure 3.3b. Forces Acting on the Steel.

In order for static equilibrium to occur, the summation of forces must be set equal to zero.

$$(\sigma_c + \Delta\sigma_c)A_c - \sigma_c A_c - \pi d_s \tau_b \Delta x - b\tau_f \Delta x = 0 \tag{3.2}$$

This equation can be simplified to:

$$\frac{\Delta\sigma_c}{\Delta x} - \left[\frac{\pi d_s}{A_c}\right]\tau_b - \left[\frac{b}{A_c}\right]\tau_f = 0 \tag{3.3}$$

Looking at the steel in Figure 3.3b, a similar summation of forces can be set equal to zero.

$$(\sigma_s + \Delta\sigma_s)A_s - \sigma_s A_s + \pi d_s \tau_b \Delta x = 0 \tag{3.4}$$

where: A_s = the cross sectional area of the steel reinforcing, and the other variables are as previously defined.

This reduces to:

$$\frac{\Delta\sigma_s}{\Delta x} + \frac{\pi d_s}{A_s}\tau_b = 0 \tag{3.5}$$

The strain of the concrete, ϵ_c, is defined as:

$$\epsilon_c = \frac{du_c}{dx} \tag{3.6}$$

where: u_c = displacement of the concrete.

The stress in the concrete, σ_c, is equal to:

$$\sigma_c = E_c \epsilon_c \tag{3.7}$$

where: E_c = the elastic modulus of the concrete.

or:

$$\sigma_c = E_c \frac{du_c}{dx} \tag{3.8}$$

Differentiating Equation 3.8 once yields:

$$\frac{d\sigma_c}{dx} = E_c \frac{d^2 u_c}{dx^2} \tag{3.9}$$

By allowing $\frac{\Delta\sigma_c}{\Delta x}$ in Equation 3.3 to become sufficiently small, it can be replaced by $\frac{d\sigma_c}{dx}$. Equation 3.3 can then be rewritten as:

$$E_c \frac{d^2 u_c}{dx^2} - \left[\frac{\pi d_s}{A_c}\right] \tau_b - \left[\frac{b}{A_c}\right] \tau_f = 0 \tag{3.10}$$

which reduces to:

$$\frac{d^2 u_c}{dx^2} - \left[\frac{\pi d_s}{E_c A_c}\right] \tau_b - \left[\frac{b}{E_c A_c}\right] \tau_f = 0 \tag{3.11}$$

Similarly, for the steel:

$$\frac{d\sigma_s}{dx} = E_s \frac{d^2 u_s}{dx^2} \tag{3.12}$$

where: u_s = the displacement of the steel, and
E_s = the elastic modulus of the steel.

By performing the same substitution as was used for the concrete, Equation 3.5 becomes:

$$\frac{d^2 u_s}{dx^2} + \left[\frac{\pi d_s}{E_s A_s}\right] \tau_b = 0 \tag{3.13}$$

Equations 3.11 and 3.13 are the general differential equations that govern the behavior of the prism. All that remains is to substitute the correct expressions for τ_b and τ_f and solve the differential equations for the displacements u_c and u_s. Both τ_b and τ_f are modeled by the three-part linear functions that were described in Chapter II and are shown again in Figures 3.4a and 3.4b. For τ_b, the function is described as:

$$\tau_b = \begin{cases} K_1(u_c - u_s), & \text{if } 0 \le (u_c - u_s) \le \delta_b \\ C_1 + K_2(u_c - u_s), & \text{if } \delta_b \le (u_c - u_s) \le \delta_{bl} \\ 0, & \text{if } (u_c - u_s) > \delta_{bl} \end{cases} \tag{3.14}$$

Similarly, the function for τ_f can be expressed as:

$$\tau_f = \begin{cases} K_3 u_c, & \text{if } 0 \le u_c \le \delta_f \\ C_2 + K_4 u_c, & \text{if } \delta_f \le u_c \le \delta_{fl} \\ 0, & \text{if } u_c > \delta_{fl} \end{cases} \tag{3.15}$$

Combining the Stress Functions

By simple multiplication, it is seen that there are 9 different possible combinations the two stress functions can have with respect to each other, and each combination causes the general differential equations to have a different solution. The controlling factors are the displacements of the concrete and of the steel. It is also possible for several different combinations to be needed to describe the complete behavior of the prism from the crack face to the center of the slab.

Which combinations are necessary depends not only on the stress functions themselves, but also upon the magnitude of the thermal and shrinkage loading that the prism is subjected to. As the magnitude of the loading increases, the number of different combinations necessary to describe the behavior increases from one to as many as five. Each

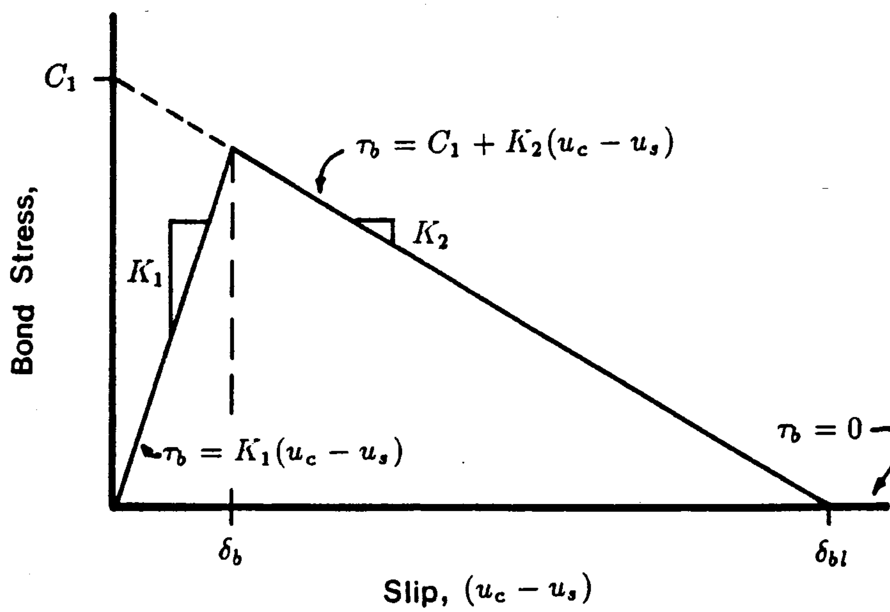


Figure 3.4a. The Bond Stress Function.

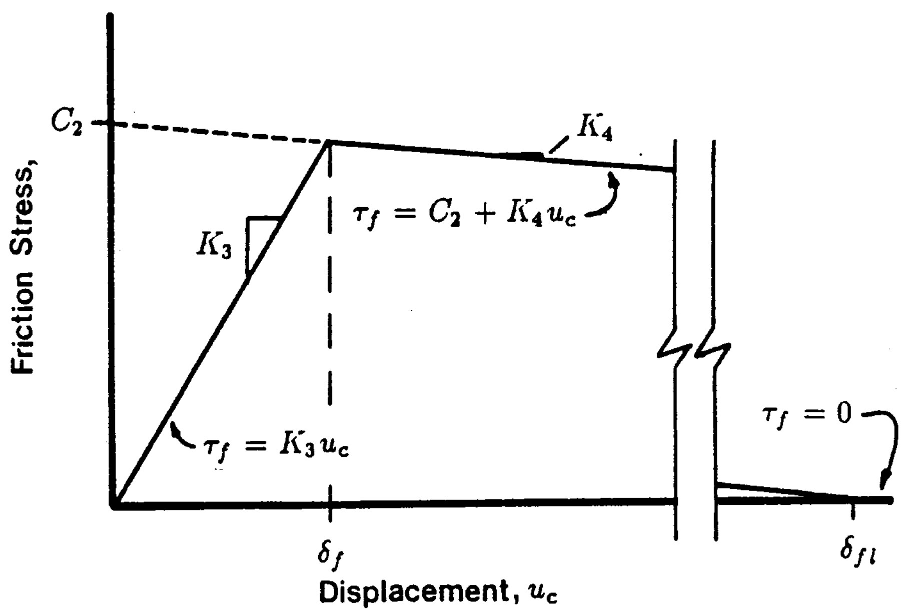


Figure 3.4b. The Friction Stress Function.

unique set of combinations is termed a "case." For the expected range of loading and the bond stress and friction stress functions that normally occur in CRCP, there are 9 different cases that the prism may be in. In Figure 3.5, each of the cases is shown along with its typical bond stress and friction stress distributions. The distinguishing characteristics between each case are the locations of the interfaces between the combinations, indicated by the dimensions ℓ_1, ℓ_2, and ℓ_3.

The length ℓ_1 is the distance from the crack to the point at which the relative slip between the steel reinforcing and the concrete, $(u_c - u_s)$, is equal to δ_b, which is the slip at which the maximum bond stress occurs. The length ℓ_2 corresponds to the point at which the relative slip between the steel and the concrete is equal to δ_{bl}, the slip at which the bond stress has just decreased to zero. The length ℓ_3 corresponds to the point at which the displacement of the concrete, u_c, is equal to δ_f, the displacement at which the maximum friction stress occurs.

Case 1 only has one zone (i.e. only one combination is used) and the magnitudes of the slips are less than δ_b and δ_f. Case 2 has two zones, since the relative slip between the steel and the concrete has exceeded δ_b for a distance ℓ_1 from the center of the crack. Case 3 has three zones and is a continuation of Case 2, since the slip between the concrete and the steel has also increased beyond δ_{bl} for a distance equal to ℓ_2.

Case 4 also has three zones, and is similar to Case 2 except that the concrete has displaced more than δ_f for a distance equal to ℓ_3, but ℓ_3 is less than ℓ_1. Case 5 is a continuation of Case 3 that has four zones because the concrete has again displaced more than δ_f. Case 6 is a continuation of Case 4 or Case 5 that has ℓ_1 greater than ℓ_3, which is greater than ℓ_2.

Case 7 has only two zones and is a special case where the concrete slip has exceeded δ_f, but the relative slip between the steel and the concrete is still less than δ_b. Case 7 is rare. Case 8 is similar to Case 4, except that now ℓ_3 is greater than ℓ_1. Case 9 can be a

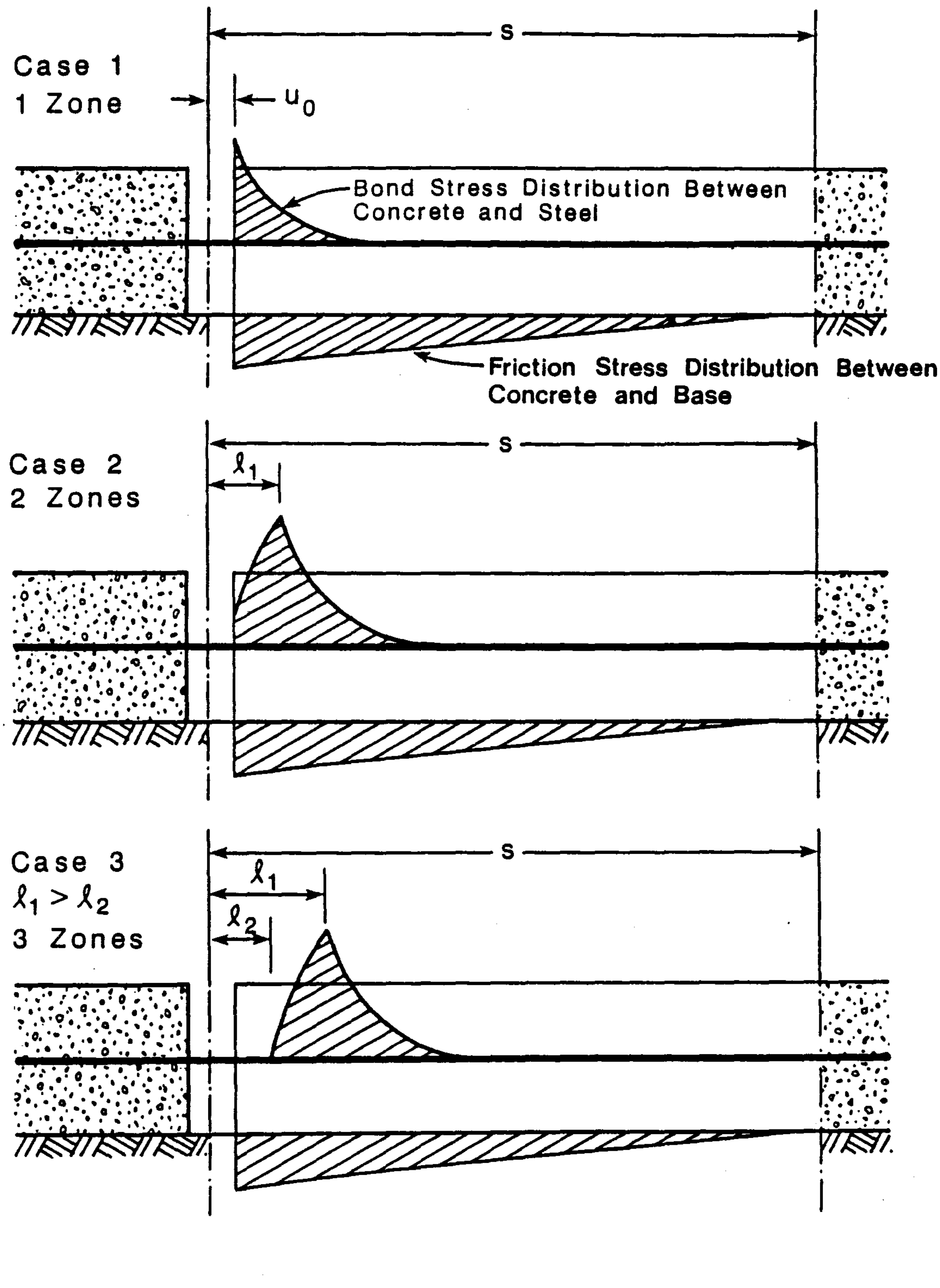


Figure 3.5a. Cases Describing the Different Combinations of the Bond Stress and Friction Stress Distributions. Cases 1 through 3.

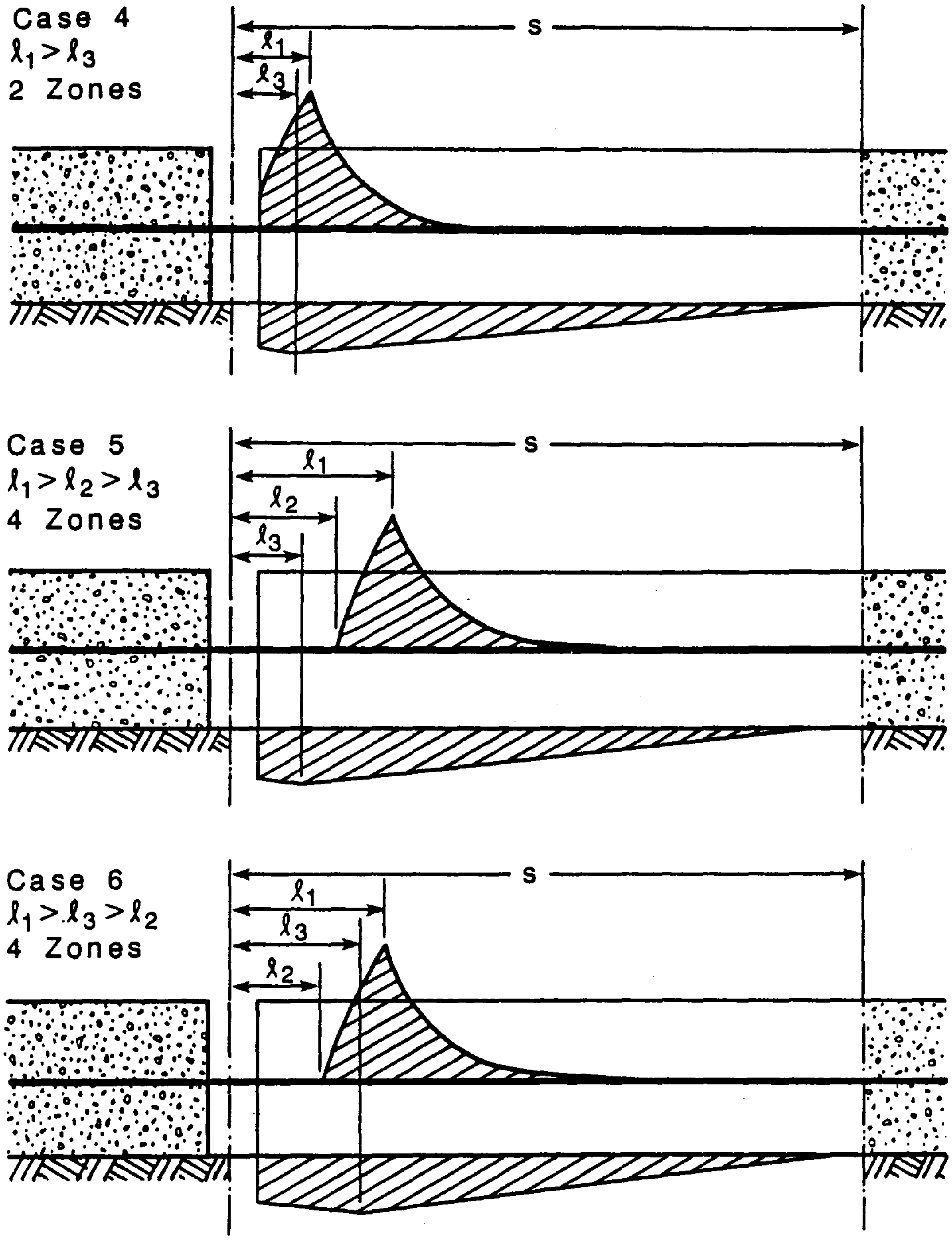


Figure 3.5b. Cases Describing the Different Combinations of the Bond Stress and Friction Stress Distributions. Cases 4 through 6.

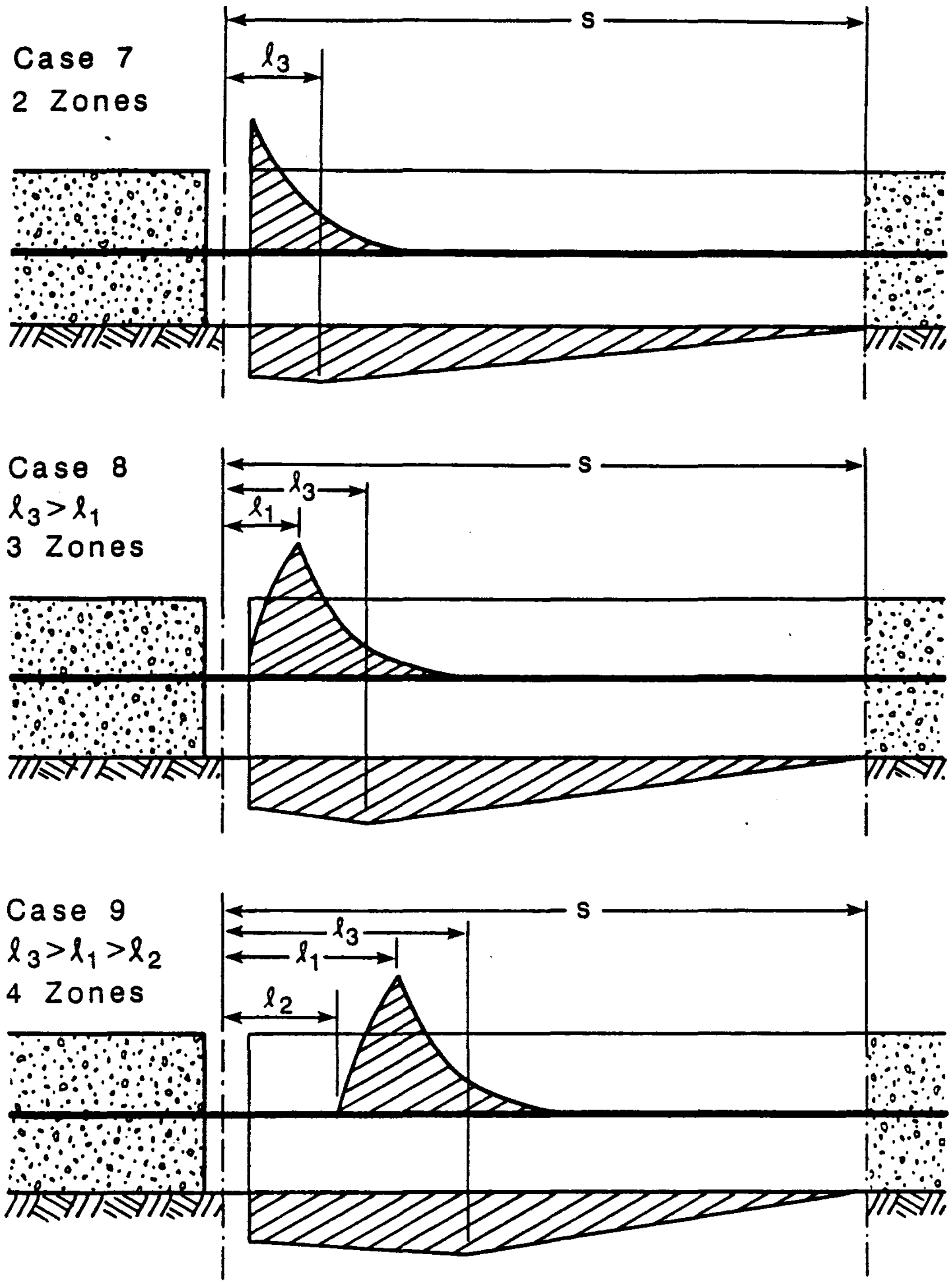


Figure 3.5c. Cases Describing the Different Combinations of the Bond Stress and Friction Stress Distributions. Cases 7 through 9.

continuation of either Case 6 or Case 8, with ℓ_3 greater than ℓ_1, which is greater than ℓ_2.

As the magnitude of the combined thermal and shrinkage loading increases, the prism's final stress state will progress from Case 1 through to Case 9. The shape of the friction stress function theoretically allows additional cases beyond Case 9 to occur. However, if the function is a realistic model of the actual behavior between the concrete and the base material, it is extremely doubtful that any case beyond Case 9 will occur, and for this reason, these additional cases were not considered.

SOLUTION OF THE CASE 6 EQUATION SYSTEM

The solution of the Case 6 equations will be presented to demonstrate the technique that was used to solve all 9 cases. General solutions for every case, with all necessary variables defined, are listed in Appendix A. Case 6 has four zones, as can be seen in Figure 3.6. The solution procedure will begin with Zone 1 and work through to Zone 4, following with the method of linking the zones together and solving for the coefficients.

Zone 1

In Zone 1, the bond stress function is:

$$\tau_b = K_1(u_c - u_s) \tag{3.16}$$

and the friction stress function is:

$$\tau_f = K_3 u_c \tag{3.17}$$

Substituting Equations 3.16 and 3.17 into Equation 3.11 obtains:

$$\frac{d^2 u_c}{dx^2} - \left[\frac{\pi d_s}{E_c A_c}\right](K_1(u_c - u_s)) - \left[\frac{b}{E_c A_c}\right](K_3 u_c) = 0 \tag{3.18}$$

which reduces to:

$$\frac{d^2u_c}{dx^2} - \left[\frac{K_1\pi d_s + K_3 b}{E_c A_c}\right] u_c + \left[\frac{K_1\pi d_s}{E_c A_c}\right] u_s = 0 \qquad (3.19)$$

Substituting Equations 3.16 into Equation 3.13 obtains:

$$\frac{d^2u_s}{dx^2} + \left[\frac{\pi d_s}{E_s A_s}\right] (K_1(u_c - u_s)) = 0 \qquad (3.20)$$

which reduces to:

$$\frac{d^2u_s}{dx^2} + \left[\frac{K_1\pi d_s}{E_s A_s}\right] u_c - \left[\frac{K_1\pi d_s}{E_s A_s}\right] u_s = 0 \qquad (3.21)$$

Equations 3.19 and 3.21 are the governing differential equations for Zone 1. Each of the equations involves two dependent variables, u_c and u_s, and an independent variable, x, and they constitute a system of simultaneous linear differential equations. The procedure of solution that will be followed is that which is outlined by Feineman et al. (__).

In operator form, the equations can be written as:

$$(D^2 - a_1)u_s \quad + \quad a_1 u_c \quad = 0 \qquad (3.22a)$$
$$d_1 u_s \quad + \quad (D^2 - c_1)u_c \quad = 0 \qquad (3.22b)$$

where: D = differential operator

$$a_1 = \left[\frac{K_1\pi d_s}{E_s A_s}\right]$$

$$c_1 = \left[\frac{K_1\pi d_s + K_3 b}{E_c A_c}\right]$$

$$d_1 = \left[\frac{K_1\pi d_s}{E_c A_c}\right]$$

The characteristic determinant is:

$$f(D) = \begin{vmatrix} (D^2 - a_1) & a_1 \\ d_1 & (D^2 - c_1) \end{vmatrix} \qquad (3.23)$$

Expanded, this becomes:

$$f(D) = D^4 - (a_1 + c_1)D^2 + (a_1 c_1 - a_1 d_1) \qquad (3.24)$$

The degree of the differential operator, D, is 4, which indicates that there will be four independent arbitrary constants in the complementary functions.

The auxiliary equation is:

$$f(m) = m^4 - (a_1 + c_1)m^2 + (a_1 c_1 - a_1 d_1) = 0 \qquad (3.24)$$

An evaluation of the constants shows that:

1. $a_1 > c_1$
2. $c_1 > d_1$
3. $(a_1 + c_1) > 0$, and
4. $(a_1 c_1 - a_1 d_1) > 0$

The roots of the auxiliary equation are:

$$m = \pm\sqrt{\frac{(a_1 + c_1)}{2} \pm \frac{(a_1 + c_1)}{2}\sqrt{1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2}}} \qquad (3.25)$$

where: $\sqrt{1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2}} < 1$

All of the roots are real, and can be written explicitly as:

$$g_{61} = m_1 = \left(\frac{(a_1 + c_1)}{2} + \frac{(a_1 + c_1)}{2}\left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}} \qquad (3.26)$$

$$-g_{61} = m_2 = -m_1 \tag{3.27}$$

$$h_{61} = m_3 = \left(\frac{(a_1 + c_1)}{2} - \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}} \tag{3.28}$$

$$-h_{61} = m_4 = -m_3 \tag{3.29}$$

Please note that the roots m_1 through m_4 will now be referred to as g_{61}, $-g_{61}$, h_{61}, and $-h_{61}$ to avoid confusion with other zones.

It follows that the complementary functions are:

$$u_{cc} = A_{c61} e^{g_{61} x} + B_{c61} e^{-g_{61} x} + C_{c61} e^{h_{61} x} + D_{c61} e^{-h_{61} x} \tag{3.30}$$

and:

$$u_{sc} = A_{s61} e^{g_{61} x} + B_{s61} e^{-g_{61} x} + C_{s61} e^{h_{61} x} + D_{s61} e^{-h_{61} x} \tag{3.31}$$

The constants $(A_{c61}, \ldots, D_{s61})$ must be linked together so that the solutions satisfy the original equations. This is accomplished by setting the original system of equations equal to zero so that:

$$(D^2 - a_1) u_{sc} + a_1 u_{cc} = 0 \tag{3.32a}$$

$$d_1 u_{sc} + (D^2 - c_1) u_{cc} = 0 \tag{3.32b}$$

By performing the indicated operations on Equation 3.32a, Equation 3.33 is obtained.

$$\left((g_{61}^2 - a_1) A_{s61} + a_1 A_{c61}\right) e^{g_{61} x} + \left((g_{61}^2 - a_1) B_{s61} + a_1 B_{c61}\right) e^{-g_{61} x} + \left((h_{61}^2 - a_1) C_{s61} + a_1 C_{c61}\right) e^{h_{61} x} + \left((h_{61}^2 - a_1) D_{s61} + a_1 D_{c61}\right) e^{-h_{61} x} = 0 \tag{3.33}$$

For this to be true for all values of x:

$$(g_{61}^2 - a_1) A_{s61} + a_1 A_{c61} = 0 \quad (3.34a)$$

$$(g_{61}^2 - a_1) B_{s61} + a_1 B_{c61} = 0 \quad (3.34b)$$

$$(h_{61}^2 - a_1) C_{s61} + a_1 C_{c61} = 0 \quad (3.34c)$$

$$(h_{61}^2 - a_1) D_{s61} + a_1 D_{c61} = 0 \quad (3.34d)$$

It follows that:

$$A_{s61} = \left[a_1/(a_1 - g_{61}^2)\right] A_{c61} \quad (3.35a)$$

$$B_{s61} = \left[a_1/(a_1 - g_{61}^2)\right] B_{c61} \quad (3.35b)$$

$$C_{s61} = \left[a_1/(a_1 - h_{61}^2)\right] C_{c61} \quad (3.35c)$$

$$D_{s61} = \left[a_1/(a_1 - h_{61}^2)\right] D_{c61} \quad (3.35d)$$

If Equation 3.32b is used, identical results will be obtained, although the constants that are used will be different.

For simplicity:

$$r_1 = \left[a_1/(a_1 - g_{61}^2)\right] \quad (3.36)$$

and:

$$s_1 = \left[a_1/(a_1 - h_{61}^2)\right] \quad (3.37)$$

The general solutions for Case 6, Zone 1 are:

$$u_{c61} = A_{c61} e^{g_{61} x} + B_{c61} e^{-g_{61} x} + C_{c61} e^{h_{61} x} + D_{c61} e^{-h_{61} x} \quad (3.38)$$

and:

$$u_{s61} = r_1 A_{c61} e^{g_{61} x} + r_1 B_{c61} e^{-g_{61} x} + s_1 C_{c61} e^{h_{61} x} + s_1 D_{c61} e^{-h_{61} x} \quad (3.39)$$

Zone 2

In Zone 2, the friction stress function remains the same as in Zone 1, but the bond stress function is now:

$$\tau_b = C_1 + K_2(u_c - u_s) \quad (3.40)$$

The governing differential equations in Zone 2 are:

$$\frac{d^2u_c}{dx^2} - \left[\frac{K_2\pi d_s + K_3 b}{E_c A_c}\right] u_c + \left[\frac{K_2\pi d_s}{E_c A_c}\right] u_s = \left[\frac{C_1\pi d_s}{E_c A_c}\right] \qquad (3.41)$$

$$\frac{d^2u_s}{dx^2} + \left[\frac{K_2\pi d_s}{E_s A_s}\right] u_c - \left[\frac{K_2\pi d_s}{E_s A_s}\right] u_s = \left[\frac{-C_1\pi d_s}{E_s A_s}\right] \qquad (3.42)$$

In operator form, the equations are written as:

$$(D^2 - a_2)u_s + a_2 u_c = b_2 \qquad (3.43a)$$

$$d_2 u_s + (D^2 - c_2)u_c = e_2 \qquad (3.43b)$$

where: $a_2 = \left[\frac{K_2\pi d_s}{E_s A_s}\right]$

$b_2 = \left[\frac{-C_1\pi d_s}{E_s A_s}\right]$

$c_2 = \left[\frac{K_2\pi d_s + K_3 b}{E_c A_c}\right]$

$d_2 = \left[\frac{K_2\pi d_s}{E_c A_c}\right]$

$e_2 = \left[\frac{C_1\pi d_s}{E_c A_c}\right]$

The auxiliary equation is:

$$f(m) = m^4 - (a_2 + c_2)m^2 + (a_2 c_2 - a_2 d_2) = 0 \qquad (3.44)$$

with

1. $(a_2 + c_2) < 0$
2. $\sqrt{1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2}} > 1$

The roots of the auxiliary equation are:

$$g_{62} = m_1 = \left(\frac{(a_2 + c_2)}{2} - \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}} \qquad (3.45)$$

$$-g_{62} = m_2 = -m_1 \qquad (3.46)$$

$$m_3 = i \left(\frac{-(a_2 + c_2)}{2} - \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}} \qquad (3.47)$$

$$m_4 = -m_3 \qquad (3.48)$$

Unlike zone 1, two of the roots are imaginary, m_3 and m_4. By letting:

$$h_{62} = m_3 / i \qquad (3.49)$$

the complimentary functions will be:

$$u_{cc} = A_{c62} e^{g_{62} x} + B_{c62} e^{-g_{62} x} + C_{c62} e^{h_{62} ix} + D_{c62} e^{-h_{62} ix} \qquad (3.50)$$

$$u_{sc} = A_{s62} e^{g_{62} x} + B_{s62} e^{-g_{62} x} + C_{s62} e^{h_{62} ix} + D_{s62} e^{-h_{62} ix} \qquad (3.51)$$

To reduce the functions to a useable form requires a little work and a few substitutions. First, it is known that:

$$e^{h_{62} ix} = \cos h_{62} x + i \sin h_{62} x \qquad (3.52)$$

and:

$$e^{-h_{62} ix} = \cos h_{62} x - i \sin h_{62} x \qquad (3.53)$$

Second, C_{c62} and D_{c62} are complex numbers and it will be assumed that:

$$C_{c62} = (a + bi) \qquad (3.54)$$

and:

$$D_{c62} = (c + di) \tag{3.55}$$

Substituting Equations 3.52 through 3.55 into Equation 3.50 gives:

$$\begin{aligned} u_{cc} = {} & A_{c62}\, e^{g_{62}x} + B_{c62}\, e^{-g_{62}x} + \\ & (a + bi)(\cos h_{62}x + i \sin h_{62}x) + \\ & (c + di)(\cos h_{62}x - i \sin h_{62}x) \end{aligned} \tag{3.56}$$

which becomes:

$$\begin{aligned} u_{cc} = {} & A_{c62}\, e^{g_{62}x} + B_{c62}\, e^{-g_{62}x} + \\ & a\cos h_{62}x + i\, a\sin h_{62}x + i\, b\cos h_{62}x - b\sin h_{62}x + \\ & c\cos h_{62}x - i\, c\sin h_{62}x + i\, d\cos h_{62}x + d\sin h_{62}x \end{aligned} \tag{3.57}$$

Since it is known that u_c is a real valued function, the imaginary portions will be cancelled out and like terms combined to get:

$$\begin{aligned} u_{cc} = {} & A_{c62}\, e^{g_{62}x} + B_{c62}\, e^{-g_{62}x} + \\ & (a + c)\, \cos h_{62}x + (d - b)\, \sin h_{62}x \end{aligned} \tag{3.58}$$

By redefining the constants to be:

$$C_{c62} = (a + c) \tag{3.59}$$

and:

$$D_{c62} = (d - b) \tag{3.60}$$

It is seen that:

$$u_{cc} = A_{c62}\, e^{g_{62}x} + B_{c62}\, e^{-g_{62}x} + C_{c62} \cos h_{62}x + D_{c62} \sin h_{62}x \tag{3.61}$$

Similarly:

$$u_{sc} = A_{s62}\, e^{g_{62}x} + B_{s62}\, e^{-g_{62}x} + C_{s62} \cos h_{62}x + D_{s62} \sin h_{62}x \tag{3.62}$$

The constants are linked together by the same method used in zone 1, which yields:

$$r_2 = \left[a_2/(a_2 - g_{62}^2)\right] \tag{3.63}$$

$$s_2 = \left[a_2/(a_2 + h_{62}^2)\right] \tag{3.64}$$

and:

$$A_{s62} = r_2 A_{c62} \tag{3.65}$$

$$B_{s62} = r_2 B_{c62} \tag{3.66}$$

$$C_{s62} = s_2 C_{c62} \tag{3.67}$$

$$D_{s62} = s_2 D_{c62} \tag{3.68}$$

The Zone 2 equations are nonhomogeneous, which requires that particular integrals can be evaluated and added to the complementary functions to obtain the complete general solutions. Substituting the particular integrals, u_{cp} and u_{sp}, into Equations 3.43a and 3.43b gives:

$$(D^2 - a_2) u_{sp} \quad + \quad a_2 u_{cp} \quad = b_2 \tag{3.69}$$

$$d_2 u_{sp} \quad + \quad (D^2 - c_2) u_{cp} \quad = e_2 \tag{3.70}$$

which reduces to:

$$-a_2 u_{sp} \quad + \quad a_2 u_{cp} \quad = b_2 \tag{3.71}$$

$$d_2 u_{sp} \quad + \quad -c_2 u_{cp} \quad = e_2 \tag{3.72}$$

By solving simultaneously:

$$u_{cp} = \left[\frac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)} \right] \tag{3.73}$$

and:

$$u_{sp} = \left[\frac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)} \right] \tag{3.74}$$

The complete general solutions for Zone 2 are:

$$u_{c62} = A_{c62} e^{g_{62} x} + B_{c62} e^{-g_{62} x} + C_{c62} \cos h_{62} x + D_{c62} \sin h_{62} x + \left[\frac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)} \right] \tag{3.75}$$

$$u_{s62} = r_2 A_{c62} e^{g_{62} x} + r_2 B_{c62} e^{-g_{62} x} + s_2 C_{c62} \cos h_{62} x + s_2 D_{c62} \sin h_{62} x + \left[\frac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)} \right] \tag{3.76}$$

Zone 3

In Zone 3, the bond stress function remains the same as in Zone 2, but the friction stress function is now:

$$\tau_f = C_2 + K_4 u_c \tag{3.77}$$

This causes the complete general solutions for Zone 3, which are found by the same method as used for Zone 2, to be:

$$u_{c63} = A_{c63} \cos g_{63} x + B_{c63} \sin g_{63} x + C_{c63} \cos h_{63} x + D_{c63} \sin h_{63} x + \left[\frac{b_3 d_3 + a_3 e_3}{a_3 (d_3 - c_3)} \right] \tag{3.78}$$

$$u_{s63} = r_3 A_{c63} \cos g_{63} x + r_3 B_{c63} \sin g_{63} x + s_3 C_{c63} \cos h_{63} x + s_3 D_{c63} \sin h_{63} x + \left[\frac{a_3 e_3 + b_3 c_3}{a_3 (d_3 - c_3)} \right] \tag{3.79}$$

where: $$a_3 = \left[\frac{K_2 \pi d_s}{E_s A_s} \right]$$

$$b_3 = \left[\frac{-C_1 \pi d_s}{E_s A_s} \right]$$

$$c_3 = \left[\frac{K_2 \pi d_s + K_4 b}{E_c A_c} \right]$$

$$d_3 = \left[\frac{K_2 \pi d_s}{E_c A_c} \right]$$

$$e_3 = \left[\frac{C_1 \pi d_s + C_2 b}{E_c A_c} \right]$$

$$r_3 = \left[a_3/(a_3 + g_{63}^2)\right]$$

$$s_3 = \left[a_3/(a_3 + h_{63}^2)\right]$$

Zone 4

In Zone 4, the friction stress function remains the same as in Zone 3, but the bond stress function is now:

$$\tau_b = 0 \tag{3.80}$$

which eliminates the steel displacement equation from the system of equations. For the concrete, the governing equation is:

$$\frac{d^2 u_c}{dx^2} - a_4 u_c = b_4 \tag{3.81}$$

where: $a_4 = \left[\frac{K_4 b}{E_c A_c}\right]$

$$b_4 = \left[\frac{C_2 b}{E_c A_c}\right]$$

If we let:

$$g_{64} = \sqrt{-a_4} \tag{3.82}$$

then the complete general solution for zone 4 is easily seen to be:

$$u_{c64} = A_{c64} \cos g_{64} x + B_{c64} \sin g_{64} x - \left[\frac{b_4}{a_4}\right] \tag{3.83}$$

In summary, the general solutions for all of the zones in Case 6 are:

Zone 1:

$$u_{c61} = A_{c61} e^{g_{61} x} + B_{c61} e^{-g_{61} x} + C_{c61} e^{h_{61} x} + D_{c61} e^{-h_{61} x} \tag{3.38}$$

$$u_{s61} = r_1 A_{c61} e^{g_{61} x} + r_1 B_{c61} e^{-g_{61} x} + s_1 C_{c61} e^{h_{61} x} + s_1 D_{c61} e^{-h_{61} x} \tag{3.39}$$

Zone 2:

$$u_{c62} = A_{c62} e^{g_{62} x} + B_{c62} e^{-g_{62} x} + C_{c62} \cos h_{62} x + D_{c62} \sin h_{62} x + \left[\frac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)} \right] \tag{3.75}$$

$$u_{s62} = r_2 A_{c62} e^{g_{62} x} + r_2 B_{c62} e^{-g_{62} x} + s_2 C_{c62} \cos h_{62} x + s_2 D_{c62} \sin h_{62} x + \left[\frac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)} \right] \tag{3.76}$$

Zone 3:

$$u_{c63} = A_{c63} \cos g_{63} x + B_{c63} \sin g_{63} x + C_{c63} \cos h_{63} x + D_{c63} \sin h_{63} x + \left[\frac{b_3 d_3 + a_3 e_3}{a_3 (d_3 - c_3)} \right] \tag{3.78}$$

$$u_{s63} = r_3 A_{c63} \cos g_{63} x + r_3 B_{c63} \sin g_{63} x + s_3 C_{c63} \cos h_{63} x + s_3 D_{c63} \sin h_{63} x + \left[\frac{a_3 e_3 + b_3 c_3}{a_3 (d_3 - c_3)} \right] \tag{3.79}$$

Zone 4:

$$u_{c64} = A_{c64} \cos g_{64} x + B_{c64} \sin g_{64} x - \left[\frac{b_4}{a_4} \right] \tag{3.83}$$

All that remains is to apply the boundary conditions to each of the above solutions and then solve for the exact values of the coefficients.

The Boundary Conditions

The general solutions of the Case 6 equations have a total of 14 unknown coefficients, which means there have to be 14 independent boundary conditions available to ensure that all of the coefficients can be found. In Figure 3.6, it is seen that at the midpoint of the slab, $x = S$, the displacements of both the concrete and the steel have to be equal to zero because of symmetry. This gives the first two boundary conditions.

(1) $\quad u_{c61} \Big|_{x = S} = 0$

(2) $\quad u_{s61} \Big|_{x = S} = 0$

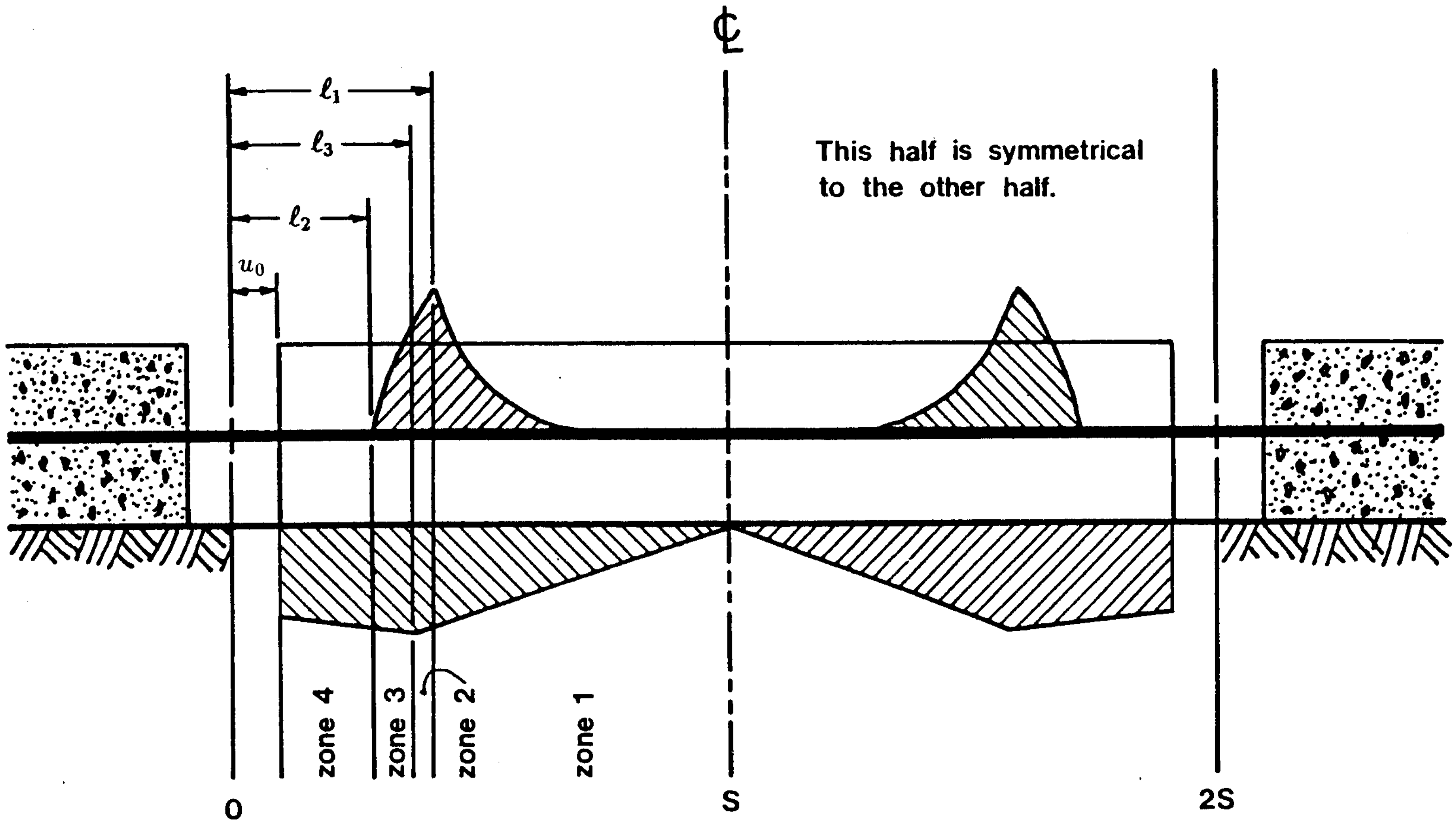


Figure 3.6. Description of Case 6.

Ten more boundary conditions are generated by the fact that the steel and concrete displacement functions are smooth and continuous over the length of the prism. This requires that at the interfaces between zones, the displacements are equal and the slopes of the displacements (the first derivatives of the displacement functions) are equal. These boundary conditions are:

(3) $u_{c61}\Big|_{x=\ell_1} = u_{c62}\Big|_{x=\ell_1}$ (4) $u_{s61}\Big|_{x=\ell_1} = u_{s62}\Big|_{x=\ell_1}$

(5) $\dfrac{du_{c61}}{dx}\Big|_{x=\ell_1} = \dfrac{du_{c62}}{dx}\Big|_{x=\ell_1}$ (6) $\dfrac{du_{s61}}{dx}\Big|_{x=\ell_1} = \dfrac{du_{s62}}{dx}\Big|_{x=\ell_1}$

(7) $u_{c62}\Big|_{x=\ell_3} = u_{c63}\Big|_{x=\ell_3}$ (8) $u_{s62}\Big|_{x=\ell_3} = u_{s63}\Big|_{x=\ell_3}$

(9) $\dfrac{du_{c62}}{dx}\Big|_{x=\ell_3} = \dfrac{du_{c63}}{dx}\Big|_{x=\ell_3}$ (10) $\dfrac{du_{s62}}{dx}\Big|_{x=\ell_3} = \dfrac{du_{s63}}{dx}\Big|_{x=\ell_3}$

(11) $u_{c63}\Big|_{x=\ell_2} = u_{c64}\Big|_{x=\ell_2}$ (12) $\dfrac{du_{c63}}{dx}\Big|_{x=\ell_2} = \dfrac{du_{c64}}{dx}\Big|_{x=\ell_2}$

Again in Figure 3.6, it is seen that at $x = u_0$, the displacement of the concrete is equal to u_0, which is another boundary condition.

(13) $u_{c64}\Big|_{x=u_0} = u_0$

And finally, at the center of the crack, $x = 0$, the steel displacement is equal to zero and the steel strain is equal to ϵ_{st}. The steel strain is constant from $x = 0$ to $x = \ell_2$, which allows the steel displacement at $x = \ell_2$ to be equal to $\epsilon_{st}\ell_2$. This is the final boundary condition.

(14) $u_{s63}\Big|_{x=\ell_2} = \epsilon_{st}\ell_2$

Written out, the boundary conditions appear as:

(1) $$A_{c61}\,e^{g_{61}S} + B_{c61}\,e^{-g_{61}S} + C_{c61}\,e^{h_{61}S} + D_{c61}\,e^{-h_{61}S} = 0$$

(2) $$r_1 A_{c61}\,e^{g_{61}S} + r_1 B_{c61}\,e^{-g_{61}S} + s_1 C_{c61}\,e^{h_{61}S} + s_1 D_{c61}\,e^{-h_{61}S} = 0$$

(3) $$A_{c61}\,e^{g_{61}\ell_1} + B_{c61}\,e^{-g_{61}\ell_1} + C_{c61}\,e^{h_{61}\ell_1} + D_{c61}\,e^{-h_{61}\ell_1} = A_{c62}\,e^{g_{62}\ell_1} + B_{c62}\,e^{-g_{62}\ell_1} + C_{c62}\cos h_{62}\ell_1 + D_{c62}\sin h_{62}\ell_1 + \left[\frac{b_2 d_2 + a_2 e_2}{a_2(d_2 - c_2)}\right]$$

(4) $$r_1 A_{c61}\,e^{g_{61}\ell_1} + r_1 B_{c61}\,e^{-g_{61}\ell_1} + s_1 C_{c61}\,e^{h_{61}\ell_1} + s_1 D_{c61}\,e^{-h_{61}\ell_1} = r_2 A_{c62}\,e^{g_{62}\ell_1} + r_2 B_{c62}\,e^{-g_{62}\ell_1} + s_2 C_{c62}\cos h_{62}\ell_1 + s_2 D_{c62}\sin h_{62}\ell_1 + \left[\frac{a_2 e_2 + b_2 c_2}{a_2(d_2 - c_2)}\right]$$

(5) $$g_{61} A_{c61}\,e^{g_{61}\ell_1} - g_{61} B_{c61}\,e^{-g_{61}\ell_1} + h_{61} C_{c61}\,e^{h_{61}\ell_1} - h_{61} D_{c61}\,e^{-h_{61}\ell_1} = g_{62} A_{c62}\,e^{g_{62}\ell_1} - g_{62} B_{c62}\,e^{-g_{62}\ell_1} - h_{62} C_{c62}\sin h_{62}\ell_1 + h_{62} D_{c62}\cos h_{62}\ell_1$$

(6) $$r_1 g_{61} A_{c61}\,e^{g_{61}\ell_1} - r_1 g_{61} B_{c61}\,e^{-g_{61}\ell_1} + s_1 h_{61} C_{c61}\,e^{h_{61}\ell_1} - s_1 h_{61} D_{c61}\,e^{-h_{61}\ell_1} = r_2 g_{62} A_{c62}\,e^{g_{62}\ell_1} - r_2 g_{62} B_{c62}\,e^{-g_{62}\ell_1} - s_2 h_{62} C_{c62}\sin h_{62}\ell_1 + s_2 h_{62} D_{c62}\cos h_{62}\ell_1$$

(7) $$A_{c62}\,e^{g_{62}\ell_3} + B_{c62}\,e^{-g_{62}\ell_3} + C_{c62}\cos h_{62}\ell_3 + D_{c62}\sin h_{62}\ell_3 = A_{c63}\cos g_{63}\ell_3 + B_{c63}\sin g_{63}\ell_3 + C_{c63}\cos h_{63}\ell_3 + D_{c63}\sin h_{63}\ell_3 + \left[\frac{b_3 d_3 + a_3 e_3}{a_3(d_3 - c_3)}\right] - \left[\frac{b_2 d_2 + a_2 e_2}{a_2(d_2 - c_2)}\right]$$

(8) $$r_2 A_{c62}\,e^{g_{62}\ell_3} + r_2 B_{c62}\,e^{-g_{62}\ell_3} + s_2 C_{c62}\cos h_{62}\ell_3 + s_2 D_{c62}\sin h_{62}\ell_3 = r_3 A_{c63}\cos g_{63}\ell_3 + r_3 B_{c63}\sin g_{63}\ell_3 + s_3 C_{c63}\cos h_{63}\ell_3 + s_3 D_{c63}\sin h_{63}\ell_3 + \left[\frac{a_3 e_3 + b_3 c_3}{a_3(d_3 - c_3)}\right] - \left[\frac{a_2 e_2 + b_2 c_2}{a_2(d_2 - c_2)}\right]$$

(9) $g_{62} A_{c62} e^{g_{62} \ell_3} - g_{62} B_{c62} e^{-g_{62} \ell_3} - h_{62} C_{c62} \sin h_{62} \ell_3 +$
$h_{62} D_{c62} \cos h_{62} \ell_3 = -g_{63} A_{c63} \sin g_{63} \ell_3 + g_{63} B_{c63} \cos g_{63} \ell_3 -$
$h_{63} C_{c63} \sin h_{63} \ell_3 + h_{63} D_{c63} \cos h_{63} \ell_3$

(10) $r_2 g_{62} A_{c62} e^{g_{62} \ell_3} - r_2 g_{62} B_{c62} e^{-g_{62} \ell_3} - s_2 h_{62} C_{c62} \sin h_{62} \ell_3 +$
$s_2 h_{62} D_{c62} \cos h_{62} \ell_3 = -r_3 g_{63} A_{c63} \sin g_{63} \ell_3 + r_3 g_{63} B_{c63} \cos g_{63} \ell_3 -$
$s_3 h_{63} C_{c63} \sin h_{63} \ell_3 + s_3 h_{63} D_{c63} \cos h_{63} \ell_3$

(11) $A_{c63} \cos g_{63} \ell_2 + B_{c63} \sin g_{63} \ell_2 + C_{c63} \cos h_{63} \ell_2 + D_{c63} \sin h_{63} \ell_2 =$
$A_{c64} \cos g_{64} \ell_2 + B_{c64} \sin g_{64} \ell_2 - \left[\frac{b_4}{a_4} \right] - \left[\frac{b_3 d_3 + a_3 e_3}{a_3 (d_3 - c_3)} \right]$

(12) $- g_{63} A_{c63} \sin g_{63} \ell_2 + g_{63} B_{c63} \cos g_{63} \ell_2 - h_{63} C_{c63} \sin h_{63} \ell_2 +$
$h_{63} D_{c63} \cos h_{63} \ell_2 = -g_{64} A_{c64} \sin g_{64} \ell_2 + g_{64} B_{c64} \cos g_{64} \ell_2$

(13) $A_{c64} \cos g_{64} u_0 + B_{c64} \sin g_{64} u_0 - \left[\frac{b_4}{a_4} \right] = u_0$

(14) $r_3 A_{c63} \cos g_{63} \ell_2 + r_3 B_{c63} \sin g_{63} \ell_2 +$
$s_3 C_{c63} \cos h_{63} \ell_2 + s_3 D_{c63} \sin h_{63} \ell_2 + \left[\frac{a_3 e_3 + b_3 c_3}{a_3 (d_3 - c_3)} \right] = \epsilon_{st} \ell_2$

The Matrix Method of Solution

The above system of linear equations must be solved simultaneously in order that the coefficients be found. This is most efficiently accomplished by matrix methods. Placing the above equations into an equivalent matrix form obtains:

$$[\mathbf{A}][\mathbf{E}] = [\mathbf{D}] \qquad (3.85)$$

where: $[\mathbf{A}]$ = a 14 by 14 matrix of constants

$[\mathbf{E}]$ = a 14 by 1 matrix of unknown coefficients, and

$[\mathbf{D}]$ = a 14 by 1 matrix of constants.

To find $[\mathbf{E}]$, the inverse of $[\mathbf{A}]$, $[\mathbf{A}]^{-1}$, must be found and then multiplied by $[\mathbf{D}]$, which is written as:

$$[\mathbf{E}] = [\mathbf{A}]^{-1}[\mathbf{D}] \tag{3.86}$$

The actual A matrix for Case 6 is too large to be written out as a whole, so 4 submatrices, $[\mathbf{A_1}]$ through $[\mathbf{A_4}]$, will be shown and used to represent the entire matrix.

$$[\mathbf{A_1}] = \begin{bmatrix} e^{g_{61}S} & e^{-g_{61}S} & e^{h_{61}S} & e^{-h_{61}S} \\ r_1 e^{g_{61}S} & r_1 e^{-g_{61}S} & s_1 e^{h_{61}S} & s_1 e^{-h_{61}S} \\ e^{g_{61}\ell_1} & e^{-g_{61}\ell_1} & e^{h_{61}\ell_1} & e^{-h_{61}\ell_1} \\ r_1 e^{g_{61}\ell_1} & r_1 e^{-g_{61}\ell_1} & s_1 e^{h_{61}\ell_1} & s_1 e^{-h_{61}\ell_1} \\ g_{61} e^{g_{61}\ell_1} & -g_{61} e^{-g_{61}\ell_1} & h_{61} e^{h_{61}\ell_1} & -h_{61} e^{-h_{61}\ell_1} \\ r_1 g_{61} e^{g_{61}\ell_1} & -r_1 g_{61} e^{-g_{61}\ell_1} & s_1 h_{61} e^{h_{61}\ell_1} & -s_1 h_{61} e^{-h_{61}\ell_1} \end{bmatrix}$$

$$
[\mathbf{A_2}] = \begin{bmatrix}
e^{g_{62}\ell_1} & e^{-g_{62}\ell_1} & \cos h_{62}\ell_1 & \sin h_{62}\ell_1 \\
r_2 e^{g_{62}\ell_1} & r_2 e^{-g_{62}\ell_1} & s_2 \cos h_{62}\ell_1 & s_2 \sin h_{62}\ell_1 \\
g_{62} e^{g_{62}\ell_1} & -g_{62} e^{-g_{62}\ell_1} & -h_{62} \sin h_{62}\ell_1 & h_{62} \cos h_{62}\ell_1 \\
r_2 g_{62} e^{g_{62}\ell_1} & -r_2 g_{62} e^{-g_{62}\ell_1} & -s_2 h_{62} \sin h_{62}\ell_1 & s_2 h_{62} \cos h_{62}\ell_1 \\
e^{g_{62}\ell_3} & e^{-g_{62}\ell_3} & \cos h_{62}\ell_3 & \sin h_{62}\ell_3 \\
r_2 e^{g_{62}\ell_3} & r_2 e^{-g_{62}\ell_3} & s_2 \cos h_{62}\ell_3 & s_2 \sin h_{62}\ell_3 \\
g_{62} e^{g_{62}\ell_3} & -g_{62} e^{-g_{62}\ell_3} & -h_{62} \sin h_{62}\ell_3 & h_{62} \cos h_{62}\ell_3 \\
r_2 g_{62} e^{g_{62}\ell_3} & -r_2 g_{62} e^{-g_{62}\ell_3} & -s_2 h_{62} \sin h_{62}\ell_3 & s_2 h_{62} \cos h_{62}\ell_3
\end{bmatrix}
$$

$$
[\mathbf{A_3}] = \begin{bmatrix}
\cos g_{63}\ell_3 & \sin g_{63}\ell_3 & \cos h_{63}\ell_3 & \sin h_{63}\ell_3 \\
r_3 \cos g_{63}\ell_3 & r_3 \sin g_{63}\ell_3 & s_3 \cos h_{63}\ell_3 & s_3 \sin h_{63}\ell_3 \\
-g_{63} \sin g_{63}\ell_3 & g_{63} \cos g_{63}\ell_3 & -h_{63} \sin h_{63}\ell_3 & h_{63} \cos h_{63}\ell_3 \\
-r_3 g_{63} \sin g_{63}\ell_3 & r_3 g_{63} \cos g_{63}\ell_3 & -s_3 h_{63} \sin h_{63}\ell_3 & s_3 h_{63} \cos h_{63}\ell_3 \\
\cos g_{63}\ell_2 & \sin g_{63}\ell_2 & \cos h_{63}\ell_2 & \sin h_{63}\ell_2 \\
-g_{63} \sin g_{63}\ell_2 & g_{63} \cos g_{63}\ell_2 & -h_{63} \sin h_{63}\ell_2 & h_{63} \cos h_{63}\ell_2 \\
0 & 0 & 0 & 0 \\
r_3 \cos g_{63}\ell_2 & r_3 \sin g_{63}\ell_2 & s_3 \cos h_{63}\ell_2 & s_3 \sin h_{63}\ell_2
\end{bmatrix}
$$

$$
[\mathbf{A_4}] = \begin{bmatrix}
\cos g_{64}\ell_2 & \sin g_{64}\ell_2 \\
-g_{64} \sin g_{64}\ell_2 & g_{64} \cos g_{64}\ell_2 \\
\cos g_{64}u_0 & \sin g_{64}u_0 \\
0 & 0
\end{bmatrix}
$$

The $[\mathbf{A}]$ matrix can now be represented as:

$$
[\mathbf{A}] = \left[\begin{array}{cccccccccccccc}
 & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & [\mathbf{A_1}] & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & & [\mathbf{A_2}] & & & & & & & 0 & 0 \\
0 & 0 & 0 & 0 & & & & & & & & & 0 & 0 \\
0 & 0 & 0 & 0 & & & & & & & & & 0 & 0 \\
0 & 0 & 0 & 0 & & & & & & & & & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & [\mathbf{A_3}] & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & [\mathbf{A_4}] & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & &
\end{array}\right]
$$

The $[\mathbf{D}]$ matrix for Case 6 is:

$$
[\mathbf{D}] = \begin{bmatrix}
0 \\
0 \\
\left[\dfrac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)}\right] \\
\left[\dfrac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)}\right] \\
0 \\
0 \\
\left[\dfrac{b_3 d_3 + a_3 e_3}{a_3 (d_3 - c_3)} - \dfrac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)}\right] \\
\left[\dfrac{a_3 e_3 + b_3 c_3}{a_3 (d_3 - c_3)} - \dfrac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)}\right] \\
0 \\
0 \\
\left[-\dfrac{b_4}{a_4} - \dfrac{b_3 d_3 + a_3 e_3}{a_3 (d_3 - c_3)}\right] \\
0 \\
\left[u_0 + \dfrac{b_4}{a_4}\right] \\
\left[\epsilon_{st} \ell_2 - \dfrac{a_3 e_3 + b_3 c_3}{a_3 (d_3 - c_3)}\right]
\end{bmatrix}
$$

The $[\mathbf{E}]$ matrix for Case 6 is:

$$[\mathbf{E}] = \begin{bmatrix} A_{c61} \\ B_{c61} \\ C_{c61} \\ D_{c61} \\ A_{c62} \\ B_{c62} \\ C_{c62} \\ D_{c62} \\ A_{c63} \\ B_{c63} \\ C_{c63} \\ D_{c63} \\ A_{c64} \\ B_{c64} \end{bmatrix}$$

The computations that are necessary to find the $[\mathbf{A}]^{-1}$ matrix are best performed by a computer, and thus will not be presented here. It is important to note that the method described above is used for every case by the computer model. Only the size and the individual elements of each matrix are different. Calculating the coefficients for the displacement functions is a fundamental step that may have to be repeated several hundred or thousand times in the overall procedure before the final and correct solution is determined. Chapter IV describes the method that is used to arrive at the correct solution.

CHAPTER IV

DETERMINING THE CORRECT AND UNIOUE SOLUTION

ENERGY CONSIDERATIONS

Two separate equilibrium conditions are used to determine the correct solution to the displacement functions. The first is a force equilibrium. The model assumes that the prism comes to rest after reacting to a given change in its environmental and internal system. Mathematically,the summation of forces acting upon the prism must equal zero for the prism to be at rest. This equilibrium condition is automatically satisfied by the displacement functions, since the original differential equations were derived by setting the summation of forces equal to zero. In addition, the prism must also satisfy an energy equilibrium. All of the energy that was available to displace the slab through a drop in temperature and drying shrinkage must be accounted for. In the model, energy can be used up or stored as:

1. Potential energy in the concrete and the steel
2. Frictional work energy lost during slab movement
3. Stress relief energy lost because of slab movement.

A summation of these three types of energy over the entire prism must equal the total thermal and shrinkage energy that is available to the slab. Since the force equilibrium is automatically satisfied by any solution of the displacement functions, the energy equilibrium is the only deciding criteria in finding the correct and unique solution for a given set of environmental and internal conditions.

Potential Energy

The potential energy stored in the concrete and the steel is quite readily determined if the stress states of the concrete and the steel are known. For a unit volume of concrete, the potential energy stored is equal to:

$$\mathbf{E}_{pot_c} = \int_0^{\epsilon_1} \sigma_c \, d\epsilon_c \tag{4.1}$$

Similarly, for a unit volume of steel:

$$\mathbf{E}_{pot_s} = \int_0^{\epsilon_2} \sigma_s \, d\epsilon_s \tag{4.2}$$

Figures 4.1a and 4.1b graphically show that the potential energy stored is equal to the area underneath the stress-strain curve. For the steel and the concrete the loading curve is linear in tension, which simplifies Eqs. 4.1 and 4.2 to:

$$\mathbf{E}_{pot_c} = \sigma_c^2/2E_c \tag{4.3}$$

$$\mathbf{E}_{pot_s} = \sigma_s^2/2E_s \tag{4.4}$$

since $\epsilon_c = \sigma_c/E_c$ and $\epsilon_s = \sigma_s/E_s$. At the point where there is no bond, the steel can possibly yield and the model does not apply in that case.

To determine the potential energy stored by the concrete and the steel in the prism it is necessary to integrate the above functions over the total volumes of the concrete and the steel respectively. Since the stress distributions in both the steel and the concrete only vary with respect to x, the total potential energy stored in the prism is equal to:

$$\mathbf{E}_{\text{pot}} = A_c \int_0^S \sigma_c^2/2E_c \, dx + A_s \int_0^S \sigma_s^2/2E_s \, dx \tag{4.5}$$

where S = the length of the prism.

Frictional Work Energy

The frictional work energy that is expended by the prism as it moves to its final state of displacement is again defined as an integration of the area beneath a set of curves. However, rather than the area underneath the stress-strain curves, it is the area beneath the bond stress and friction stress functions. Figures 4.2a and 4.2b

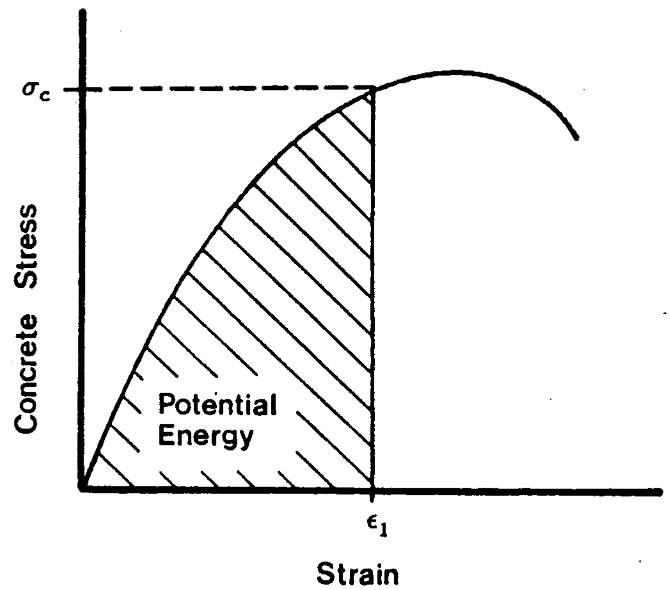


Figure 4.1a. Potential Energy Stored by Concrete at a Strain ϵ_1 .

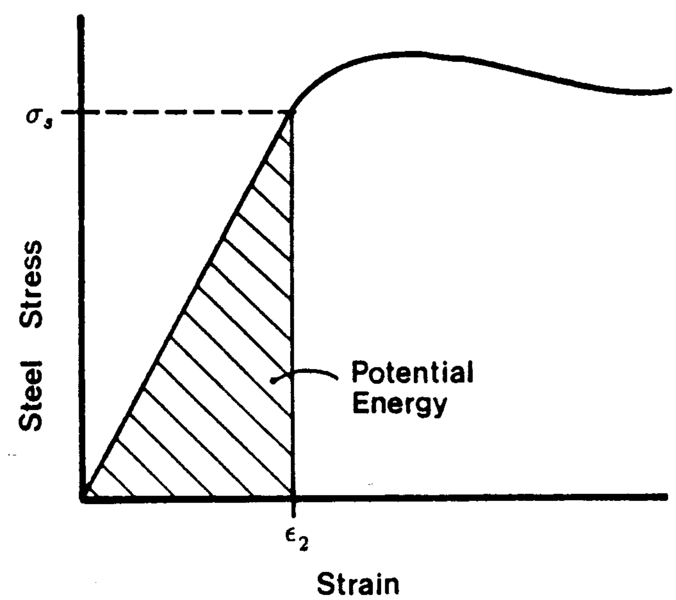


Figure 4.1b. Potential Energy Stored by Steel at a Strain ϵ_2 .

display this graphically. By integrating over the total contact area between the steel and the concrete, and the concrete and the base material, the total frictional work energy expended by the prism is seen to be :

$$\mathrm{E_{fric}} = \pi d_s \int_0^S \int_0^{u_c - u_s} \tau_b \, d\tau_b dx + b \int_0^S \int_0^{u_c} \tau_f \, d\tau_f \, dx \qquad (4.6)$$

Stress Relief Energy

Both potential energy and frictional work energy are commonly understood and easily quantified, but together they do not account for all of the energy that is stored or lost by the prism. A third type of energy, stress relief energy, has to be defined to account for the remainder of the energy available to the slab.

If an infinite amount of friction existed between the prism and the base material, as well as between the steel and the concrete inside the prism, the prism would not be able to move when subjected to thermal loading. All of the energy that is available to the prism because of the thermal loading would be stored as potential energy in the concrete and the steel. The stress levels in the concrete and the steel would be:

$$\sigma_c = E_c \alpha_c \Delta T \qquad (4.7)$$

$$\sigma_s = E_s \alpha_s \Delta T \qquad (4.8)$$

where: α_c = the coefficient of thermal contraction of the concrete

α_s = the coefficient of thermal contraction of the steel and,

ΔT = the temperature drop.

In addition, no frictional work energy would have been expended by the prism, since it was unable to move. If the extreme opposite case were considered, where there was no friction between the prism and the

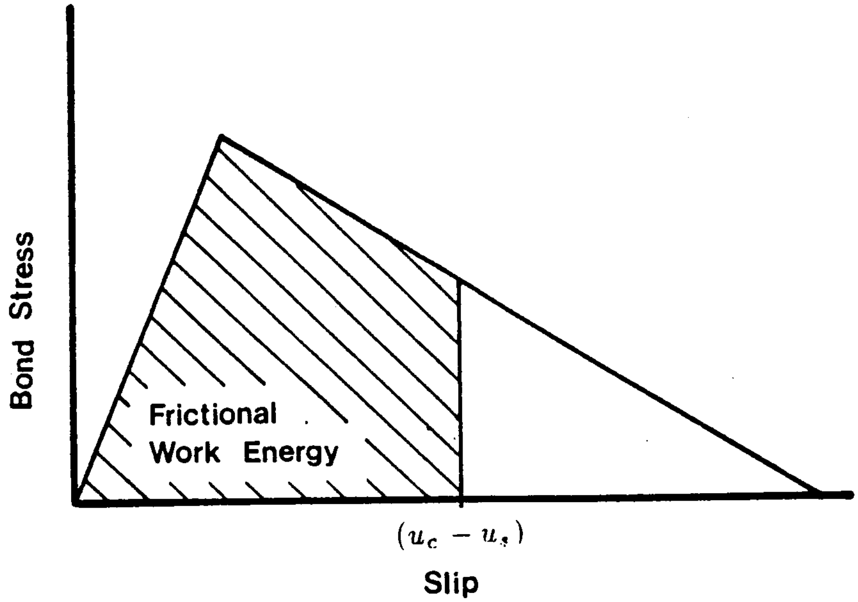


Figure 4.2a. Frictional Work Energy Expended Between the Steel and the Concrete to Achieve a Relative Slip of $(u_c - u_s)$.

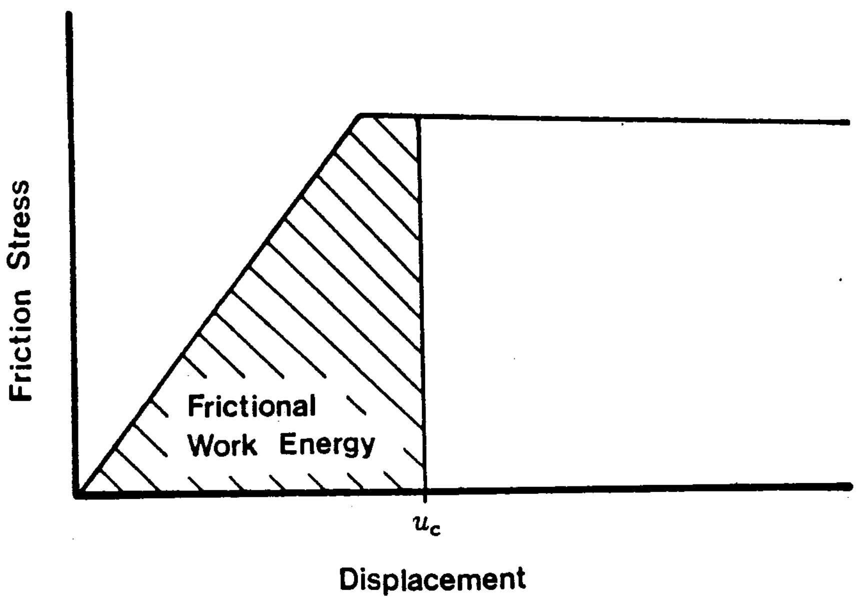


Figure 4.2b. Frictional Work Energy Expended Between the Concrete and the Base to Achieve a Displacement of u_c.

base material, or between the steel and the concrete, and the same thermal loading was applied to the prism, the concrete would contract. The strain in the concrete would be constant throughout its length and would equal:

$$\epsilon_c = \alpha_c \Delta T \tag{4.9}$$

The maximum displacement of the concrete would occur at the crack face and would be equal to:

$$u_{c_{max}} = S\alpha_c \Delta T \tag{4.10}$$

The stress level of the concrete would be equal to zero at all locations, meaning that no potential energy was stored by the concrete. The stress level of the steel would again be $\sigma_s = E_s \alpha_s \Delta T$, since the ends of the steel are rigidly anchored. No frictional work energy was expended by the prism because no friction or bond forces were generated by the displacement of the concrete. The potential energy that was stored in the steel was the same in both cases.

The potential energy that was stored by the concrete in the first case is equal in magnitude to the energy that was lost by the concrete as it moved in the second case. For a unit volume of concrete, the energy that was lost in the second case is equal to:

$$\mathbf{E}_{rel} = \int_0^{\epsilon_c} E_c \epsilon_c \, d\epsilon_c \tag{4.11}$$

For the purposes of this report, this energy is defined as stress relief energy, and as can be seen in Equation 4.11, it is a function of the concrete strain, ϵ_c. Figure 4.3 shows that the assumed linear elastic behavior of the concrete allows this relationship to be simplified to:

$$\mathbf{E}_{rel} = \frac{1}{2} E_c \epsilon_c^2 \tag{4.12}$$

In a case where a normal amount of friction occurs between the prism and the base material, as well as there being a normal bond

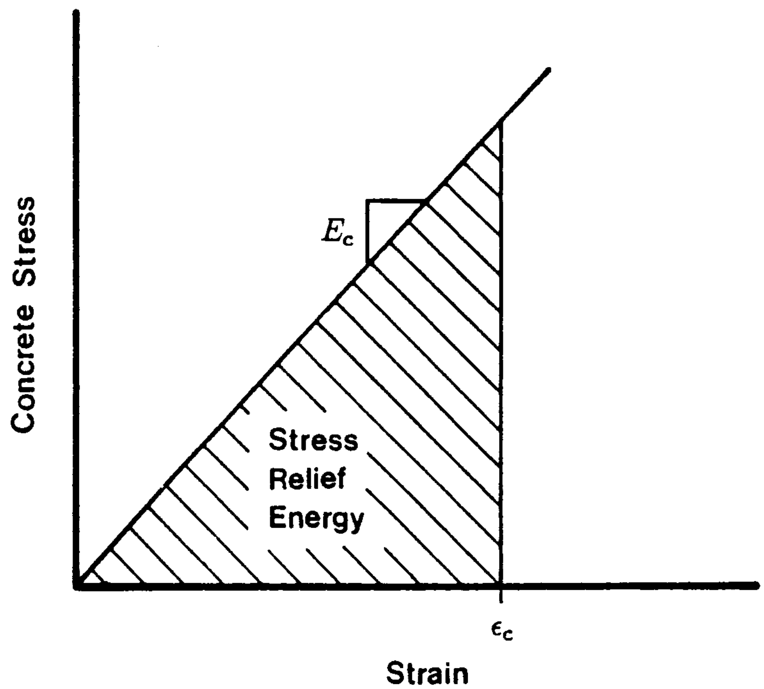


Figure 4.3. Stress Relief Energy Lost by the Concrete at a Strain ϵ_c.

stress-slip relationship between the steel and the concrete, the total stress relief energy that is lost by the concrete is found by integrating the above function over the total volume of concrete. Thus, the total stress relief energy is equal to:

$$\mathbf{E}_{\text{relief}} = A_c \int_0^S \frac{1}{2} E_c \epsilon_c^2 \, dx \tag{4.13}$$

THE EQUIVALENT LOAD METHOD

It is seen that potential energy is a function of stress, frictional work energy is a function of displacement, and stress relief energy is a function of strain. The "equivalent load" method, as described in Chapter 3, enables the displacement functions of the concrete and the steel to be found. Since the first derivative of the displacement is equal to the strain, it would seem that differentiating the displacement functions would yield the correct strain functions for the concrete and the steel, and if these functions were multiplied by their respectful moduli of elasticity, the stress functions could be found. However, this is not necessarily true.

The Stress and Strain Transformations

The equivalent load method assumed that the thermal and shrinkage effects could be modeled by a conventional load acting uniaxially upon the prism (14). The difference between a conventional loaded structure and a thermal or shrinkage loaded structure is that under conventional loads, the stress is a direct function of the strain, whereas under thermal or shrinkage loading, the stress is not directly related to the strain. The equivalent load method requires that the true stress be found by transforming the "displacement stress," which is found by differentiating the displacement function and multiplying it by the modulus of elasticity (14). For a simple case with only thermal loading and no friction effects, the transformation is:

$$\sigma = \sigma' - E\alpha\Delta T \tag{4.14}$$

where: σ = the true stress, and

σ' = the displacement stress.

When friction effects are considered, the transformation is not as simple or straightforward.

In the TTICRCP model, the displacement functions are the foundation for calculating the actual stress and strain distributions that occur. In the concrete, the true strain function is the first derivative of the displacement function, as would be expected. The true stress function is found by the equation:

$$\sigma_c(x) = E_c\epsilon_c(x) - E_c\epsilon_c(u_0) \tag{4.15}$$

where: $\sigma_c(x)$ = the true concrete stress at x ,

$\epsilon_c(x)$ = the true concrete strain at x , and

$\epsilon_c(u_0)$ = the true concrete strain at u_0.

If there were no friction or bond forces, the true concrete strain at $x = u_0$, $\epsilon_c(u_0)$, would be equal to $\alpha_c\Delta T$, and Equation 4.15 would reduce to:

$$\sigma_c = \sigma'_c - E_c\alpha_c\Delta T \tag{4.16}$$

In the steel, the true strain function is again equal to the first derivative of the displacement function, and the true stress is found from the equation:

$$\sigma_s = \sigma'_s - E_s\alpha_s\Delta T \tag{4.17}$$

With the functions for true stress, true strain, and displacement now available, the calculation of the potential, frictional work , and stress relief energies is relatively straightforward. Since the displacement functions are summations of simple exponential, sine and cosine functions, it is possible to work out the exact integrals for determining the energies. However, the sheer volume of calculations and the vast number of terms that are required for each integral prohibit the use of exact integrals in the model. Approximating the integrals is much more efficient.

Total Available Energy

The total energy that is available to the prism from the thermal and shrinkage effects is equal to:

$$\mathbf{E}_{\text{total}} = \frac{1}{2}SA_c(\alpha_c \Delta T)^2 + \frac{1}{2}SA_s(\alpha_s \Delta T)^2 + \frac{1}{2}SA_c(\epsilon_{shr})^2 \qquad (4.18)$$

If the summation of the potential, frictional work, and stress relief energies is equal to the total available energy, then the correct displacement functions have been found, and consequently, the correct stress and strain functions are also known.

THE ITERATIVE SOLUTION TECHNIQUE

The matrix method outlined in Chapter III calculates the coefficients for the displacement functions. The coefficients, however, are not the only values that are unknown. The correct coefficients will be found only if the following values are known.

1. u_0, the displacement of the concrete at the crack face
2. ℓ_1, ℓ_2, and ℓ_3, the locations of the zone interfaces, and
3. ϵ_{st}, the steel strain at the crack.

No additional equations exist for determining the above values, or even how they relate to each other. In addition, these values and the displacement functions that they help define are highly dependent on each other. To overcome this problem, a simple, iterative search technique is used to approximate the unknown values and find the coefficients of the displacement functions at the same time. The basic steps of this technique are:

1. A small, initial value for u_0 is assumed, and ℓ_1, ℓ_2, ℓ_3, and ϵ_{st} are set equal to zero.

2. The matrix method is used to find the coefficients for the displacement functions.

3. The potential, frictional work, and stress relief energies are calculated.

4. The energies are summed and compared to the total available energy.

5. If the sum does not equal the total available energy, u_0 is increased by a small increment, and ℓ_1, ℓ_2, ℓ_3, and ϵ_{st} are approximated by the values calculated by the previous displacement functions. New coefficients for the displacement functions are then found.

6. This iterative process continues until the total available energy is equal to the sum of the potential, frictional work, and stress relief energies.

The increment size for u_0 must be kept small to ensure that the approximations of ℓ_1, ℓ_2, ℓ_3, and ϵ_{st} are accurate. If accuracy is not maintained, the iterative method will blow up very quickly and it will be impossible to find the correct solution. For this reason, other techniques which might promise to converge upon the correct solution much faster have great difficulty. Several different techniques were tried during the evolution of the model, but the small-step, iterative technique proved to be the best.

TIME DEPENDENCY

To allow the model to realistically predict the behavior of a CRCP slab, it is necessary to incorporate some additional features along with the mathematical model that generates the displacement functions of the concrete and the steel. The most important of these features is time dependency, which allows the model to calculate the daily behavior of the CRCP slab as it is affected by changes in material properties and environmental and internal conditions.

Material Properties

The material properties that the model allows to be time dependent are:

1. The concrete's compressive strength.
2. The concrete's tensile strength
3. The concrete's elastic modulus
4. The concrete's coefficient of thermal contraction
5. All of the descriptive coefficients of the bond stress and friction stress functions.

The TTICRCP program allows the user to input individual daily values for each of the above material properties for each day of the analysis period. In addition, the model can generate approximate values for the tensile strength and the elastic modulus, given the compressive strength of the concrete. The function that approximates the tensile strength of the concrete is:

$$f_t' = 7.5\sqrt{f_c'} \tag{4.19}$$

where: f_t' = the tensile strength of the concrete and

f_c' = the compressive strength of the concrete.

The elastic modulus is approximated by the function:

$$E_c = 57,000\sqrt{f_c'} \tag{4.20}$$

Both the tensile strength and the elastic modulus functions are relationships recommended by the American Concrete Institute for instances where more accurate data is not available (16).

Environmental and Internal Conditions

The environmental and internal conditions that are time dependent are the daily minimum temperature and the magnitude of the drying shrinkage. The concrete is assumed to have a fixed "cure" temperature which is used as a reference point to determine the daily temperature drop. The minimum daily temperatures can be given individually for each day of the analysis period, or the model can generate continuously increasing or decreasing temperature patterns by defining the beginning and ending minimum temperatures.

Drying shrinkage is predicted by an empirical equation recommended by ACI Committee 209 (16), which states:

$$(\epsilon_{shr})_t = \frac{t}{35 + t} (\epsilon_{shr})_{ult} \qquad (4.21)$$

where: ϵ_{shr} = the shrinkage strain

t = time in days

$(\epsilon_{shr})_{ult}$ = the ultimate shrinkage for drying at 40% relative humidity.

The effect of relative humidity is accounted for by a correction factor that is multiplied with the drying shrinkage obtained from Equation 4.21.

$$C.F. = 3.00 - 0.03H \text{ , for a relative humidity} > 80\% \qquad (4.22)$$

$$C.F. = 1.40 - 0.01H \text{ , for a relative humidity} < 80\% \qquad (4.23)$$

where: $C.F.$ = the correction factor and

H = the relative humidity expressed as a percentage.

The model allows the relative humidity to be varied by day or held constant over the length of the analysis period.

WHEEL LOADS

In addition to time dependency, it is desirable to account for the stress induced by wheel loads and how that stress may affect the cracking behavior of the CRCP slab. A convenient and simple method of approximating the tensile stress in the bottom fiber of the pavement slab induced by wheel loading is by Westergaard's equation for interior loading (3). It shows the tensile strength to be equal to:

$$\sigma_c = 0.3162\frac{P}{d^2}\left[\log_{10}(d^3) - 4\log_{10}k + 6.478\right] \quad (4.24)$$

where:
- P = the applied wheel load (lbs)
- d = the thickness of the slab (inches)
- a = the radius of the tire contact area (inches), and
- k = the modulus of subgrade reaction

The model will replace the radius, a, with an equivalent radius, b, which equals:

$$b = \sqrt{1.6a^2 + d^2} - 0.675d \quad (4.25)$$

when $a \geq 1.724d$.

The replacement of the radius allows a more accurate tensile stress to be calculated (3).

Since most pavements are not subjected to wheel loads during the first days that follow their placing, the TTICRCP model allows the user to choose which day the wheel loading will begin. The wheel load stress is added to the stress that occurs at the center of the slab (at $x = S$) to determine if the tensile strength of the concrete is exceeded. If the tensile strength is exceeded, the slab is cracked in half. It should be noted that regardless of wheel loading, the TTICRCP model always assumes that the pavement slab cracks in half immediately if the concrete stress at the center of the slab exceeds the concrete's tensile strength. This does not necessarily reflect actual behavior, since even though a crack may be started when the tensile strength is exceeded, several thermal cycles may be necessary to propogate the crack through the depth of the slab.

The model can also predict the behavior of CRCP slabs that are reinforced with two or more layers of steel, rather then just one layer. The practice of using two layers of steel reinforcing is becoming quite common, particularly for thick slabs.

The large number of input parameters required by the model allow it to model realistically a wide variety of pavement designs and environmental conditions. Chapter 5 will discuss the effect of a select number of input parameters on CRCP behavior as predicted by the model.

CHAPTER V

FINDINGS AND RESULTS

PURPOSE AND SCOPE OF CHAPTER

In this chapter, a select amount of compiled output from the TTICRCP model is presented with a twofold purpose. First, the data is presented to briefly demonstrate the capability of the model to handle a wide variety of input characteristics and situations. Second, the influence of several important input parameters on the final solutions will be shown to hopefully advance the understanding of the complex CRCP structure and highlight areas that need additional study.

There are many output values from the model that have significance and are worthy of examination. Those that will be presented include:

1. The concrete and steel displacement distributions
2. The concrete and steel stress distributions
3. The steel stress at the crack
4. The concrete stress at mid-slab
5. The crack width
6. The stress transfer length.

The Stress Transfer Length

The stress transfer length is a characteristic of the final solutions that is analogous to the development length that is used in the CRCP-1, 2, and 3 computer models (3). It is defined, for the purpose of this report, as the distance from the crack to the point at which the relative slip between the steel and the concrete is first equal to zero. Virtually all of the stress transfer that occurs between the concrete and the steel occurs within the stress transfer length. In the CRCP-1, 2, and 3 models, the development length is the length over which an average bond stress is assumed to be present between the steel and the concrete. As it will be seen, however, the magnitude of the bond stress over the stress transfer length is not an average value, but is instead distinctly variable.

Input Parameters

Several input parameters can have a significant influence on the final solutions. Those that will be looked at in this chapter are:

1. The magnitude of the temperature drop
2. The amount of drying shrinkage strain
3. The steel percentage
4. The crack spacing
5. The number of steel layers.

It would also be beneficial to see the effect that material properties, such as compressive strength, elastic modulus, bond stress and friction stress characteristics, have on the final solutions. However, at the present time, there is very limited data available on the bond stress-slip relationship and how it is influenced by the other material properties. It is not possible to accurately predict how the bond stress function will change as the other material properties are varied. For this reason, the effect of material properties on the final solutions was not examined. It should also be noted that the data that will be presented in this chapter is best used only to note general trends and effects. Much of the input data that was used was fabricated for these demonstrative purposes, and although it is realistically valued, it is not valid, and the results may not indicate actual performance.

PRESENTATION AND EXPLANATION OF RESULTS

Extreme Boundary Conditions

As a test to ensure that the mathematical model used to generate the displacement functions for the prism was correctly predicting the stress and strain distributions of the concrete and the steel, two separate cases were run that represent extreme boundary conditions of a thermally loaded slab. The first case assumed that infinite friction exists between both the concrete and the base material, as well as between the steel reinforcing and the concrete. The second case assumed that zero friction, rather than infinite friction, exists. The elastic modulus of the concrete was chosen to be 3.5×10^6 psi, and the coefficient of thermal contraction was set equal to 4.0×10^{-6} in/in°F.

Infinite Friction: In theory, infinite friction prevents the concrete slab from displacing itself at all, so for the given temperature drop of 40°F, the stress in the concrete throughout the slab should be:

$$\sigma_c = E_c \alpha_c \Delta T \tag{5.1}$$

$$= (3.5 \text{ x } 10 \text{ psi}) \ (4.0 \text{ x } 10 \text{ in/in°F}) \ (40°\text{F})$$

$$= 560.0 \text{ psi}$$

The steel, having an elastic modulus of 29.0×10^6 psi and a coefficient of thermal contraction of 6.0×10^{-6} in/in°F, would not be affected by the concrete, since no movement can occur between the two. This allows the stress in the steel to be constant over the length of the bar and equal to:

$$\sigma_s = E_s \alpha_s \Delta T \tag{5.2}$$

$$= (29.0 \times 10^6 \text{ psi}) \ (6.0 \times 10^{-6} \text{ in/in°DF}) \ (40°\text{F})$$

$$= 6960.0 \text{ psi}$$

To approximate conditions of infinte friction in the model, the bond stress coefficient, K_1, was set equal to 3.0×10^{15} pci, and the friction stress coefficient, K_3, was set equal to 5.0×10^{13} pci. Figure 5.1 shows that both the concrete stress and the steel stress are constant from virtually the crack face to the center of the slab, which

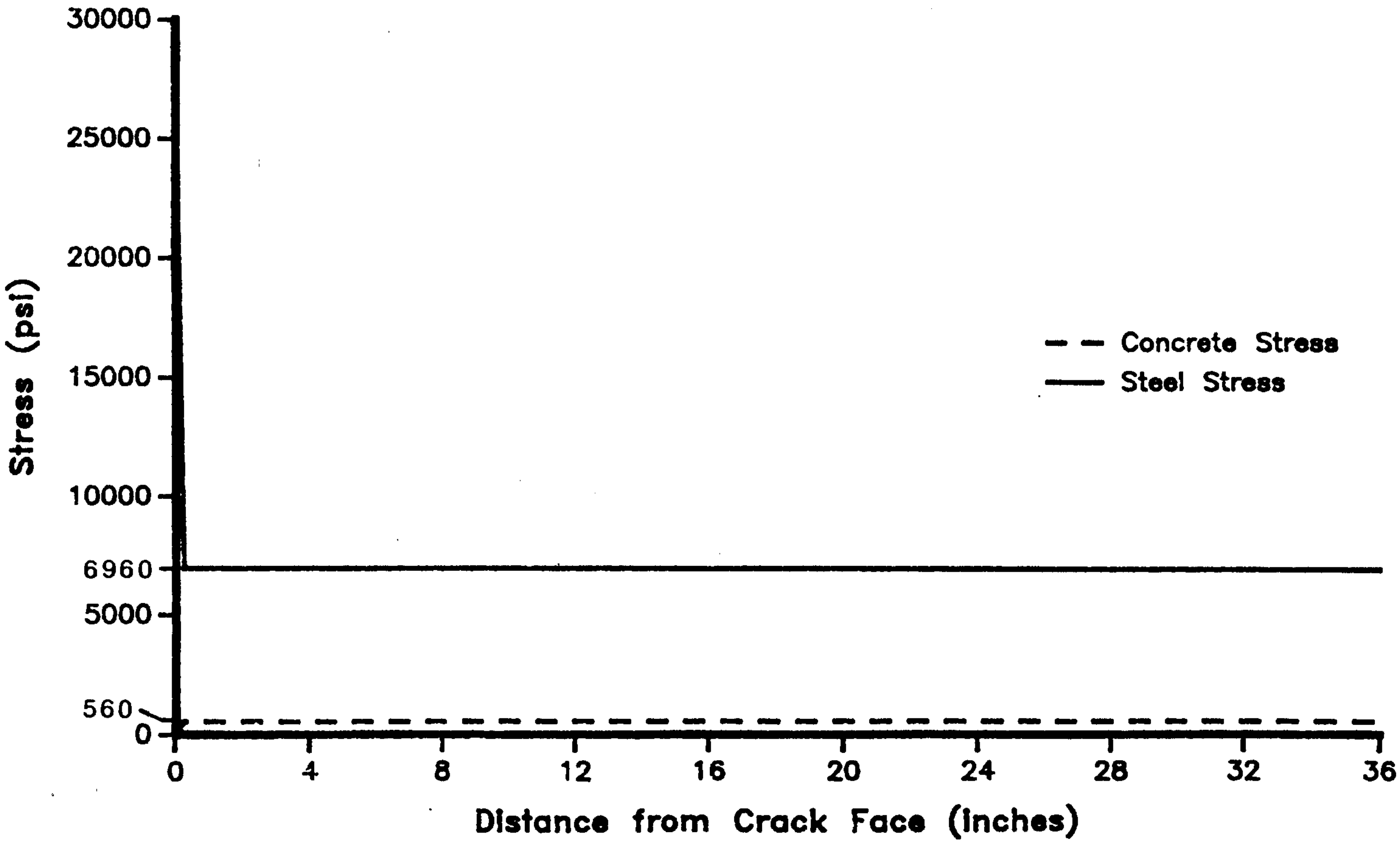


Figure 5.1. Concrete and Steel Stress Distributions for Infinite Friction Test Case.

has a crack spacing of 72 inches. The magnitude of the concrete stress is 560 psi, equal to the theoretical concrete stress. The steel has a stress of 6960 psi, which also matches its theoretical stress.

Zero Friction: In the case with zero friction, the concrete is theoretically unrestrained, and will move until its internal stress is equal to zero. The steel, because it is still rigidly anchored at both ends, will have the same magnitude of stress as in the case with infinite friction. The computer model cannot handle a case with zero friction, so instead, the friction was allowed only to come close to zero. The bond stress coefficient, K_1, was set equal to 1.0 x 10^{-3} pci and the friction coefficient, K_3, was set equal to 1.0 x 10^{-4} pci. Figure 5.2 shows that the model again behaves as it theoretically should, with the concrete stress equal to 0 psi throughout the length of the slab, and the steel stress is equal to 6960 psi.

Influence of the Temperature Drop

The effect of the magnitude of the temperature drop is seen in Figures 5.3 through 5.10. The material properties and other input parameters that were used are listed in Table 5.1. The slab length was set equal to 72 inches to correspond to a normal final crack spacing that occurs in CRCP. The temperature drop was varied from 0° to 70°F.

Concrete and Steel Displacements: Figure 5.3 shows the concrete displacement distribution for one half of the slab at temperature drops of 10, 30, 50 and 70°F. The other half of the slab is symmetrical. As it shows, the concrete displacement is continuously decreasing from the crack face to the center of the slab. The slope of the displacement curve at any of the temperature drops also decreases from the crack face to the center of the slab, although only slightly. This allows the concrete displacement to be almost linear.

Figure 5.4 shows the steel displacement distributions at the same magnitudes of temperature drop. The maximum steel displacement increases with the temperature drop, as did the concrete's. However, the steel's displacement distribution is not linear. It is a peaked distribution, with the displacements at both the crack face and at the

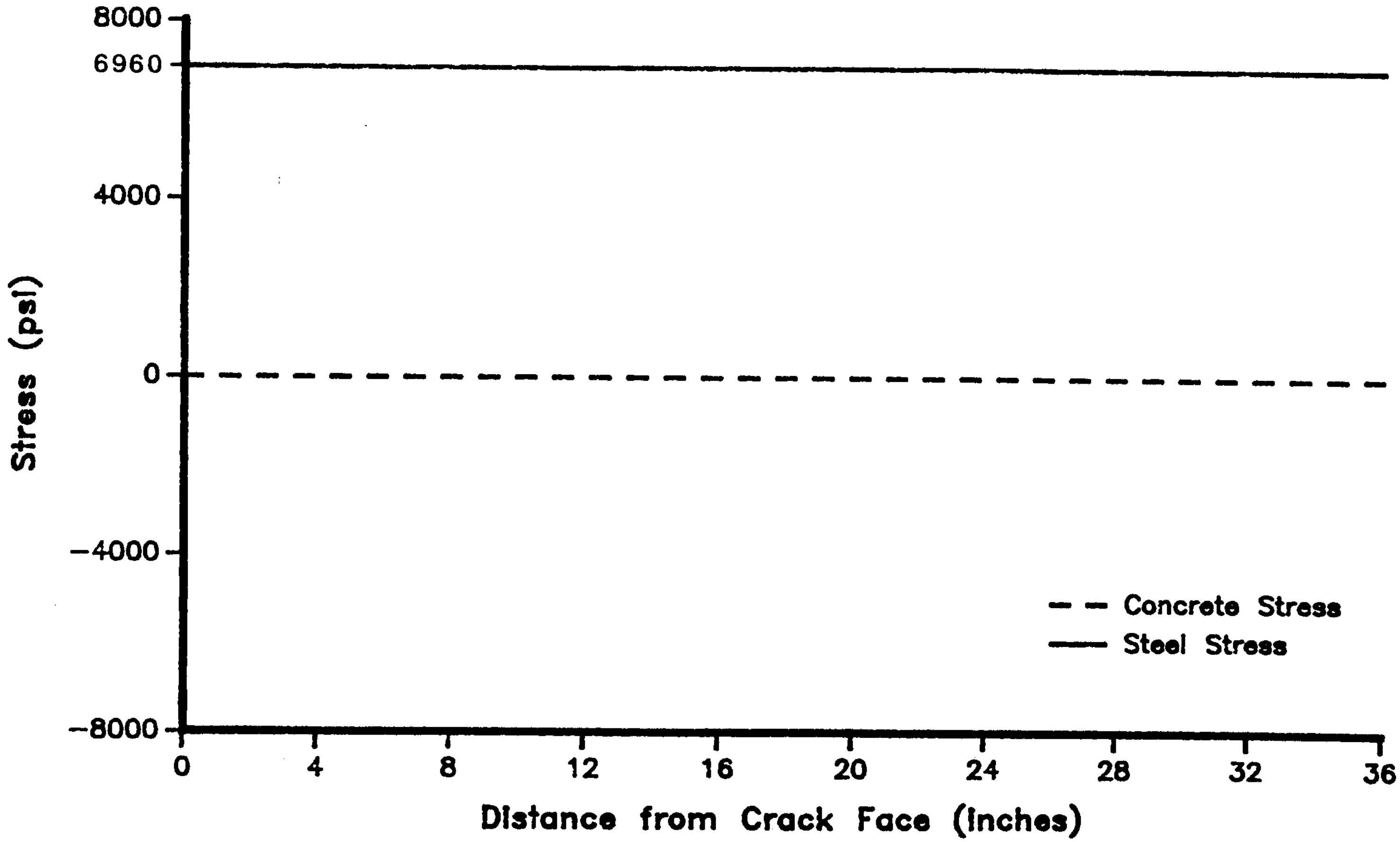


Figure 5.2. Concrete and Steel Stress Distributions for Zero Friction Test Case.

Table 5.1. Material Properties and Other Input Parameters Used for the Analysis of the Influence of Temperature Drop.

Concrete Elastic Modulus: E_c = 3.5 x 10^6 psi

Steel Elastic Modulus: E_s = 29.0 x 10^6 psi

Concrete Coefficient of Thermal Contraction: α_c = 4.0 x 10^{-6} in/in°F

Steel Coefficient of Thermal Contraction: α_s = 6.0 x 10^{-6} in/in°F

Concrete Compressive Strength: f_c' = 5000.0 psi
Concrete Tensile Strength: f_t' = 700.0 psi

Steel Bar Diameter: d_s = 0.75 in
Steel Bar Spacing: b = 6.0 in
Steel Percentage: = 0.61%

Slab Thickness: d = 12.0 in
Crack Spacing: $2S$ = 72.0 in

Curing Temperature = 80° F
Temperature Drop Varied from 0 to 70° F

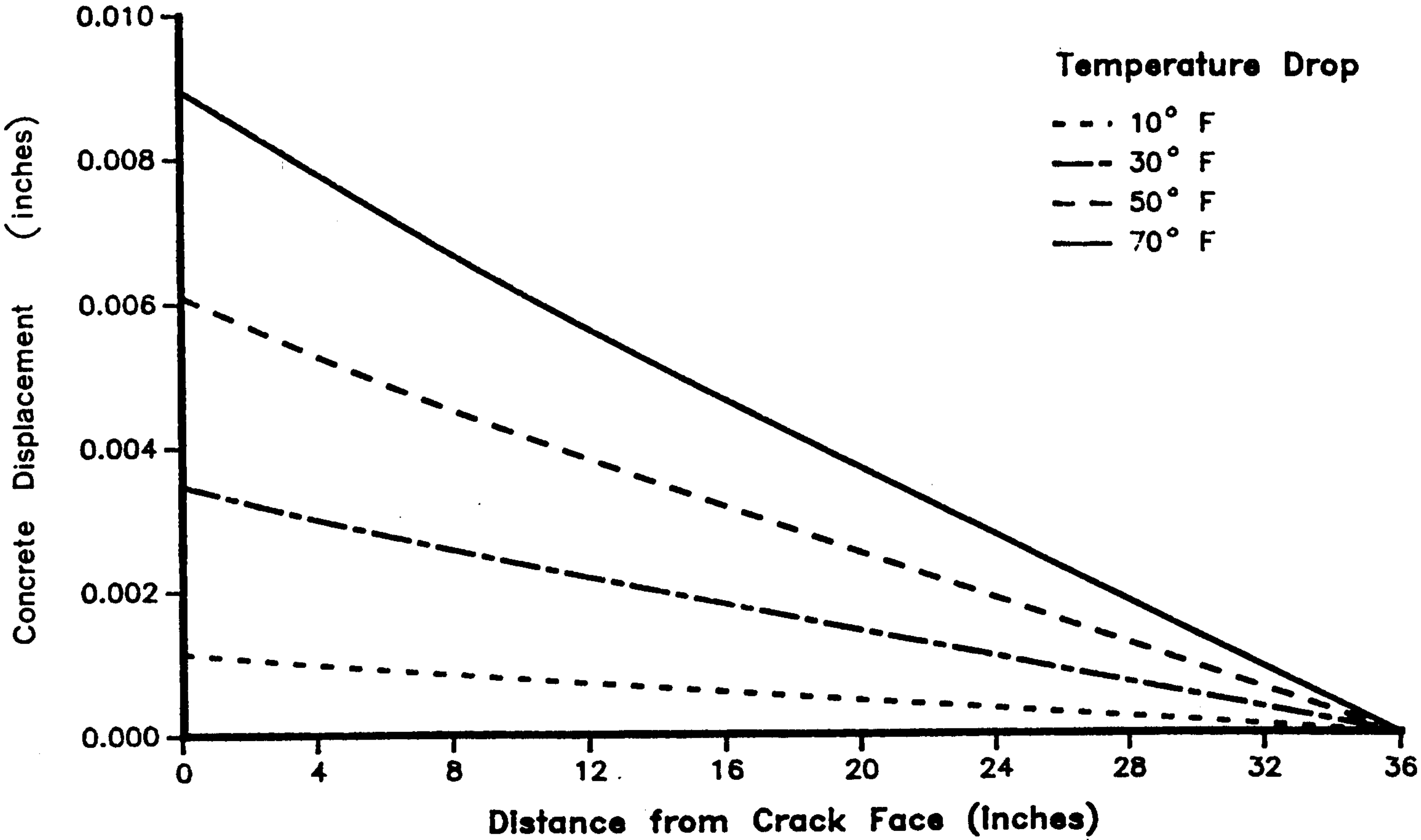


Figure 5.3. Concrete Displacement Distributions at Temperature Drops of 10°, 30°, 50°, and 70°F.

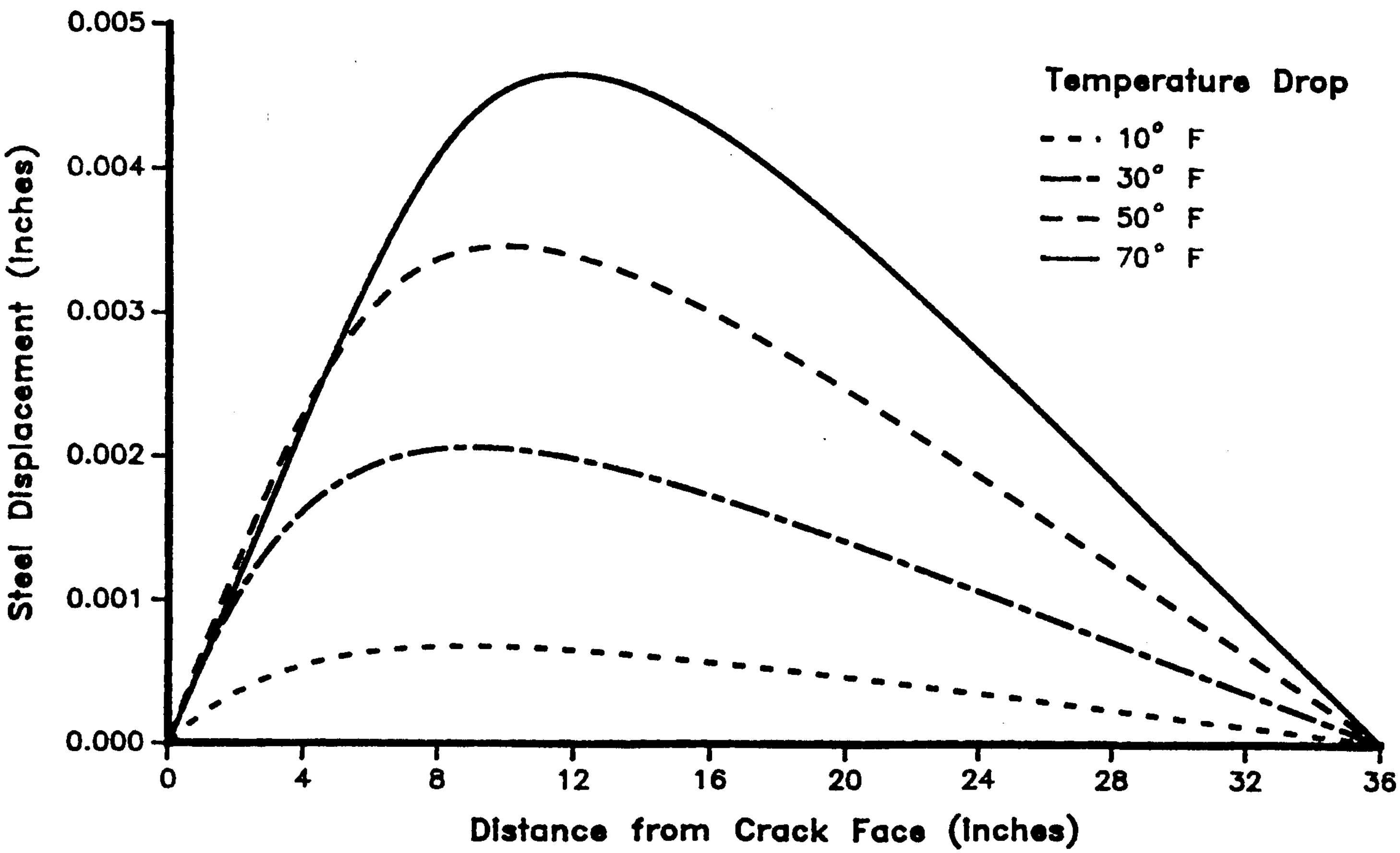


Figure 5.4. Steel Displacement Distributions at Temperature Drops of 10°, 30°, 50°, and 70°F.

center of the slab equal to zero. This corresponds correctly to the boundary conditions that were used to solve for the coefficients of the displacement functions. The point at which the maximum steel displacement occurs seems to progress further into the slab with an increased temperature drop.

Figure 5.5 shows the concrete and steel displacement distributions superimposed on each other at 30° and 70°F temperature drops. The relative magnitudes of the distributions can be easily seen.

Concrete and Steel Stress: The concrete stress distributions at the identical temperature drops of 10°, 30°, 50°, and 70°F are plotted in Figure 5.6. As is observed, the concrete distribution changes dramatically as the temperature drop increases. This behavior is caused by the bond stress and friction stress relationships that exist in the slab. As the temperature drop increases, the debonding that occurs between the steel and the concrete is noticeable, particularly in the first few inches in from the crack face. The maximum relative slip between the steel and the concrete occurs in this zone.

The effect of the debonding is even more noticeable in the steel stress distributions, which are shown in Figure 5.7. At the higher temperature drops, it is seen that the stress distribution begins to flatten off as it approaches its peak value at the crack ($x = 0$). On the 70°F curve, the slope of the stress curve is 0 within the first 4 inches from the crack. This shows that the steel and the concrete have totally debonded from each other, and that no bond forces between the steel and the concrete exist. The flattening of the curve (concave downwards) indicates that the relative slip between the steel and the concrete is in the second part of the three-part bond stress function, where the bond stress decreases as the slip increases.

Figure 5.8 shows how the magnitude of the concrete stress at the center of the slab increases with increasing temperature drops. The curve appears to be made of two linear portions that are connected through a transition zone that occurs between temperature drops of 20° and 50°F. The steel stress at the crack exhibits the same behavior in Figure 5.9. The transition zone again occurs between 20° and 50°F.

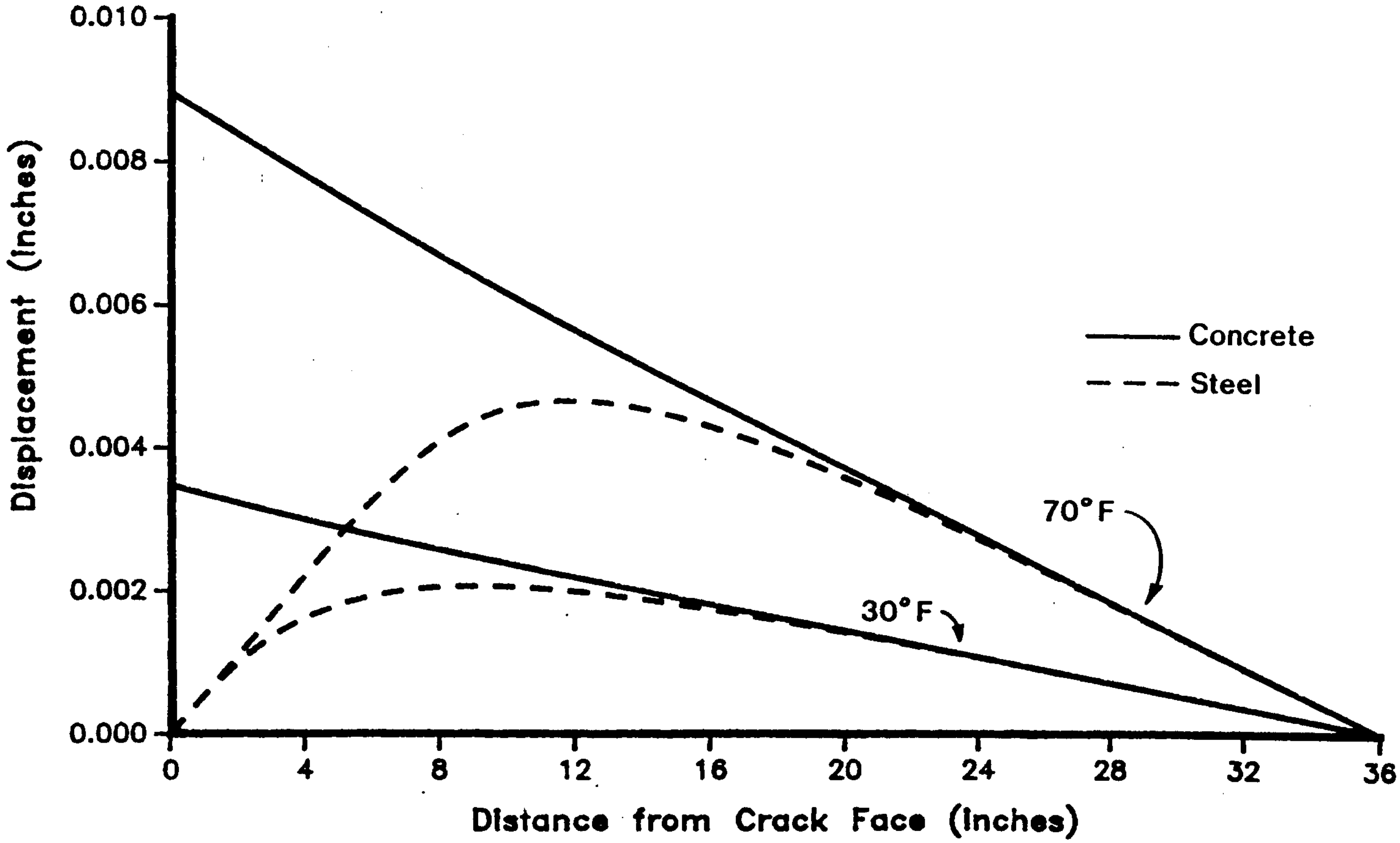


Figure 5.5. Concrete and Steel Displacement Distributions at Temperature Drops of 30° and 70°F.

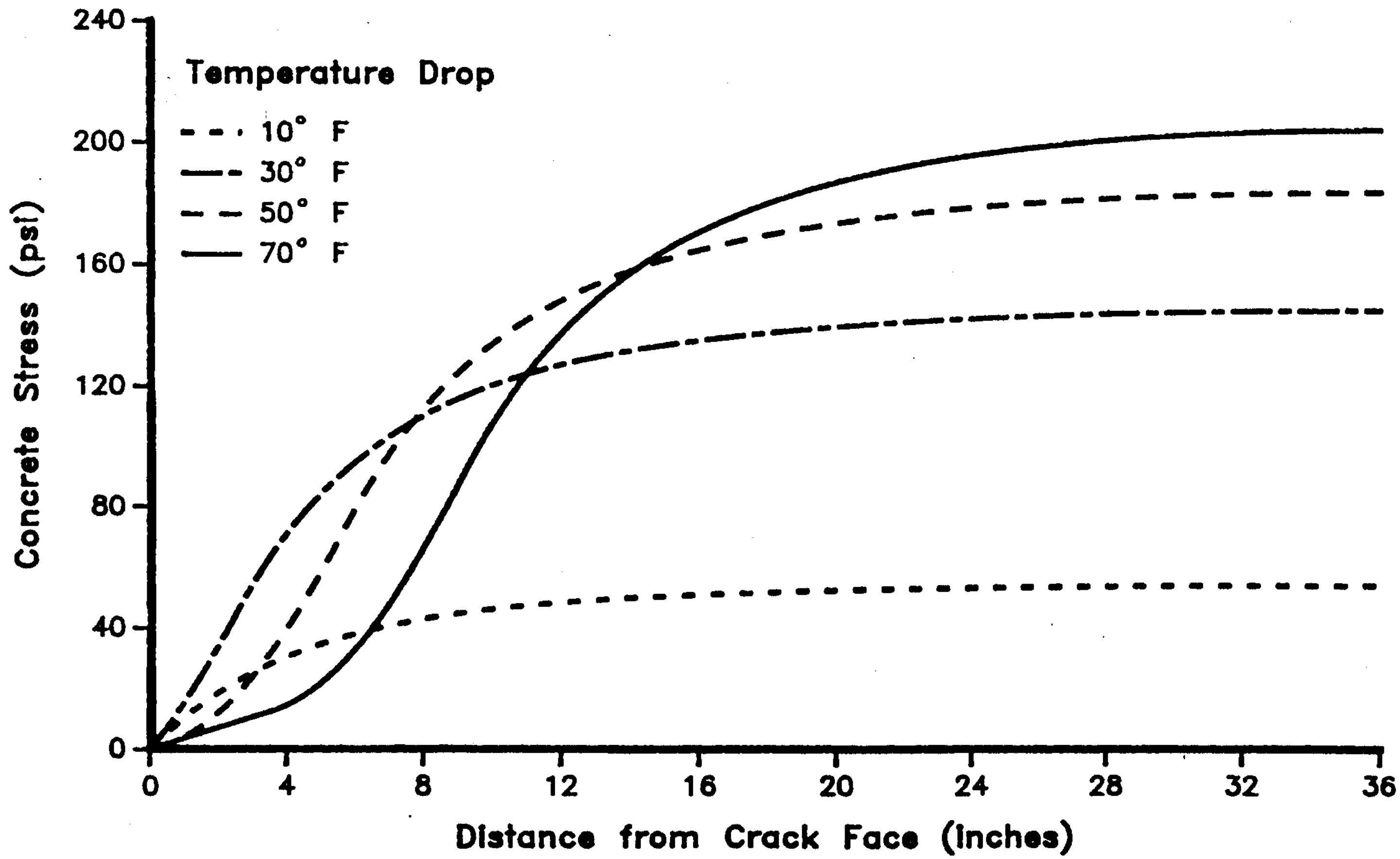


Figure 5.6. Concrete Stress Distributions at Temperature Drops of 10°, 30° 50°, and 70°F.

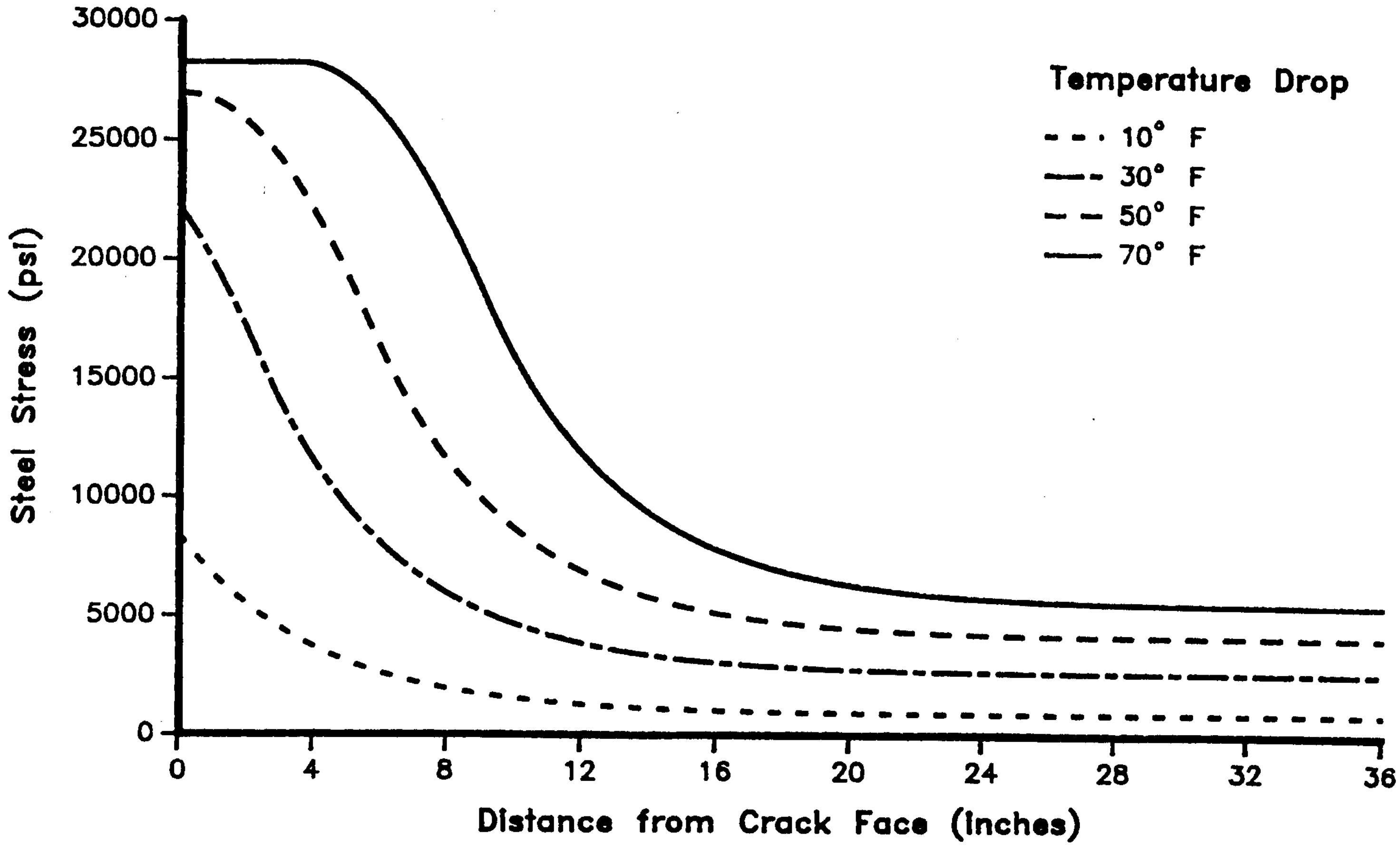


Figure 5.7. Steel Stress Distributions at Temperature Drops of 10°, 30°, 50°, and 70°F.

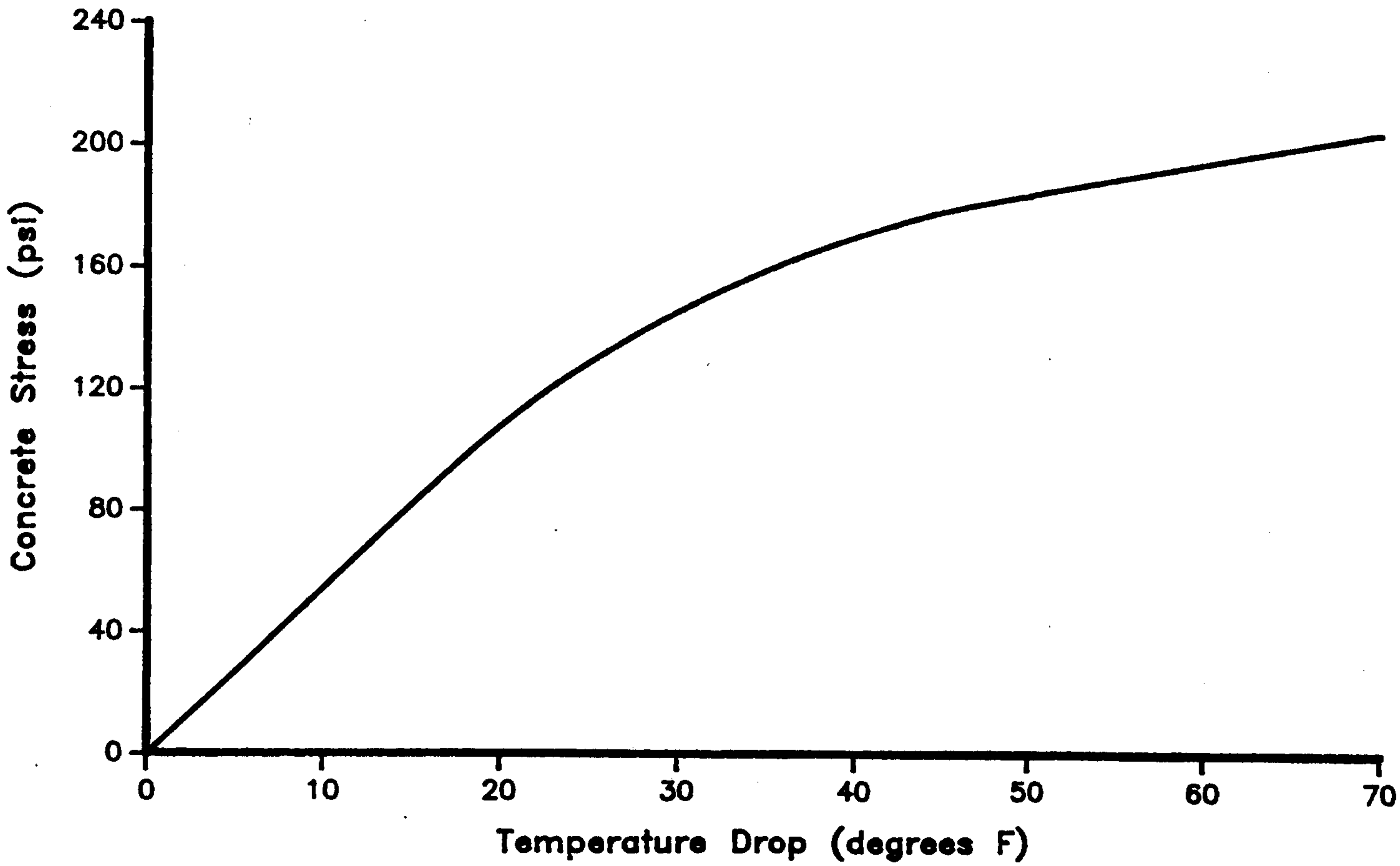


Figure 5.8. Concrete Stress at Mid-Slab vs. Temperature Drop.

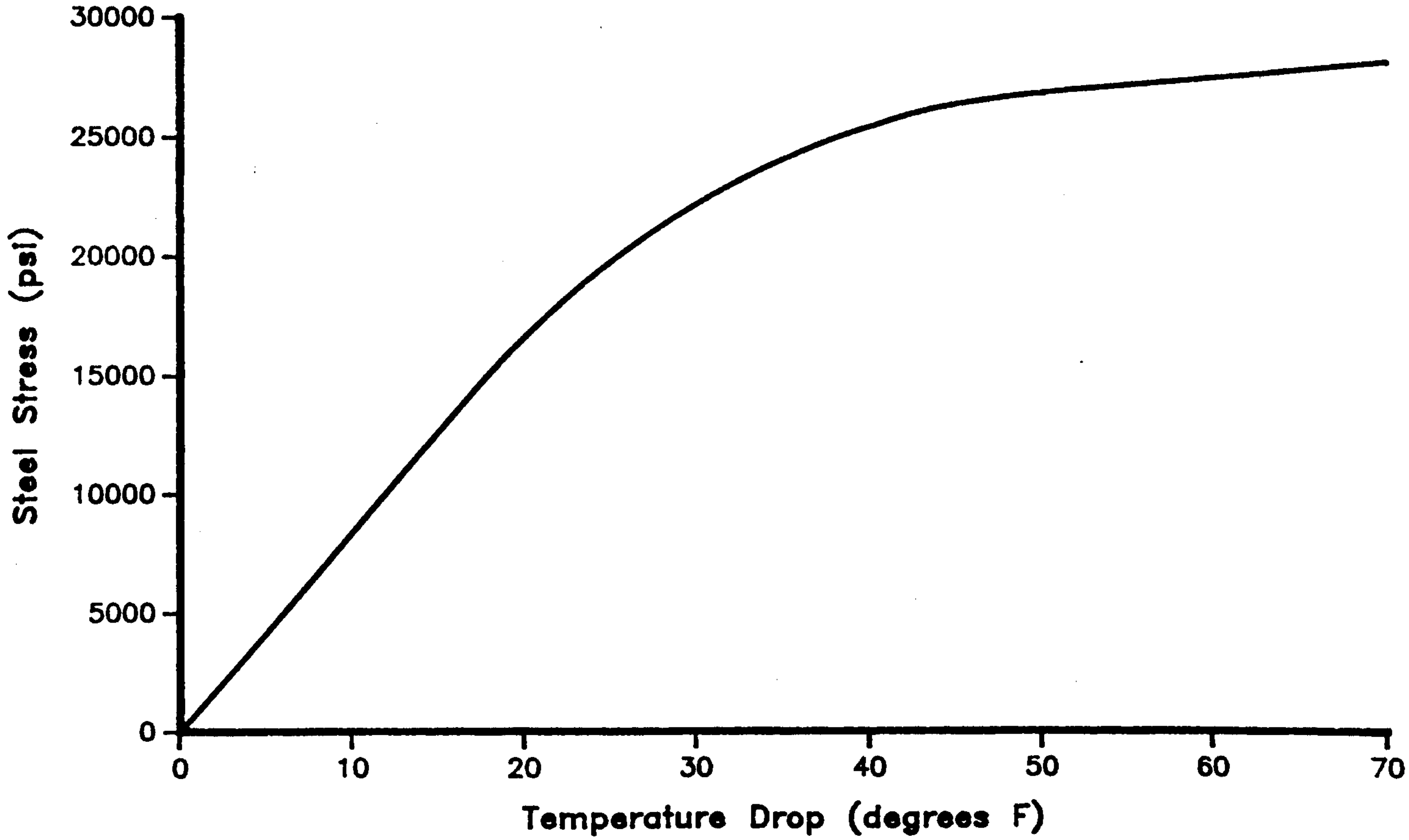


Figure 5.9. Steel Stress at Crack vs. Temperature Drop.

This is explainable when it is known that up until the temperature drop equals 20°F, the final solution is in Case 1. (See Chapter 3 for a complete description of the cases that the final solution may be in.) From 20° to 50°, the solution is in Case 2, and beyond 50°, the solution is in Case 3. This does not mean, however, that the terminal case is Case 3. The solution will progress into Case 5, and possibly into Cases 6 and 9 if the temperature drop is allowed to increase more than 70°F.

In Case 1, an increase of the displacements causes a roughly proportionate increase in the total bond and friction forces that act against the displacements. This causes the concrete stress at mid-slab and the steel stress at the crack to increase linearly as the temperature drop increases below 20°F. When the temperature drop becomes greater than 20°F, the solution is in Case 2, which means that an increase in the displacements causes the total bond and friction forces to still increase, but the rate of increase is decreasing. This causes curvilinear behavior in both the concrete and the steel. Beyond 50°F, the concrete displacement is large enough so that total debonding begins to occur between the steel and the concrete near the crack. Any additional increases in displacement will not cause the total bond force to increase, since the area underneath the bond stress distribution will now remain virtually constant. (This is verified in Figure 5.19.) The total friction force will still increase, though. This causes the linear behavior evident on the curves above 50°F. As the solutions progress through additional cases, the magnitude of the mid-slab concrete stress and the steel stress at the crack will be affected similarly. The effect is not limited to just the concrete stress at mid-slab and the steel stress at the crack, but influences the entire stress distributions of both the concrete and the steel.

Effect on Crack Opening: Figure 5.10 shows the effect of temperature drop on the crack opening, which is equal to twice the concrete displacement at the crack, u_0. This curve also shows the influence of the effect discussed above for the steel and concrete stresses, although it is much less noticeable. The crack opening appears to increase almost linearly as the magnitude of the temperature drop increases, and close examination shows that from 0° to 20°F, it

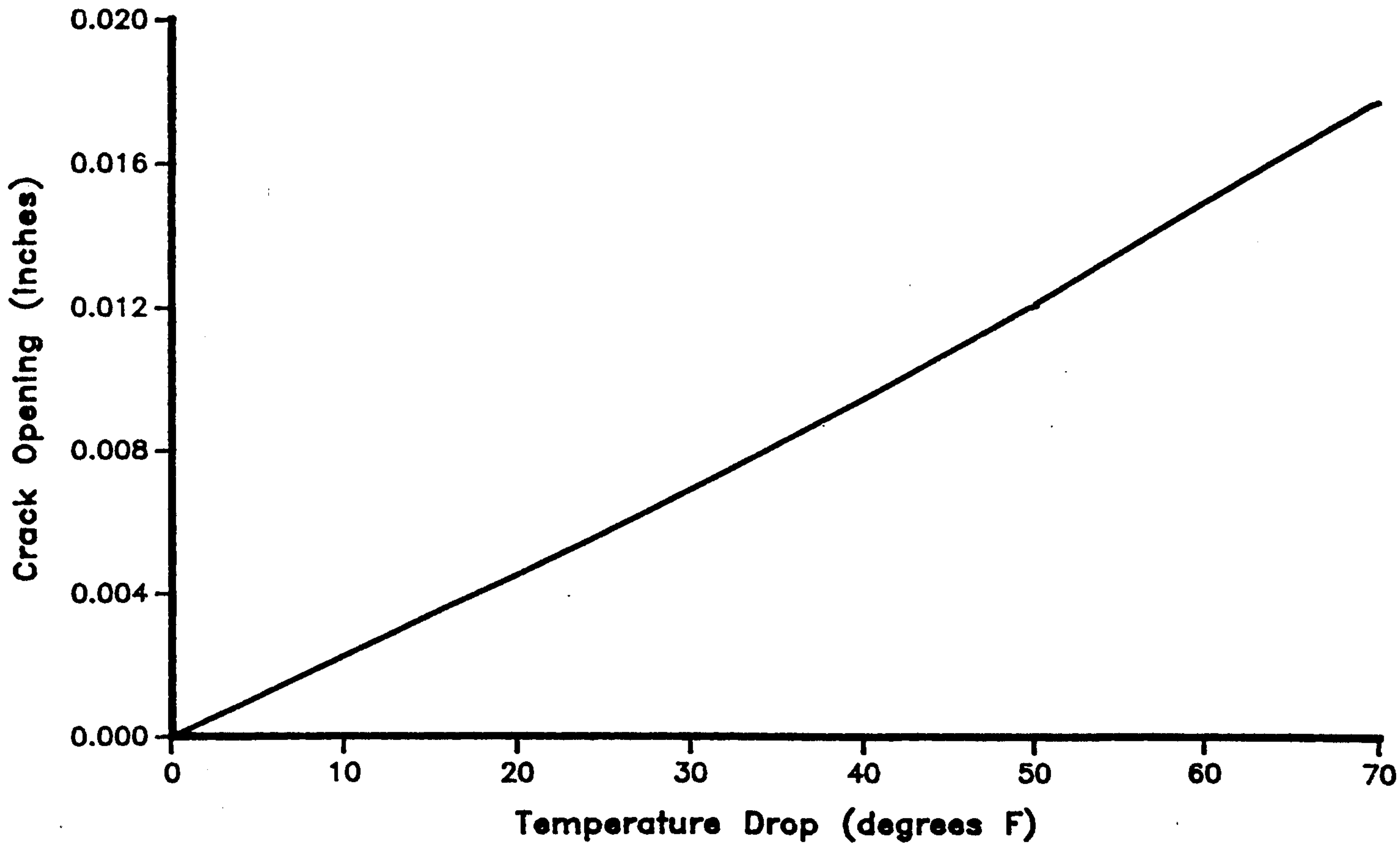


Figure 5.10. Crack Opening vs. Temperature Drop.

does increase linearly. From 20° to 50°, the slope of the curve gently increases, and the curve becomes linear again above 50°.

Influence of the Steel Percentage

Steel percentage is an important consideration in the design of CRCP, and Figures 5.11 through 5.13 briefly show its influence. The same material properties and other input parameters that are listed in Table 5.1 were used for the analysis, except that the temperature drop was held constant at 40°F, and the steel percentage was allowed to vary from 0.125% to 1.0%.

Figure 5.11 shows the influence of steel percentage on the concrete stress at mid-slab. A nearly linear relationship exists, with the lowest concrete stress occurring at the lowest steel percentage. The steel stress at the crack is not greatly influenced by the steel percentage, as Figure 5.12 shows. A slight decrease in the steel stress as the steel percentage increases can be noted, though. In Figure 5.13, it is seen that the crack opening decreases almost linearly as the steel percentage increases. These figures combine to show that the presence of the steel, and the bond forces that are generated by it, have a significant effect on the movement of the concrete. With a low amount of steel, the concrete is able to displace itself more, and consequently, the stress level throughout the concrete is lower than with a high amount of steel.

Influence of Crack Spacing

Figures 5.14 through 5.16 show the effects that the crack spacing has on the concrete and the steel. Again, the material properties and other input parameters that are listed in Table 5.1 were used, except that the temperature drop was 40°F and the crack spacing was allowed to vary from 24 to 216 inches.

The concrete stress at mid-slab is seen in Figure 5.14 to be greater at longer slab lengths. It decreases linearly with the crack spacing until the crack spacing is equal to 96 inches, at which point the concrete stress begins to slowly drop off. Below 48 inches, the

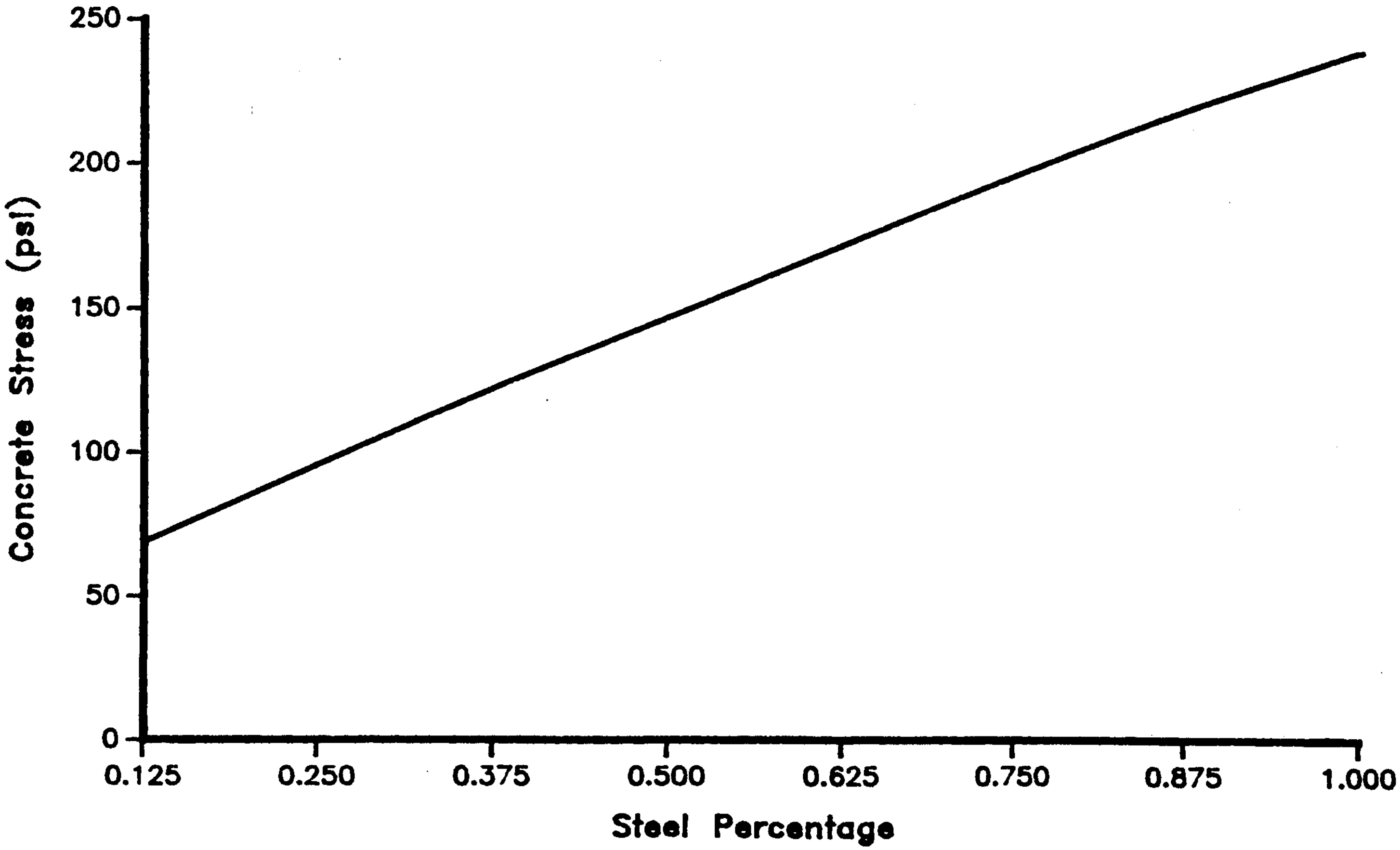


Figure 5.11. Concrete Stress at Mid-Slab vs. Steel Percentage.

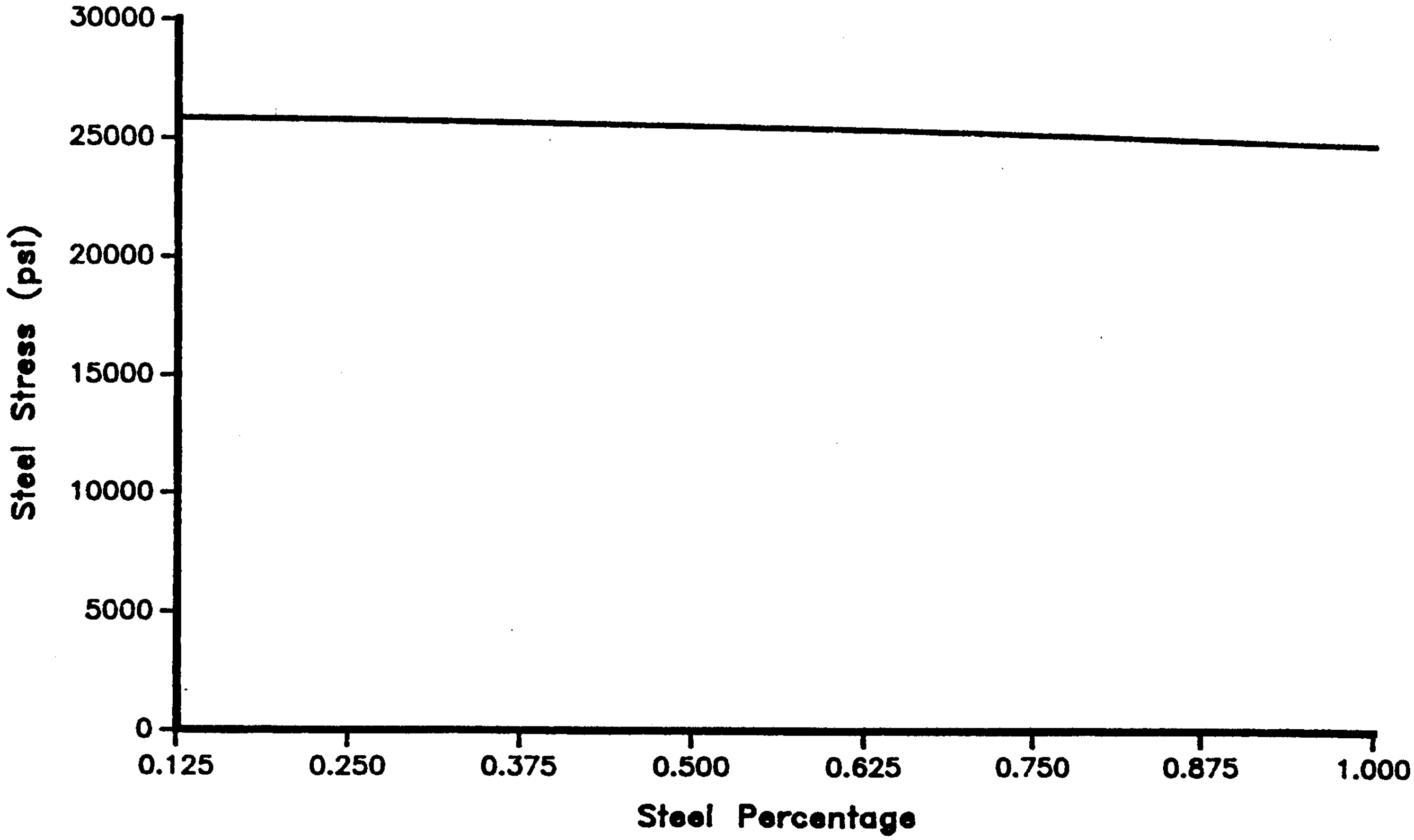


Figure 5.12. Steel Stress at Crack vs. Steel Percentage.

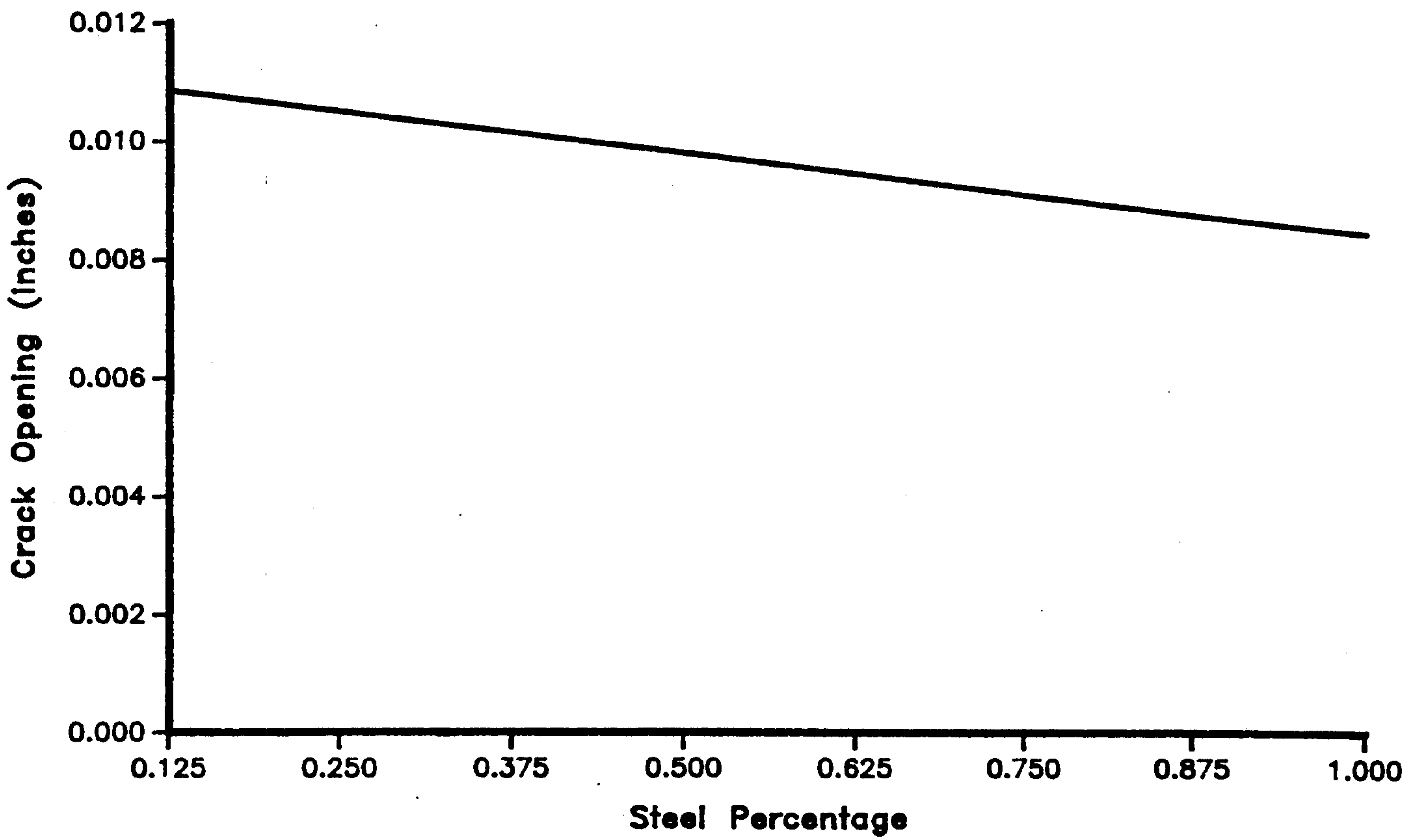


Figure 5.13. Crack Opening vs. Steel Percentage.

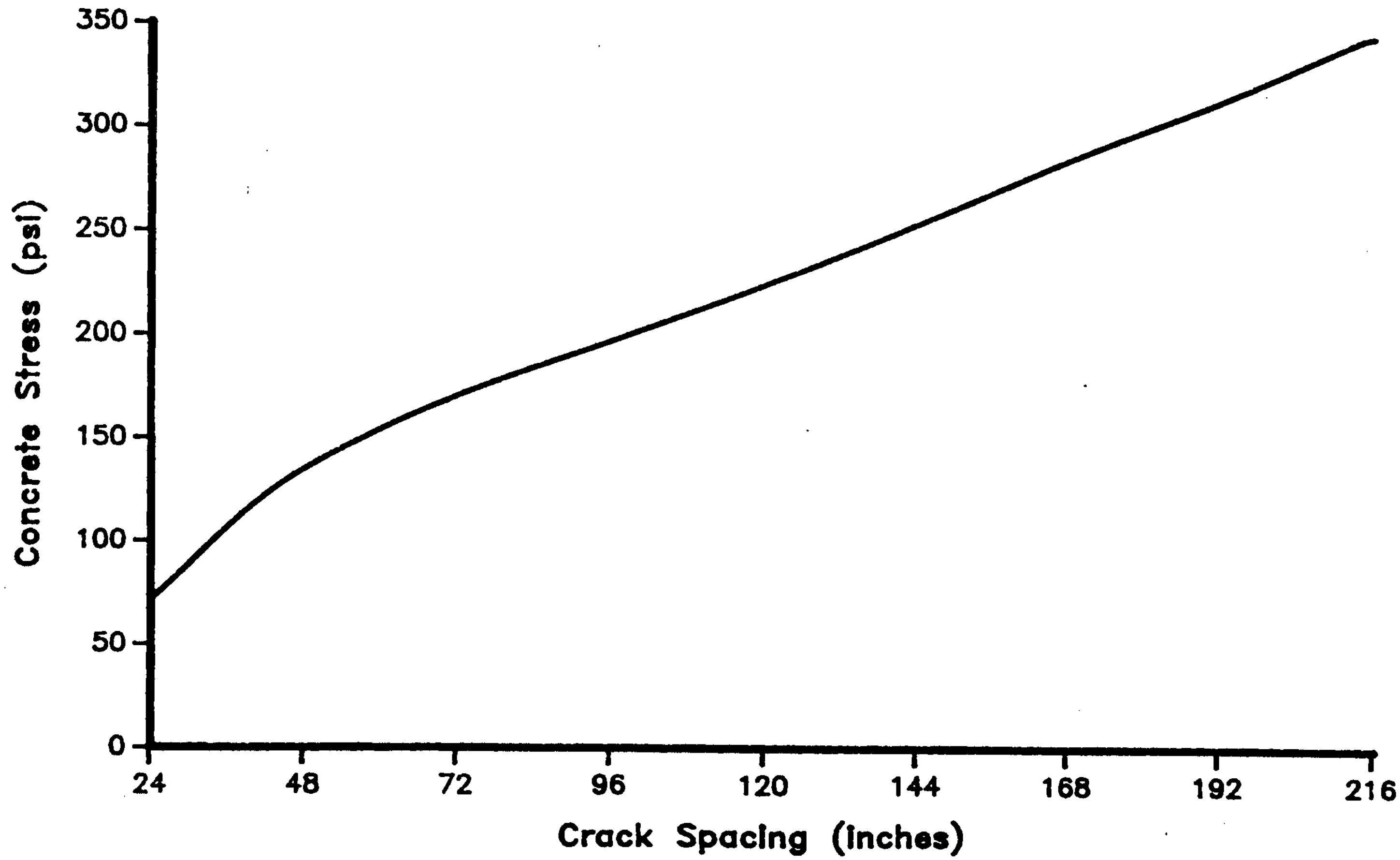


Figure 5.14. Concrete Stress at Mid-Slab vs. Crack Spacing.

rate of drop off is quite sharp. Figure 5.15 shows that the steel stress at the crack is constant for crack spacings down to 96 inches, at which point it begins to decrease before dropping off sharply at crack spacings less than 48 inches. The crack opening, which is seen in Figure 5.16, exhibits the same phenomenon, although the effect is very slight. This type of behavior again shows the influence of the bond stress and friction stress distributions that were previously seen in Figures 5.8 through 5.10. For crack spacings greater than 96 inches, the final solution is in Case 3. Between 96 and 48 inches, the solution is in Case 2, and for crack spacings less than 48 inches, the solution is in Case 1.

The Stress Transfer Length

The TTICRCP program indicates that the stress transfer length is dependent, in some amount, on the crack spacing, as Figure 5.17 shows. The stress transfer length is equal to one half of the crack spacing for crack spacings of 77 inches or less, which is to say that the only point where the relative slip between the steel and the concrete is equal to 0 occurs at the center of the slab. This is also where the boundary conditions dictate that the displacements of both the steel and the concrete be equal to zero. As the crack spacing increases past 77 inches, the stress transfer length drops sharply before leveling out at a value of approximately 30 inches. The abruptness of this drop is, for now, unexplainable.

Figure 5.18 shows how the stress transfer length is influenced by the temperature drop. A slab length of 120 inches was used to generate this data, along with the material properties and other input parameters listed in Table 5.1. The stress transfer length is not influenced by small temperature drops (less than 14°F) and only gently increases as the temperature drop is increased above 14°F. This behavior can possibly be understood better by seeing how the bond stress distribution changes between the steel and the concrete as the temperature drops.

Bond Stress Distribution: As is observed in Figure 5.19, the location of the peak bond force progresses uniformly into the slab as

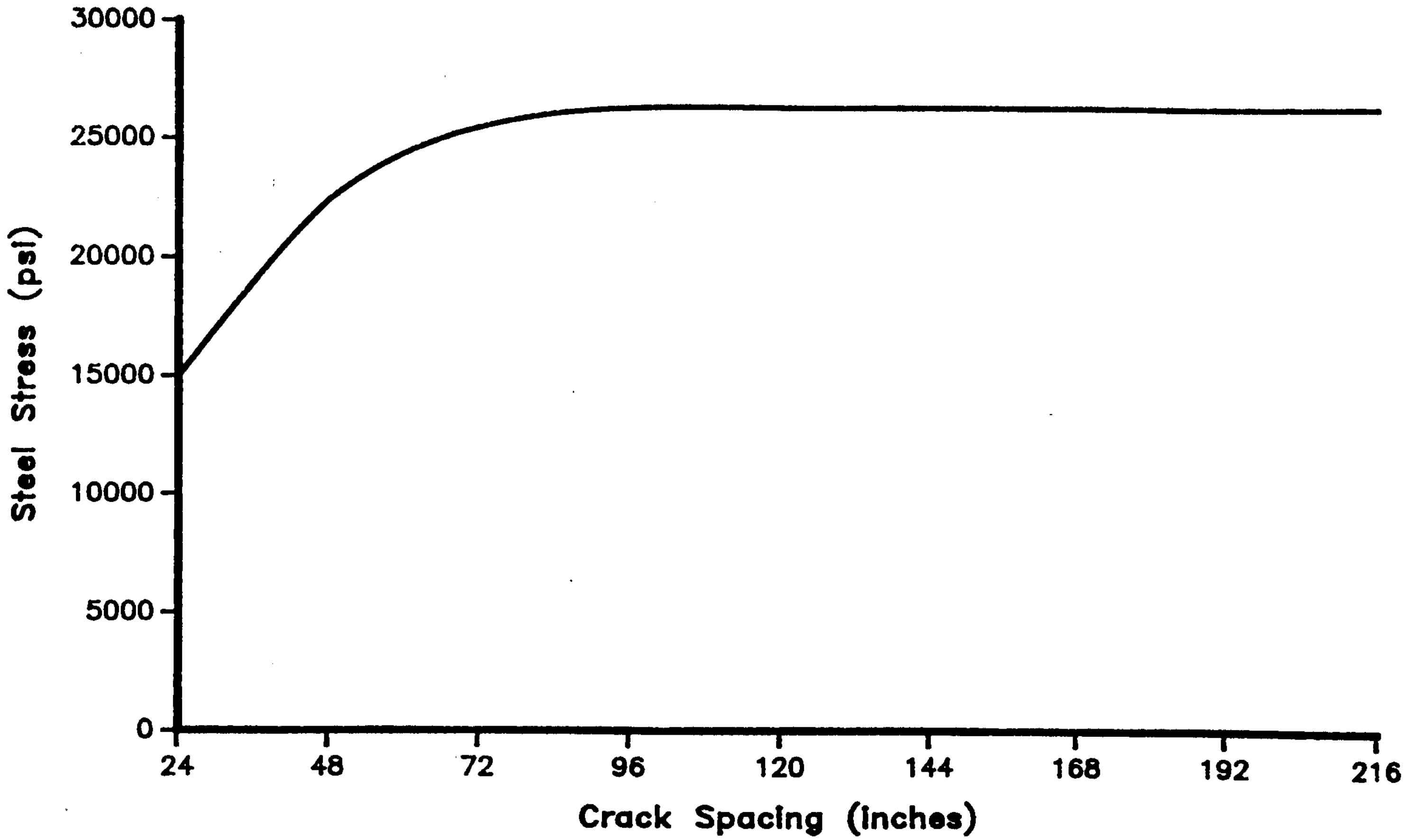


Figure 5.15. Steel Stress at Crack vs. Crack Spacing.

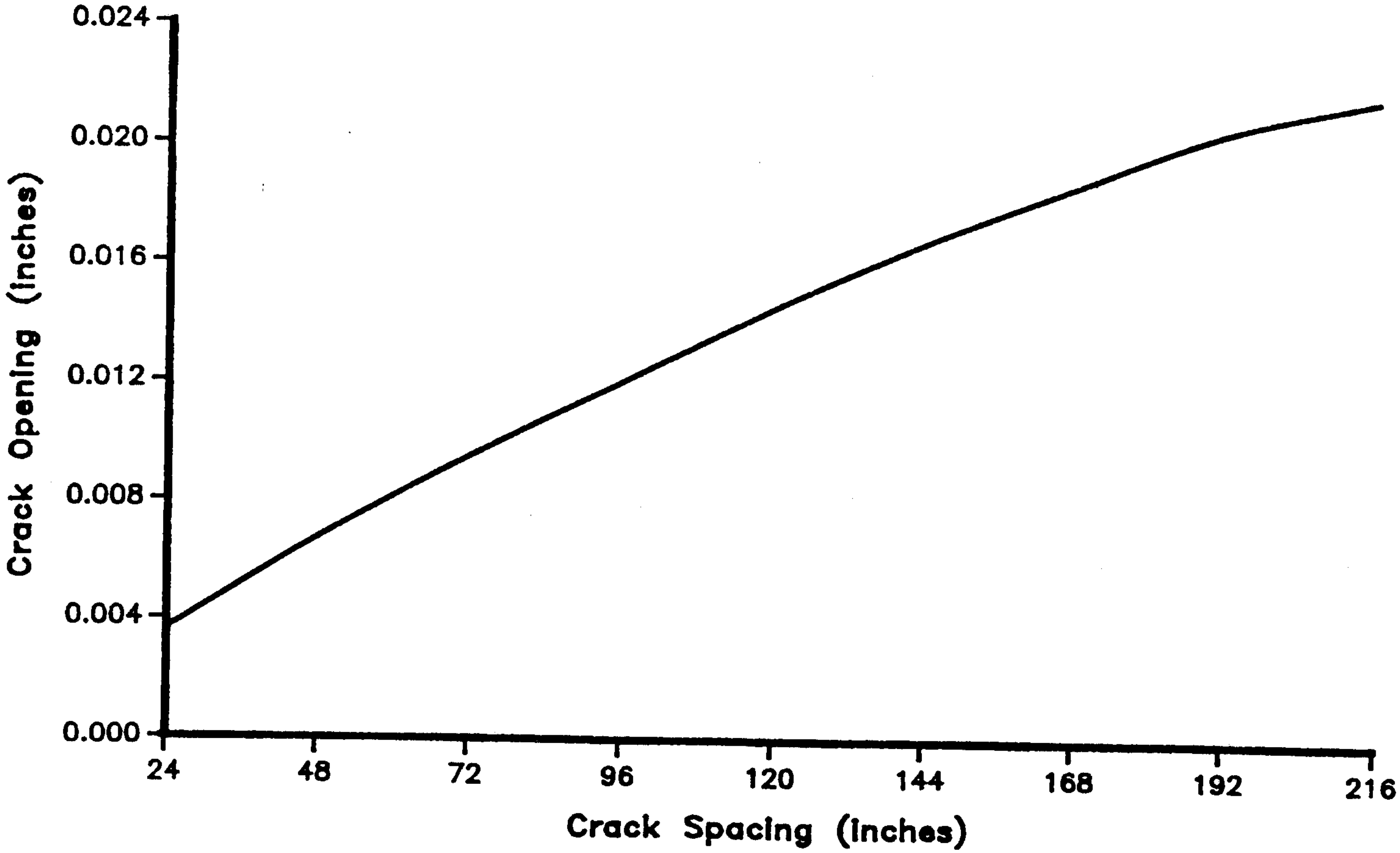


Figure 5.16. Crack Opening vs. Crack Spacing.

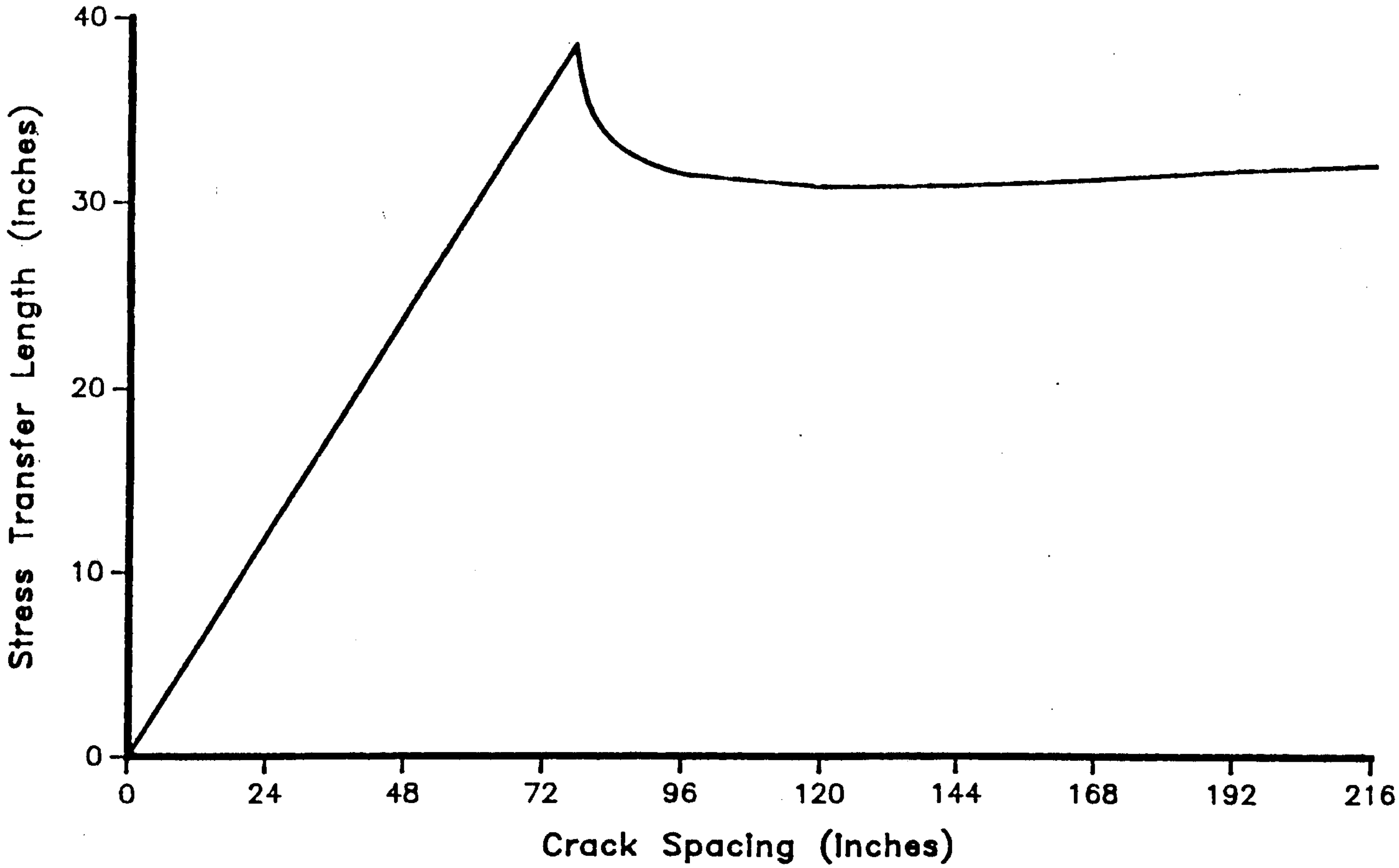


Figure 5.17. Stress Transfer Length vs. Crack Spacking.

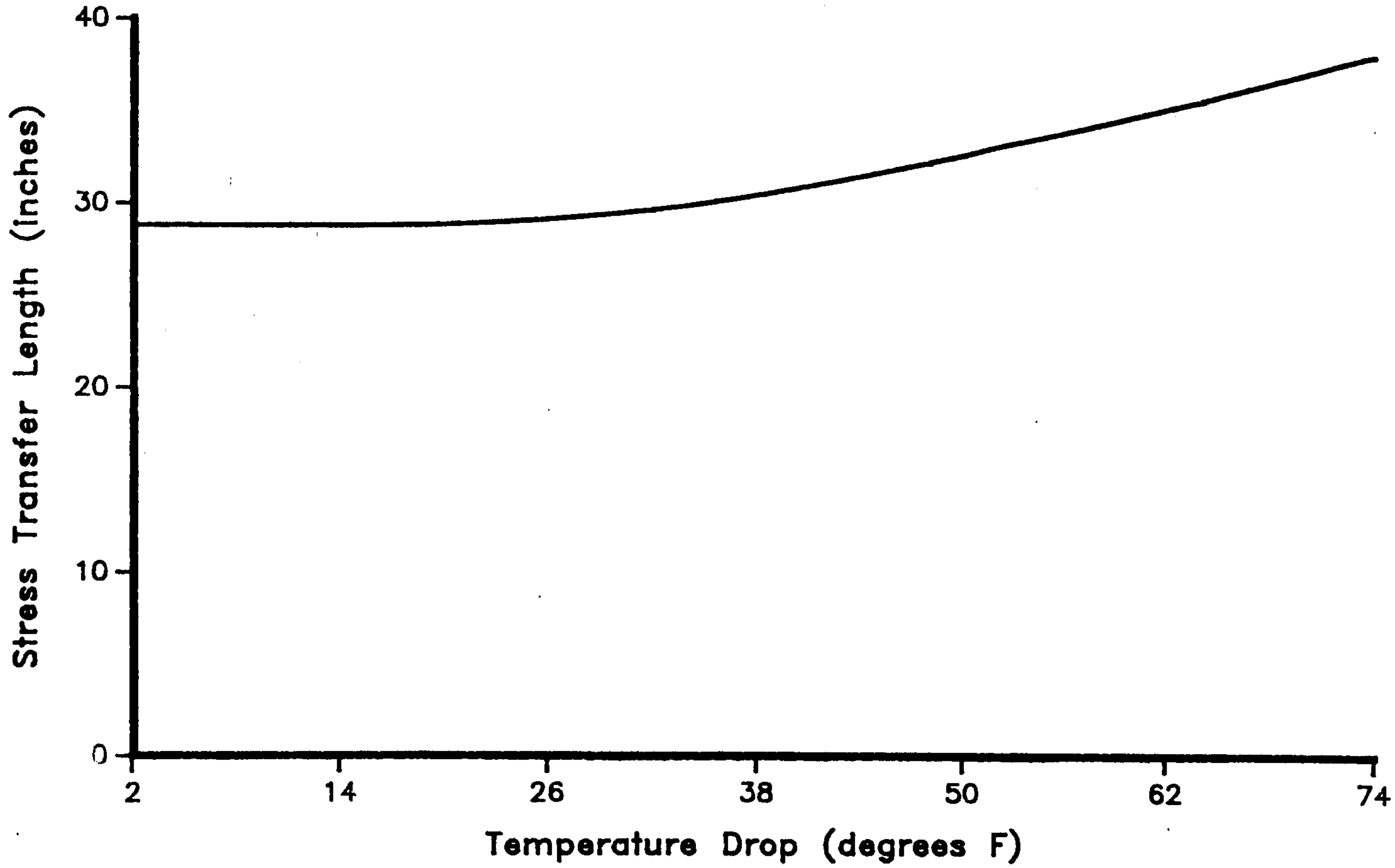


Figure 5.18. Stress Transfer Length vs. Temperature Drop.

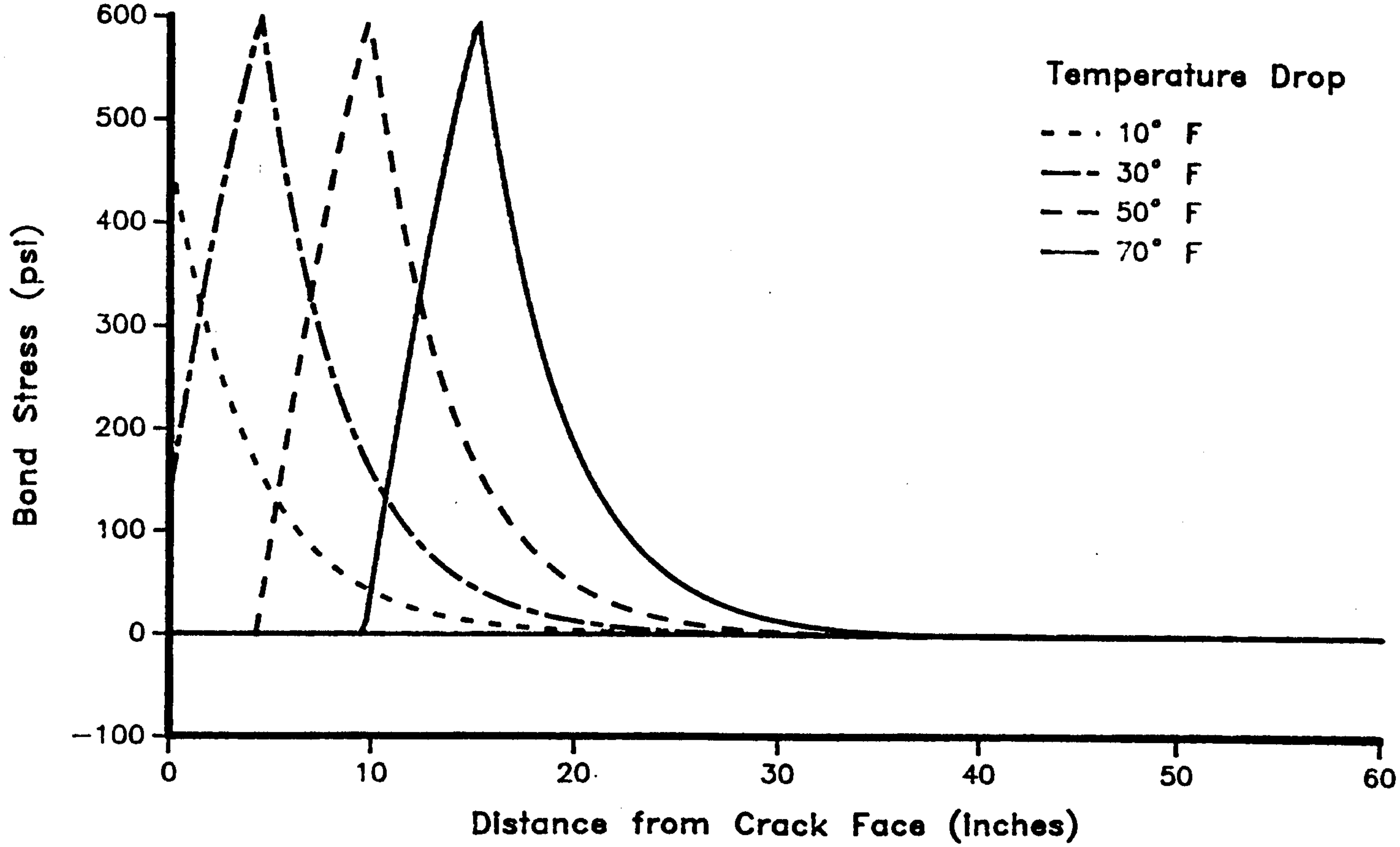


Figure 5.19. Bond Stress Distributions at Temperature Drops of 10°, 30°, 50°, and 70°F.

the temperature drop increases. The entire bond stress distribution appears to maintain the same shape as it moves into the slab as well. The increase in the stress transfer length begins after the solution has passed through Case 1 (this occurs at $\Delta T = 14°F$), and increases along with the progression of the bond stress distribution into the slab. The stress transfer length increases at a slower rate than the progression of the peak bond stress occurs, which indicates that the shape of the distribution is changing slightly as it moves in.

The actual numerical solution of the bond stress distribution also indicates that the bond stress first passes through 0.0 (which defines the stress transfer length) and is then slightly negative. If the slab length is sufficiently long enough, the bond stress will return to 0.0 before the center of the slab. If not, the bond stress will be equal to 0.0 at the center of the slab, where both the steel and the concrete displacements are 0.0. The magnitude of the negative bond stress is very small, only a few psi. Effectively, in a real slab, it would be impossible to measure this bond stress, and for all purposes, the steel and the concrete should be considered completely bonded together.

The Friction Stress Distribution: Figure 5.20 shows the corresponding friction stress distributions to the bond stress distributions in Figure 5.19. It is seen that the friction stress is a great deal less than the bond stress, and the shape of the friction stress curve is proportional to the concrete displacement distribution for lesser temperature drops. The 70°F curve shows the peaked distribution that occurs at higher temperature drops.

Comparison of Pavements With One and Two Layers of Steel

All of the data presented thus far has disregarded the effect of time, drying shrinkage, and wheel loads on the behavior of the CRCP. To demonstrate the capability of the model to handle these very important parameters, a comparison was made between a design with one layer of steel reinforcing and a design with two layers of steel. The two pavements were looked at over a 40 day analysis period that corresponded to the first 40 days of the pavements' lives. The steel percentage

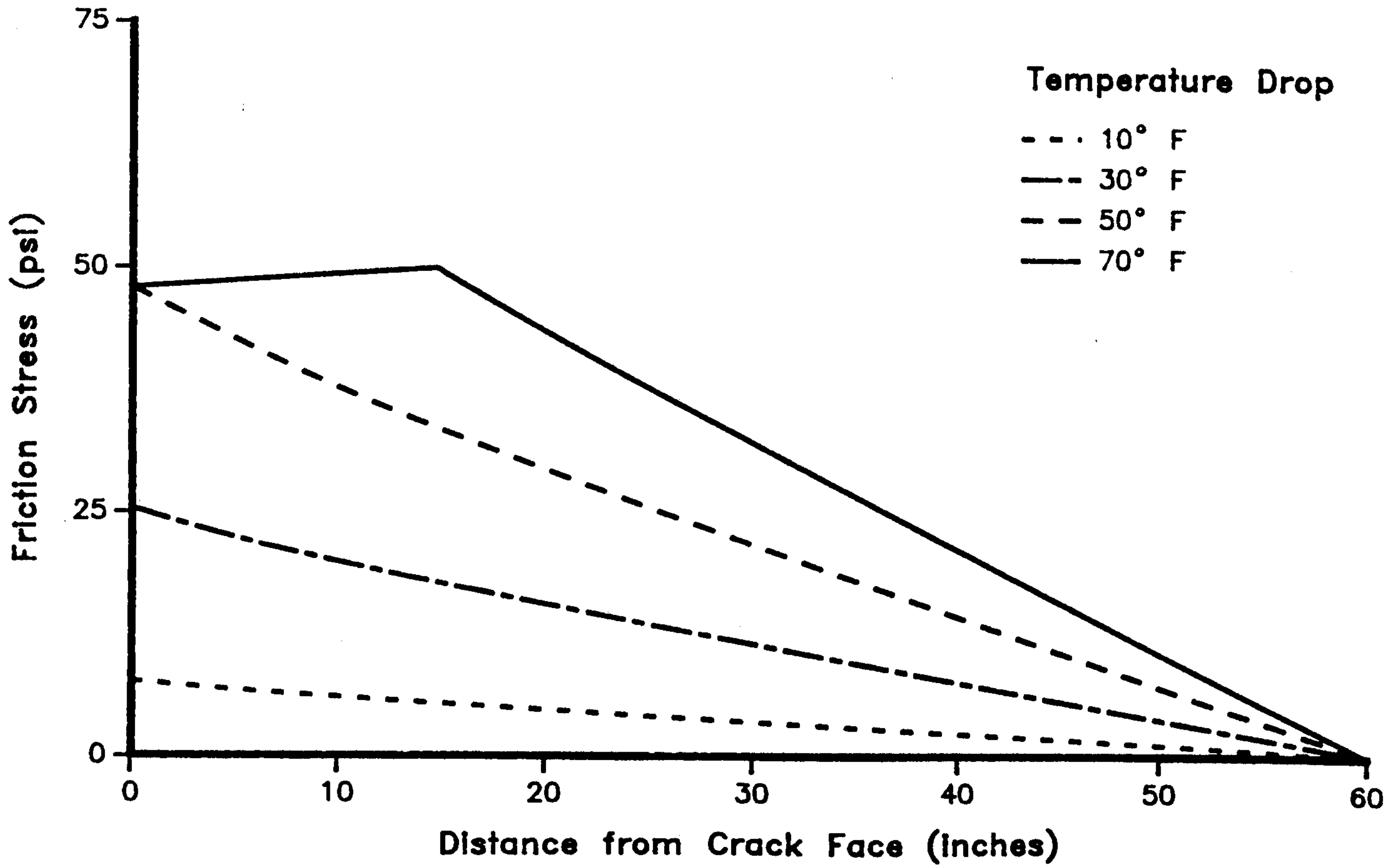


Figure 5.20. Friction Stress Distributions at Temperature Drops of 10°, 30°, 50°, and 70°F.

between the two pavements was held constant. Table 5.2 lists the other pertinent input parameters.

The temperature drops and shrinkage strains that the pavements were subjected to are shown in Figures 5.21a and 5.21b. As it is seen, the temperature drop increased from 20°F on Day 1 to 60°F on Day 40. The drying shrinkage also increased continually from 7.2×10^{-6} in./in. to 138.1×10^{-6} in./in. over the course of the analysis period. A 9 kip wheel load was assumed to begin loading the pavement on Day 29 and to continue loading through Day 40.

The crack spacing was initially 60,000 inches, and Figure 5.22 shows how the crack spacing decreased identically through the analysis for both pavements. The cracking that occurred on Day 29 was directly attributable to the wheel load. Without the wheel load, the pavement would not have cracked for several more days.

Figure 5.23 shows how the crack openings varied over the 40 days. There is virtually no difference between the pavement with one layer of steel and the pavement with two layers of steel. It can be seen that the cracking of the pavement has a significant effect on the magnitude of the crack opening.

The steel stress at the crack was the value that showed the greatest difference between the two pavements, as Figure 5.24 indicates. Even still, the behavior of both pavements was quite similar, with the steel stress of the pavement with one layer of reinforcing only about 2500 psi below the steel stress of the pavement with two layers of steel for the entire analysis period.

Figure 5.25 shows the concrete stress at mid-slab of both pavements plotted with the tensile strength of the concrete. Also, the effective concrete stress due to the wheel load is shown from Day 29 on. Again, there was virtually no difference between the one and two layer pavements, with the concrete stresses always within 4 psi of each other.

To gain a complete understanding of the behavior of both pavement systems may require the comparison of Figures 5.22 through 5.25 with each other. For example, the concrete stress at mid-slab, shown in

Table 5.2. Material Properties and Other Input Parameters Used for the Comparison of Pavements with One and Two Layers of Steel Reinforcement Over a 40 Day Analysis Period.

Elastic Modulus of the Steel: E_s = 29.0 x 10^6 psi

Concrete Coefficient of Thermal Contraction: α_c = 4.0 x 10^{-6} in/in°F

Steel Coefficient of Thermal Contraction: α_s = 6.0 x 10^{-6} in/in°F

28 Day Compressive Strength of the Concrete: f_c' = 5000.0 psi

Tensile Strength, Elastic Modulus of the Concrete, and Bond and Friction Coefficients are Calculated by the TTICRCP program.

Thickness of the Slabs: d = 12.0 in

Steel Diameter: d_s = 0.75 inches for the One Layer Design
= 0.625 inches for the Two Layer Design

Steel Spacing: b = 6.0 inches for the One Layer Design
= 9.0 inches for the Two Layer Design

Wheel Load: P = 9000.0 lbs
Radius of Contact Area: a = 5.0 in
Modulus of Subgrade Reaction: k = 150.0 lb/in^2/in
Loading Begins on Day 29

Relative Humidity = 70.0%

Concrete Curing Temperature = 80° F

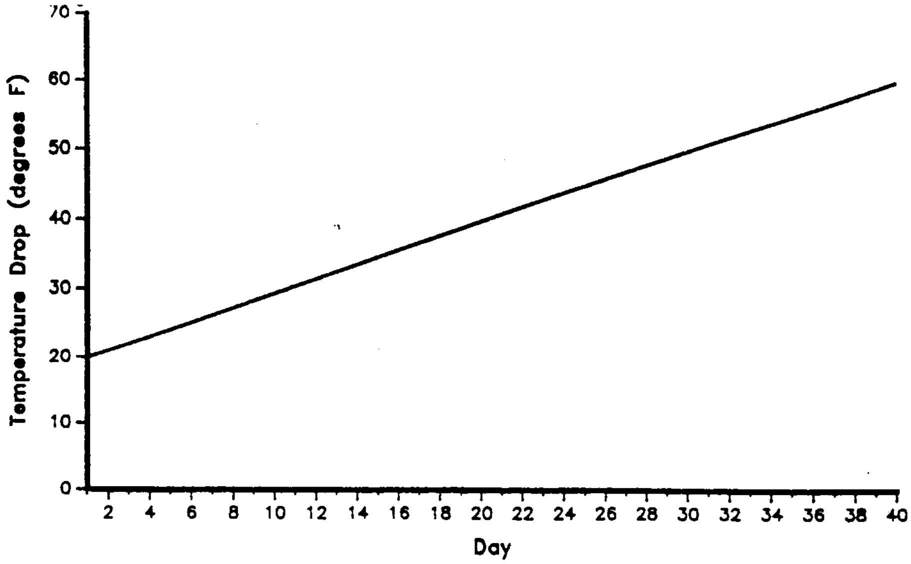


Figure 5.21a. Temperature Drop Distribution for the 40 Day Analysis Period.

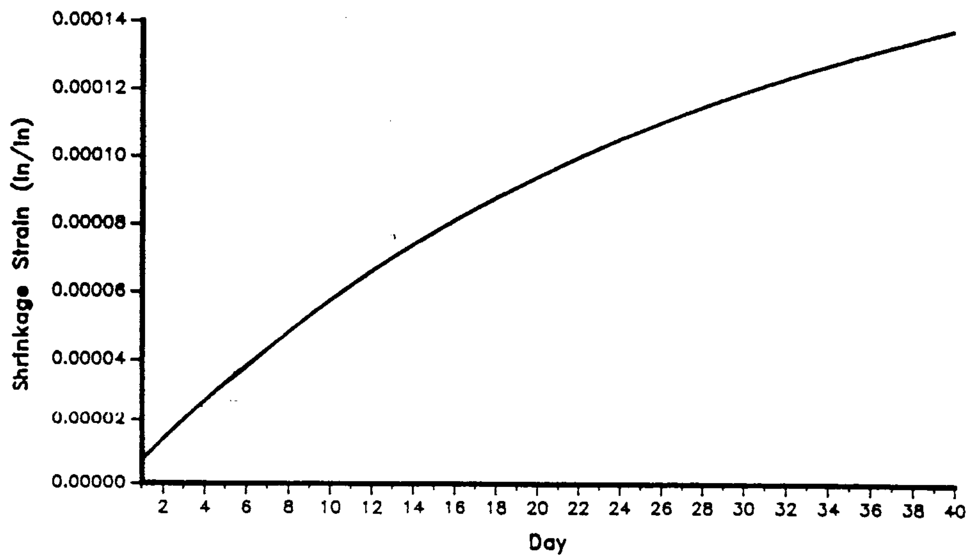


Figure 5.21b. Shrinkage Strain Distribution for the 40 Day Analysis Period.

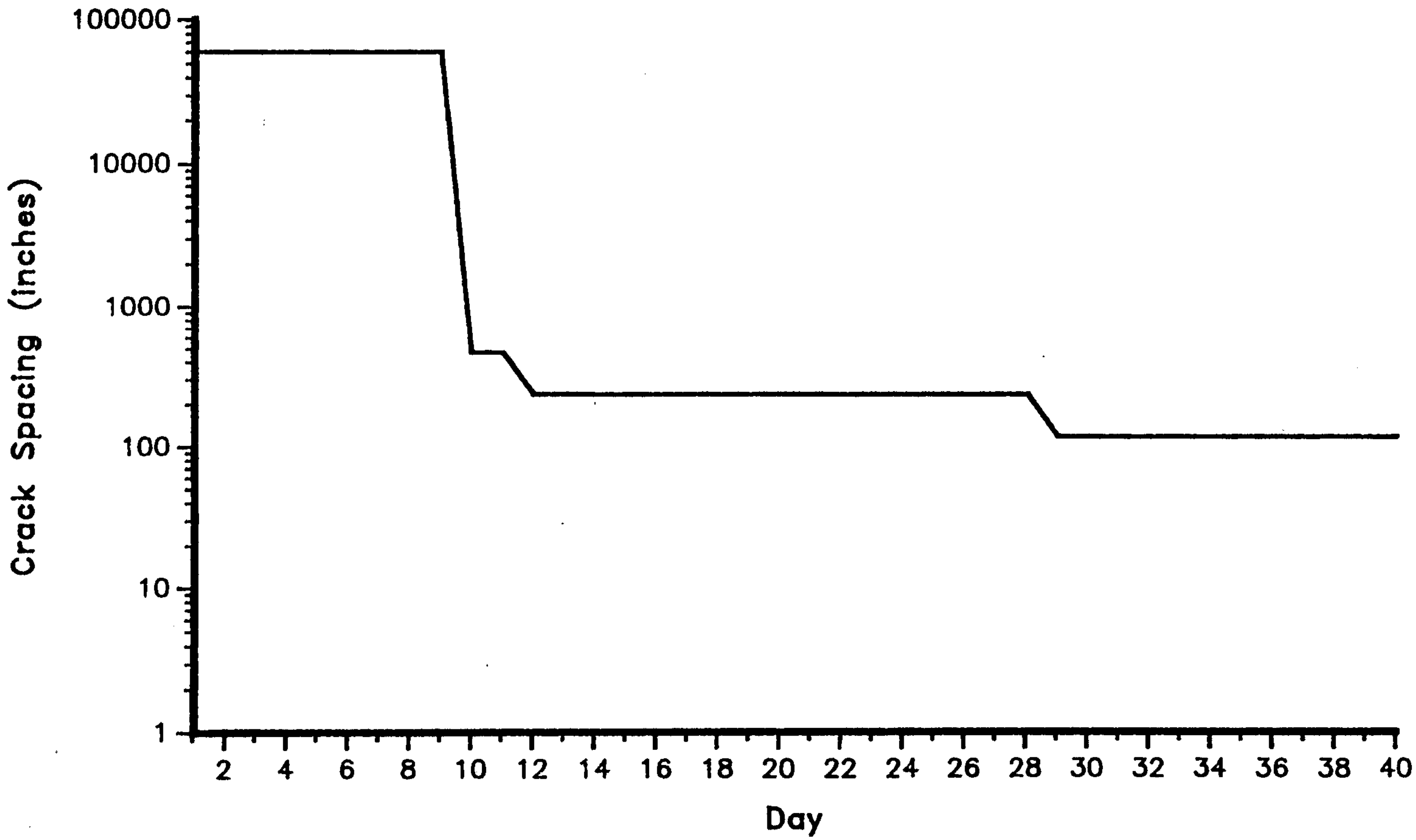


Figure 5.22 Variation of the Crack Spacing of Both Pavements over the Analysis Period.

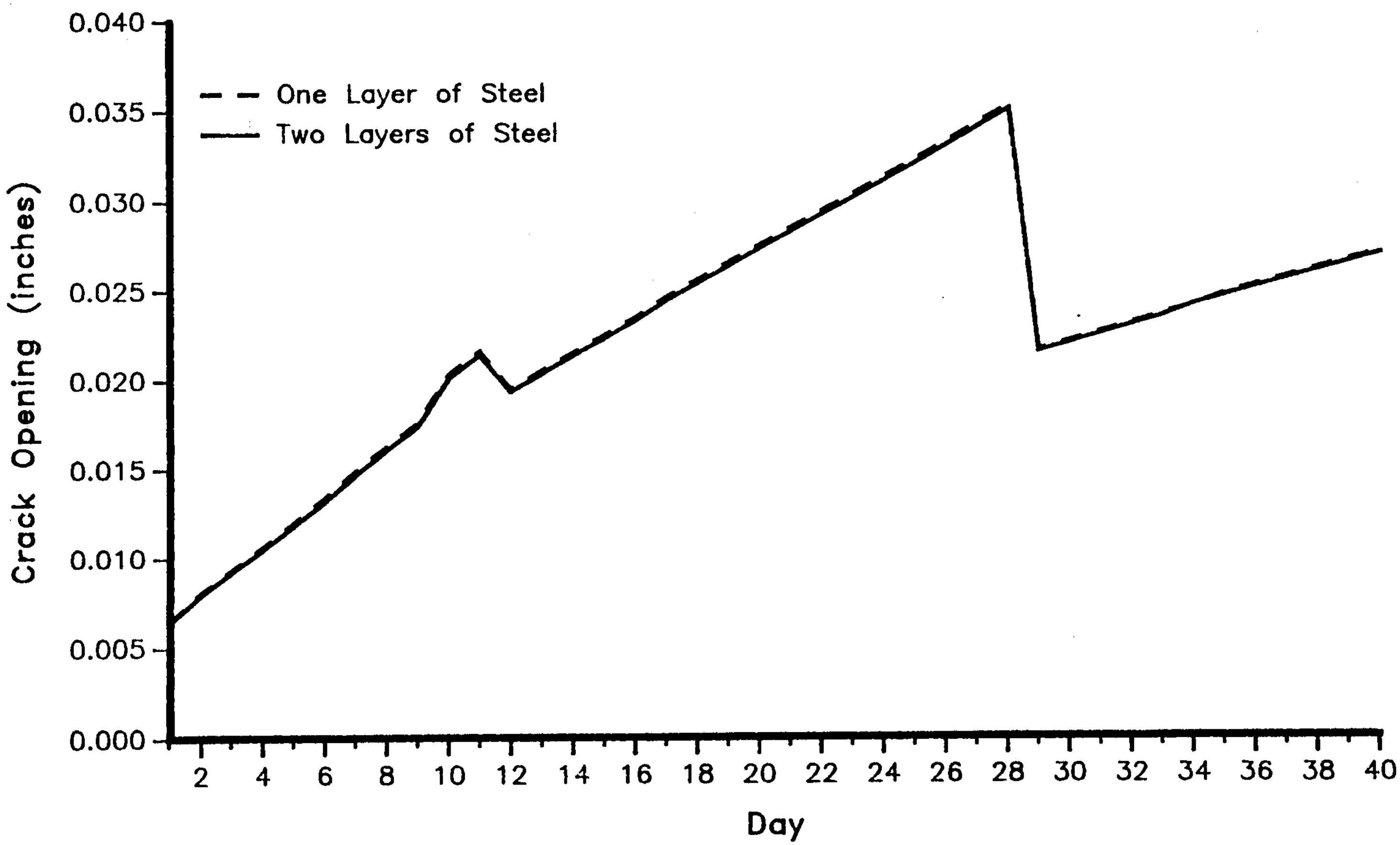


Figure 5.23. Variation of the Crack Opening of the Pavements over the Analysis Period.

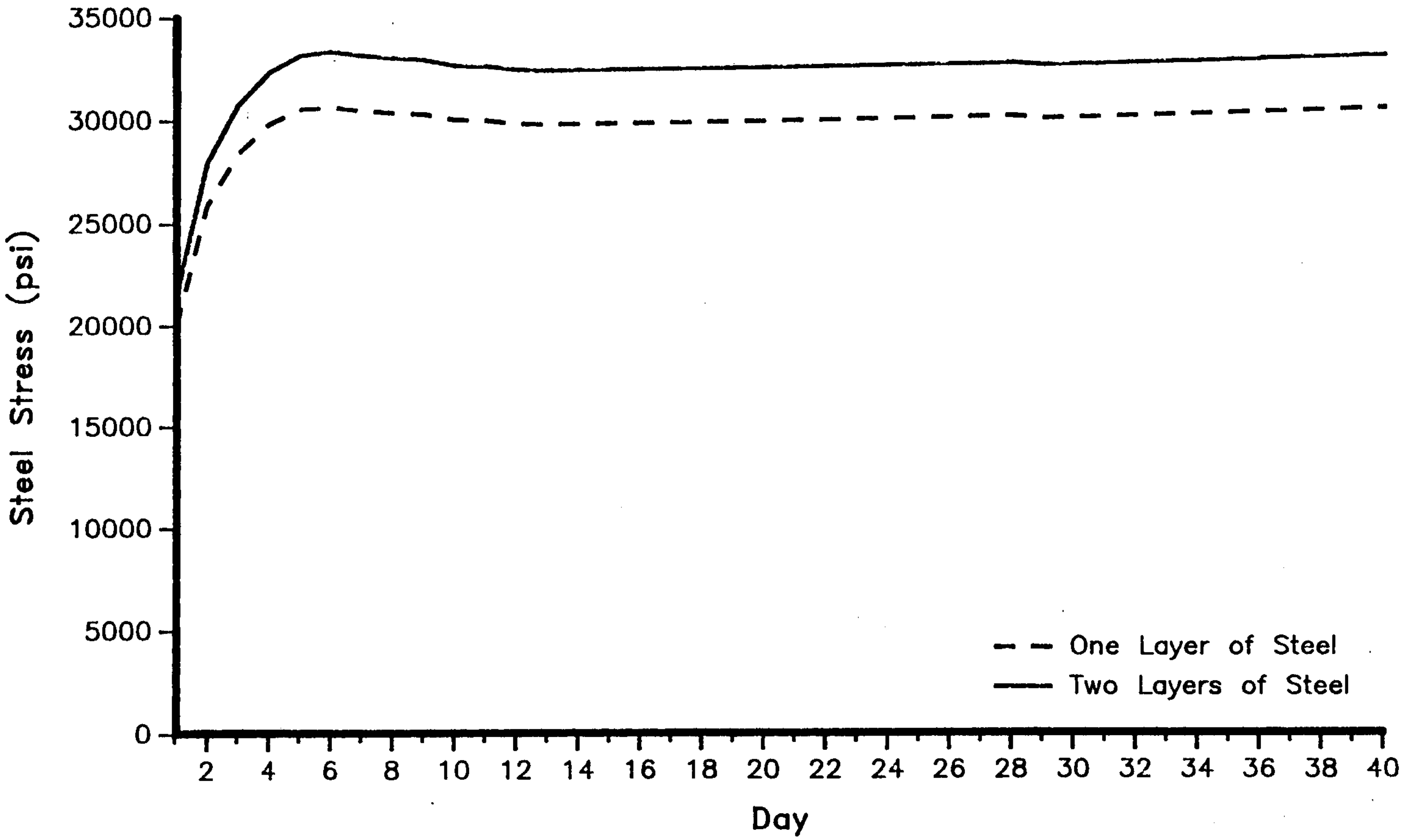


Figure 5.24. Variation of the Steel Stress at the Crack for the Two Pavements over the Analysis Period.

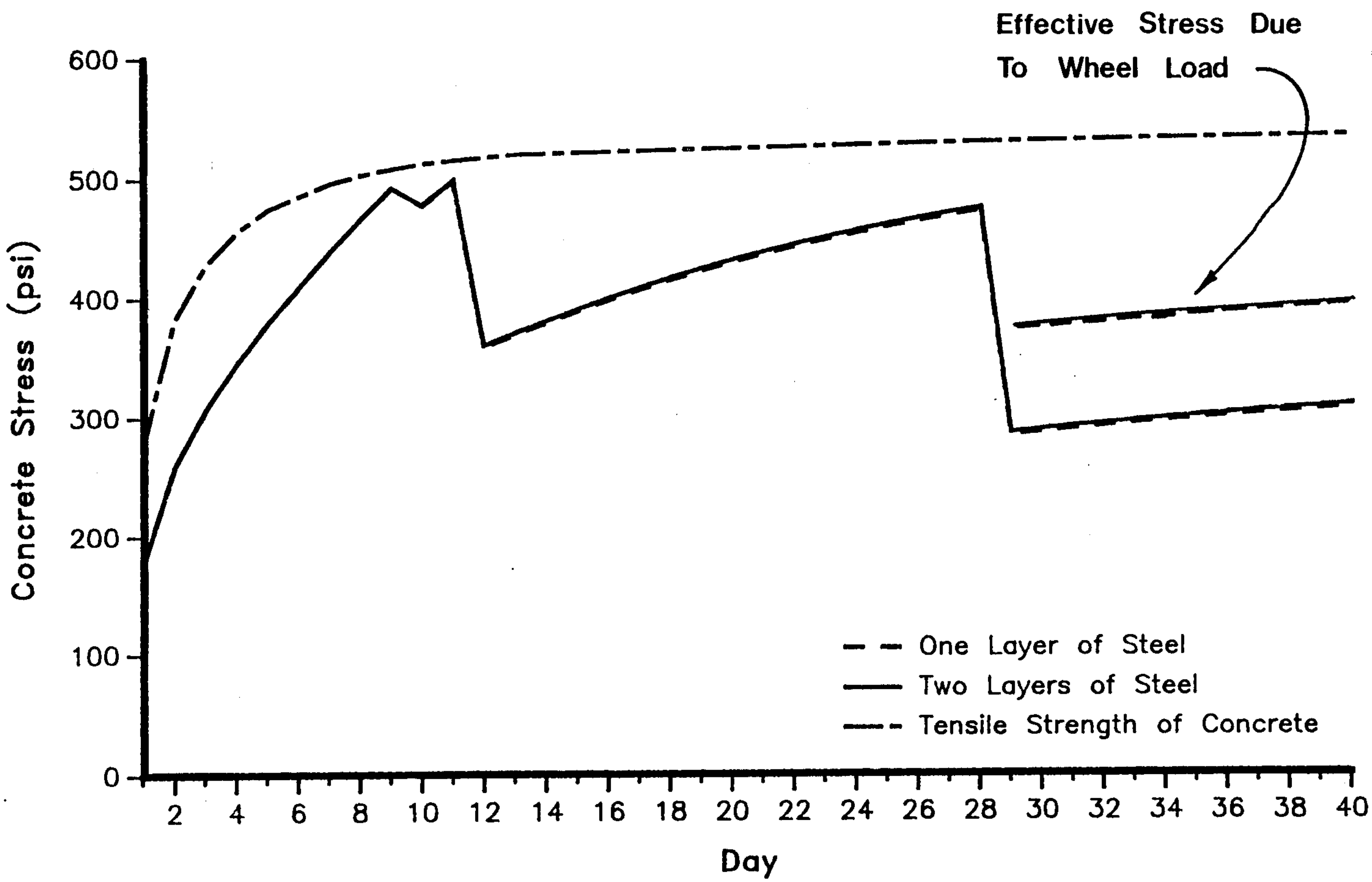


Figure 5.25. Variation of the Concrete Stress at Mid-Slab for the Two Pavements over the Analysis Period, with the Effective Stress Due to Wheel Loading Shown.

Figure 5.25, is seen to never exceed the tensile strength of the concrete. In actuality, the concrete stress at mid-slab does exceed the tensile strength many times, but each time the strength is exceeded, the TTICRCP model cracks the pavement slab in half. The slab may have to be cracked in half several times before the concrete stress at mid-slab falls below the tensile strength of the concrete. A comparison of Figure 5.22, the crack spacing of both pavements with respect to time, with Figure 5.25 verifies this.

Another reason for the similarity between the two types of pavement is that the steel percentage has been maintained constant in this analysis. From separate calculations using a simple free body diagram, similar ratios of steel and concrete stresses are observed between the two pavement types. Despite the similarity, however, the authors believe that the effect of the geometrical position of the steel layers is beneficial in reducing cracking due to curling and warping, and help extend the life of the CRC pavement.

SUMMARY

This chapter has presented a brief overview of the capabilities of the TTICRCP model, with a wide range of input and output data shown. The influence of many key input parameters has been seen, and the ability of the model to handle time dependency, wheel loads, and multiple layers of steel reinforcing was demonstrated. However, the full potential of the model remains to be explored. Recommendations for further development and use of the model, as well as a brief summary, will be included in Chapter VI.

CHAPTER VI

SUMMARY AND RECOMMENDATIONS

SUMMARY

Continuously reinforced concrete pavement is a complex structural system whose behavior is dependent on the interaction of many environmental, material, and mechanical variables. Predicting the reaction of the CRCP to changes in these variables is a difficult, but very necessary, task in the design and evaluation of new CRCP. Ouite often, the only predictor available is empirical observations from previously cast pavements, which provide a very limited range of information. A rational, computer-based method of predicting CRCP performance is available in the CRCP-1, 2, and 3 models developed by the Center for Transportation Research at the University of Texas, Austin.

The CRCP-1, 2, and 3 models are able to predict the behavior of the pavement system under a wide range of conditions, with the accuracy of the models being limited, in some part, by the mathematical simplicity of the model. Many of the assumptions that were made to develop the CRCP model compromise its accuracy,and although most are needed to reduce the complexity of the theory, others allow the introduction of gross approximations of actual behavior. The assumption that the bond stress distribution which occurs between the steel and the concrete can be modeled by an average bond stress acting over a calculated development length is an example.

To improve upon the existing CRCP models, it was thought possible to more accurately describe the actual bond stress distribution that occurs between the steel and the concrete. A search of the pertinent and current literature suggested that the local bond stress vs. slip relationship which exists between the concrete and the steel could be used as the basic building block on which to build the new model. The bond stress function was mathematically modeled as a three-part linear function, and a similar relationship was chosen to represent the friction stress-slip behavior that occurs between the concrete and the base material.

A set of differential equations was then generated to describe the concrete and steel behavior. The nature of the bond stress and friction stress functions allows the final solutions of the differential equations to be in one of nine different cases. The pavement slab in each case is divided into zones, with the most advanced cases having four zones. A system of linear, second order differential equations describes the concrete and steel behavior in each zone, and the equations for each zone are linked together by the boundary conditions to allow for their simultaneous solution. The exact solutions are found by using a matrix method to solve for the unknown, constant coefficients of the displacement functions.

The unique solution that correctly describes the concrete and steel behavior must satisfy two equilibrium conditions. First, the summation of forces acting on the slab must equal zero. This is automatically satisfied by any solution of the differential equations. Second, all of the energy that was made available to the slab through temperature drops and drying shrinkage must be accounted for. Three types of energy are used by the slab: potential energy stored in the concrete and the steel, frictional work energy expended by the slab as it moves, and stress relief energy, which is directly related to the physical strain that concrete exhibits.

To find the solution that satisfies the above two equilibrium conditions, a simple stepwise iteration method is used in the TTICRCP program. Since there can be as many as five additional unknown values along with the unknown coefficients, the iteration must proceed slowly to ensure that the approximations of the unknown values remain accurate. The correct solution of the equation system provides displacement and stress distributions for both the concrete and the steel. In additon, the bond stress and friction stress distributions can also be found.

The model is also capable of incorporating time dependent behavior and wheel loads into the solutions, which allows for the prediction of actual pavement system performance. The wheel load stress at the bottom fiber of the slab is approximated by Westergaard's equation for interior loading.

The range and capabilities of the model were briefly demonstrated, with the influence of several input parameters on key output values being shown and described. Also, the ability of the model to predict the CRCP system bahavior over a 40 day analysis period was shown.

The model provides the means to analyze a wide variety of pavement designs under the influence of extremely variable environmental conditions. Since the displacement functions that are solved by the model are closed-form, very detailed stress and displacement distributions can be generated for the steel and the concrete. The bond stress and friction stress distributions are also detailed and distinctly variable.

As with the CRCP models, many simplifying assumptions had to be made to mathematically develop the TTICRCP model, and they will limit the accuracy of the model to an extent. These assumptions include assuming linear elastic behavior of the materials and allowing the temperature differential and shrinkage strains to be one dimensional, which eliminates the effect of slab warpage. In addition, there is currently a lack of accurate data available to describe the bond stress-slip relationship fully. This will limit the accuracy of the model until sufficient data has been collected. The potential of this model to realistically predict the behavior of CRCP is great, and further developments will only serve to enhance its accuracy.

RECOMMENDATIONS

The TTICRCP model has not yet been validated by comparison with actual field data. This should be considered a necessary step if the full potential of the model is to be realized.

Several enhancements to the model would improve its accuracy and performance. They include:

1. Incorporating a more realistic model of the drying shrinkage that the slab experiences in place of the function that is now used. Of particular interest is the effect of daily changes in the relative humidity and other environmental parameters.

2. Modifying the model, and possibly reworking the general differential equations, to allow for the bond stress function to be dependent upon x as well as on the relative slip between the steel and the concrete.

3. Finding a faster method of iterating to find the correct solution. Currently, the model takes up to several hours of CPU time on a VAX 8800 to work through a given analysis, and finding the correct solution for each day's displacements and stresses takes approximately 3 to 5 minutes of CPU time.

4. Including the effect of the fracture of the concrete due to repeated tensile stresses and incorporating the energy used in the fracturing process as a part of the energy balance on those days when the slab cracks in tension. This will permit the slab to crack without exceeding the tensile strength of the concrete, as is common in all fatigue processes.

It is hoped that the abilities of the model will be further explored in future research, and that CRCP design methods can be improved by the TTICRCP solutions.

REFERENCES

1. McCullough, B.F. and Ledbetter, W.B. "LTS Design of Continuously Reinforced Concrete Pavement," Journal of the Highway Division, ASCE Proceedings, 1960, pp. 1-24.

2. Transportation Research Board. NCHRP Synthesis of Highway Practice 60: Failure and Repair of Continuously Reinforced Concrete Pavement, National Research Council, Washington, D.C. 1979, 42 pp.

3. Ma, J. and McCullough, B.F. CRCP-2, An Improved Computer Program for the Analysis of Continuously Reinforced Concrete Pavements, Research Report No. 177-9, Center for Highway Research, Austin, Texas, 1977, 174 pp.

4. ACI Committee 408. "Bond Stress - The State of the Art," ACI Journal, Proceedings, V. 63, No. 11, Nov. 1966, pp. 1161-1189.

5. Lutz, L.A. "The Mechanics of Bond and Slip of Deformed Reinforcing Bars in Concrete," Ph.D. Dissertation, Cornell University, Ithaca, NY, 1966.

6. Mylrea, T.D. "Bond and Anchorage," ACI Journal, Proceedings, Vol. 44, March 1948, pp. 521-552.

7. Watstein, D. "Bond Stress in Concrete Pull-Out Specimens," ACI Journal, Proceedings, Vol. 38, Sept. 1941, pp. 37-50.

8. Watstein, D. "Distribution of Bond Stresses in Concrete Pull-Out Specimens," ACI Journal, Proceedings, Vol. 43, May 1947, pp. 37-50.

9. Mains, R.M. "Measurement of the Distribution of Tensile Stresses Along Reinforcing Bars," ACI Journal, Proceedings, Vol. 48, Nov. 1951, pp. 225-252.

10. Wahla, M.I. "Direct Measurement of Bond Slip in Reinforced Concrete," Ph.D. Dissertation, Cornell University, Ithaca, NY, 1970.

11. Nilson, A.H. Bond Stress Relations in Reinforced Concrete, Research Report No. 345, Dept. of Structural Eng., Cornell University, Ithaca, NY, 1971.

12. Edwards, A.D. and Yannapoulos, P.J. "Local Bond-Stress to Slip Relationships for Hot Rolled Deformed Bars and Mild Steel Plain Bars," ACI Journal, Proceedings, Vol. 76, Feb. 1979, pp. 297-309.

13. Diaz, A.M., Burns, N.H. and McCullough, B.F. Behavior of Long Prestressed Pavement Slabs and Design Methodology, Research Report 401-3, Center for Transportation Research, The University of Texas, Austin, 1986, 284 pp.

14. Burgreen, D. Elements of Thermal Stress Analysis, C.P. Press, Jamaica, NY, 1971.

15. Feineman, G. et al. Applied Differential Equations, Spartan Books, Inc., Washington, D.C., 1965.

16. Mindess, S. and Young, F.J. Concrete, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.

APPENDIX A

General Solutions of Cases 1 through 9

Case 1 General Solutions

Zone 1

$$u_{c11} = A_{c11}e^{g_{11}x} + B_{c11}e^{-g_{11}x} + C_{c11}e^{h_{11}x} + D_{c11}e^{-h_{11}x}$$

$$u_{s11} = r_1 A_{c11}e^{g_{11}x} + r_1 B_{c11}e^{-g_{11}x} + s_1 C_{c11}e^{h_{11}x} + s_1 D_{c11}e^{-h_{11}x}$$

$$g_{11} = \left(\frac{(a_1+c_1)}{2} + \frac{(a_1+c_1)}{2}\left(1 - \frac{4(a_1c_1 - a_1d_1)}{(a_1+c_1)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$h_{11} = \left(\frac{(a_1+c_1)}{2} - \frac{(a_1+c_1)}{2}\left(1 - \frac{4(a_1c_1 - a_1d_1)}{(a_1+c_1)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$a_1 = \left[\frac{K_1\pi d_s}{E_s A_s}\right] \qquad c_1 = \left[\frac{K_1\pi d_s + K_3 b}{E_c A_c}\right]$$

$$d_1 = \left[\frac{K_1\pi d_s}{E_c A_c}\right]$$

$$r_1 = \left[a_1/(a_1 - g_{11}^2)\right] \qquad s_1 = \left[a_1/(a_1 - h_{11}^2)\right]$$

Case 2 General Solutions

Zone 1

$$u_{c21} = A_{c21} e^{g_{21} x} + B_{c21} e^{-g_{21} x} + C_{c21} e^{h_{21} x} + D_{c21} e^{-h_{21} x}$$

$$u_{s21} = r_1 A_{c21} e^{g_{21} x} + r_1 B_{c21} e^{-g_{21} x} + s_1 C_{c21} e^{h_{21} x} + s_1 D_{c21} e^{-h_{21} x}$$

$$g_{21} = \left(\frac{(a_1 + c_1)}{2} + \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$h_{21} = \left(\frac{(a_1 + c_1)}{2} - \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_1 = \left[\frac{K_1 \pi d_s}{E_s A_s} \right] \qquad c_1 = \left[\frac{K_1 \pi d_s + K_3 b}{E_c A_c} \right]$$

$$d_1 = \left[\frac{K_1 \pi d_s}{E_c A_c} \right]$$

$$r_1 = \left[a_1 / (a_1 - g_{21}^2) \right] \qquad s_1 = \left[a_1 / (a_1 - h_{21}^2) \right]$$

Zone 2

$$u_{c22} = A_{c22} e^{g_{22} x} + B_{c22} e^{-g_{22} x} + C_{c22} \cos h_{22} x + D_{c22} \sin h_{22} x + \left[\frac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)} \right]$$

$$u_{s22} = r_2 A_{c22} e^{g_{22} x} + r_2 B_{c22} e^{-g_{22} x} + s_2 C_{c22} \cos h_{22} x + s_2 D_{c22} \sin h_{22} x + \left[\frac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)} \right]$$

$$g_{22} = \left(\frac{(a_2 + c_2)}{2} - \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$h_{22} = \left(\frac{-(a_2 + c_2)}{2} - \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_2 = \left[\frac{K_2 \pi d_s}{E_s A_s} \right] \qquad b_2 = \left[\frac{-C_1 \pi d_s}{E_s A_s} \right]$$

$$c_2 = \left[\frac{K_2 \pi d_s + K_3 b}{E_c A_c} \right] \qquad d_2 = \left[\frac{K_2 \pi d_s}{E_c A_c} \right]$$

$$e_2 = \left[\frac{C_1 \pi d_s}{E_c A_c} \right]$$

$$r_2 = \left[a_2 / (a_2 - g_{22}^2) \right] \qquad s_2 = \left[a_2 / (a_2 + h_{22}^2) \right]$$

Case 3 General Solutions

Zone 1

$$u_{c31} = A_{c31}\,e^{g_{31}x} + B_{c31}\,e^{-g_{31}x} + C_{c31}\,e^{h_{31}x} + D_{c31}\,e^{-h_{31}x}$$

$$u_{s31} = r_1\,A_{c31}\,e^{g_{31}x} + r_1\,B_{c31}\,e^{-g_{31}x} + s_1\,C_{c31}\,e^{h_{31}x} + s_1\,D_{c31}\,e^{-h_{31}x}$$

$$g_{31} = \left(\frac{(a_1+c_1)}{2} + \frac{(a_1+c_1)}{2}\left(1 - \frac{4\,(a_1c_1 - a_1d_1)}{(a_1+c_1)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$h_{31} = \left(\frac{(a_1+c_1)}{2} - \frac{(a_1+c_1)}{2}\left(1 - \frac{4\,(a_1c_1 - a_1d_1)}{(a_1+c_1)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$a_1 = \left[\frac{K_1\pi d_s}{E_s A_s}\right] \qquad c_1 = \left[\frac{K_1\pi d_s + K_3 b}{E_c A_c}\right]$$

$$d_1 = \left[\frac{K_1\pi d_s}{E_c A_c}\right]$$

$$r_1 = \left[a_1/(a_1 - g_{31}^2)\right] \qquad s_1 = \left[a_1/(a_1 - h_{31}^2)\right]$$

Zone 2

$$u_{c32} = A_{c32}\,e^{g_{32}x} + B_{c32}\,e^{-g_{32}x} + C_{c32}\cos h_{32}x + D_{c32}\sin h_{32}x + \left[\frac{b_2d_2 + a_2e_2}{a_2(d_2 - c_2)}\right]$$

$$u_{s32} = r_2\,A_{c32}\,e^{g_{32}x} + r_2\,B_{c32}\,e^{-g_{32}x} + s_2\,C_{c32}\cos h_{32}x + s_2\,D_{c32}\sin h_{32}x + \left[\frac{a_2e_2 + b_2c_2}{a_2(d_2 - c_2)}\right]$$

$$g_{32} = \left(\frac{(a_2+c_2)}{2} - \frac{(a_2+c_2)}{2}\left(1 - \frac{4\,(a_2c_2 - a_2d_2)}{(a_2+c_2)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$h_{32} = \left(\frac{-(a_2+c_2)}{2} - \frac{(a_2+c_2)}{2}\left(1 - \frac{4\,(a_2c_2 - a_2d_2)}{(a_2+c_2)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$a_2 = \left[\frac{K_2\pi d_s}{E_s A_s}\right] \qquad b_2 = \left[\frac{-C_1\pi d_s}{E_s A_s}\right]$$

$$c_2 = \left[\frac{K_2\pi d_s + K_3 b}{E_c A_c}\right] \qquad d_2 = \left[\frac{K_2\pi d_s}{E_c A_c}\right]$$

$$e_2 = \left[\frac{C_1\pi d_s}{E_c A_c}\right]$$

$$r_2 = \left[a_2/(a_2 - g_{32}^2)\right] \qquad s_2 = \left[a_2/(a_2 + h_{32}^2)\right]$$

Case 3 Solutions Continued...

Zone 3

$$u_{c33} = A_{c33} e^{g_{33} x} + B_{c33} e^{-g_{33} x}$$

$$g_{33} = (a_3)^{\frac{1}{2}}$$

$$a_3 = \left[\frac{K_3 b}{E_c A_c} \right]$$

Case 4 General Solutions

Zone 1

$$u_{c41} = A_{c41} e^{g_{41} x} + B_{c41} e^{-g_{41} x} + C_{c41} e^{h_{41} x} + D_{c41} e^{-h_{41} x}$$

$$u_{s41} = r_1 A_{c41} e^{g_{41} x} + r_1 B_{c41} e^{-g_{41} x} + s_1 C_{c41} e^{h_{41} x} + s_1 D_{c41} e^{-h_{41} x}$$

$$g_{41} = \left(\frac{(a_1 + c_1)}{2} + \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$h_{41} = \left(\frac{(a_1 + c_1)}{2} - \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_1 = \left[\frac{K_1 \pi d_s}{E_s A_s} \right] \qquad c_1 = \left[\frac{K_1 \pi d_s + K_3 b}{E_c A_c} \right]$$

$$d_1 = \left[\frac{K_1 \pi d_s}{E_c A_c} \right]$$

$$r_1 = \left[a_1 / (a_1 - g_{41}^2) \right] \qquad s_1 = \left[a_1 / (a_1 - h_{41}^2) \right]$$

Zone 2

$$u_{c42} = A_{c42} e^{g_{42} x} + B_{c42} e^{-g_{42} x} + C_{c42} \cos h_{42} x + D_{c42} \sin h_{42} x + \left[\frac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)} \right]$$

$$u_{s42} = r_2 A_{c42} e^{g_{42} x} + r_2 B_{c42} e^{-g_{42} x} + s_2 C_{c42} \cos h_{42} x + s_2 D_{c42} \sin h_{42} x + \left[\frac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)} \right]$$

$$g_{42} = \left(\frac{(a_2 + c_2)}{2} - \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$h_{42} = \left(\frac{-(a_2 + c_2)}{2} - \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_2 = \left[\frac{K_2 \pi d_s}{E_s A_s} \right] \qquad b_2 = \left[\frac{-C_1 \pi d_s}{E_s A_s} \right]$$

$$c_2 = \left[\frac{K_2 \pi d_s + K_3 b}{E_c A_c} \right] \qquad d_2 = \left[\frac{K_2 \pi d_s}{E_c A_c} \right]$$

$$e_2 = \left[\frac{C_1 \pi d_s}{E_c A_c} \right]$$

$$r_2 = \left[a_2 / (a_2 - g_{42}^2) \right] \qquad s_2 = \left[a_2 / (a_2 + h_{42}^2) \right]$$

Case 4 Solutions Continued...

Zone 3

$$u_{c43} = A_{c43} \cos g_{43} x + B_{c43} \sin g_{43} x + C_{c43} \cos h_{43} x + D_{c43} \sin h_{43} x + \left[\frac{b_3 d_3 + a_3 e_3}{a_3 (d_3 - c_3)}\right]$$

$$u_{s43} = r_3 A_{c43} \cos g_{43} x + r_3 B_{c43} \sin g_{43} x + s_3 C_{c43} \cos h_{43} x + s_3 D_{c43} \sin h_{43} x + \left[\frac{a_3 e_3 + b_3 c_3}{a_3 (d_3 - c_3)}\right]$$

$$g_{43} = \left(\frac{-(a_3 + c_3)}{2} + \frac{(a_3 + c_3)}{2}\left(1 - \frac{4(a_3 c_3 - a_3 d_3)}{(a_3 + c_3)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$h_{43} = \left(\frac{-(a_3 + c_3)}{2} - \frac{(a_3 + c_3)}{2}\left(1 - \frac{4(a_3 c_3 - a_3 d_3)}{(a_3 + c_3)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$a_3 = \left[\frac{K_2 \pi d_s}{E_s A_s}\right] \qquad b_3 = \left[\frac{-C_1 \pi d_s}{E_s A_s}\right]$$

$$c_3 = \left[\frac{K_2 \pi d_s + K_4 b}{E_c A_c}\right] \qquad d_3 = \left[\frac{K_2 \pi d_s}{E_c A_c}\right]$$

$$e_3 = \left[\frac{C_1 \pi d_s + C_2 b}{E_c A_c}\right]$$

$$r_3 = \left[a_3 / (a_3 + g_{43}^2)\right] \qquad s_3 = \left[a_3 / (a_3 + h_{43}^2)\right]$$

Case 5 General Solutions

Zone 1

$$u_{c51} = A_{c51}e^{g_{51}x} + B_{c51}e^{-g_{51}x} + C_{c51}e^{h_{51}x} + D_{c51}e^{-h_{51}x}$$

$$u_{s51} = r_1 A_{c51}e^{g_{51}x} + r_1 B_{c51}e^{-g_{51}x} + s_1 C_{c51}e^{h_{51}x} + s_1 D_{c51}e^{-h_{51}x}$$

$$g_{51} = \left(\frac{(a_1+c_1)}{2} + \frac{(a_1+c_1)}{2}\left(1 - \frac{4(a_1c_1 - a_1d_1)}{(a_1+c_1)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$h_{51} = \left(\frac{(a_1+c_1)}{2} - \frac{(a_1+c_1)}{2}\left(1 - \frac{4(a_1c_1 - a_1d_1)}{(a_1+c_1)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$a_1 = \left[\frac{K_1 \pi d_s}{E_s A_s}\right] \qquad c_1 = \left[\frac{K_1 \pi d_s + K_3 b}{E_c A_c}\right]$$

$$d_1 = \left[\frac{K_1 \pi d_s}{E_c A_c}\right]$$

$$r_1 = \left[a_1/(a_1 - g_{51}^2)\right] \qquad s_1 = \left[a_1/(a_1 - h_{51}^2)\right]$$

Zone 2

$$u_{c52} = A_{c52}e^{g_{52}x} + B_{c52}e^{-g_{52}x} + C_{c52}\cos h_{52}x + D_{c52}\sin h_{52}x + \left[\frac{b_2d_2 + a_2e_2}{a_2(d_2 - c_2)}\right]$$

$$u_{s52} = r_2 A_{c52}e^{g_{52}x} + r_2 B_{c52}e^{-g_{52}x} + s_2 C_{c52}\cos h_{52}x + s_2 D_{c52}\sin h_{52}x + \left[\frac{a_2e_2 + b_2c_2}{a_2(d_2 - c_2)}\right]$$

$$g_{52} = \left(\frac{(a_2+c_2)}{2} - \frac{(a_2+c_2)}{2}\left(1 - \frac{4(a_2c_2 - a_2d_2)}{(a_2+c_2)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$h_{52} = \left(\frac{-(a_2+c_2)}{2} - \frac{(a_2+c_2)}{2}\left(1 - \frac{4(a_2c_2 - a_2d_2)}{(a_2+c_2)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$a_2 = \left[\frac{K_2 \pi d_s}{E_s A_s}\right] \qquad b_2 = \left[\frac{-C_1 \pi d_s}{E_s A_s}\right]$$

$$c_2 = \left[\frac{K_2 \pi d_s + K_3 b}{E_c A_c}\right] \qquad d_2 = \left[\frac{K_2 \pi d_s}{E_c A_c}\right]$$

$$e_2 = \left[\frac{C_1 \pi d_s}{E_c A_c}\right]$$

$$r_2 = \left[a_2/(a_2 - g_{52}^2)\right] \qquad s_2 = \left[a_2/(a_2 + h_{52}^2)\right]$$

Case 5 Solutions Continued...

Zone 3

$$u_{c53} = A_{c53}\, e^{g_{53} x} + B_{c53}\, e^{-g_{53} x}$$

$$g_{53} = (a_3)^{\frac{1}{2}}$$

$$a_3 = \left[\frac{K_3\, b}{E_c\, A_c}\right]$$

Zone 4

$$u_{c54} = A_{c54} \cos g_{54} x + B_{c54} \sin g_{54} x - \left[\frac{b_4}{a_4}\right]$$

$$g_{54} = (-a_4)^{\frac{1}{2}}$$

$$a_4 = \left[\frac{K_4\, b}{E_c\, A_c}\right] \qquad b_4 = \left[\frac{C_2\, b}{E_c\, A_c}\right]$$

Case 6 General Solutions

Zone 1

$$u_{c61} = A_{c61} e^{g_{61} x} + B_{c61} e^{-g_{61} x} + C_{c61} e^{h_{61} x} + D_{c61} e^{-h_{61} x}$$

$$u_{s61} = r_1 A_{c61} e^{g_{61} x} + r_1 B_{c61} e^{-g_{61} x} + s_1 C_{c61} e^{h_{61} x} + s_1 D_{c61} e^{-h_{61} x}$$

$$g_{61} = \left(\frac{(a_1 + c_1)}{2} + \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2}\right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$h_{61} = \left(\frac{(a_1 + c_1)}{2} - \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2}\right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_1 = \left[\frac{K_1 \pi d_s}{E_s A_s}\right] \qquad c_1 = \left[\frac{K_1 \pi d_s + K_3 b}{E_c A_c}\right]$$

$$d_1 = \left[\frac{K_1 \pi d_s}{E_c A_c}\right]$$

$$r_1 = \left[a_1 / (a_1 - g_{61}^2)\right] \qquad s_1 = \left[a_1 / (a_1 - h_{61}^2)\right]$$

Zone 2

$$u_{c62} = A_{c62} e^{g_{62} x} + B_{c62} e^{-g_{62} x} + C_{c62} \cos h_{62} x + D_{c62} \sin h_{62} x + \left[\frac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)}\right]$$

$$u_{s62} = r_2 A_{c62} e^{g_{62} x} + r_2 B_{c62} e^{-g_{62} x} + s_2 C_{c62} \cos h_{62} x + s_2 D_{c62} \sin h_{62} x + \left[\frac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)}\right]$$

$$g_{62} = \left(\frac{(a_2 + c_2)}{2} - \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2}\right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$h_{62} = \left(\frac{-(a_2 + c_2)}{2} - \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2}\right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_2 = \left[\frac{K_2 \pi d_s}{E_s A_s}\right] \qquad b_2 = \left[\frac{-C_1 \pi d_s}{E_s A_s}\right]$$

$$c_2 = \left[\frac{K_2 \pi d_s + K_3 b}{E_c A_c}\right] \qquad d_2 = \left[\frac{K_2 \pi d_s}{E_c A_c}\right]$$

$$e_2 = \left[\frac{C_1 \pi d_s}{E_c A_c}\right]$$

$$r_2 = \left[a_2 / (a_2 - g_{62}^2)\right] \qquad s_2 = \left[a_2 / (a_2 + h_{62}^2)\right]$$

Case 6 Solutions Continued...

Zone 3

$$u_{c63} = A_{c63}\cos g_{63}x + B_{c63}\sin g_{63}x + C_{c63}\cos h_{63}x + D_{c63}\sin h_{63}x + \left[\frac{b_3 d_3 + a_3 e_3}{a_3(d_3 - c_3)}\right]$$

$$u_{s63} = r_3 A_{c63}\cos g_{63}x + r_3 B_{c63}\sin g_{63}x + s_3 C_{c63}\cos h_{63}x + s_3 D_{c63}\sin h_{63}x + \left[\frac{a_3 e_3 + b_3 c_3}{a_3(d_3 - c_3)}\right]$$

$$g_{63} = \left(\frac{-(a_3+c_3)}{2} + \frac{(a_3+c_3)}{2}\left(1 - \frac{4(a_3c_3 - a_3d_3)}{(a_3+c_3)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$h_{63} = \left(\frac{-(a_3+c_3)}{2} - \frac{(a_3+c_3)}{2}\left(1 - \frac{4(a_3c_3 - a_3d_3)}{(a_3+c_3)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$a_3 = \left[\frac{K_2\pi d_s}{E_s A_s}\right] \qquad b_3 = \left[\frac{-C_1\pi d_s}{E_s A_s}\right]$$

$$c_3 = \left[\frac{K_2\pi d_s + K_4 b}{E_c A_c}\right] \qquad d_3 = \left[\frac{K_2\pi d_s}{E_c A_c}\right]$$

$$e_3 = \left[\frac{C_1\pi d_s + C_2 b}{E_c A_c}\right]$$

$$r_3 = \left[a_3/(a_3 + g_{63}^2)\right] \qquad s_3 = \left[a_3/(a_3 + h_{63}^2)\right]$$

Zone 4

$$u_{c64} = A_{c64}\cos g_{64}x + B_{c64}\sin g_{64}x - \left[\frac{b_4}{a_4}\right]$$

$$g_{64} = (-a_4)^{\frac{1}{2}}$$

$$a_4 = \left[\frac{K_4 b}{E_c A_c}\right] \qquad b_4 = \left[\frac{C_2 b}{E_c A_c}\right]$$

Case 7 General Solutions

Zone 1

$$u_{c71} = A_{c71} e^{g_{71} x} + B_{c71} e^{-g_{71} x} + C_{c71} e^{h_{71} x} + D_{c71} e^{-h_{71} x}$$

$$u_{s71} = r_1 A_{c71} e^{g_{71} x} + r_1 B_{c71} e^{-g_{71} x} + s_1 C_{c71} e^{h_{71} x} + s_1 D_{c71} e^{-h_{71} x}$$

$$g_{71} = \left(\frac{(a_1 + c_1)}{2} + \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$h_{71} = \left(\frac{(a_1 + c_1)}{2} - \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_1 = \left[\frac{K_1 \pi d_s}{E_s A_s} \right] \qquad c_1 = \left[\frac{K_1 \pi d_s + K_3 b}{E_c A_c} \right]$$

$$d_1 = \left[\frac{K_1 \pi d_s}{E_c A_c} \right]$$

$$r_1 = \left[a_1 / (a_1 - g_{71}^2) \right] \qquad s_1 = \left[a_1 / (a_1 - h_{71}^2) \right]$$

Zone 2

$$u_{c72} = A_{c72} e^{g_{72} x} + B_{c72} e^{-g_{72} x} + C_{c72} \cos h_{72} x + D_{c72} \sin h_{72} x + \left[\frac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)} \right]$$

$$u_{s72} = r_2 A_{c72} e^{g_{72} x} + r_2 B_{c72} e^{-g_{72} x} + s_2 C_{c72} \cos h_{72} x + s_2 D_{c72} \sin h_{72} x + \left[\frac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)} \right]$$

$$g_{72} = \left(\frac{(a_2 + c_2)}{2} + \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$h_{72} = \left(\frac{-(a_2 + c_2)}{2} + \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_2 = \left[\frac{K_1 \pi d_s}{E_s A_s} \right] \qquad c_2 = \left[\frac{K_1 \pi d_s + K_4 b}{E_c A_c} \right]$$

$$d_2 = \left[\frac{K_1 \pi d_s}{E_c A_c} \right] \qquad e_2 = \left[\frac{C_2 b}{E_c A_c} \right]$$

$$r_2 = \left[a_2 / (a_2 - g_{72}^2) \right] \qquad s_2 = \left[a_2 / (a_2 + h_{72}^2) \right]$$

Case 8 General Solutions

Zone 1

$$u_{c81} = A_{c81} e^{g_{81} x} + B_{c81} e^{-g_{81} x} + C_{c81} e^{h_{81} x} + D_{c81} e^{-h_{81} x}$$

$$u_{s81} = r_1 A_{c81} e^{g_{81} x} + r_1 B_{c81} e^{-g_{81} x} + s_1 C_{c81} e^{h_{81} x} + s_1 D_{c81} e^{-h_{81} x}$$

$$g_{81} = \left(\frac{(a_1 + c_1)}{2} + \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2}\right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$h_{81} = \left(\frac{(a_1 + c_1)}{2} - \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2}\right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_1 = \left[\frac{K_1 \pi d_s}{E_s A_s}\right] \qquad c_1 = \left[\frac{K_1 \pi d_s + K_3 b}{E_c A_c}\right]$$

$$d_1 = \left[\frac{K_1 \pi d_s}{E_c A_c}\right]$$

$$r_1 = \left[a_1 / (a_1 - g_{81}^2)\right] \qquad s_1 = \left[a_1 / (a_1 - h_{81}^2)\right]$$

Zone 2

$$u_{c82} = A_{c82} e^{g_{82} x} + B_{c82} e^{-g_{82} x} + C_{c82} \cos h_{82} x + D_{c82} \sin h_{82} x + \left[\frac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)}\right]$$

$$u_{s82} = r_2 A_{c82} e^{g_{82} x} + r_2 B_{c82} e^{-g_{82} x} + s_2 C_{c82} \cos h_{82} x + s_2 D_{c82} \sin h_{82} x + \left[\frac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)}\right]$$

$$g_{82} = \left(\frac{(a_2 + c_2)}{2} + \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2}\right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$h_{82} = \left(\frac{-(a_2 + c_2)}{2} + \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2}\right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_2 = \left[\frac{K_1 \pi d_s}{E_s A_s}\right] \qquad c_2 = \left[\frac{K_1 \pi d_s + K_4 b}{E_c A_c}\right]$$

$$d_2 = \left[\frac{K_1 \pi d_s}{E_c A_c}\right] \qquad e_2 = \left[\frac{C_2 b}{E_c A_c}\right]$$

$$r_2 = \left[a_2 / (a_2 - g_{82}^2)\right] \qquad s_2 = \left[a_2 / (a_2 + h_{82}^2)\right]$$

Case 8 Solutions Continued...

Zone 3

$$u_{c83} = A_{c83}\cos g_{83}x + B_{c83}\sin g_{83}x + C_{c83}\cos h_{83}x + D_{c83}\sin h_{83}x + \left[\frac{b_3 d_3 + a_3 e_3}{a_3(d_3 - c_3)}\right]$$

$$u_{s83} = r_3 A_{c83}\cos g_{83}x + r_3 B_{c83}\sin g_{83}x + s_3 C_{c83}\cos h_{83}x + s_3 D_{c83}\sin h_{83}x + \left[\frac{a_3 e_3 + b_3 c_3}{a_3(d_3 - c_3)}\right]$$

$$g_{83} = \left(\frac{-(a_3+c_3)}{2} + \frac{(a_3+c_3)}{2}\left(1 - \frac{4(a_3c_3 - a_3d_3)}{(a_3+c_3)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$h_{83} = \left(\frac{-(a_3+c_3)}{2} - \frac{(a_3+c_3)}{2}\left(1 - \frac{4(a_3c_3 - a_3d_3)}{(a_3+c_3)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$a_3 = \left[\frac{K_2\pi d_s}{E_s A_s}\right] \qquad b_3 = \left[\frac{-C_1\pi d_s}{E_s A_s}\right]$$

$$c_3 = \left[\frac{K_2\pi d_s + K_4 b}{E_c A_c}\right] \qquad d_3 = \left[\frac{K_2\pi d_s}{E_c A_c}\right]$$

$$e_3 = \left[\frac{C_1\pi d_s + C_2 b}{E_c A_c}\right]$$

$$r_3 = \left[a_3/(a_3 + g_{83}^2)\right] \qquad s_3 = \left[a_3/(a_3 + h_{83}^2)\right]$$

Case 9 General Solutions

Zone 1

$$u_{c91} = A_{c91} e^{g_{91} x} + B_{c91} e^{-g_{91} x} + C_{c91} e^{h_{91} x} + D_{c91} e^{-h_{91} x}$$
$$u_{s91} = r_1 A_{c91} e^{g_{91} x} + r_1 B_{c91} e^{-g_{91} x} + s_1 C_{c91} e^{h_{91} x} + s_1 D_{c91} e^{-h_{91} x}$$

$$g_{91} = \left(\frac{(a_1 + c_1)}{2} + \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$
$$h_{91} = \left(\frac{(a_1 + c_1)}{2} - \frac{(a_1 + c_1)}{2} \left(1 - \frac{4(a_1 c_1 - a_1 d_1)}{(a_1 + c_1)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_1 = \left[\frac{K_1 \pi d_s}{E_s A_s} \right] \qquad c_1 = \left[\frac{K_1 \pi d_s + K_3 b}{E_c A_c} \right]$$
$$d_1 = \left[\frac{K_1 \pi d_s}{E_c A_c} \right]$$
$$r_1 = \left[a_1 / (a_1 - g_{91}^2) \right] \qquad s_1 = \left[a_1 / (a_1 - h_{91}^2) \right]$$

Zone 2

$$u_{c92} = A_{c92} e^{g_{92} x} + B_{c92} e^{-g_{92} x} + C_{c92} \cos h_{92} x + D_{c92} \sin h_{92} x + \left[\frac{b_2 d_2 + a_2 e_2}{a_2 (d_2 - c_2)} \right]$$
$$u_{s92} = r_2 A_{c92} e^{g_{92} x} + r_2 B_{c92} e^{-g_{92} x} + s_2 C_{c92} \cos h_{92} x + s_2 D_{c92} \sin h_{92} x + \left[\frac{a_2 e_2 + b_2 c_2}{a_2 (d_2 - c_2)} \right]$$

$$g_{92} = \left(\frac{(a_2 + c_2)}{2} + \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$
$$h_{92} = \left(\frac{-(a_2 + c_2)}{2} + \frac{(a_2 + c_2)}{2} \left(1 - \frac{4(a_2 c_2 - a_2 d_2)}{(a_2 + c_2)^2} \right)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$a_2 = \left[\frac{K_1 \pi d_s}{E_s A_s} \right] \qquad c_2 = \left[\frac{K_1 \pi d_s + K_4 b}{E_c A_c} \right]$$
$$d_2 = \left[\frac{K_1 \pi d_s}{E_c A_c} \right] \qquad e_2 = \left[\frac{C_2 b}{E_c A_c} \right]$$
$$r_2 = \left[a_2 / (a_2 - g_{92}^2) \right] \qquad s_2 = \left[a_2 / (a_2 + h_{92}^2) \right]$$

Case 9 Solutions Continued...

Zone 3

$$u_{c93} = A_{c93} \cos g_{93} x + B_{c93} \sin g_{93} x + C_{c93} \cos h_{93} x + D_{c93} \sin h_{93} x + \left[\frac{b_3 d_3 + a_3 e_3}{a_3 (d_3 - c_3)}\right]$$

$$u_{s93} = r_3 A_{c93} \cos g_{93} x + r_3 B_{c93} \sin g_{93} x + s_3 C_{c93} \cos h_{93} x + s_3 D_{c93} \sin h_{93} x + \left[\frac{a_3 e_3 + b_3 c_3}{a_3 (d_3 - c_3)}\right]$$

$$g_{93} = \left(\frac{-(a_3 + c_3)}{2} + \frac{(a_3 + c_3)}{2}\left(1 - \frac{4(a_3 c_3 - a_3 d_3)}{(a_3 + c_3)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$h_{93} = \left(\frac{-(a_3 + c_3)}{2} - \frac{(a_3 + c_3)}{2}\left(1 - \frac{4(a_3 c_3 - a_3 d_3)}{(a_3 + c_3)^2}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}$$

$$a_3 = \left[\frac{K_2 \pi d_s}{E_s A_s}\right] \qquad b_3 = \left[\frac{-C_1 \pi d_s}{E_s A_s}\right]$$

$$c_3 = \left[\frac{K_2 \pi d_s + K_4 b}{E_c A_c}\right] \qquad d_3 = \left[\frac{K_2 \pi d_s}{E_c A_c}\right]$$

$$e_3 = \left[\frac{C_1 \pi d_s + C_2 b}{E_c A_c}\right]$$

$$r_3 = \left[a_3 / (a_3 + g_{93}^2)\right] \qquad s_3 = \left[a_3 / (a_3 + h_{93}^2)\right]$$

Zone 4

$$u_{c94} = A_{c94} \cos g_{94} x + B_{c94} \sin g_{94} x - \left[\frac{b_4}{a_4}\right]$$

$$g_{94} = (-a_4)^{\frac{1}{2}}$$

$$a_4 = \left[\frac{K_4 b}{E_c A_c}\right] \qquad b_4 = \left[\frac{C_2 b}{E_c A_c}\right]$$

APPENDIX B

Using the TTICRCP Program

USING THE TTICRCP PROGRAM

The TTICRCP program is written in FORTRAN and requires extended floating point capability (Range = 10^{+308} to 10^{-308}) to run. If the program is not compiled or run in extended floating point mode, there is an excellent chance of having an arithmetic overflow error. This typically appears when setting up the boundary conditions at x = S in the VAL subroutines. To run TTICRCP without extended floating point capabilities will require altering the program slightly. These modifications may affect the accuracy of the solutions to a slight extent.

The input file for TTICRCP is a simple, non-formatted list of the input data. No data files are opened inside the program. TTICRCP only requires that all data be given through unit #5, whatever that is assigned to be. Similarly, all of the output from TTICRCP is channeled into unit #6. Data files and output files can be, and definitely should be, used. However, the assignment of the input file to unit #5 and the output file to unit #6 must be accomplished outside of TTICRCP.

The Input File

The order that the data is read in by TTICRCP is given below. It is recommended that the data be entered as one value per line, as shown, even though it is possible to place more than one value on a line. If that is not satisfactory, the INPUT subroutine is well documented and a look at the READ statements can tell you which values may be placed on the same line. The variable name, a brief description of the variable, and the line number or location of that value in the input file is given. Also, the various options that are available for specifying the input data are described. The options are controlled by the flag variables FLG1 through FLG10. At the present time, several options are not implemented. Hopefully, these options will be activated in the near future.

The input file should resemble:

Line 1:	NUM	The number of runs that are included in the dataset. A complete dataset must be included for each run. NUM is only read once.
2:	SPACE	The initial crack spacing of the pavement at the beginning of the analysis period.
3:	STLDIA	The steel bar diameter.
4:	LAYERS	The number of steel layers.
5:	ES	The elastic modulus of the steel.
6:	ALPHAS	The steel's coefficient of thermal contraction.
7:	WIDT	The steel bar spacing.
8:	THICK	The thickness of the concrete slab.
9:	ULTSHR	The ultimate shrinkage coefficient.
10:	CURTEMP	The curing temperature of the pavement. This is used as a reference temperature to calculate the daily temperature drop.

Lines 2 - 10 are the non-time-dependent parameters. They will always be inputed in the above order.

As was mentioned above, there is a multitude of options for inputing the time-dependent parameters. The options that are selected are chosen by the use of flag variables, FLG1 through FLG10. Generally, these variables are set equal to 0.0 or 1.0, or possibly 2.0, to choose the option that will be used to input the data.

Line 11:	FLG1	If FLG1 = 1.0, then the analysis period is assumed to be one day. If FLG1 = 0.0, the length of the analysis period will be inputed by the user.

When FLG1 = 1.0, the rest of the parameters that describe the problem should be inputed in the following order. No options are available to change or modify this order.

FPRMC	The compressive strength of the concrete.
FPRMT	The tensile strength of the concrete.
ELASCON	The elastic modulus of the concrete.
ALPHACON	The concrete's coefficient of thermal contraction.
BONSLP(1,1)	K_1, the positive slope of the bond stress function.
BONSLP(2,1)	K_2, the negative slope of the bond stress function.
BONSLP(3,1)	K_3, the positive slope of the friction stress function.
BONSLP(4,1)	K_4, the negative slope of the friction stress function.
STLSLIP(1)	Delta b, the slip at which the maximum bond occurs.
CONSLIP(1)	Delta f, the slip at which the maximum friction occurs.
AMAGLOD	The magnitude of the wheel load. If a wheel load is not going to be applied, AMAGLOD should be set equal to 0.0.
SHRINKAGE	The shrinkage strain in the concrete.
MINTEMP	The day's minimum temperature.
RADLOD	The radius of the contact area of the wheel load.
BASEK	The modulus of subgrade reaction. Even if there is no wheel load, BASEK must be set equal to a value that is greater than 0.0. This is important!
TRIP	The number of increments that the output will give to describe the stress and displacement functions. The larger TRIP is, the more voluminous the output file will be.

This is probably a good place to restate that the mathematical model that was developed for TTICRCP assumed that K_1 and K_3 are always positive and K_1 is much greater than K_3. (How much is "much greater" is a question that has not been answered. If realistic values for K_1 and K_3 are used, there shouldn't be a problem.) Similarly, K_2 and K_4 **must** be negative and $|K_2|$ should be greater than $|K_4|$. If these assumptions are violated, TTICRCP will not work correctly and a fatal error will probably occur. Now, back to the input file...

If FLG1 = 0.0, then a whole slew of possibilities exist. Input the following:

MTIME	The number of days in the analysis period. Currently, the arrays in TTICRCP are dimensioned so that the maximum length of the analysis period is 100 days. If the analysis period has to be longer, the arrays will have to be redimensioned, but that should be the only necessary modification.
FLG2	Should be set equal to 1.0, since only one option presently exists for inputing the compressive strength data.
FPRMC(1) thru FPRMC(MTIME)	The compressive strength of the concrete must be given for each day in the analysis period.
FLG3	Can be 0.0 or 1.0.

If FLG3 = 1.0, the concrete's tensile strength must be inputed for each day.

FPRMT(1) thru FPRMT(MTIME)	Concrete tensile strength.

If FLG3 = 0.0, the tensile strength is approximated within TTICRCP from the compressive strength data.

FLG4	Can be 0.0 or 1.0

If FLG4 = 1.0, the concrete's elastic modulus must be given for each day.

ELASCON(1) thru ELASCON(MTIME)	The elastic modulus of the concrete.

IF FLG4 = 0.0, the elastic modulus is calculated from the compressive strength data.

FLG5	Can be 1.0 or 2.0.

If FLG5 = 1.0, then the concrete's coefficient of thermal contraction must be given for each day.

ALPHACON(1) Concrete's coefficient of thermal contraction.
thru
ALPHACON(MTIME)

If FLG5 = 2.0 the concrete's coefficient of thermal contraction is assumed to be constant over the entire length of the analysis period, and only one value needs to be given.

ALPHAC Concrete's coefficient of thermal contraction.

FLG6 Can be 0.0 or 1.0.

If FLG6 = 1.0, then the bond stress and friction stress coefficients, K_1, K_2, K_3, and K_4, must be inputed for each day.

BONSLP(1,1) K_1, the positive slope of the bond stress function.
BONSLP(2,1) K_2, the negative slope of the bond stress function.
BONSLP(3,1) K_3, the positive slope of the friction stress function.
BONSLP(4,1) K_4, the negative slope of the friction stress function.
thru

BONSLP(1,MTIME)
BONSLP(2,MTIME)
BONSLP(3,MTIME)
BONSLP(4,MTIME)

If FLG6 = 0.0 the bond stress and friction stress coefficients are approximated automatically by TTICRCP.

FLG7 Can be 0.0 or 1.0.

If FLG7 = 1.0, the slip value at which the maximum bond stress occurs and the displacement at which the maximum friction stress occurs, delta b and delta f, must be given for each day.

STLSLIP(1)	Delta b, the slip at which the maximum bond stress occurs.
CONSLIP(1)	Delta f, the displacement at which the maximum friction stress occurs.
thru	
STLSLIP(MTIME)	
CONSLIP(MTIME)	

If FLG7 = 0.0, the slip and displacement values are calculated by TTICRCP.

FLG8	Can be 0.0 or 1.0.

If FLG8 = 1.0, the minimum temperature for each day must be inputed.

MINTEMP(1)	The day's minimum temperature.
thru	
MINTEMP(MTIME)	

If FLG8 = 0.0, TTICRCP assumes that the daily minimum temperature distribution over the analysis period is linear. It is defined by the temperature on the first day of the analysis period and the temperature on the last day of the analysis period.

STARTEM	Minimum temperature on the first day.
STOPEM	Minimum temperature on the last day.

FLG9	Can be 0.0 or 1.0.

If FLG9 = 1.0, the relative humidity each day must be inputed.

RELHUM(1) thru RELHUM(MTIME)	The day's relative humidity.

If FLG9 = 0.0, the relative humidity is assumed to be constant over the analysis period.

RELAHU	Constant value for the relative humidity.
FLG10	Can be 0.0 or 1.0.

If FLG10 = 1.0, wheel loading is assumed to occur.

KDAYLOBEG	The day the wheel loading begins.
AMAGALOD	The magnitude of the wheel load.
RADLOD	Radius of the contact area.
BASEK	The modulus of subgrade reaction.

If FLG10 = 0.0, wheel loading is not included in the analysis.

TRIP	The number of increments that the program will use to spew out the displacement and stress distributions.

TRIP is always the last value in the dataset, no matter if it is a single-day analysis or a multi-day analysis. If more than 1 run is to be included in the dataset, all of the lines from Line 2 through the line on which TRIP appears need to be repeated. (This is not saying that each dataset must be identical to the other datasets. Rather, it is just a restating of a point made at the beginning of this section, which says that a complete dataset must be included for each run. The data values and number of lines in each dataset are unique to that dataset, and can be any valid combination of parameters and FLG variables.)

Examples

Two examples are now given to conclude this guide. They demonstrate how an actual input file is composed and how output from TTICRCP is presented. Example 1 is a single-day analysis, and Example 2 is a multi-day analysis.

Example 1: Input file

This is an input file for a single-day analysis. The actual data file contains only the numerical data. The text has been included for clarity

1	- Number of runs in the dataset
72.0	- Initial crack spacing
0.75	- Steel bar diameter
1	- Number of layers of Steel
29000000.0	- Steel's elastic modulus
0.000006	- Steel's coefficient of thermal contraction
6.0	- Spacing of the steel bars
12.0	- Thickness of the slab
0.0	- Ultimate shrinkage coefficient
80.0	- Curing temperature
1.0	- FLG1
5000.0	- Compressive strength of the concrete
700.0	- Tensile strength of the concrete
3500000.0	- Concrete's elastic modulus
0.000004	- Concrete's coefficient of thermal contraction
300000.0	- K1: coefficient for bond stress function
-100000.0	- K2: coefficient for bond stress function
5000.0	- K3: coefficient for friction stress function
-50.0	- K4: coefficient for friction stress function
0.002	- delta b: point of max. bond stress
0.01	- delta f: point of max. friction stress
0.0	- Magnitude of wheel load
0.00015	- Shrinkage strain in the concrete
40.0	- Minimum temperature
0.0	- Radius of tire contact area
1.0	- Modulus of subgrade reaction (must be > 0)
144.0	- Number of increments of x in output

Example 1: Output file

==

THE INPUT PARAMETERS ARE:

Length of Analysis Period = 1 day(s)
Initial Crack Spacing = 72.0 inches
Slab Thickness = 12.0 inches

Steel Bar Spacing = 6.0 inches
Steel Bar Diameter = 0.750 inches
Layers of Steel = 1

Cross-Sectional Area of the Steel = 0.442 sq. inches
Cross-Sectional Area of the Concrete = 71.56 sq. inches
Steel Percentage = 0.61%

Steel Modulus of Elasticity = 29000000.0 psi
Steel Coefficient of Thermal Expansion = 0.0000060 inches/inch-degree F

Concrete Curing Temperature = 80.0 degrees F
Initial Shrinkage of Concrete = 0.0001500 inches/inch
Ultimate Shrinkage of Concrete = 0.0003700 inches/inch

Wheel Loading of the Pavement Begins on Day 1
Magnitude of Wheel Load = 0.0 lbs
Radius of Tire Contact Area = 0.0 inches
Modulus of Subgrade Reaction = 1.0 lb/in≠2/in

Concrete Properties and Environmental Conditions

Day	Ec	f'c	f't	Alpha	Rel. Hum.	Min. Temp.
1	3500000.0	5000.0	700.0	0.0000040	100.0	40.00

Local Bond Stress / Slip Coefficients

Day	K1	K2	STLSLP	K3	K4	CONSLP
1	300000.0	-100000.0	0.0020	5000.0	-50.0	0.0100

==

****

Day 1

Temperature Drop = 40.0 Degrees F

Compressive Strength of the Concrete = 5000.0 psi
Tensile Strength of the Concrete = 700.0 psi
Elastic Modulus of the Concrete = 3500000.0 psi

K1 = 300000.0 pci STLSLP = 0.0020 inches
K2 = -100000.0 pci STLLIM = 0.0080 inches
K3 = 5000.0 pci CONSLP = 0.0100 inches
K4 = -50.0 pci CONLIM = 1.0100 inches

The Solution is in Case 2

The Crack Opening = 0.01335 inches
The Crack Spacing = 72.0 inches
The Calculated Value of U0 = 0.00667667
VL1 = 6.12 inches VL2 = 0.00 inches VL3 = 0.00 inches

The Shrinkage Strain = 0.0001500 inches/inch
Load-induced Stress at the Center of the Slab = 0.00 psi

The Total Energy Available = 216.8 inch-lbs
The Frictional Energy Used = 32.6 inch-lbs
The Potential Energy Stored = 27.5 inch-lbs
The Stress Relief Energy Used = 156.7 inch-lbs

The Force Exerted by the Subgrade = 3440.7 lbs

X	US	UC	BOND STR.	FRIC. STR.	ST. STRESS	CO. STRESS
0.00	0.000000	0.006678	132.179	33.391	27806.13	-0.05
0.25	0.000179	0.006620	155.850	33.102	27614.09	1.83
0.50	0.000356	0.006563	179.327	32.814	27390.62	3.91
0.75	0.000531	0.006505	202.584	32.526	27135.99	6.16
1.00	0.000704	0.006448	225.590	32.239	26850.51	8.60
1.25	0.000874	0.006391	248.319	31.954	26534.54	11.23
1.50	0.001041	0.006334	270.744	31.669	26188.46	14.03
1.75	0.001205	0.006277	292.836	31.385	25812.70	17.01
2.00	0.001366	0.006220	314.570	31.102	25407.72	20.17
2.25	0.001523	0.006164	335.920	30.821	24974.02	23.49
2.50	0.001677	0.006108	356.859	30.540	24512.12	26.99
2.75	0.001826	0.006052	377.362	30.261	24022.59	30.65
3.00	0.001971	0.005997	397.405	29.984	23506.02	34.47
3.25	0.002111	0.005941	416.963	29.707	22963.06	38.45
3.50	0.002247	0.005886	436.013	29.432	22394.35	42.58
3.75	0.002377	0.005832	454.532	29.159	21800.59	46.86
4.00	0.002502	0.005777	472.497	28.887	21182.51	51.28
4.25	0.002622	0.005723	489.888	28.617	20540.85	55.84

4.50	0.002736	0.005670	506.682	28.348	19876.41	60.54
4.75	0.002845	0.005616	522.860	28.082	19189.98	65.37
5.00	0.002947	0.005563	538.402	27.816	18482.40	70.33
5.25	0.003044	0.005511	553.289	27.553	17754.53	75.40
5.50	0.003133	0.005458	567.504	27.292	17007.26	80.59
5.75	0.003217	0.005406	581.029	27.032	16241.49	85.89
6.00	0.003293	0.005355	593.848	26.774	15458.16	91.29
6.25	0.003363	0.005304	582.160	26.518	14666.13	96.74
6.50	0.003426	0.005253	547.982	26.265	13912.93	101.94
6.75	0.003483	0.005203	515.804	26.013	13203.95	106.86
7.00	0.003534	0.005152	485.507	25.762	12536.61	111.53
7.25	0.003579	0.005103	456.982	25.514	11908.48	115.94
7.50	0.003620	0.005053	430.125	25.266	11317.26	120.12
7.75	0.003655	0.005004	404.839	25.021	10760.78	124.09
8.00	0.003685	0.004955	381.033	24.776	10237.03	127.84
8.25	0.003711	0.004907	358.619	24.533	9744.08	131.40
8.50	0.003733	0.004858	337.516	24.291	9280.13	134.78
8.75	0.003751	0.004810	317.648	24.051	8843.48	137.98
9.00	0.003766	0.004762	298.942	23.811	8432.55	141.02
9.25	0.003777	0.004715	281.332	23.573	8045.82	143.90
9.50	0.003785	0.004667	264.751	23.335	7681.87	146.64
9.75	0.003789	0.004620	249.141	23.099	7339.38	149.24
10.00	0.003791	0.004573	234.445	22.863	7017.09	151.72
10.25	0.003790	0.004526	220.609	22.628	6713.81	154.06
10.50	0.003787	0.004479	207.584	22.395	6428.44	156.30
10.75	0.003781	0.004432	195.320	22.161	6159.92	158.42
11.00	0.003773	0.004386	183.776	21.929	5907.27	160.44
11.25	0.003763	0.004339	172.907	21.697	5669.55	162.37
11.50	0.003751	0.004293	162.675	21.466	5445.90	164.20
11.75	0.003737	0.004247	153.042	21.236	5235.48	165.95
12.00	0.003721	0.004201	143.974	21.006	5037.54	167.61
12.25	0.003704	0.004155	135.437	20.777	4851.32	169.20
12.50	0.003685	0.004110	127.401	20.548	4676.15	170.72
12.75	0.003665	0.004064	119.836	20.320	4511.37	172.16
13.00	0.003643	0.004019	112.714	20.093	4356.39	173.54
13.25	0.003620	0.003973	106.010	19.865	4210.62	174.86
13.50	0.003595	0.003928	99.700	19.639	4073.52	176.12
13.75	0.003570	0.003883	93.759	19.413	3944.59	177.33
14.00	0.003543	0.003837	88.168	19.187	3823.34	178.48
14.25	0.003516	0.003792	82.905	18.961	3709.33	179.58
14.50	0.003487	0.003747	77.950	18.736	3602.12	180.64
14.75	0.003458	0.003702	73.287	18.512	3501.33	181.65
15.00	0.003428	0.003657	68.898	18.287	3406.57	182.62
15.25	0.003397	0.003613	64.767	18.063	3317.49	183.56
15.50	0.003365	0.003568	60.879	17.840	3233.75	184.45
15.75	0.003333	0.003523	57.220	17.616	3155.04	185.31
16.00	0.003299	0.003479	53.776	17.393	3081.07	186.13
16.25	0.003266	0.003434	50.535	17.171	3011.55	186.92
16.50	0.003231	0.003390	47.484	16.948	2946.22	187.68
16.75	0.003196	0.003345	44.614	16.726	2884.84	188.41
17.00	0.003161	0.003301	41.913	16.504	2827.18	189.12
17.25	0.003125	0.003256	39.371	16.282	2773.01	189.80
17.50	0.003089	0.003212	36.979	16.061	2722.12	190.45
17.75	0.003052	0.003168	34.729	15.840	2674.33	191.08

18.00	0.003015	0.003124	32.611	15.619	2629.45	191.69
18.25	0.002978	0.003080	30.619	15.398	2587.31	192.27
18.50	0.002940	0.003035	28.745	15.177	2547.75	192.84
18.75	0.002901	0.002991	26.982	14.957	2510.61	193.38
19.00	0.002863	0.002947	25.323	14.737	2475.75	193.91
19.25	0.002824	0.002903	23.763	14.517	2443.04	194.42
19.50	0.002785	0.002859	22.295	14.297	2412.34	194.91
19.75	0.002746	0.002815	20.915	14.077	2383.55	195.38
20.00	0.002706	0.002772	19.617	13.858	2356.53	195.84
20.25	0.002666	0.002728	18.396	13.639	2331.20	196.29
20.50	0.002626	0.002684	17.248	13.419	2307.44	196.72
20.75	0.002586	0.002640	16.169	13.200	2285.17	197.13
21.00	0.002546	0.002596	15.154	12.982	2264.30	197.54
21.25	0.002505	0.002553	14.200	12.763	2244.73	197.93
21.50	0.002464	0.002509	13.304	12.544	2226.40	198.31
21.75	0.002424	0.002465	12.461	12.326	2209.23	198.67
22.00	0.002383	0.002421	11.669	12.107	2193.15	199.03
22.25	0.002341	0.002378	10.924	11.889	2178.10	199.37
22.50	0.002300	0.002334	10.225	11.671	2164.00	199.71
22.75	0.002259	0.002291	9.567	11.453	2150.81	200.03
23.00	0.002217	0.002247	8.950	11.235	2138.47	200.34
23.25	0.002176	0.002204	8.370	11.018	2126.93	200.65
23.50	0.002134	0.002160	7.826	10.800	2116.13	200.94
23.75	0.002092	0.002116	7.314	10.582	2106.04	201.23
24.00	0.002050	0.002073	6.834	10.365	2096.62	201.51
24.25	0.002008	0.002030	6.384	10.148	2087.81	201.78
24.50	0.001966	0.001986	5.961	9.930	2079.58	202.04
24.75	0.001924	0.001943	5.564	9.713	2071.90	202.29
25.00	0.001882	0.001899	5.192	9.496	2064.73	202.54
25.25	0.001840	0.001856	4.843	9.279	2058.04	202.78
25.50	0.001797	0.001812	4.515	9.062	2051.81	203.01
25.75	0.001755	0.001769	4.208	8.846	2045.99	203.23
26.00	0.001713	0.001726	3.920	8.629	2040.58	203.45
26.25	0.001670	0.001682	3.651	8.412	2035.53	203.66
26.50	0.001628	0.001639	3.398	8.196	2030.83	203.86
26.75	0.001585	0.001596	3.161	7.979	2026.46	204.06
27.00	0.001543	0.001553	2.940	7.763	2022.40	204.25
27.25	0.001500	0.001509	2.732	7.546	2018.62	204.43
27.50	0.001458	0.001466	2.538	7.330	2015.11	204.61
27.75	0.001415	0.001423	2.356	7.114	2011.84	204.78
28.00	0.001372	0.001380	2.186	6.898	2008.82	204.94
28.25	0.001330	0.001336	2.028	6.681	2006.01	205.10
28.50	0.001287	0.001293	1.879	6.465	2003.40	205.26
28.75	0.001244	0.001250	1.740	6.249	2000.99	205.41
29.00	0.001201	0.001207	1.610	6.033	1998.76	205.55
29.25	0.001159	0.001164	1.489	5.818	1996.69	205.69
29.50	0.001116	0.001120	1.376	5.602	1994.79	205.82
29.75	0.001073	0.001077	1.270	5.386	1993.02	205.94
30.00	0.001030	0.001034	1.172	5.170	1991.39	206.06
30.25	0.000987	0.000991	1.080	4.954	1989.89	206.18
30.50	0.000944	0.000948	0.994	4.739	1988.51	206.29
30.75	0.000902	0.000905	0.914	4.523	1987.24	206.39
31.00	0.000859	0.000861	0.839	4.307	1986.07	206.49
31.25	0.000816	0.000818	0.769	4.092	1985.00	206.59

31.50	0.000773	0.000775	0.704	3.876	1984.02	206.68
31.75	0.000730	0.000732	0.643	3.661	1983.12	206.76
32.00	0.000687	0.000689	0.586	3.445	1982.30	206.84
32.25	0.000644	0.000646	0.533	3.230	1981.56	206.92
32.50	0.000601	0.000603	0.483	3.014	1980.88	206.99
32.75	0.000558	0.000560	0.436	2.799	1980.27	207.05
33.00	0.000515	0.000517	0.392	2.584	1979.72	207.11
33.25	0.000472	0.000474	0.350	2.368	1979.22	207.17
33.50	0.000430	0.000431	0.311	2.153	1978.78	207.22
33.75	0.000387	0.000388	0.274	1.938	1978.39	207.26
34.00	0.000344	0.000344	0.239	1.722	1978.05	207.30
34.25	0.000301	0.000301	0.206	1.507	1977.75	207.34
34.50	0.000258	0.000258	0.174	1.292	1977.50	207.37
34.75	0.000215	0.000215	0.143	1.076	1977.29	207.39
35.00	0.000172	0.000172	0.113	0.861	1977.12	207.42
35.25	0.000129	0.000129	0.084	0.646	1976.99	207.43
35.50	0.000086	0.000086	0.056	0.431	1976.89	207.44
35.75	0.000043	0.000043	0.028	0.215	1976.84	207.45
36.00	0.000000	0.000000	0.000	0.000	1976.82	207.45

****

The Final Crack Spacing = 72.0 inches

Example 2: Input file

This input file is for a multi-day analysis. The actual data file contains only the numerical data. The text is included for clarity.

```
1                       - Number of runs in dataset
60000.0                 - Initial crack spacing
0.75                    - Steel Bar Diameter
1                       - Number of layers of steel
29000000.0              - Elastic modulus of the steel
0.000006                - Steel's coefficient of thermal contraction
6.0                     - Spacing of the steel bars
12.0                    - Thickness of the slab
0.00037                 - Ultimate shrinkage coefficient
80.0                    - Curing temperature
0.0                     - FLG1
5.0                     - Number of days in analysis period
1.0                     - FLG2
1375.0                  - Compressive strength of concrete: Day 1
2594.0                  -                                   Day 2
3250.0                  -                                   Day 3
3688.0                  -                                   Day 4
4000.0                  -                                   Day 5
0.0                     - FLG3
0.0                     - FLG4
2.0                     - FLG5
0.000004                - Concrete's coefficient of thermal contraction
0.0                     - FLG6
0.0                     - FLG7
0.0                     - FLG8
60.0                    - Minimum temperature on Day 1
20.0                    - Minimum temperature on Day 5
0.0                     - FLG9
70.0                    - Relative humidity
1.0                     - FLG10
4                       - Day loading begins
9000.0                  - Magnitude of wheel load
5.0                     - Radius of tire contact area
150.0                   - Modulus of subgrade reaction
40.0                    - Number of increments of x for output
```

Example 2: Output file

```
================================================================================
THE INPUT PARAMETERS ARE:

Length of Analysis Period =   5 day(s)
Initial Crack Spacing = 60000.0 inches
Slab Thickness = 12.0 inches

Steel Bar Spacing =  6.0 inches
Steel Bar Diameter = 0.750 inches
Layers of Steel = 1

Cross-Sectional Area of the Steel = 0.442 sq. inches
Cross-Sectional Area of the Concrete =  71.56 sq. inches
Steel Percentage = 0.61%

Steel Modulus of Elasticity = 29000000.0 psi
Steel Coefficient of Thermal Expansion = 0.0000060 inches/inch-degree F

Concrete Curing Temperature =  80.0 degrees F
Initial Shrinkage of Concrete  = 0.0000000 inches/inch
Ultimate Shrinkage of Concrete = 0.0003700 inches/inch

Wheel Loading of the Pavement Begins on Day   4
Magnitude of Wheel Load =  9000.0 lbs
Radius of Tire Contact Area =  5.0 inches
Modulus of Subgrade Reaction = 150.0 lb/in**2/in
```

Concrete Properties and Environmental Conditions

Day	Ec	f'c	f't	Alpha	Rel. Hum.	Min. Temp.
1	2113616.6	1375.0	278.1	0.0000040	70.0	52.00
2	2903085.6	2594.0	382.0	0.0000040	70.0	44.00
3	3249500.0	3250.0	427.6	0.0000040	70.0	36.00
4	3461547.6	3688.0	455.5	0.0000040	70.0	28.00
5	3604996.5	4000.0	474.3	0.0000040	70.0	20.00

Local Bond Stress / Slip Coefficients

Day	K1	K2	STLSLP	K3	K4	CONSLP
1	203945.5	-101972.7	0.0030	3483.2	-348.3	0.0151
2	280122.3	-140061.1	0.0026	4037.3	-403.7	0.0133
3	313548.2	-156774.1	0.0024	4280.4	-428.0	0.0125
4	334009.0	-167004.5	0.0023	4429.2	-442.9	0.0120
5	347850.5	-173925.3	0.0023	4529.8	-453.0	0.0117

==

****

Day 1

Temperature Drop = 28.0 Degrees F

Compressive Strength of the Concrete = 1375.0 psi
Tensile Strength of the Concrete = 278.1 psi
Elastic Modulus of the Concrete = 2113616.6 psi

K1 = 203945.5 pci STLSLP = 0.0030 inches
K2 = -101972.7 pci STLLIM = 0.0089 inches
K3 = 3483.2 pci CONSLP = 0.0151 inches
K4 = -348.3 pci CONLIM = 0.1662 inches

The Solution is in Case 2

The Crack Opening = 0.00905 inches
The Crack Spacing = 60000.0 inches
The Calculated Value of UO = 0.00452377
VL1 = 2.12 inches VL2 = 0.00 inches VL3 = 0.00 inches

The Shrinkage Strain = 0.0000072 inches/inch

The Total Energy Available = 28576.0 inch-lbs
The Frictional Energy Used = 21.6 inch-lbs
The Potential Energy Stored = 28542.2 inch-lbs
The Stress Relief Energy Used = 12.2 inch-lbs

The Force Exerted by the Subgrade = 8743.5 lbs

X	US	UC	BOND STR.	FRIC. STR.	ST. STRESS	CO. STRESS
0.00	0.000000	0.004524	450.367	15.759	25748.21	-0.07
750.00	0.000001	0.000001	-0.001	0.003	4871.71	237.20
1500.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
2250.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
3000.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
3750.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
4500.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
5250.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
6000.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
6750.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
7500.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
8250.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
9000.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
9750.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
10500.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
11250.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
12000.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
12750.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
13500.00	0.000000	0.000000	0.000	0.000	4872.00	237.22

14250.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
15000.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
15750.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
16500.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
17250.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
18000.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
18750.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
19500.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
20250.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
21000.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
21750.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
22500.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
23250.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
24000.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
24750.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
25500.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
26250.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
27000.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
27750.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
28500.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
29250.00	0.000000	0.000000	0.000	0.000	4872.00	237.22
30000.00	0.000000	0.000000	0.000	0.000	4872.00	237.22

****

****

Day 2

Temperature Drop = 36.0 Degrees F

Compressive Strength of the Concrete = 2594.0 psi
Tensile Strength of the Concrete = 382.0 psi
Elastic Modulus of the Concrete = 2903085.6 psi

K1 = 280122.3 pci   STLSLP = 0.0026 inches
K2 = -140061.1 pci   STLLIM = 0.0078 inches
K3 = 4037.3 pci   CONSLP = 0.0133 inches
K4 = -403.7 pci   CONLIM = 0.1460 inches

The Solution is in Case 3

The Crack Opening = 0.01634 inches
The Crack Spacing = 234.4 inches
The Calculated Value of UO = 0.00817222
VL1 = 6.09 inches   VL2 = 0.33 inches   VL3 = 0.00 inches

The Shrinkage Strain = 0.0000140 inches/inch

The Total Energy Available = 254.8 inch-lbs
The Frictional Energy Used = 59.8 inch-lbs
The Potential Energy Stored = 129.1 inch-lbs
The Stress Relief Energy Used = 65.9 inch-lbs

The Force Exerted by the Subgrade = 9802.9 lbs

X	US	UC	BOND STR.	FRIC. STR.	ST. STRESS	CO. STRESS
0.00	0.000000	0.008174	0.000	32.999	32422.92	-0.02
2.93	0.002566	0.007720	370.248	31.169	29840.19	23.80
5.86	0.004558	0.007310	706.754	29.514	21256.18	84.25
8.79	0.005587	0.006967	386.594	28.129	12594.28	144.80
11.72	0.005981	0.006670	192.949	26.929	8238.16	178.45
14.65	0.006059	0.006399	95.426	25.836	6071.69	198.31
17.58	0.005979	0.006145	46.347	24.807	5007.66	211.09
20.51	0.005823	0.005901	21.681	23.822	4498.15	220.21
23.44	0.005632	0.005665	9.316	22.870	4267.05	227.37
26.37	0.005424	0.005436	3.151	21.945	4175.22	233.44
29.30	0.005212	0.005212	0.108	21.043	4152.57	238.86
32.23	0.004999	0.004994	-1.362	20.162	4163.77	243.85
35.16	0.004788	0.004781	-2.041	19.301	4191.07	248.53
38.09	0.004580	0.004572	-2.323	18.458	4225.52	252.96
41.02	0.004376	0.004368	-2.405	17.633	4262.63	257.16
43.95	0.004176	0.004167	-2.389	16.825	4300.18	261.16
46.88	0.003979	0.003971	-2.323	16.032	4337.04	264.97
49.80	0.003787	0.003779	-2.234	15.255	4372.66	268.59
52.73	0.003597	0.003590	-2.135	14.492	4406.81	272.03

55.66	0.003411	0.003404	-2.030	13.743	4439.35	275.30
58.59	0.003228	0.003222	-1.924	13.006	4470.24	278.40
61.52	0.003049	0.003042	-1.819	12.282	4499.49	281.32
64.45	0.002872	0.002866	-1.714	11.570	4527.09	284.08
67.38	0.002698	0.002692	-1.610	10.868	4553.06	286.67
70.31	0.002526	0.002521	-1.508	10.177	4577.42	289.11
73.24	0.002357	0.002352	-1.407	9.495	4600.20	291.38
76.17	0.002190	0.002185	-1.308	8.823	4621.41	293.50
79.10	0.002025	0.002021	-1.209	8.158	4641.07	295.47
82.03	0.001862	0.001858	-1.112	7.501	4659.20	297.28
84.96	0.001701	0.001697	-1.016	6.852	4675.82	298.94
87.89	0.001541	0.001538	-0.920	6.208	4690.94	300.45
90.82	0.001383	0.001380	-0.826	5.571	4704.58	301.81
93.75	0.001226	0.001223	-0.732	4.939	4716.75	303.03
96.68	0.001070	0.001068	-0.639	4.311	4727.46	304.10
99.61	0.000915	0.000913	-0.547	3.688	4736.72	305.02
102.54	0.000762	0.000760	-0.455	3.068	4744.54	305.80
105.47	0.000608	0.000607	-0.363	2.451	4750.93	306.44
108.40	0.000456	0.000455	-0.272	1.836	4755.90	306.94
111.33	0.000304	0.000303	-0.181	1.223	4759.44	307.29
114.26	0.000152	0.000151	-0.091	0.611	4761.56	307.50
117.19	0.000000	0.000000	0.000	0.000	4762.27	307.57

****

****

Day 3

Temperature Drop = 44.0 Degrees F

Compressive Strength of the Concrete = 3250.0 psi
Tensile Strength of the Concrete = 427.6 psi
Elastic Modulus of the Concrete = 3249500.0 psi

K1 = 313548.2 pci STLSLP = 0.0024 inches
K2 = -156774.1 pci STLLIM = 0.0073 inches
K3 = 4280.4 pci CONSLP = 0.0125 inches
K4 = -428.0 pci CONLIM = 0.1372 inches

The Solution is in Case 3

The Crack Opening = 0.02326 inches
The Crack Spacing = 234.4 inches
The Calculated Value of U0 = 0.01163167
VL1 = 9.57 inches VL2 = 4.13 inches VL3 = 0.00 inches

The Shrinkage Strain = 0.0000204 inches/inch

The Total Energy Available = 427.7 inch-lbs
The Frictional Energy Used = 110.7 inch-lbs
The Potential Energy Stored = 170.7 inch-lbs
The Stress Relief Energy Used = 146.4 inch-lbs

The Force Exerted by the Subgrade = 14968.2 lbs

X	US	UC	BOND STR.	FRIC. STR.	ST. STRESS	CO. STRESS
0.00	0.000000	0.011634	0.000	49.797	32635.04	-0.05
2.93	0.002523	0.011081	0.000	47.429	32635.04	11.89
5.86	0.005021	0.010539	278.855	45.112	31340.85	31.25
8.79	0.007080	0.010032	681.078	42.941	23671.77	89.40
11.72	0.008161	0.009588	447.428	41.042	13880.84	160.16
14.65	0.008510	0.009194	214.617	39.354	8925.99	200.62
17.58	0.008504	0.008828	101.698	37.787	6560.29	224.70
20.51	0.008330	0.008480	46.978	36.296	5449.98	240.65
23.44	0.008078	0.008144	20.510	34.857	4947.60	252.49
26.37	0.007792	0.007817	7.754	33.460	4738.86	262.17
29.30	0.007493	0.007499	1.652	32.097	4671.23	270.64
32.23	0.007191	0.007188	-1.221	30.765	4670.70	278.37
35.16	0.006891	0.006883	-2.528	29.462	4701.36	285.57
38.09	0.006595	0.006585	-3.077	28.187	4745.81	292.38
41.02	0.006303	0.006293	-3.259	26.936	4795.62	298.84
43.95	0.006017	0.006007	-3.265	25.710	4846.75	304.99
46.88	0.005736	0.005726	-3.187	24.508	4897.23	310.85
49.80	0.005460	0.005450	-3.070	23.327	4946.14	316.42
52.73	0.005188	0.005179	-2.935	22.167	4993.06	321.72

55.66	0.004921	0.004913	-2.792	21.027	5037.81	326.75
58.59	0.004659	0.004651	-2.647	19.907	5080.31	331.51
61.52	0.004401	0.004393	-2.503	18.804	5120.54	336.02
64.45	0.004147	0.004139	-2.359	17.717	5158.52	340.27
67.38	0.003896	0.003889	-2.217	16.647	5194.27	344.27
70.31	0.003649	0.003643	-2.077	15.592	5227.81	348.02
73.24	0.003406	0.003399	-1.938	14.551	5259.18	351.53
76.17	0.003165	0.003159	-1.801	13.523	5288.39	354.80
79.10	0.002927	0.002922	-1.666	12.507	5315.47	357.83
82.03	0.002692	0.002687	-1.532	11.502	5340.46	360.62
84.96	0.002459	0.002455	-1.400	10.507	5363.36	363.18
87.89	0.002229	0.002225	-1.268	9.522	5384.20	365.51
90.82	0.002000	0.001997	-1.138	8.546	5403.00	367.62
93.75	0.001773	0.001770	-1.009	7.577	5419.78	369.49
96.68	0.001548	0.001545	-0.881	6.615	5434.55	371.14
99.61	0.001325	0.001322	-0.754	5.659	5447.32	372.57
102.54	0.001102	0.001100	-0.627	4.708	5458.11	373.78
105.47	0.000880	0.000879	-0.501	3.761	5466.93	374.77
108.40	0.000660	0.000658	-0.375	2.818	5473.77	375.53
111.33	0.000439	0.000439	-0.250	1.877	5478.66	376.08
114.26	0.000220	0.000219	-0.125	0.938	5481.59	376.41
117.19	0.000000	0.000000	0.000	0.000	5482.57	376.51

****

****

Day 4

Temperature Drop = 52.0 Degrees F

Compressive Strength of the Concrete = 3688.0 psi
Tensile Strength of the Concrete = 455.5 psi
Elastic Modulus of the Concrete = 3461547.6 psi

K1 = 334009.0 pci STLSLP = 0.0023 inches
K2 = -167004.5 pci STLLIM = 0.0070 inches
K3 = 4429.2 pci CONSLP = 0.0120 inches
K4 = -442.9 pci CONLIM = 0.1318 inches

The Solution is in Case 3

The Crack Opening = 0.01882 inches
The Crack Spacing = 117.2 inches
The Calculated Value of U0 = 0.00941000
VL1 = 7.62 inches VL2 = 2.34 inches VL3 = 0.00 inches

The Shrinkage Strain = 0.0000266 inches/inch
Load-induced Stress at the Center of the Slab = 90.69 psi

The Total Energy Available = 319.1 inch-lbs
The Frictional Energy Used = 65.0 inch-lbs
The Potential Energy Stored = 63.0 inch-lbs
The Stress Relief Energy Used = 191.1 inch-lbs

The Force Exerted by the Subgrade = 6844.6 lbs

X	US	UC	BOND STR.	FRIC. STR.	ST. STRESS	CO. STRESS
0.00	0.000000	0.009412	0.000	41.688	32613.67	-0.03
1.46	0.001190	0.009089	0.000	40.257	32613.67	5.00
2.93	0.002380	0.008768	100.431	38.836	32457.10	10.82
4.39	0.003526	0.008452	344.747	37.433	30707.79	26.31
5.86	0.004537	0.008144	564.981	36.071	27133.29	52.89
7.32	0.005326	0.007850	745.970	34.767	21982.40	89.04
8.79	0.005836	0.007571	579.615	33.535	16568.91	126.65
10.25	0.006115	0.007307	398.264	32.364	12792.96	154.01
11.72	0.006234	0.007053	273.331	31.237	10199.80	173.93
13.18	0.006245	0.006805	187.277	30.142	8421.39	188.68
14.65	0.006180	0.006564	128.015	29.072	7204.14	199.83
16.11	0.006065	0.006326	87.216	28.019	6373.29	208.46
17.58	0.005915	0.006092	59.140	26.981	5808.40	215.33
19.04	0.005741	0.005860	39.831	25.955	5426.50	220.94
20.51	0.005551	0.005630	26.564	24.938	5170.38	225.64
21.97	0.005350	0.005403	17.460	23.930	5000.64	229.69
23.44	0.005143	0.005177	11.224	22.928	4890.13	233.25
24.90	0.004931	0.004952	6.966	21.933	4820.14	236.44

26.37	0.004716	0.004729	4.069	20.943	4777.76	239.33
27.83	0.004500	0.004506	2.111	19.959	4754.12	241.99
29.30	0.004283	0.004285	0.799	18.980	4743.10	244.45
30.76	0.004065	0.004065	-0.067	18.005	4740.48	246.74
32.23	0.003848	0.003846	-0.626	17.034	4743.36	248.87
33.69	0.003630	0.003628	-0.975	16.067	4749.72	250.87
35.16	0.003414	0.003410	-1.177	15.103	4758.21	252.73
36.62	0.003197	0.003193	-1.280	14.143	4767.86	254.46
38.09	0.002981	0.002977	-1.313	13.187	4778.03	256.08
39.55	0.002766	0.002762	-1.300	12.233	4788.26	257.58
41.02	0.002551	0.002547	-1.253	11.281	4798.25	258.96
42.48	0.002336	0.002333	-1.185	10.332	4807.78	260.23
43.95	0.002122	0.002119	-1.101	9.386	4816.72	261.38
45.41	0.001909	0.001906	-1.006	8.442	4824.95	262.43
46.88	0.001696	0.001693	-0.904	7.499	4832.42	263.36
48.34	0.001483	0.001481	-0.798	6.558	4839.07	264.18
49.80	0.001271	0.001269	-0.688	5.618	4844.87	264.90
51.27	0.001058	0.001057	-0.575	4.680	4849.81	265.50
52.73	0.000846	0.000845	-0.462	3.743	4853.86	265.99
54.20	0.000635	0.000634	-0.347	2.806	4857.02	266.37
55.66	0.000423	0.000422	-0.232	1.871	4859.28	266.65
57.13	0.000211	0.000211	-0.116	0.935	4860.64	266.81
58.59	0.000000	0.000000	0.000	0.000	4861.09	266.86

****

****

Day 5

Temperature Drop = 60.0 Degrees F

Compressive Strength of the Concrete = 4000.0 psi
Tensile Strength of the Concrete = 474.3 psi
Elastic Modulus of the Concrete = 3604996.5 psi

K1 = 347850.5 pci   STLSLP = 0.0023 inches
K2 = -173925.3 pci   STLLIM = 0.0068 inches
K3 = 4529.8 pci   CONSLP = 0.0117 inches
K4 = -453.0 pci   CONLIM = 0.1282 inches

The Solution is in Case 3

The Crack Opening = 0.02256 inches
The Crack Spacing = 117.2 inches
The Calculated Value of UO = 0.01127833
VL1 = 9.57 inches   VL2 = 4.39 inches   VL3 = 0.00 inches

The Shrinkage Strain = 0.0000324 inches/inch
Load-induced Stress at the Center of the Slab = 90.69 psi

The Total Energy Available = 443.2 inch-lbs
The Frictional Energy Used = 86.3 inch-lbs
The Potential Energy Stored = 71.8 inch-lbs
The Stress Relief Energy Used = 285.2 inch-lbs

The Force Exerted by the Subgrade = 8419.6 lbs

X	US	UC	BOND STR.	FRIC. STR.	ST. STRESS	CO. STRESS
0.00	0.000000	0.011281	0.000	51.102	32862.62	-0.05
1.46	0.001133	0.010912	0.000	49.430	32862.62	6.13
2.93	0.002265	0.010545	0.000	47.768	32862.62	12.09
4.39	0.003398	0.010181	0.000	46.118	32862.64	17.86
5.86	0.004514	0.009820	256.703	44.482	31854.14	29.65
7.32	0.005528	0.009466	494.782	42.879	28900.65	53.25
8.79	0.006349	0.009124	697.059	41.329	24216.63	87.34
10.25	0.006899	0.008797	660.089	39.849	18488.92	127.68
11.72	0.007191	0.008485	450.240	38.437	14203.48	158.95
13.18	0.007303	0.008184	306.724	37.073	11282.01	181.62
14.65	0.007292	0.007891	208.587	35.746	9293.31	198.37
16.11	0.007197	0.007604	141.496	34.445	7942.37	211.02
17.58	0.007047	0.007322	95.644	33.165	7027.39	220.82
19.04	0.006858	0.007042	64.321	31.901	6410.29	228.63
20.51	0.006643	0.006766	42.939	30.650	5996.61	235.02
21.97	0.006411	0.006493	28.358	29.410	5721.73	240.41
23.44	0.006168	0.006221	18.428	28.179	5541.46	245.06
24.90	0.005917	0.005951	11.681	26.956	5425.55	249.16

26.37	0.005662	0.005682	7.110	25.740	5353.31	252.84
27.83	0.005404	0.005415	4.029	24.531	5310.59	256.19
29.30	0.005144	0.005150	1.967	23.328	5287.71	259.27
30.76	0.004884	0.004885	0.601	22.130	5278.05	262.12
32.23	0.004623	0.004622	-0.288	20.937	5277.08	264.78
33.69	0.004362	0.004360	-0.852	19.748	5281.71	267.25
35.16	0.004102	0.004098	-1.193	18.565	5289.82	269.55
36.62	0.003842	0.003838	-1.381	17.385	5299.95	271.69
38.09	0.003582	0.003578	-1.466	16.209	5311.13	273.69
39.55	0.003324	0.003319	-1.479	15.036	5322.67	275.53
41.02	0.003065	0.003061	-1.444	13.867	5334.11	277.24
42.48	0.002808	0.002804	-1.376	12.701	5345.14	278.80
43.95	0.002551	0.002547	-1.285	11.538	5355.55	280.23
45.41	0.002294	0.002291	-1.179	10.377	5365.18	281.51
46.88	0.002038	0.002035	-1.062	9.218	5373.94	282.66
48.34	0.001782	0.001780	-0.939	8.062	5381.76	283.68
49.80	0.001527	0.001525	-0.810	6.907	5388.59	284.55
51.27	0.001272	0.001270	-0.679	5.753	5394.41	285.29
52.73	0.001017	0.001016	-0.545	4.601	5399.19	285.90
54.20	0.000763	0.000762	-0.410	3.450	5402.91	286.37
55.66	0.000508	0.000508	-0.273	2.300	5405.58	286.71
57.13	0.000254	0.000254	-0.137	1.150	5407.19	286.91
58.59	0.000000	0.000000	0.000	0.000	5407.72	286.98

****

The Final Crack Spacing =   117.2 inches

APPENDIX C

Listing of the TTICRCP Program

```
C=====================================================================C
C                                                                     C
C  TTICRCP is rationally-based computer model that predicts the behavior  C
C  of Continuously Reinforced Concrete Pavements.  It was developed and    C
C  written by Richard P. Palmer at the Texas Transportation Institute,     C
C  Texas A & M University, College Station, Texas.  November 1987.         C
C                                                                     C
C  Program TTICRCP is the main control block of the program           C
C                                                                     C
      PROGRAM TTICRCP
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_1A/ G11,H11,AS11,BS11,CS11,DS11
      COMMON /CASE_1B/ AC11,BC11,CC11,DC11
      COMMON /CASE_2A/ G21,H21,G22,H22,UCP22,USP22
      COMMON /CASE_2B/ AC21,BC21,CC21,DC21,AS21,BS21,CS21,DS21
      COMMON /CASE_2C/ AC22,BC22,CC22,DC22,AS22,BS22,CS22,DS22
      COMMON /CASE_3A/ G31,H31,G32,H32,G33,UCP32,USP32
      COMMON /CASE_3B/ AC31,BC31,CC31,DC31,AS31,BS31,CS31,DS31
      COMMON /CASE_3C/ AC32,BC32,CC32,DC32,AS32,BS32,CS32,DS32
      COMMON /CASE_3D/ AC33,BC33
      COMMON /CASE_4A/ G41,H41,G42,H42,G43,H43,UCP43,USP43,UCP42,USP42
      COMMON /CASE_4B/ AC41,BC41,CC41,DC41,AS41,BS41,CS41,DS41
      COMMON /CASE_4C/ AC42,BC42,CC42,DC42,AS42,BS42,CS42,DS42
      COMMON /CASE_4D/ AC43,BC43,CC43,DC43,AS43,BS43,CS43,DS43
      COMMON /CASE_5A/ G51,H51,G52,H52,G53,G54
      COMMON /CASE_5B/ AC51,BC51,CC51,DC51,AS51,BS51,CS51,DS51
      COMMON /CASE_5C/ AC52,BC52,CC52,DC52,AS52,BS52,CS52,DS52
      COMMON /CASE_5D/ AC53,BC53,AC54,BC54,UCP54,UCP52,USP52
      COMMON /CASE_6A/ G61,H61,G62,H62,G63,H63,G64
      COMMON /CASE_6B/ AC61,BC61,CC61,DC61,AS61,BS61,CS61,DS61
      COMMON /CASE_6C/ AC62,BC62,CC62,DC62,AS62,BS62,CS62,DS62
      COMMON /CASE_6D/ AC63,BC63,CC63,DC63,AS63,BS63,CS63,DS63
      COMMON /CASE_6E/ AC64,BC64,UCP64,UCP63,USP63,UCP62,USP62
      COMMON /CASE_7A/ G71,H71,G72,H72,UCP72,USP72
      COMMON /CASE_7B/ AC71,BC71,CC71,DC71,AS71,BS71,CS71,DS71
      COMMON /CASE_7C/ AC72,BC72,CC72,DC72,AS72,BS72,CS72,DS72
      COMMON /CASE_8A/ G81,H81,G82,H82,G83,H83,UCP83,USP83,UCP82,USP82
      COMMON /CASE_8B/ AC81,BC81,CC81,DC81,AS81,BS81,CS81,DS81
      COMMON /CASE_8C/ AC82,BC82,CC82,DC82,AS82,BS82,CS82,DS82
      COMMON /CASE_8D/ AC83,BC83,CC83,DC83,AS83,BS83,CS83,DS83
      COMMON /CASE_9A/ G91,H91,G92,H92,G93,H93,G94
      COMMON /CASE_9B/ AC91,BC91,CC91,DC91,AS91,BS91,CS91,DS91
      COMMON /CASE_9C/ AC92,BC92,CC92,DC92,AS92,BS92,CS92,DS92
      COMMON /CASE_9D/ AC93,BC93,CC93,DC93,AS93,BS93,CS93,DS93
      COMMON /CASE_9E/ AC94,BC94,UCP94,UCP93,USP93,UCP92,USP92
```

```
      READ(5,*) NUM
      DO 20 I = 1,NUM
      CALL INPUT(KTIME)
      CRACK = S
      READ(5,*) TRIP
      DO 10 KDAY = 1,KTIME
 5    CALL INITIALISE
      CALL TIMEDEPEND(KDAY,NODROP,CRACK)
      IF (NODROP.EQ.1) GOTO 10
      CALL CALCUO(UO)
      CALL SUMSTRAIN(KDAY)
      IF ((CENTERSTR*EC).GT.FPRMT(KDAY)) THEN
        CRACK = S/2.0
        GOTO 5
        ENDIF
      CALL OUTPUT(UO,KDAY,TRIP)
 10   CONTINUE
      CALL FINALOUT
 20   CONTINUE
      STOP
      END

C=========================================================================C
C                                                                         C
C  SUBROUTINE INITIALISE sets the value of the case coefficients = 0.0    C
C  Its main purpose is to prevent previously calculated values from       C
C  affecting the current values.                                          C
C                                                                         C
      SUBROUTINE INITIALISE
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_1A/ G11,H11,AS11,BS11,CS11,DS11
      COMMON /CASE_1B/ AC11,BC11,CC11,DC11
      COMMON /CASE_2A/ G21,H21,G22,H22,UCP22,USP22
      COMMON /CASE_2B/ AC21,BC21,CC21,DC21,AS21,BS21,CS21,DS21
      COMMON /CASE_2C/ AC22,BC22,CC22,DC22,AS22,BS22,CS22,DS22
      COMMON /CASE_3A/ G31,H31,G32,H32,G33,UCP32,USP32
      COMMON /CASE_3B/ AC31,BC31,CC31,DC31,AS31,BS31,CS31,DS31
      COMMON /CASE_3C/ AC32,BC32,CC32,DC32,AS32,BS32,CS32,DS32
      COMMON /CASE_3D/ AC33,BC33
      COMMON /CASE_4A/ G41,H41,G42,H42,G43,H43,UCP43,USP43,UCP42,USP42
      COMMON /CASE_4B/ AC41,BC41,CC41,DC41,AS41,BS41,CS41,DS41
      COMMON /CASE_4C/ AC42,BC42,CC42,DC42,AS42,BS42,CS42,DS42
      COMMON /CASE_4D/ AC43,BC43,CC43,DC43,AS43,BS43,CS43,DS43
      COMMON /CASE_5A/ G51,H51,G52,H52,G53,G54
      COMMON /CASE_5B/ AC51,BC51,CC51,DC51,AS51,BS51,CS51,DS51
```

```
COMMON /CASE_5C/ AC52,BC52,CC52,DC52,AS52,BS52,CS52,DS52
COMMON /CASE_5D/ AC53,BC53,AC54,BC54,UCP54,UCP52,USP52
COMMON /CASE_6A/ G61,H61,G62,H62,G63,H63,G64
COMMON /CASE_6B/ AC61,BC61,CC61,DC61,AS61,BS61,CS61,DS61
COMMON /CASE_6C/ AC62,BC62,CC62,DC62,AS62,BS62,CS62,DS62
COMMON /CASE_6D/ AC63,BC63,CC63,DC63,AS63,BS63,CS63,DS63
COMMON /CASE_6E/ AC64,BC64,UCP64,UCP63,USP63,UCP62,USP62
COMMON /CASE_7A/ G71,H71,G72,H72,UCP72,USP72
COMMON /CASE_7B/ AC71,BC71,CC71,DC71,AS71,BS71,CS71,DS71
COMMON /CASE_7C/ AC72,BC72,CC72,DC72,AS72,BS72,CS72,DS72
COMMON /CASE_8A/ G81,H81,G82,H82,G83,H83,UCP83,USP83,UCP82,USP82
COMMON /CASE_8B/ AC81,BC81,CC81,DC81,AS81,BS81,CS81,DS81
COMMON /CASE_8C/ AC82,BC82,CC82,DC82,AS82,BS82,CS82,DS82
COMMON /CASE_8D/ AC83,BC83,CC83,DC83,AS83,BS83,CS83,DS83
COMMON /CASE_9A/ G91,H91,G92,H92,G93,H93,G94
COMMON /CASE_9B/ AC91,BC91,CC91,DC91,AS91,BS91,CS91,DS91
COMMON /CASE_9C/ AC92,BC92,CC92,DC92,AS92,BS92,CS92,DS92
COMMON /CASE_9D/ AC93,BC93,CC93,DC93,AS93,BS93,CS93,DS93
COMMON /CASE_9E/ AC94,BC94,UCP94,UCP93,USP93,UCP92,USP92
G11 = 0.0
H11 = 0.0
AS11 = 0.0
BS11 = 0.0
CS11 = 0.0
DS11 = 0.0
AC11 = 0.0
BC11 = 0.0
CC11 = 0.0
DC11 = 0.0
G21 = 0.0
H21 = 0.0
G22 = 0.0
H22 = 0.0
UCP22 = 0.0
USP22 = 0.0
AC21 = 0.0
BC21 = 0.0
CC21 = 0.0
DC21 = 0.0
AS21 = 0.0
BS21 = 0.0
CS21 = 0.0
DS21 = 0.0
AC22 = 0.0
BC22 = 0.0
CC22 = 0.0
DC22 = 0.0
AS22 = 0.0
BS22 = 0.0
CS22 = 0.0
DS22 = 0.0
G31 = 0.0
H31 = 0.0
G32 = 0.0
```

```
H32 = 0.0
G33 = 0.0
UCP32 = 0.0
USP32 = 0.0
AC31 = 0.0
BC31 = 0.0
CC31 = 0.0
DC31 = 0.0
AS31 = 0.0
BS31 = 0.0
CS31 = 0.0
DS31 = 0.0
AC32 = 0.0
BC32 = 0.0
CC32 = 0.0
DC32 = 0.0
AS32 = 0.0
BS32 = 0.0
CS32 = 0.0
DS32 = 0.0
AC33 = 0.0
BC33 = 0.0
G41 = 0.0
H41 = 0.0
G42 = 0.0
H42 = 0.0
G43 = 0.0
H43 = 0.0
UCP43 = 0.0
USP43 = 0.0
UCP42 = 0.0
USP42 = 0.0
AC41 = 0.0
BC41 = 0.0
CC41 = 0.0
DC41 = 0.0
AS41 = 0.0
BS41 = 0.0
CS41 = 0.0
DS41 = 0.0
AC42 = 0.0
BC42 = 0.0
CC42 = 0.0
DC42 = 0.0
AS42 = 0.0
BS42 = 0.0
CS42 = 0.0
DS42 = 0.0
AC43 = 0.0
BC43 = 0.0
CC43 = 0.0
DC43 = 0.0
AS43 = 0.0
BS43 = 0.0
```

CS43 = 0.0
DS43 = 0.0
G51 = 0.0
H51 = 0.0
G52 = 0.0
H52 = 0.0
G53 = 0.0
G54 = 0.0
AC51 = 0.0
BC51 = 0.0
CC51 = 0.0
DC51 = 0.0
AS51 = 0.0
BS51 = 0.0
CS51 = 0.0
DS51 = 0.0
AC52 = 0.0
BC52 = 0.0
CC52 = 0.0
DC52 = 0.0
AS52 = 0.0
BS52 = 0.0
CS52 = 0.0
DS52 = 0.0
AC53 = 0.0
BC53 = 0.0
AC54 = 0.0
BC54 = 0.0
UCP54 = 0.0
UCP52 = 0.0
USP52 = 0.0
G61 = 0.0
H61 = 0.0
G62 = 0.0
H62 = 0.0
G63 = 0.0
H63 = 0.0
G64 = 0.0
AC61 = 0.0
BC61 = 0.0
CC61 = 0.0
DC61 = 0.0
AS61 = 0.0
BS61 = 0.0
CS61 = 0.0
DS61 = 0.0
AC62 = 0.0
BC62 = 0.0
CC62 = 0.0
DC62 = 0.0
AS62 = 0.0
BS62 = 0.0
CS62 = 0.0
DS62 = 0.0

AC63 = 0.0
BC63 = 0.0
CC63 = 0.0
DC63 = 0.0
AS63 = 0.0
BS63 = 0.0
CS63 = 0.0
DS63 = 0.0
AC64 = 0.0
BC64 = 0.0
UCP64 = 0.0
UCP63 = 0.0
USP63 = 0.0
UCP62 = 0.0
USP62 = 0.0
G71 = 0.0
H71 = 0.0
G72 = 0.0
H72 = 0.0
UCP72 = 0.0
USP72 = 0.0
AC71 = 0.0
BC71 = 0.0
CC71 = 0.0
DC71 = 0.0
AS71 = 0.0
BS71 = 0.0
CS71 = 0.0
DS71 = 0.0
AC72 = 0.0
BC72 = 0.0
CC72 = 0.0
DC72 = 0.0
AS72 = 0.0
BS72 = 0.0
CS72 = 0.0
DS72 = 0.0
G81 = 0.0
H81 = 0.0
G82 = 0.0
H82 = 0.0
G83 = 0.0
H83 = 0.0
UCP83 = 0.0
USP83 = 0.0
UCP82 = 0.0
USP82 = 0.0
AC81 = 0.0
BC81 = 0.0
CC81 = 0.0
DC81 = 0.0
AS81 = 0.0
BS81 = 0.0
CS81 = 0.0

DS81 = 0.0
AC82 = 0.0
BC82 = 0.0
CC82 = 0.0
DC82 = 0.0
AS82 = 0.0
BS82 = 0.0
CS82 = 0.0
DS82 = 0.0
AC83 = 0.0
BC83 = 0.0
CC83 = 0.0
DC83 = 0.0
AS83 = 0.0
BS83 = 0.0
CS83 = 0.0
DS83 = 0.0
G91 = 0.0
H91 = 0.0
G92 = 0.0
H92 = 0.0
G93 = 0.0
H93 = 0.0
G94 = 0.0
AC91 = 0.0
BC91 = 0.0
CC91 = 0.0
DC91 = 0.0
AS91 = 0.0
BS91 = 0.0
CS91 = 0.0
DS91 = 0.0
AC92 = 0.0
BC92 = 0.0
CC92 = 0.0
DC92 = 0.0
AS92 = 0.0
BS92 = 0.0
CS92 = 0.0
DS92 = 0.0
AC93 = 0.0
BC93 = 0.0
CC93 = 0.0
DC93 = 0.0
AS93 = 0.0
BS93 = 0.0
CS93 = 0.0
DS93 = 0.0
AC94 = 0.0
BC94 = 0.0
UCP94 = 0.0
UCP93 = 0.0
USP93 = 0.0
UCP92 = 0.0

```
      USP92 = 0.0
      STSTRAIN = 0.0
      CENTERSTR = 0.0
      STRESS = 0.0
      VL1 = 0.0
      VL2 = 0.0
      VL3 = 0.0
      RETURN
      END

C========================================================================C
C                                                                        C
C  TIMDEPEND increments all of the time dependent values.                C
C                                                                        C
C                                                                        C
      SUBROUTINE TIMEDEPEND(KDAY,NODROP,CRACK)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
C
      IF (S.NE.CRACK) THEN
        S = CRACK
        GOTO 10
        ENDIF
      EC = ELASCON(KDAY)
      ALPHA = ALPHACON(KDAY)
      XK1 = BONSLP(1,KDAY)
      XK2 = BONSLP(2,KDAY)
      XK3 = BONSLP(3,KDAY)
      XK4 = BONSLP(4,KDAY)
      STLSLP = STLSLIP(KDAY)
      CONSLP = CONSLIP(KDAY)
      STLLIM = STLSLP - XK1*STLSLP/XK2
      CONLIM = CONSLP - XK3*CONSLP/XK4
      XC1 = -XK2*STLLIM
      XC2 = -XK4*CONLIM
      DELTEM = CURTEMP - DAYTEMP(KDAY)
      IF (DELTEM.LE.0.0) THEN
        NODROP = 1
       ELSE
        NODROP = 0
       ENDIF
      IF (RELHUM(KDAY).GT.80.0) THEN
        COFAC = 3.0 - 0.03*RELHUM(KDAY)
       ELSE
        COFAC = 1.4 - 0.01*RELHUM(KDAY)
       ENDIF
```

```
      ATIME = KDAY
      SHRINK1 = COFAC*ULTSHR*(ATIME/(35.0+ATIME) -
     +                        (ATIME-1.0)/(34.0+ATIME))
      SHRINKAGE = SHRINKAGE + SHRINK1
 10   TAE = 0.5*EC*ACON*S*((ALPHA*DELTEM)**2.0 + SHRINKAGE**2.0)
      STRN = ALPHAS*DELTEM
      RETURN
      END

C========================================================================C
C                                                                        C
C  SUMSTRAIN sums the strain at the center of the slab due to the        C
C  temperature drop and shrinkage with the strain induced by external    C
C  wheel loads.                                                          C
C                                                                        C
      SUBROUTINE SUMSTRAIN(KDAY)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      IF (KDAY.LT.KDAYLOBEG) THEN
        CENTERSTR = CENTERSTR + 0.0
       ELSE
        IF (RADLOD.LE.(1.724*THICK)) THEN
          B = (1.6*RADLOD**2.0+THICK**2.0)**0.5 - 0.675*THICK
         ELSE
          B = RADLOD
         ENDIF
        STRESS = LOG10(THICK**3.0)-4.0*LOG10(B)-LOG10(BASEK)+6.478
        STRESS = 0.3162*AMAGLOD*STRESS/THICK**2.0
        CENTERSTR = CENTERSTR + STRESS/EC
       ENDIF
      RETURN
      END

C========================================================================C
C                                                                        C
C  FINALOUT prints out the final crack spacing.                          C
C                                                                        C
C                                                                        C
      SUBROUTINE FINALOUT
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
```

```
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      WRITE(6,1) 2.0*S
 1    FORMAT('0','The Final Crack Spacing = ',F7.1,' inches')
      WRITE(6,2)
 2    FORMAT('1')
      RETURN
      END

C=====================================================================C
C                                                                     C
C  CALCUO iterates to find the correct values for UO, VL1, VL2, and VL3.  C
C                                                                     C
C                                                                     C
      SUBROUTINE CALCUO(UO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_1A/ G11,H11,AS11,BS11,CS11,DS11
      COMMON /CASE_1B/ AC11,BC11,CC11,DC11
      COMMON /CASE_2A/ G21,H21,G22,H22,UCP22,USP22
      COMMON /CASE_2B/ AC21,BC21,CC21,DC21,AS21,BS21,CS21,DS21
      COMMON /CASE_2C/ AC22,BC22,CC22,DC22,AS22,BS22,CS22,DS22
      COMMON /CASE_3A/ G31,H31,G32,H32,G33,UCP32,USP32
      COMMON /CASE_3B/ AC31,BC31,CC31,DC31,AS31,BS31,CS31,DS31
      COMMON /CASE_3C/ AC32,BC32,CC32,DC32,AS32,BS32,CS32,DS32
      COMMON /CASE_3D/ AC33,BC33
      COMMON /CASE_4A/ G41,H41,G42,H42,G43,H43,UCP43,USP43,UCP42,USP42
      COMMON /CASE_4B/ AC41,BC41,CC41,DC41,AS41,BS41,CS41,DS41
      COMMON /CASE_4C/ AC42,BC42,CC42,DC42,AS42,BS42,CS42,DS42
      COMMON /CASE_4D/ AC43,BC43,CC43,DC43,AS43,BS43,CS43,DS43
      COMMON /CASE_5A/ G51,H51,G52,H52,G53,G54
      COMMON /CASE_5B/ AC51,BC51,CC51,DC51,AS51,BS51,CS51,DS51
      COMMON /CASE_5C/ AC52,BC52,CC52,DC52,AS52,BS52,CS52,DS52
      COMMON /CASE_5D/ AC53,BC53,AC54,BC54,UCP54,UCP52,USP52
      COMMON /CASE_6A/ G61,H61,G62,H62,G63,H63,G64
      COMMON /CASE_6B/ AC61,BC61,CC61,DC61,AS61,BS61,CS61,DS61
      COMMON /CASE_6C/ AC62,BC62,CC62,DC62,AS62,BS62,CS62,DS62
      COMMON /CASE_6D/ AC63,BC63,CC63,DC63,AS63,BS63,CS63,DS63
      COMMON /CASE_6E/ AC64,BC64,UCP64,UCP63,USP63,UCP62,USP62
      COMMON /CASE_7A/ G71,H71,G72,H72,UCP72,USP72
      COMMON /CASE_7B/ AC71,BC71,CC71,DC71,AS71,BS71,CS71,DS71
      COMMON /CASE_7C/ AC72,BC72,CC72,DC72,AS72,BS72,CS72,DS72
      COMMON /CASE_8A/ G81,H81,G82,H82,G83,H83,UCP83,USP83,UCP82,USP82
      COMMON /CASE_8B/ AC81,BC81,CC81,DC81,AS81,BS81,CS81,DS81
      COMMON /CASE_8C/ AC82,BC82,CC82,DC82,AS82,BS82,CS82,DS82
```

```
      COMMON /CASE_8D/ AC83,BC83,CC83,DC83,AS83,BS83,CS83,DS83
      COMMON /CASE_9A/ G91,H91,G92,H92,G93,H93,G94
      COMMON /CASE_9B/ AC91,BC91,CC91,DC91,AS91,BS91,CS91,DS91
      COMMON /CASE_9C/ AC92,BC92,CC92,DC92,AS92,BS92,CS92,DS92
      COMMON /CASE_9D/ AC93,BC93,CC93,DC93,AS93,BS93,CS93,DS93
      COMMON /CASE_9E/ AC94,BC94,UCP94,UCP93,USP93,UCP92,USP92
      TOLERANCE = 0.049
      HIUPPER = (2.0*TAE/(S*ACON*EC))**0.5
 9    UPPER = HIUPPER*S
      ALOWER = 0.0
      UO = 0.00005
      UL1 = 0.0
      UL2 = 0.0
      UL3 = 0.0
      VL1 = 0.0
      VL2 = 0.0
      VL3 = 0.0
      KASE = 1
      STEP1 = UO*10.0
10    STEP1 = STEP1/10.0
20    BSTSTR = STSTRAIN
      CALL SELECT(UO,UL1,UL2,UL3)
      KKASE = KASE
      CALL CHECK(UO,UL1,UL2,UL3)
      IF (KKASE.NE.KASE) THEN
        IF (KASE.EQ.0) THEN
          WRITE(6,21) KKASE,KASE
21        FORMAT('0','PREVIOUS CASE = ',I2,'  CASE = ',I2)
          GOTO 28
          ENDIF
        VL1 = UL1
        VL2 = UL2
        VL3 = UL3
        IF (STEP1.LE.(ALPHA*DELTEM/9.0)) GOTO 20
        STEP1 = STEP1/3.0
        GOTO 20
        ENDIF
28    IF (KASE.EQ.0) THEN
        WRITE(6,*)
        WRITE(6,29)
29      FORMAT(1X,'KASE = 0.0,  MAJOR ERROR HAS OCCURRED')
        RETURN
        ENDIF
30    ENERBAL = TAE - FRICTION - STREEN - POTENTIAL
      IF (ABS(ENERBAL).LT.TOLERANCE) GOTO 100
      IF (UO.GT.(HIUPPER*S)) GOTO 100
      IF (ENERBAL.GT.0.0) THEN
        ALOWER = UO
        AUL1 = VL1
        AUL2 = VL2
        AUL3 = VL3
        ASTSTR = BSTSTR
        MKASE = KASE
        IF((UO+STEP1).GT.UPPER) THEN
```

```
          UO = UPPER
         ELSE
          UO = UO + STEP1
         ENDIF
        VL1 = UL1
        VL2 = UL2
        VL3 = UL3
        GOTO 20
       ENDIF
      IF (ENERBAL.LT.0.0) THEN
        UO = ALOWER
        VL1 = AUL1
        VL2 = AUL2
        VL3 = AUL3
        STSTRAIN = ASTSTR
        KASE = MKASE
        GOTO 10
        ENDIF
 100   RETURN
      END

C===========================================================================C
C                                                                           C
C  SELECT calls the VAL and FRICSHUN subroutines depending on the case.    C
C                                                                           C
C                                                                           C
      SUBROUTINE SELECT(UO,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_1A/ G11,H11,AS11,BS11,CS11,DS11
      COMMON /CASE_1B/ AC11,BC11,CC11,DC11
      COMMON /CASE_2A/ G21,H21,G22,H22,UCP22,USP22
      COMMON /CASE_2B/ AC21,BC21,CC21,DC21,AS21,BS21,CS21,DS21
      COMMON /CASE_2C/ AC22,BC22,CC22,DC22,AS22,BS22,CS22,DS22
      COMMON /CASE_3A/ G31,H31,G32,H32,G33,UCP32,USP32
      COMMON /CASE_3B/ AC31,BC31,CC31,DC31,AS31,BS31,CS31,DS31
      COMMON /CASE_3C/ AC32,BC32,CC32,DC32,AS32,BS32,CS32,DS32
      COMMON /CASE_3D/ AC33,BC33
      COMMON /CASE_4A/ G41,H41,G42,H42,G43,H43,UCP43,USP43,UCP42,USP42
      COMMON /CASE_4B/ AC41,BC41,CC41,DC41,AS41,BS41,CS41,DS41
      COMMON /CASE_4C/ AC42,BC42,CC42,DC42,AS42,BS42,CS42,DS42
      COMMON /CASE_4D/ AC43,BC43,CC43,DC43,AS43,BS43,CS43,DS43
      COMMON /CASE_5A/ G51,H51,G52,H52,G53,G54
      COMMON /CASE_5B/ AC51,BC51,CC51,DC51,AS51,BS51,CS51,DS51
      COMMON /CASE_5C/ AC52,BC52,CC52,DC52,AS52,BS52,CS52,DS52
      COMMON /CASE_5D/ AC53,BC53,AC54,BC54,UCP54,UCP52,USP52
```

```
COMMON /CASE_6A/ G61,H61,G62,H62,G63,H63,G64
COMMON /CASE_6B/ AC61,BC61,CC61,DC61,AS61,BS61,CS61,DS61
COMMON /CASE_6C/ AC62,BC62,CC62,DC62,AS62,BS62,CS62,DS62
COMMON /CASE_6D/ AC63,BC63,CC63,DC63,AS63,BS63,CS63,DS63
COMMON /CASE_6E/ AC64,BC64,UCP64,UCP63,USP63,UCP62,USP62
COMMON /CASE_7A/ G71,H71,G72,H72,UCP72,USP72
COMMON /CASE_7B/ AC71,BC71,CC71,DC71,AS71,BS71,CS71,DS71
COMMON /CASE_7C/ AC72,BC72,CC72,DC72,AS72,BS72,CS72,DS72
COMMON /CASE_8A/ G81,H81,G82,H82,G83,H83,UCP83,USP83,UCP82,USP82
COMMON /CASE_8B/ AC81,BC81,CC81,DC81,AS81,BS81,CS81,DS81
COMMON /CASE_8C/ AC82,BC82,CC82,DC82,AS82,BS82,CS82,DS82
COMMON /CASE_8D/ AC83,BC83,CC83,DC83,AS83,BS83,CS83,DS83
COMMON /CASE_9A/ G91,H91,G92,H92,G93,H93,G94
COMMON /CASE_9B/ AC91,BC91,CC91,DC91,AS91,BS91,CS91,DS91
COMMON /CASE_9C/ AC92,BC92,CC92,DC92,AS92,BS92,CS92,DS92
COMMON /CASE_9D/ AC93,BC93,CC93,DC93,AS93,BS93,CS93,DS93
COMMON /CASE_9E/ AC94,BC94,UCP94,UCP93,USP93,UCP92,USP92
IF (KASE.EQ.1) THEN
  CALL VAL1(UO)
  CALL FRICSHUN1(UO,UL1,UL2,UL3)
  ENDIF
IF (KASE.EQ.2) THEN
  CALL VAL2(UO)
  CALL FRICSHUN2(UO,UL1,UL2,UL3)
  ENDIF
IF (KASE.EQ.3) THEN
  CALL VAL3(UO)
  CALL FRICSHUN3(UO,UL1,UL2,UL3)
  ENDIF
IF (KASE.EQ.4) THEN
  CALL VAL4(UO)
  CALL FRICSHUN4(UO,UL1,UL2,UL3)
  ENDIF
IF (KASE.EQ.5) THEN
  CALL VAL5(UO)
  CALL FRICSHUN5(UO,UL1,UL2,UL3)
  ENDIF
IF (KASE.EQ.6) THEN
  CALL VAL6(UO)
  CALL FRICSHUN6(UO,UL1,UL2,UL3)
  ENDIF
IF (KASE.EQ.7) THEN
  CALL VAL7(UO)
  CALL FRICSHUN7(UO,UL1,UL2,UL3)
  ENDIF
IF (KASE.EQ.8) THEN
  CALL VAL8(UO)
  CALL FRICSHUN8(UO,UL1,UL2,UL3)
  ENDIF
IF (KASE.EQ.9) THEN
  CALL VAL9(UO)
  CALL FRICSHUN9(UO,UL1,UL2,UL3)
  ENDIF
 RETURN
```

```
      END
C=====================================================================C
C                                                                     C
C  CHECK determines what case the solution is currently in.          C
C                                                                     C
C                                                                     C
      SUBROUTINE CHECK(UO,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
C
      IF (UL1.LE.UO.AND.UL2.LE.UO.AND.UL3.LE.UO) KASE = 1
      IF (UL1.GT.UO.AND.UL2.LE.UO.AND.UL3.LE.UO) KASE = 2
      IF (UL1.LE.UO.AND.UL2.LE.UO.AND.UL3.GT.UO) KASE = 7
      IF (UL1.GT.UO.AND.UL2.GT.UO.AND.UL3.LE.UO) KASE = 3
      IF (UL1.GT.UO.AND.UL2.LE.UO.AND.UL3.GT.UO.AND.UL3.LE.UL1) KASE = 4
      IF (UL1.GT.UO.AND.UL2.LE.UO.AND.UL3.GT.UO.AND.UL3.GT.UL1) KASE = 8
      IF (UL1.GT.UO.AND.UL2.GT.UO.AND.UL3.GT.UO) THEN
         IF (UL1.GT.UL2.AND.UL2.GT.UL3) KASE = 5
         IF (UL1.GT.UL2.AND.UL3.GT.UL2.AND.UL3.LE.UL1) KASE = 6
         IF (UL1.GT.UL2.AND.UL3.GT.UL1) KASE = 9
        ENDIF
      IF (UL2.GT.UL1) KASE = 0
      RETURN
      END
C=====================================================================C
C                                                                     C
C  FRICSHUN1 calculates all of the energy used or lost by the slab   C
C  when the solution is in Case 1.  It also determines the locations of  C
C  the zone interfaces, if they exist.  Additional values (DUCDX,    C
C  STSTRAIN, CENTERSTR) are calculated for use in other parts of the  C
C  program.                                                           C
C                                                                     C
C                                                                     C
      SUBROUTINE FRICSHUN1(UO,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
```

```
      COMMON /CASE_1A/ G11,H11,AS11,BS11,CS11,DS11
      COMMON /CASE_1B/ AC11,BC11,CC11,DC11
C
      GY1 = G11*S
      HY1 = H11*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0

C
      DUCDX = G11*AC11*EXP(G11*UO)-
     +        G11*BC11*EXP(-G11*UO)+
     +        H11*CC11*EXP(H11*UO)-
     +        H11*DC11*EXP(-H11*UO)
      DUCDX = -DUCDX
C
      STSTRAIN = G11*AS11*EXP(G11*UO)-
     +           G11*BS11*EXP(-G11*UO)+
     +           H11*CS11*EXP(H11*UO)-
     +           H11*DS11*EXP(-H11*UO)
C
      CENTERSTR = G11*AC11*EXP(GY1)-
     +            G11*BC11*EXP(-GY1)+
     +            H11*CC11*EXP(HY1)-
     +            H11*DC11*EXP(-HY1)
      CENTERSTR = DUCDX + CENTERSTR
      FRICTION = 0.0
      STREEN = 0.0
      POTENTIAL = 0.0
      FORCE = 0.0
      X = S
      ZZ1 = 0.0
      ZZ2 = 0.0
      ZZ3 = 0.0
      DELX = S/1000.0
 10   ZIG = 0.0
      GY1 = G11*X
      HY1 = H11*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      U1ST = AS11*EXP(GY1)+
     +       BS11*EXP(-GY1)+
     +       CS11*EXP(HY1)+
     +       DS11*EXP(-HY1)
      U1CO = AC11*EXP(GY1)+
     +       BC11*EXP(-GY1)+
     +       CC11*EXP(HY1)+
     +       DC11*EXP(-HY1)
      DU1CO = G11*AC11*EXP(GY1)-
     +        G11*BC11*EXP(-GY1)+
     +        H11*CC11*EXP(HY1)-
     +        H11*DC11*EXP(-HY1)
      DU1ST = G11*AS11*EXP(GY1)-
     +        G11*BS11*EXP(-GY1)+
     +        H11*CS11*EXP(HY1)-
```

```
     +        H11*DS11*EXP(-HY1)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U1CO,U1ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
      IF (X.LE.0.0) RETURN
   40 FRICTION = FRICTION + 0.5*XK3*WIDT*DELX*U1CO**2.0 +
     +           0.5*XK1*PI*STLDIA*DELX*(U1CO-U1ST)**2.0
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU1CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU1CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU1ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U1CO
      X = X - DELX
      IF (X.GE.0.0) GOTO 10
      RETURN
      END


C=========================================================================C
C                                                                         C
C  FRICSHUN2 calculates all of the energy used or lost by the slab       C
C  when the solution is in Case 2.  It also determines the locations of  C
C  the zone interfaces, if they exist.  Additional values (DUCDX,        C
C  STSTRAIN, CENTERSTR) are calculated for use in other parts of the     C
C  program.                                                               C
C                                                                         C
C                                                                         C
      SUBROUTINE FRICSHUN2(UO,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_2A/ G21,H21,G22,H22,UCP22,USP22
      COMMON /CASE_2B/ AC21,BC21,CC21,DC21,AS21,BS21,CS21,DS21
      COMMON /CASE_2C/ AC22,BC22,CC22,DC22,AS22,BS22,CS22,DS22
C
      GY1 = G21*S
      HY1 = H21*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      DUCDX = G22*AC22*EXP(G22*UO)-
     +        G22*BC22*EXP(-G22*UO)-
     +        H22*CC22*SIN(H22*UO)+
     +        H22*DC22*COS(H22*UO)
      DUCDX = -DUCDX
      STSTRAIN =  G22*AS22*EXP(G22*UO)-
     +            G22*BS22*EXP(-G22*UO)-
     +            H22*CS22*SIN(H22*UO)+
     +            H22*DS22*COS(H22*UO)
```

```
C
      CENTERSTR = G21*AC21*EXP(GY1)-
     +            G21*BC21*EXP(-GY1)+
     +            H21*CC21*EXP(H21*S)-
     +            H21*DC21*EXP(-H21*S)
      CENTERSTR = DUCDX + CENTERSTR
C
      FRICTION = 0.0
      STREEN = 0.0
      POTENTIAL = 0.0
      FORCE = 0.0
      X = S
      DELX = S/1000.0
      ZZ1 = 0.0
      ZZ2 = 0.0
      ZZ3 = 0.0
 10   ZIG = 0.0
      GY1 = G21*X
      HY1 = H21*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL1) THEN
        U2CO = AC22*EXP(G22*X)+
     +         BC22*EXP(-G22*X)+
     +         CC22*COS(H22*X)+
     +         DC22*SIN(H22*X)+UCP22
        U2ST = AS22*EXP(G22*X)+
     +         BS22*EXP(-G22*X)+
     +         CS22*COS(H22*X)+
     +         DS22*SIN(H22*X)+USP22
        DU2CO = G22*AC22*EXP(G22*X)-
     +          G22*BC22*EXP(-G22*X)-
     +          H22*CC22*SIN(H22*X)+
     +          H22*DC22*COS(H22*X)
        DU2ST = G22*AS22*EXP(G22*X)-
     +          G22*BS22*EXP(-G22*X)-
     +          H22*CS22*SIN(H22*X)+
     +          H22*DS22*COS(H22*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U2CO,U2ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
 100  FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 +
     +           XK2*((STLLIM-(U2CO-U2ST))**2.0-(STLLIM-STLSLP)**2.0))+
     +           XK3*WIDT*U2CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU2CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU2CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU2ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U2CO
      ELSE
        U1CO = AC21*EXP(GY1)+
     +         BC21*EXP(-GY1)+
     +         CC21*EXP(HY1)+
     +         DC21*EXP(-HY1)
        U1ST = AS21*EXP(GY1)+
     +         BS21*EXP(-GY1)+
```

```
     +         CS21*EXP(HY1)+
     +         DS21*EXP(-HY1)
        DU1CO = G21*AC21*EXP(GY1)-
     +          G21*BC21*EXP(-GY1)+
     +          H21*CC21*EXP(HY1)-
     +          H21*DC21*EXP(-HY1)
        DU1ST = G21*AS21*EXP(GY1)-
     +          G21*BS21*EXP(-GY1)+
     +          H21*CS21*EXP(HY1)-
     +          H21*DS21*EXP(-HY1)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U1CO,U1ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
      IF (X.LE.0.0) RETURN
 130  FRICTION = FRICTION + 0.5 *DELX*(XK1*PI*STLDIA*(U1CO-U1ST)**2.0+
     +           XK3*WIDT*U1CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU1CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU1CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU1ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U1CO
      ENDIF
      X = X - DELX
      IF (X.GE.0.0) GOTO 10
      RETURN
      END

C========================================================================C
C                                                                        C
C  FRICSHUN3 calculates all of the energy used or lost by the slab       C
C  when the solution is in Case 3.  It also determines the locations of  C
C  the zone interfaces, if they exist.  Additional values (DUCDX,        C
C  STSTRAIN, CENTERSTR) are calculated for use in other parts of the     C
C  program.                                                              C
C                                                                        C
C                                                                        C
      SUBROUTINE FRICSHUN3(UO,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_3A/ G31,H31,G32,H32,G33,UCP32,USP32
      COMMON /CASE_3B/ AC31,BC31,CC31,DC31,AS31,BS31,CS31,DS31
      COMMON /CASE_3C/ AC32,BC32,CC32,DC32,AS32,BS32,CS32,DS32
      COMMON /CASE_3D/ AC33,BC33
C
      GY1 = G31*S
      HY1 = H31*S
      IF (GY1.GE.700.0) GYl = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
```

```
C
      DUCDX =   G33*AC33*EXP(G33*UO)  -
     +          G33*BC33*EXP(-G33*UO)
      DUCDX = -DUCDX
      STSTRAIN =   G32*AS32*EXP(G32*VL2)-
     +             G32*BS32*EXP(-G32*VL2)-
     +             H32*CS32*SIN(H32*VL2)+
     +             H32*DS32*COS(H32*VL2)
C
      CENTERSTR = G31*AC31*EXP(GY1)-
     +            G31*BC31*EXP(-GY1)+
     +            H31*CC31*EXP(HY1)-
     +            H31*DC31*EXP(-HY1)
      CENTERSTR = DUCDX + CENTERSTR
C
      FRICTION = 0.0
      STREEN = 0.0
      POTENTIAL = 0.0
      FORCE = 0.0
      X = S
      DELX = S/1000.0
      ZZ1 = 0.0
      ZZ2 = 0.0
      ZZ3 = 0.0
 10   ZIG = 0.0
      GY1 = G31*X
      HY1 = H31*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL2) THEN
        U3CO = AC33*EXP(G33*X)+
     +         BC33*EXP(-G33*X)
        U3ST = STSTRAIN*X
        DU3CO = G33*AC33*EXP(G33*X)-
     +          G33*BC33*EXP(-G33*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U3CO,U3ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
 70   FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 -
     +           XK2*(STLLIM-STLSLP)**2.0)+WIDT*(XK3*U3CO**2.0))
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU3CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU3CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*STSTRAIN**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U3CO
      ELSE
      IF (X.LE.VL1) THEN
        U2CO = AC32*EXP(G32*X)+
     +         BC32*EXP(-G32*X)+
     +         CC32*COS(H32*X)+
     +         DC32*SIN(H32*X)+UCP32
        U2ST = AS32*EXP(G32*X)+
     +         BS32*EXP(-G32*X)+
     +         CS32*COS(H32*X)+
     +         DS32*SIN(H32*X)+USP32
        DU2CO = G32*AC32*EXP(G32*X)-
```

```
     +          G32*BC32*EXP(-G32*X)-
     +          H32*CC32*SIN(H32*X)+
     +          H32*DC32*COS(H32*X)
        DU2ST = G32*AS32*EXP(G32*X)-
     +          G32*BS32*EXP(-G32*X)-
     +          H32*CS32*SIN(H32*X)+
     +          H32*DS32*COS(H32*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U2CO,U2ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
 100  FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 +
     +           XK2*((STLLIM-(U2CO-U2ST))**2.0-(STLLIM-STLSLP)**2.0))+
     +           XK3*WIDT*U2CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU2CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU2CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU2ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U2CO
      ELSE
        U1CO = AC31*EXP(GY1)+
     +         BC31*EXP(-GY1)+
     +         CC31*EXP(HY1)+
     +         DC31*EXP(-HY1)
        U1ST = AS31*EXP(GY1)+
     +         BS31*EXP(-GY1)+
     +         CS31*EXP(HY1)+
     +         DS31*EXP(-HY1)
        DU1CO = G31*AC31*EXP(GY1)-
     +          G31*BC31*EXP(-GY1)+
     +          H31*CC31*EXP(HY1)-
     +          H31*DC31*EXP(-HY1)
        DU1ST = G31*AS31*EXP(GY1)-
     +          G31*BS31*EXP(-GY1)+
     +          H31*CS31*EXP(HY1)-
     +          H31*DS31*EXP(-HY1)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U1CO,U1ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
      IF (X.LE.0.0) RETURN
 130  FRICTION = FRICTION + 0.5 *DELX*(XK1*PI*STLDIA*(U1CO-U1ST)**2.0+
     +           XK3*WIDT*U1CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU1CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU1CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU1ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U1CO
      ENDIF
      ENDIF
      X = X - DELX
      IF (X.GE.0.0) GOTO 10
      RETURN
      END

C=================================================================C
C                                                                 C
C  FRICSHUN4 calculates all of the energy used or lost by the slab     C
C  when the solution is in Case 4.  It also determines the locations of   C
C  the zone interfaces, if they exist.  Additional values (DUCDX,        C
```

```
C  STSTRAIN, CENTERSTR) are calculated for use in other parts of the      C
C  program.                                                                C
C                                                                          C
C                                                                          C
      SUBROUTINE FRICSHUN4(UO,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_4A/ G41,H41,G42,H42,G43,H43,UCP43,USP43,UCP42,USP42
      COMMON /CASE_4B/ AC41,BC41,CC41,DC41,AS41,BS41,CS41,DS41
      COMMON /CASE_4C/ AC42,BC42,CC42,DC42,AS42,BS42,CS42,DS42
      COMMON /CASE_4D/ AC43,BC43,CC43,DC43,AS43,BS43,CS43,DS43
C
      GY1 = G41*S
      HY1 = H41*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      DUCDX = -G43*AC43*SIN(G43*UO) +
     +         G43*BC43*COS(G43*UO) -
     +         H43*CC43*SIN(H43*UO) +
     +         H43*DC43*COS(H43*UO)
      DUCDX = -DUCDX
      STSTRAIN = -G43*AS43*SIN(G43*UO)+
     +            G43*BS43*COS(G43*UO)-
     +            H43*CS43*SIN(H43*UO)+
     +            H43*DS43*COS(H43*UO)
C
      CENTERSTR = G41*AC41*EXP(GY1)-
     +            G41*BC41*EXP(-GY1)+
     +            H41*CC41*EXP(HY1)-
     +            H41*DC41*EXP(-HY1)
      CENTERSTR = DUCDX + CENTERSTR
C
      FRICTION = 0.0
      STREEN = 0.0
      POTENTIAL = 0.0
      FORCE = 0.0
      X = S
      DELX = S/1000.0
      ZZ1 = 0.0
      ZZ2 = 0.0
      ZZ3 = 0.0
 10   ZIG = 0.0
      GY1 = G41*X
      HY1 = H41*X
      IF (GY1.GE.700.0) GY1 = 700.0
```

```
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL3) THEN
        U3CO = AC43*COS(G43*X)+
     +         BC43*SIN(G43*X)+
     +         CC43*COS(H43*X)+
     +         DC43*SIN(H43*X)+UCP43
        U3ST = AS43*COS(G43*X)+
     +         BS43*SIN(G43*X)+
     +         CS43*COS(H43*X)+
     +         DS43*SIN(H43*X)+USP43
      DU3CO = -G43*AC43*SIN(G43*X) +
     +         G43*BC43*COS(G43*X) -
     +         H43*CC43*SIN(H43*X) +
     +         H43*DC43*COS(H43*X)
      DU3ST = -G43*AS43*SIN(G43*X) +
     +         G43*BS43*COS(G43*X) -
     +         H43*CS43*SIN(H43*X) +
     +         H43*DS43*COS(H43*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U3CO,U3ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
70    FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 +
     +           XK2*((STLLIM-(U3CO-U3ST))**2.0-(STLLIM-STLSLP)**2.0))+
     +           WIDT*(XK3*CONSLP**2.0+XK4*((CONLIM-U3CO)**2.0-
     +           (CONLIM-CONSLP)**2.0)))
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU3CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU3CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU3ST**2.0
      FORCE = FORCE + DELX*WIDT*(XC2+XK4*U3CO)
      ELSE
      IF (X.LE.VL1) THEN
        U2CO = AC42*EXP(G42*X)+
     +         BC42*EXP(-G42*X)+
     +         CC42*COS(H42*X)+
     +         DC42*SIN(H42*X)+UCP42
        U2ST = AS42*EXP(G42*X)+
     +         BS42*EXP(-G42*X)+
     +         CS42*COS(H42*X)+
     +         DS42*SIN(H42*X)+USP42
        DU2CO = G42*AC42*EXP(G42*X)-
     +          G42*BC42*EXP(-G42*X)-
     +          H42*CC42*SIN(H42*X)+
     +          H42*DC42*COS(H42*X)
        DU2ST = G42*AS42*EXP(G42*X)-
     +          G42*BS42*EXP(-G42*X)-
     +          H42*CS42*SIN(H42*X)+
     +          H42*DS42*COS(H42*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U2CO,U2ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
100   FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 +
     +           XK2*((STLLIM-(U2CO-U2ST))**2.0-(STLLIM-STLSLP)**2.0))+
     +           XK3*WIDT*U2CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU2CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU2CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU2ST**2.0
```

```
      FORCE = FORCE + DELX*WIDT*XK3*U2CO
      ELSE
        U1CO = AC41*EXP(GY1)+
     +         BC41*EXP(-GY1)+
     +         CC41*EXP(HY1)+
     +         DC41*EXP(-HY1)
        U1ST = AS41*EXP(GY1)+
     +         BS41*EXP(-GY1)+
     +         CS41*EXP(HY1)+
     +         DS41*EXP(-HY1)
        DU1CO = G41*AC41*EXP(GY1)-
     +          G41*BC41*EXP(-GY1)+
     +          H41*CC41*EXP(HY1)-
     +          H41*DC41*EXP(-HY1)
        DU1ST = G41*AS41*EXP(GY1)-
     +          G41*BS41*EXP(-GY1)+
     +          H41*CS41*EXP(HY1)-
     +          H41*DS41*EXP(-HY1)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U1CO,U1ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
      IF (X.LE.0.0) RETURN
 130  FRICTION = FRICTION + 0.5 *DELX*(XK1*PI*STLDIA*(U1CO-U1ST)**2.0+
     +           XK3*WIDT*U1CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU1CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU1CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU1ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U1CO
      ENDIF
      ENDIF
      X = X - DELX
      IF (X.GE.0.0) GOTO 10
      RETURN
      END

C=======================================================================C
C                                                                       C
C  FRICSHUN5 calculates all of the energy used or lost by the slab      C
C  when the solution is in Case 5.  It also determines the locations of C
C  the zone interfaces, if they exist.  Additional values (DUCDX,       C
C  STSTRAIN, CENTERSTR) are calculated for use in other parts of the    C
C  program.                                                             C
C                                                                       C
C                                                                       C
      SUBROUTINE FRICSHUN5(UO,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
```

```
      COMMON /CASE_5A/ G51,H51,G52,H52,G53,G54
      COMMON /CASE_5B/ AC51,BC51,CC51,DC51,AS51,BS51,CS51,DS51
      COMMON /CASE_5C/ AC52,BC52,CC52,DC52,AS52,BS52,CS52,DS52
      COMMON /CASE_5D/ AC53,BC53,AC54,BC54,UCP54,UCP52,USP52
C
      GY1 = G51*S
      HY1 = H51*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      DUCDX = -G54*AC54*SIN(G54*UO) +
     +         G54*BC54*COS(G54*UO)
      DUCDX = -DUCDX
      STSTRAIN =  G52*AS52*EXP(G52*VL2)-
     +            G52*BS52*EXP(-G52*VL2)-
     +            H52*CS52*SIN(H52*VL2)+
     +            H52*DS52*COS(H52*VL2)
C
      CENTERSTR = G51*AC51*EXP(GY1)-
     +            G51*BC51*EXP(-GY1)+
     +            H51*CC51*EXP(HY1)-
     +            H51*DC51*EXP(-HY1)
      CENTERSTR = DUCDX + CENTERSTR
C
      FRICTION = 0.0
      STREEN = 0.0
      POTENTIAL = 0.0
      FORCE = 0.0
      X = S
      DELX = S/1000.0
      ZZ1 = 0.0
      ZZ2 = 0.0
      ZZ3 = 0.0
 10   ZIG = 0.0
      GY1 = G51*X
      HY1 = H51*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL3) THEN
        U4CO = AC54*COS(G54*X)+
     +         BC54*SIN(G54*X)+UCP54
        U4ST = STSTRAIN*X
        DU4CO = -G54*AC54*SIN(G54*X)+
     +           G54*BC54*COS(G54*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U4CO,U4ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
 40   FRICTION = FRICTION + 0.5*XK1*PI*DELX*STLDIA*STLSLP**2.0 -
     +           0.5*XK2*PI*DELX*STLDIA*(STLLIM-STLSLP)**2.0 +
     +           0.5*XK3*WIDT*DELX*CONSLP**2.0 +0.5*XK4*WIDT*DELX*
     +           ((CONLIM-U4CO)**2.0-(CONLIM-CONSLP)**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU4CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU4CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*STSTRAIN**2.0
      FORCE = FORCE + DELX*WIDT*(XC2+XK4*U4CO)
```

```
      ELSE
      IF (X.LE.VL2) THEN
        U3CO = AC53*EXP(G53*X)+
     +         BC53*EXP(-G53*X)
        U3ST = STSTRAIN*X
        DU3CO = G53*AC53*EXP(G53*X)-
     +          G53*BC53*EXP(-G53*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U3CO,U3ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
70    FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 -
     +           XK2*(STLLIM-STLSLP)**2.0)+WIDT*(XK3*U3CO**2.0))
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU3CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU3CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*STSTRAIN**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U3CO
      ELSE
      IF (X.LE.VL1) THEN
        U2CO = AC52*EXP(G52*X)+
     +         BC52*EXP(-G52*X)+
     +         CC52*COS(H52*X)+
     +         DC52*SIN(H52*X)+UCP52
        U2ST = AS52*EXP(G52*X)+
     +         BS52*EXP(-G52*X)+
     +         CS52*COS(H52*X)+
     +         DS52*SIN(H52*X)+USP52
        DU2CO = G52*AC52*EXP(G52*X)-
     +          G52*BC52*EXP(-G52*X)-
     +          H52*CC52*SIN(H52*X)+
     +          H52*DC52*COS(H52*X)
        DU2ST = G52*AS52*EXP(G52*X)-
     +          G52*BS52*EXP(-G52*X)-
     +          H52*CS52*SIN(H52*X)+
     +          H52*DS52*COS(H52*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U2CO,U2ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
100   FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 +
     +           XK2*((STLLIM-(U2CO-U2ST))**2.0-(STLLIM-STLSLP)**2.0))+
     +           XK3*WIDT*U2CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU2CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU2CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU2ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U2CO
      ELSE
        U1CO = AC51*EXP(GY1)+
     +         BC51*EXP(-GY1)+
     +         CC51*EXP(HY1)+
     +         DC51*EXP(-HY1)
        U1ST = AS51*EXP(GY1)+
     +         BS51*EXP(-GY1)+
     +         CS51*EXP(HY1)+
     +         DS51*EXP(-HY1)
        DU1CO = G51*AC51*EXP(GY1)-
     +          G51*BC51*EXP(-GY1)+
     +          H51*CC51*EXP(HY1)-
```

```
     +          H51*DC51*EXP(-HY1)
        DU1ST = G51*AS51*EXP(GY1)-
     +          G51*BS51*EXP(-GY1)+
     +          H51*CS51*EXP(HY1)-
     +          H51*DS51*EXP(-HY1)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U1CO,U1ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
      IF (X.LE.0.0) RETURN
 130  FRICTION = FRICTION + 0.5 *DELX*(XK1*PI*STLDIA*(U1CO-U1ST)**2.0+
     +           XK3*WIDT*U1CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU1CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU1CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU1ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U1CO
      ENDIF
      ENDIF
      ENDIF
      X = X - DELX
      IF (X.GE.0.0) GOTO 10
      RETURN
      END

C======================================================================C
C                                                                      C
C  FRICSHUN6 calculates all of the energy used or lost by the slab     C
C  when the solution is in Case 6.  It also determines the locations of C
C  the zone interfaces, if they exist.  Additional values (DUCDX,      C
C  STSTRAIN, CENTERSTR) are calculated for use in other parts of the   C
C  program.                                                            C
C                                                                      C
C                                                                      C
      SUBROUTINE FRICSHUN6(UO,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_6A/ G61,H61,G62,H62,G63,H63,G64
      COMMON /CASE_6B/ AC61,BC61,CC61,DC61,AS61,BS61,CS61,DS61
      COMMON /CASE_6C/ AC62,BC62,CC62,DC62,AS62,BS62,CS62,DS62
      COMMON /CASE_6D/ AC63,BC63,CC63,DC63,AS63,BS63,CS63,DS63
      COMMON /CASE_6E/ AC64,BC64,UCP64,UCP63,USP63,UCP62,USP62
C
      GY1 = G61*S
      HY1 = H61*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      DUCDX = -G64*AC64*SIN(G64*UO) +
```

```
     +         G64*BC64*COS(G64*UO)
      DUCDX = -DUCDX
      STSTRAIN = -G63*AS63*SIN(G63*VL2)+
     +            G63*BS63*COS(G63*VL2)-
     +            H63*CS63*SIN(H63*VL2)+
     +            H63*DS63*COS(H63*VL2)
C
      CENTERSTR = G61*AC61*EXP(GY1)-
     +            G61*BC61*EXP(-GY1)+
     +            H61*CC61*EXP(HY1)-
     +            H61*DC61*EXP(-HY1)
      CENTERSTR = DUCDX + CENTERSTR
C
      FRICTION = 0.0
      STREEN = 0.0
      POTENTIAL = 0.0
      FORCE = 0.0
      X = S
      DELX = S/1000.0
      ZZ1 = 0.0
      ZZ2 = 0.0
      ZZ3 = 0.0
 10   ZIG = 0.0
      GY1 = G61*X
      HY1 = H61*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL2) THEN
        U4CO = AC64*COS(G64*X)+
     +         BC64*SIN(G64*X)+UCP64
        U4ST = STSTRAIN*X
        DU4CO = -G64*AC64*SIN(G64*X)+
     +           G64*BC64*COS(G64*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U4CO,U4ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
 40   FRICTION = FRICTION + 0.5*XK1*PI*DELX*STLDIA*STLSLP**2.0 -
     +           0.5*XK2*PI*DELX*STLDIA*(STLLIM-STLSLP)**2.0 +
     +           0.5*XK3*WIDT*DELX*CONSLP**2.0 +0.5*XK4*WIDT*DELX*
     +           ((CONLIM-U4CO)**2.0-(CONLIM-CONSLP)**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU4CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU4CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*STSTRAIN**2.0
      FORCE = FORCE + DELX*WIDT*(XC2+XK4*U4CO)
      ELSE
      IF (X.LE.VL3) THEN
        U3CO = AC63*COS(G63*X)+
     +         BC63*SIN(G63*X)+
     +         CC63*COS(H63*X)+
     +         DC63*SIN(H63*X)+UCP63
        U3ST = AS63*COS(G63*X)+
     +         BS63*SIN(G63*X)+
     +         CS63*COS(H63*X)+
     +         DS63*SIN(H63*X)+USP63
        DU3CO = -G63*AC63*SIN(G63*X)+
```

```
     +          G63*BC63*COS(G63*X)-
     +          H63*CC63*SIN(H63*X)+
     +          H63*DC63*COS(H63*X)
        DU3ST = -G63*AS63*SIN(G63*X)+
     +          G63*BS63*COS(G63*X)-
     +          H63*CS63*SIN(H63*X)+
     +          H63*DS63*COS(H63*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U3CO,U3ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
70    FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 +
     +           XK2*((STLLIM-(U3CO-U3ST))**2.0-(STLLIM-STLSLP)**2.0))+
     +           WIDT*(XK3*CONSLP**2.0+XK4*((CONLIM-U3CO)**2.0-
     +           (CONLIM-CONSLP)**2.0)))
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU3CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU3CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU3ST**2.0
      FORCE = FORCE + DELX*WIDT*(XC2+XK4*U3CO)
      ELSE
      IF (X.LE.VL1) THEN
        U2CO = AC62*EXP(G62*X)+
     +         BC62*EXP(-G62*X)+
     +         CC62*COS(H62*X)+
     +         DC62*SIN(H62*X)+UCP62
        U2ST = AS62*EXP(G62*X)+
     +         BS62*EXP(-G62*X)+
     +         CS62*COS(H62*X)+
     +         DS62*SIN(H62*X)+USP62
        DU2CO = G62*AC62*EXP(G62*X)-
     +          G62*BC62*EXP(-G62*X)-
     +          H62*CC62*SIN(H62*X)+
     +          H62*DC62*COS(H62*X)
        DU2ST = G62*AS62*EXP(G62*X)-
     +          G62*BS62*EXP(-G62*X)-
     +          H62*CS62*SIN(H62*X)+
     +          H62*DS62*COS(H62*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U2CO,U2ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
100   FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 +
     +           XK2*((STLLIM-(U2CO-U2ST))**2.0-(STLLIM-STLSLP)**2.0))+
     +           XK3*WIDT*U2CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU2CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU2CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU2ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U2CO
      ELSE
        U1CO = AC61*EXP(GY1)+
     +         BC61*EXP(-GY1)+
     +         CC61*EXP(HY1)+
     +         DC61*EXP(-HY1)
        U1ST = AS61*EXP(GY1)+
     +         BS61*EXP(-GY1)+
     +         CS61*EXP(HY1)+
     +         DS61*EXP(-HY1)
        DU1CO = G61*AC61*EXP(GY1)-
```

```
     +          G61*BC61*EXP(-GY1)+
     +          H61*CC61*EXP(HY1)-
     +          H61*DC61*EXP(-HY1)
        DU1ST = G61*AS61*EXP(GY1)-
     +          G61*BS61*EXP(-GY1)+
     +          H61*CS61*EXP(HY1)-
     +          H61*DS61*EXP(-HY1)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U1CO,U1ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
      IF (X.LE.0.0) RETURN
 130  FRICTION = FRICTION + 0.5 *DELX*(XK1*PI*STLDIA*(U1CO-U1ST)**2.0+
     +           XK3*WIDT*U1CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU1CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU1CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU1ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U1CO
      ENDIF
      ENDIF
      ENDIF
      X = X - DELX
      IF (X.GE.0.0) GOTO 10
      RETURN
      END

C=====================================================================C
C                                                                     C
C  FRICSHUN7 calculates all of the energy used or lost by the slab    C
C  when the solution is in Case 7.  It also determines the locations of C
C  the zone interfaces, if they exist.  Additional values (DUCDX,     C
C  STSTRAIN, CENTERSTR) are calculated for use in other parts of the  C
C  program.                                                           C
C                                                                     C
C                                                                     C
      SUBROUTINE FRICSHUN7(U0,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_7A/ G71,H71,G72,H72,UCP72,USP72
      COMMON /CASE_7B/ AC71,BC71,CC71,DC71,AS71,BS71,CS71,DS71
      COMMON /CASE_7C/ AC72,BC72,CC72,DC72,AS72,BS72,CS72,DS72
C
      GY1 = G71*S
      HY1 = H71*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      DUCDX = G72*AC72*EXP(G72*U0)-
```

```
     +         G72*BC72*EXP(-G72*UO)-
     +         H72*CC72*SIN(H72*UO)+
     +         H72*DC72*COS(H72*UO)
      DUCDX = -DUCDX
      STSTRAIN =  G72*AS72*EXP(G72*UO)-
     +            G72*BS72*EXP(-G72*UO)-
     +            H72*CS72*SIN(H72*UO)+
     +            H72*DS72*COS(H72*UO)
C
      CENTERSTR = G71*AC71*EXP(GY1)-
     +            G71*BC71*EXP(-GY1)+
     +            H71*CC71*EXP(HY1)-
     +            H71*DC71*EXP(-HY1)
      CENTERSTR = DUCDX + CENTERSTR
C
      FRICTION = 0.0
      STREEN = 0.0
      POTENTIAL = 0.0
      FORCE = 0.0
      X = S
      DELX = S/1000.0
      ZZ1 = 0.0
      ZZ2 = 0.0
      ZZ3 = 0.0
 10   ZIG = 0.0
      GY1 = G71*X
      HY1 = H71*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL3) THEN
        U2CO = AC72*EXP(G72*X)+
     +         BC72*EXP(-G72*X)+
     +         CC72*COS(H72*X)+
     +         DC72*SIN(H72*X)+UCP72
        U2ST = AS72*EXP(G72*X)+
     +         BS72*EXP(-G72*X)+
     +         CS72*COS(H72*X)+
     +         DS72*SIN(H72*X)+USP72
        DU2CO = G72*AC72*EXP(G72*X)-
     +          G72*BC72*EXP(-G72*X)-
     +          H72*CC72*SIN(H72*X)+
     +          H72*DC72*COS(H72*X)
        DU2ST = G72*AS72*EXP(G72*X)-
     +          G72*BS72*EXP(-G72*X)-
     +          H72*CS72*SIN(H72*X)+
     +          H72*DS72*COS(H72*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U2CO,U2ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
 100  FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*(U2CO-U2ST)**2.0)+
     +           WIDT*XK3*CONSLP**2.0 + WIDT*XK4*((CONLIM-U2CO)**2.0-
     +           (CONLIM-CONSLP)**2.0))
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU2CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU2CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU2ST**2.0
```

```
      FORCE = FORCE + DELX*WIDT*(XC2+XK4*U2CO)
      ELSE
        U1CO = AC71*EXP(GY1)+
     +          BC71*EXP(-GY1)+
     +          CC71*EXP(HY1)+
     +          DC71*EXP(-HY1)
        U1ST = AS71*EXP(GY1)+
     +          BS71*EXP(-GY1)+
     +          CS71*EXP(HY1)+
     +          DS71*EXP(-HY1)
        DU1CO = G71*AC71*EXP(GY1)-
     +           G71*BC71*EXP(-GY1)+
     +           H71*CC71*EXP(HY1)-
     +           H71*DC71*EXP(-HY1)
        DU1ST = G71*AS71*EXP(GY1)-
     +           G71*BS71*EXP(-GY1)+
     +           H71*CS71*EXP(HY1)-
     +           H71*DS71*EXP(-HY1)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U1CO,U1ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
      IF (X.LE.0.0) RETURN
 130  FRICTION = FRICTION + 0.5 *DELX*(XK1*PI*STLDIA*(U1CO-U1ST)**2.0+
     +            XK3*WIDT*U1CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU1CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU1CO+DUCDX)**2.0
     +           + 0.5*ASTL*DELX*ES*DU1ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U1CO
      ENDIF
      X = X - DELX
      IF (X.GE.0.0) GOTO 10
      RETURN
      END

C=====================================================================C
C                                                                     C
C  FRICSHUN8 calculates all of the energy used or lost by the slab    C
C  when the solution is in Case 8.  It also determines the locations of C
C  the zone interfaces, if they exist.  Additional values (DUCDX,     C
C  STSTRAIN, CENTERSTR) are calculated for use in other parts of the  C
C  program.                                                           C
C                                                                     C
C                                                                     C
      SUBROUTINE FRICSHUN8(UO,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_8A/ G81,H81,G82,H82,G83,H83,UCP83,USP83,UCP82,USP82
```

```
      COMMON /CASE_8B/ AC81,BC81,CC81,DC81,AS81,BS81,CS81,DS81
      COMMON /CASE_8C/ AC82,BC82,CC82,DC82,AS82,BS82,CS82,DS82
      COMMON /CASE_8D/ AC83,BC83,CC83,DC83,AS83,BS83,CS83,DS83
C
      GY1 = G81*S
      HY1 = H81*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      DUCDX = -G83*AC83*SIN(G83*UO) +
     +         G83*BC83*COS(G83*UO) -
     +         H83*CC83*SIN(H83*UO) +
     +         H83*DC83*COS(H83*UO)
      DUCDX = -DUCDX
      STSTRAIN = -G83*AS83*SIN(G83*UO)+
     +            G83*BS83*COS(G83*UO)-
     +            H83*CS83*SIN(H83*UO)+
     +            H83*DS83*COS(H83*UO)
C
      CENTERSTR = G81*AC81*EXP(GY1)-
     +            G81*BC81*EXP(-GY1)+
     +            H81*CC81*EXP(HY1)-
     +            H81*DC81*EXP(-HY1)
      CENTERSTR = DUCDX + CENTERSTR
C
      FRICTION = 0.0
      STREEN = 0.0
      POTENTIAL = 0.0
      FORCE = 0.0
      X = S
      DELX = S/1000.0
      ZZ1 = 0.0
      ZZ2 = 0.0
      ZZ3 = 0.0
 10   ZIG = 0.0
      GY1 = G81*X
      HY1 = H81*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL1) THEN
        U3CO = AC83*COS(G83*X)+
     +         BC83*SIN(G83*X)+
     +         CC83*COS(H83*X)+
     +         DC83*SIN(H83*X)+UCP83
        U3ST = AS83*COS(G83*X)+
     +         BS83*SIN(G83*X)+
     +         CS83*COS(H83*X)+
     +         DS83*SIN(H83*X)+USP83
        DU3CO = -G83*AC83*SIN(G83*X)+
     +           G83*BC83*COS(G83*X)-
     +           H83*CC83*SIN(H83*X)+
     +           H83*DC83*COS(H83*X)
        DU3ST = -G83*AS83*SIN(G83*X)+
     +           G83*BS83*COS(G83*X)-
```

```
     +           H83*CS83*SIN(H83*X)+
     +           H83*DS83*COS(H83*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U3CO,U3ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
70    FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 +
     +           XK2*((STLLIM-(U3CO-U3ST))**2.0-(STLLIM-STLSLP)**2.0))+
     +           WIDT*(XK3*CONSLP**2.0+XK4*((CONLIM-U3CO)**2.0-
     +           (CONLIM-CONSLP)**2.0)))
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU3CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU3CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU3ST**2.0
      FORCE = FORCE + DELX*WIDT*(XC2+XK4*U3CO)
      ELSE
      IF (X.LE.VL3) THEN
        U2CO = AC82*EXP(G82*X)+
     +         BC82*EXP(-G82*X)+
     +         CC82*COS(H82*X)+
     +         DC82*SIN(H82*X)+UCP82
        U2ST = AS82*EXP(G82*X)+
     +         BS82*EXP(-G82*X)+
     +         CS82*COS(H82*X)+
     +         DS82*SIN(H82*X)+USP82
        DU2CO = G82*AC82*EXP(G82*X)-
     +          G82*BC82*EXP(-G82*X)-
     +          H82*CC82*SIN(H82*X)+
     +          H82*DC82*COS(H82*X)
        DU2ST = G82*AS82*EXP(G82*X)-
     +          G82*BS82*EXP(-G82*X)-
     +          H82*CS82*SIN(H82*X)+
     +          H82*DS82*COS(H82*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U2CO,U2ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
100   FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*(U2CO-U2ST)**2.0)+
     +           WIDT*XK3*CONSLP**2.0 + WIDT*XK4*((CONLIM-U2CO)**2.0-
     +           (CONLIM-CONSLP)**2.0))
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU2CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU2CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU2ST**2.0
      FORCE = FORCE + DELX*WIDT*(XC2+XK4*U2CO)
      ELSE
        U1CO = AC81*EXP(GY1)+
     +         BC81*EXP(-GY1)+
     +         CC81*EXP(HY1)+
     +         DC81*EXP(-HY1)
        U1ST = AS81*EXP(GY1)+
     +         BS81*EXP(-GY1)+
     +         CS81*EXP(HY1)+
     +         DS81*EXP(-HY1)
        DU1CO = G81*AC81*EXP(GY1)-
     +          G81*BC81*EXP(-GY1)+
     +          H81*CC81*EXP(HY1)-
     +          H81*DC81*EXP(-HY1)
        DU1ST = G81*AS81*EXP(GY1)-
     +          G81*BS81*EXP(-GY1)+
```

```
     +          H81*CS81*EXP(HY1)-
     +          H81*DS81*EXP(-HY1)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U1CO,U1ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
      IF (X.LE.0.0) RETURN
 130  FRICTION = FRICTION + 0.5 *DELX*(XK1*PI*STLDIA*(U1CO-U1ST)**2.0+
     +           XK3*WIDT*U1CO**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU1CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU1CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU1ST**2.0
      FORCE = FORCE + DELX*WIDT*XK3*U1CO
      ENDIF
      ENDIF
      X = X - DELX
      IF (X.GE.0.0) GOTO 10
      RETURN
      END

C=====================================================================C
C                                                                     C
C  FRICSHUN9 calculates all of the energy used or lost by the slab    C
C  when the solution is in Case 9.  It also determines the locations of C
C  the zone interfaces, if they exist.  Additional values (DUCDX,     C
C  STSTRAIN, CENTERSTR) are calculated for use in other parts of the  C
C  program.                                                           C
C                                                                     C
C                                                                     C
C                                                                     C
      SUBROUTINE FRICSHUN9(UO,UL1,UL2,UL3)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_9A/ G91,H91,G92,H92,G93,H93,G94
      COMMON /CASE_9B/ AC91,BC91,CC91,DC91,AS91,BS91,CS91,DS91
      COMMON /CASE_9C/ AC92,BC92,CC92,DC92,AS92,BS92,CS92,DS92
      COMMON /CASE_9D/ AC93,BC93,CC93,DC93,AS93,BS93,CS93,DS93
      COMMON /CASE_9E/ AC94,BC94,UCP94,UCP93,USP93,UCP92,USP92
C
      GY1 = G91*S
      HY1 = H91*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      DUCDX = -G94*AC94*SIN(G94*UO) +
     +         G94*BC94*COS(G94*UO)
      DUCDX = -DUCDX
      STSTRAIN = -G93*AS93*SIN(G93*VL2)+
```

```
     +            G93*BS93*COS(G93*VL2)-
     +            H93*CS93*SIN(H93*VL2)+
     +            H93*DS93*COS(H93*VL2)
C
      CENTERSTR = G91*AC91*EXP(GY1)-
     +            G91*BC91*EXP(-GY1)+
     +            H91*CC91*EXP(HY1)-
     +            H91*DC91*EXP(-HY1)
      CENTERSTR = DUCDX + CENTERSTR
C
      FRICTION = 0.0
      STREEN = 0.0
      POTENTIAL = 0.0
      FORCE = 0.0
      X = S
      DELX = S/1000.0
      ZZ1 = 0.0
      ZZ2 = 0.0
      ZZ3 = 0.0
 10   ZIG = 0.0
      GY1 = G91*X
      HY1 = H91*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL2) THEN
        U4CO = AC94*COS(G94*X)+
     +         BC94*SIN(G94*X)+UCP94
        U4ST = STSTRAIN*X
        DU4CO = -G94*AC94*SIN(G94*X)+
     +           G94*BC94*COS(G94*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U4CO,U4ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
 40   FRICTION = FRICTION + 0.5*XK1*PI*DELX*STLDIA*STLSLP**2.0 -
     +           0.5*XK2*PI*DELX*STLDIA*(STLLIM-STLSLP)**2.0 +
     +           0.5*XK3*WIDT*DELX*CONSLP**2.0 +0.5*XK4*WIDT*DELX*
     +           ((CONLIM-U4CO)**2.0-(CONLIM-CONSLP)**2.0)
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU4CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU4CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*STSTRAIN**2.0
      FORCE = FORCE + DELX*WIDT*(XC2+XK4*U4CO)
      ELSE
      IF (X.LE.VL1) THEN
        U3CO = AC93*COS(G93*X)+
     +         BC93*SIN(G93*X)+
     +         CC93*COS(H93*X)+
     +         DC93*SIN(H93*X)+UCP93
        U3ST = AS93*COS(G93*X)+
     +         BS93*SIN(G93*X)+
     +         CS93*COS(H93*X)+
     +         DS93*SIN(H93*X)+USP93
        DU3CO = -G93*AC93*SIN(G93*X)+
     +           G93*BC93*COS(G93*X)-
     +           H93*CC93*SIN(H93*X)+
     +           H93*DC93*COS(H93*X)
```

```
        DU3ST = -G93*AS93*SIN(G93*X)+
     +           G93*BS93*COS(G93*X)-
     +           H93*CS93*SIN(H93*X)+
     +           H93*DS93*COS(H93*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U3CO,U3ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
70    FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*STLSLP**2.0 +
     +           XK2*((STLLIM-(U3CO-U3ST))**2.0-(STLLIM-STLSLP)**2.0))+
     +           WIDT*(XK3*CONSLP**2.0+XK4*((CONLIM-U3CO)**2.0-
     +           (CONLIM-CONSLP)**2.0)))
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU3CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU3CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU3ST**2.0
      FORCE = FORCE + DELX*WIDT*(XC2+XK4*U3CO)
      ELSE
      IF (X.LE.VL3) THEN
        U2CO = AC92*EXP(G92*X)+
     +         BC92*EXP(-G92*X)+
     +         CC92*COS(H92*X)+
     +         DC92*SIN(H92*X)+UCP92
        U2ST = AS92*EXP(G92*X)+
     +         BS92*EXP(-G92*X)+
     +         CS92*COS(H92*X)+
     +         DS92*SIN(H92*X)+USP92
        DU2CO = G92*AC92*EXP(G92*X)-
     +          G92*BC92*EXP(-G92*X)-
     +          H92*CC92*SIN(H92*X)+
     +          H92*DC92*COS(H92*X)
        DU2ST = G92*AS92*EXP(G92*X)-
     +          G92*BS92*EXP(-G92*X)-
     +          H92*CS92*SIN(H92*X)+
     +          H92*DS92*COS(H92*X)
      CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U2CO,U2ST,ZIG,X)
      IF (ZIG.EQ.1.0) GOTO 10
100   FRICTION = FRICTION + 0.5*DELX*(PI*STLDIA*(XK1*(U2CO-U2ST)**2.0)+
     +           WIDT*XK3*CONSLP**2.0 + WIDT*XK4*((CONLIM-U2CO)**2.0-
     +           (CONLIM-CONSLP)**2.0))
      STREEN = STREEN + 0.5*ACON*DELX*EC*DU2CO**2.0
      POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU2CO+DUCDX)**2.0
     +          + 0.5*ASTL*DELX*ES*DU2ST**2.0
      FORCE = FORCE + DELX*WIDT*(XC2+XK4*U2CO)
      ELSE
        U1CO = AC91*EXP(GY1)+
     +         BC91*EXP(-GY1)+
     +         CC91*EXP(HY1)+
     +         DC91*EXP(-HY1)
        U1ST = AS91*EXP(GY1)+
     +         BS91*EXP(-GY1)+
     +         CS91*EXP(HY1)+
     +         DS91*EXP(-HY1)
        DU1CO = G91*AC91*EXP(GY1)-
     +          G91*BC91*EXP(-GY1)+
     +          H91*CC91*EXP(HY1)-
     +          H91*DC91*EXP(-HY1)
```

```
         DU1ST = G91*AS91*EXP(GY1)-
     +           G91*BS91*EXP(-GY1)+
     +           H91*CS91*EXP(HY1)-
     +           H91*DS91*EXP(-HY1)
       CALL INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,U1CO,U1ST,ZIG,X)
       IF (ZIG.EQ.1.0) GOTO 10
       IF (X.LE.0.0) RETURN
 130   FRICTION = FRICTION + 0.5 *DELX*(XK1*PI*STLDIA*(U1CO-U1ST)**2.0+
     +            XK3*WIDT*U1CO**2.0)
       STREEN = STREEN + 0.5*ACON*DELX*EC*DU1CO**2.0
       POTENTIAL = POTENTIAL + 0.5*ACON*DELX*EC*(DU1CO+DUCDX)**2.0
     +           + 0.5*ASTL*DELX*ES*DU1ST**2.0
       FORCE = FORCE + DELX*WIDT*XK3*U1CO
       ENDIF
       ENDIF
       ENDIF
       X = X - DELX
       IF (X.GE.0.0) GOTO 10
       RETURN
       END

C=======================================================================C
C                                                                       C
C  INTERFACE finds the locations of the interfaces between the zones   C
C                                                                       C
C                                                                       C
       SUBROUTINE INTERFACE(ZZ1,ZZ2,ZZ3,UL1,UL2,UL3,DELX,UCON,USTL,ZIG,X)
       IMPLICIT REAL*8 (A-H, O-Z)
       COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
       IF (ZZ3.GT.0.0) GOTO 10
       IF (UCON.GE.CONSLP) THEN
         IF (DELX.GT.0.02) THEN
           X = X+DELX
           DELX = 0.02
           ZIG = 1.0
           GOTO 30
           ENDIF
         UL3 = X
         ZZ3 = 1.0
         ENDIF
 10    IF (ZZ2.GT.0.0) GOTO 20
       IF ((UCON-USTL).GE.STLLIM) THEN
         IF (DELX.GT.0.02) THEN
           X = X+DELX
           DELX = 0.02
           ZIG = 1.0
           GOTO 30
           ENDIF
         UL2 = X
         ZZ2 = 1.0
         ENDIF
 20    IF (ZZ1.GT.0.0) GOTO 30
       IF ((UCON-USTL).GE.STLSLP) THEN
         IF (DELX.GT.0.02) THEN
```

```
          X = X+DELX
          DELX = 0.02
          ZIG = 1.0
          GOTO 30
          ENDIF
        UL1 = X
        ZZ1 = 1.0
        ENDIF
 30   RETURN
      END

C=====================================================================C
C                                                                     C
C   This routine prints out a summary of the day's stress distributions, C
C   along with other pertinent data.                                  C
C                                                                     C
C                                                                     C
      SUBROUTINE OUTPUT (UO,KDAY,TRIP)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_1A/ G11,H11,AS11,BS11,CS11,DS11
      COMMON /CASE_1B/ AC11,BC11,CC11,DC11
      COMMON /CASE_2A/ G21,H21,G22,H22,UCP22,USP22
      COMMON /CASE_2B/ AC21,BC21,CC21,DC21,AS21,BS21,CS21,DS21
      COMMON /CASE_2C/ AC22,BC22,CC22,DC22,AS22,BS22,CS22,DS22
      COMMON /CASE_3A/ G31,H31,G32,H32,G33,UCP32,USP32
      COMMON /CASE_3B/ AC31,BC31,CC31,DC31,AS31,BS31,CS31,DS31
      COMMON /CASE_3C/ AC32,BC32,CC32,DC32,AS32,BS32,CS32,DS32
      COMMON /CASE_3D/ AC33,BC33
      COMMON /CASE_4A/ G41,H41,G42,H42,G43,H43,UCP43,USP43,UCP42,USP42
      COMMON /CASE_4B/ AC41,BC41,CC41,DC41,AS41,BS41,CS41,DS41
      COMMON /CASE_4C/ AC42,BC42,CC42,DC42,AS42,BS42,CS42,DS42
      COMMON /CASE_4D/ AC43,BC43,CC43,DC43,AS43,BS43,CS43,DS43
      COMMON /CASE_5A/ G51,H51,G52,H52,G53,G54
      COMMON /CASE_5B/ AC51,BC51,CC51,DC51,AS51,BS51,CS51,DS51
      COMMON /CASE_5C/ AC52,BC52,CC52,DC52,AS52,BS52,CS52,DS52
      COMMON /CASE_5D/ AC53,BC53,AC54,BC54,UCP54,UCP52,USP52
      COMMON /CASE_6A/ G61,H61,G62,H62,G63,H63,G64
      COMMON /CASE_6B/ AC61,BC61,CC61,DC61,AS61,BS61,CS61,DS61
      COMMON /CASE_6C/ AC62,BC62,CC62,DC62,AS62,BS62,CS62,DS62
      COMMON /CASE_6D/ AC63,BC63,CC63,DC63,AS63,BS63,CS63,DS63
      COMMON /CASE_6E/ AC64,BC64,UCP64,UCP63,USP63,UCP62,USP62
      COMMON /CASE_7A/ G71,H71,G72,H72,UCP72,USP72
      COMMON /CASE_7B/ AC71,BC71,CC71,DC71,AS71,BS71,CS71,DS71
      COMMON /CASE_7C/ AC72,BC72,CC72,DC72,AS72,BS72,CS72,DS72
      COMMON /CASE_8A/ G81,H81,G82,H82,G83,H83,UCP83,USP83,UCP82,USP82
```

```
      COMMON /CASE_8B/ AC81,BC81,CC81,DC81,AS81,BS81,CS81,DS81
      COMMON /CASE_8C/ AC82,BC82,CC82,DC82,AS82,BS82,CS82,DS82
      COMMON /CASE_8D/ AC83,BC83,CC83,DC83,AS83,BS83,CS83,DS83
      COMMON /CASE_9A/ G91,H91,G92,H92,G93,H93,G94
      COMMON /CASE_9B/ AC91,BC91,CC91,DC91,AS91,BS91,CS91,DS91
      COMMON /CASE_9C/ AC92,BC92,CC92,DC92,AS92,BS92,CS92,DS92
      COMMON /CASE_9D/ AC93,BC93,CC93,DC93,AS93,BS93,CS93,DS93
      COMMON /CASE_9E/ AC94,BC94,UCP94,UCP93,USP93,UCP92,USP92
      WRITE(6,1)
 1    FORMAT('1','****************************************************',
     +          '***************************')
      WRITE(6,36) KDAY
36    FORMAT('0','Day ',I3)
      WRITE(6,37)
37    FORMAT(1X,'-------')
      WRITE(6,*)
      WRITE(6,50) DELTEM
50    FORMAT(1X,'Temperature Drop = ',F5.1,' Degrees F')
      WRITE(6,*)
      WRITE(6,51) FPRMC(KDAY)
51    FORMAT(1X,'Compressive Strength of the Concrete = ',F6.1,' psi')
      WRITE(6,52) FPRMT(KDAY)
52    FORMAT(1X,'Tensile Strength of the Concrete = ',F6.1,' psi')
      WRITE(6,53) EC
53    FORMAT(1X,'Elastic Modulus of the Concrete = ',F9.1,' psi')
      WRITE(6,*)
      WRITE(6,54) XK1,STLSLP
      WRITE(6,55) XK2,STLLIM
      WRITE(6,56) XK3,CONSLP
      WRITE(6,57) XK4,CONLIM
54    FORMAT(1X,'K1 = ',F9.1,' pci',5X,'STLSLP = ',F6.4,' inches')
55    FORMAT(1X,'K2 = ',F9.1,' pci',5X,'STLLIM = ',F6.4,' inches')
56    FORMAT(1X,'K3 = ',F9.1,' pci',5X,'CONSLP = ',F6.4,' inches')
57    FORMAT(1X,'K4 = ',F9.1,' pci',5X,'CONLIM = ',F6.4,' inches')
      WRITE(6,*)
      WRITE(6,60) KASE
60    FORMAT(1X,'The Solution is in Case ',I2)
      WRITE(6,*)
      WRITE(6,61) 2.0*UO
61    FORMAT(1X,'The Crack Opening = ',F7.5,' inches')
      WRITE(6,62) 2.0*S
62    FORMAT(1X,'The Crack Spacing = ',F7.1,' inches')
      WRITE(6,35) UO
35    FORMAT(1X,'The Calculated Value of UO = ',F10.8)
      WRITE(6,31) VL1,VL2,VL3
31    FORMAT(1X,'VL1 = ',F6.2,' inches',3X,'VL2 = ',F6.2,' inches',
     +       3X,'VL3 = ',F6.2,' inches')
      WRITE(6,*)
      WRITE(6,70) SHRINKAGE
70    FORMAT(1X,'The Shrinkage Strain = ',F9.7,' inches/inch')
      IF (KDAY.GE.KDAYLOBEG) THEN
        WRITE(6,99) STRESS
99    FORMAT(1X,'Load-induced Stress at the Center of the Slab = ',
     +          F6.2,' psi')
```

```
        ENDIF
      WRITE(6,63) TAE
 63   FORMAT('0','The Total Energy Available  = ',F10.1,' inch-lbs')
      WRITE(6,64) FRICTION
 64   FORMAT(1X,'The Frictional Energy Used  = ',F10.1,' inch-lbs')
      WRITE(6,65) POTENTIAL
 65   FORMAT(1X,'The Potential Energy Stored  = ',F10.1,' inch-lbs')
      WRITE(6,66) STREEN
 66   FORMAT(1X,'The Stress Relief Energy Used  = ',F10.1,' inch-lbs')
      WRITE(6,*)
      WRITE(6,67) FORCE
 67   FORMAT(1X,'The Force Exerted by the Subgrade = ',F9.1,' lbs')
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,40)
      WRITE(6,41)
 40   FORMAT(1X,4X,'X',9X,'US',8X,'UC',5X,
     +'BOND STR.   FRIC. STR.  ST. STRESS  CO. STRESS')
 41   FORMAT(1X,'---------  --------  --------  ----------',
     +2X,'----------  ----------  ----------')
      IF (KASE.EQ.1) CALL OUT1(TRIP)
      IF (KASE.EQ.2) CALL OUT2(TRIP)
      IF (KASE.EQ.3) CALL OUT3(TRIP)
      IF (KASE.EQ.4) CALL OUT4(TRIP)
      IF (KASE.EQ.5) CALL OUT5(TRIP)
      IF (KASE.EQ.6) CALL OUT6(TRIP)
      IF (KASE.EQ.7) CALL OUT7(TRIP)
      IF (KASE.EQ.8) CALL OUT8(TRIP)
      IF (KASE.EQ.9) CALL OUT9(TRIP)
      WRITE(6,2)
  2   FORMAT('0','****************************************************',
     +          '***************************')
      RETURN
      END

C=====================================================================C
C                                                                     C
C  Once again, the routines are dependent upon the case that the solution C
C  is in.  When the solution is in Case 1, this routine is called to help C
C  print out the output.                                              C
C                                                                     C
C                                                                     C
      SUBROUTINE OUT1(TRIP)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_1A/ G11,H11,AS11,BS11,CS11,DS11
```

```
      COMMON /CASE_1B/ AC11,BC11,CC11,DC11
      X = UO
      DELX = S/TRIP
 10   GY1 = G11*X
      HY1 = H11*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      UST =  AS11*EXP(GY1)+
     +       BS11*EXP(-GY1)+
     +       CS11*EXP(HY1)+
     +       DS11*EXP(-HY1)
C
      UCO =   AC11*EXP(GY1)+
     +        BC11*EXP(-GY1)+
     +        CC11*EXP(HY1)+
     +        DC11*EXP(-HY1)
C
      DUST =  G11*AS11*EXP(GY1)-
     +        G11*BS11*EXP(-GY1)+
     +        H11*CS11*EXP(HY1)-
     +        H11*DS11*EXP(-HY1)
C
      DUCO =  G11*AC11*EXP(GY1)-
     +         G11*BC11*EXP(-GY1)+
     +         H11*CC11*EXP(HY1)-
     +         H11*DC11*EXP(-HY1)
C
      BONDSTR = XK1*(UCO-UST)
      FRICSTR = XK3*UCO
      POTEN = EC*(DUCDX+DUCO)
      WRITE(6,50) X,UST,UCO,BONDSTR,FRICSTR,
     +            ES*(DUST+STRN),POTEN
 50   FORMAT(1X,F9.2,2X,F8.6,2X,F8.6,2X,F10.3,2X,F10.3,2X,
     +       F10.2,2X,F10.2)
      IF (X.EQ.UO) X = 0.0
      X = X + DELX
      IF (X.LE.S) GOTO 10
      RETURN
      END

C============================================================================C
C                                                                            C
C  When the solution is in Case 2, this routine is called to help print      C
C  out the output.                                                           C
C                                                                            C
C                                                                            C
      SUBROUTINE OUT2(TRIP)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
```

```
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_2A/ G21,H21,G22,H22,UCP22,USP22
      COMMON /CASE_2B/ AC21,BC21,CC21,DC21,AS21,BS21,CS21,DS21
      COMMON /CASE_2C/ AC22,BC22,CC22,DC22,AS22,BS22,CS22,DS22
      X = UO
      DELX = S/TRIP
 10   GY1 = G21*X
      HY1 = H21*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
         IF (X.LE.VL1) THEN
          UCO = AC22*EXP(G22*X)+
     +            BC22*EXP(-G22*X)+
     +            CC22*COS(H22*X)+
     +            DC22*SIN(H22*X)+UCP22
          UST = AS22*EXP(G22*X)+
     +            BS22*EXP(-G22*X)+
     +            CS22*COS(H22*X)+
     +            DS22*SIN(H22*X)+USP22
        DUST =    G22*AS22*EXP(G22*X)-
     +            G22*BS22*EXP(-G22*X)-
     +            H22*CS22*SIN(H22*X)+
     +            H22*DS22*COS(H22*X)
        DUCO =    G22*AC22*EXP(G22*X)-
     +            G22*BC22*EXP(-G22*X)-
     +            H22*CC22*SIN(H22*X)+
     +            H22*DC22*COS(H22*X)
        BONDSTR = XC1+XK2*(UCO-UST)
        FRICSTR = XK3*UCO
         ELSE
      UST =    AS21*EXP(GY1)+
     +         BS21*EXP(-GY1)+
     +         CS21*EXP(HY1)+
     +         DS21*EXP(-HY1)
      UCO =     AC21*EXP(GY1)+
     +          BC21*EXP(-GY1)+
     +          CC21*EXP(HY1)+
     +          DC21*EXP(-HY1)
      DUST =    G21*AS21*EXP(GY1)-
     +          G21*BS21*EXP(-GY1)+
     +          H21*CS21*EXP(HY1)-
     +          H21*DS21*EXP(-HY1)
      DUCO =    G21*AC21*EXP(GY1)-
     +           G21*BC21*EXP(-GY1)+
     +           H21*CC21*EXP(HY1)-
     +           H21*DC21*EXP(-HY1)
      BONDSTR = XK1*(UCO-UST)
      FRICSTR = XK3*UCO
       ENDIF
C
      POTEN = EC*(DUCDX+DUCO)
      WRITE(6,50) X,UST,UCO,BONDSTR,FRICSTR,
```

```
     +             ES*(DUST+STRN),POTEN
 50   FORMAT(1X,F9.2,2X,F8.6,2X,F8.6,2X,F10.3,2X,F10.3,2X,
     +        F10.2,2X,F10.2)
      IF (X.EQ.UO) X = 0.0
      X = X + DELX
      IF (X.LE.S) GOTO 10
      RETURN
      END

C=====================================================================C
C                                                                     C
C  When the solution is in Case 3, this routine is called to help print  C
C  out the output.                                                    C
C                                                                     C
C                                                                     C
      SUBROUTINE OUT3(TRIP)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_3A/ G31,H31,G32,H32,G33,UCP32,USP32
      COMMON /CASE_3B/ AC31,BC31,CC31,DC31,AS31,BS31,CS31,DS31
      COMMON /CASE_3C/ AC32,BC32,CC32,DC32,AS32,BS32,CS32,DS32
      COMMON /CASE_3D/ AC33,BC33
      X = UO
      DELX = S/TRIP
 10   GY1 = G31*X
      HY1 = H31*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
        IF (X.LE.VL2) THEN
          UCO = AC33*EXP(G33*X)+
     +          BC33*EXP(-G33*X)
        DUCO =  G33*AC33*EXP(G33*X)-
     +          G33*BC33*EXP(-G33*X)
        DUST =  G32*AS32*EXP(G32*VL2)-
     +          G32*BS32*EXP(-G32*VL2)-
     +          H32*CS32*SIN(H32*VL2)+
     +          H32*DS32*COS(H32*VL2)
        UST = DUST*X
        BONDSTR = 0.0
        FRICSTR = XK3*UCO
         ELSE
         IF (X.LE.VL1) THEN
          UCO = AC32*EXP(G32*X)+
     +          BC32*EXP(-G32*X)+
     +          CC32*COS(H32*X)+
     +          DC32*SIN(H32*X)+UCP32
```

```
        UST =  AS32*EXP(G32*X)+
     +           BS32*EXP(-G32*X)+
     +           CS32*COS(H32*X)+
     +           DS32*SIN(H32*X)+USP32
       DUST =  G32*AS32*EXP(G32*X)-
     +           G32*BS32*EXP(-G32*X)-
     +           H32*CS32*SIN(H32*X)+
     +           H32*DS32*COS(H32*X)
       DUCO =  G32*AC32*EXP(G32*X)-
     +           G32*BC32*EXP(-G32*X)-
     +           H32*CC32*SIN(H32*X)+
     +           H32*DC32*COS(H32*X)
       BONDSTR = XC1+XK2*(UCO-UST)
       FRICSTR = XK3*UCO
        ELSE
      UST =  AS31*EXP(GY1)+
     +        BS31*EXP(-GY1)+
     +        CS31*EXP(HY1)+
     +        DS31*EXP(-HY1)
      UCO =   AC31*EXP(GY1)+
     +         BC31*EXP(-GY1)+
     +         CC31*EXP(HY1)+
     +         DC31*EXP(-HY1)
      DUST =  G31*AS31*EXP(GY1)-
     +         G31*BS31*EXP(-GY1)+
     +         H31*CS31*EXP(HY1)-
     +         H31*DS31*EXP(-HY1)
      DUCO =  G31*AC31*EXP(GY1)-
     +          G31*BC31*EXP(-GY1)+
     +          H31*CC31*EXP(HY1)-
     +          H31*DC31*EXP(-HY1)
      BONDSTR = XK1*(UCO-UST)
      FRICSTR = XK3*UCO
       ENDIF
       ENDIF
C
      POTEN = EC*(DUCDX+DUCO)
      WRITE(6,50) X,UST,UCO,BONDSTR,FRICSTR,
     +             ES*(DUST+STRN),POTEN
 50   FORMAT(1X,F9.2,2X,F8.6,2X,F8.6,2X,F10.3,2X,F10.3,2X,
     +        F10.2,2X,F10.2)
      IF (X.EQ.UO) X = 0.0
      X = X + DELX
      IF (X.LE.S) GOTO 10
      RETURN
      END

C==================================================================C
C                                                                  C
C  When the solution is in Case 4, this routine is called to help print  C
C  out the output.                                                 C
C                                                                  C
C                                                                  C
      SUBROUTINE OUT4(TRIP)
```

```
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_4A/ G41,H41,G42,H42,G43,H43,UCP43,USP43,UCP42,USP42
      COMMON /CASE_4B/ AC41,BC41,CC41,DC41,AS41,BS41,CS41,DS41
      COMMON /CASE_4C/ AC42,BC42,CC42,DC42,AS42,BS42,CS42,DS42
      COMMON /CASE_4D/ AC43,BC43,CC43,DC43,AS43,BS43,CS43,DS43
      X = UO
      DELX = S/TRIP
 10   GY1 = G41*X
      HY1 = H41*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
        IF (X.LE.VL3) THEN
          UCO = AC43*COS(G43*X)+
     +          BC43*SIN(G43*X)+
     +          CC43*COS(H43*X)+
     +          DC43*SIN(H43*X)+UCP43
          UST = AS43*COS(G43*X)+
     +          BS43*SIN(G43*X)+
     +          CS43*COS(H43*X)+
     +          DS43*SIN(H43*X)+USP43
        DUST = -G43*AS43*SIN(G43*X)+
     +          G43*BS43*COS(G43*X)-
     +          H43*CS43*SIN(H43*X)+
     +          H43*DS43*COS(H43*X)
        DUCO = -G43*AC43*SIN(G43*X)+
     +          G43*BC43*COS(G43*X)-
     +          H43*CC43*SIN(H43*X)+
     +          H43*DC43*COS(H43*X)
        BONDSTR = XC1+XK2*(UCO-UST)
        FRICSTR = XC2+XK4*UCO
         ELSE
         IF (X.LE.VL1) THEN
          UCO = AC42*EXP(G42*X)+
     +          BC42*EXP(-G42*X)+
     +          CC42*COS(H42*X)+
     +          DC42*SIN(H42*X)+UCP42
          UST = AS42*EXP(G42*X)+
     +          BS42*EXP(-G42*X)+
     +          CS42*COS(H42*X)+
     +          DS42*SIN(H42*X)+USP42
        DUST =  G42*AS42*EXP(G42*X)-
     +          G42*BS42*EXP(-G42*X)-
     +          H42*CS42*SIN(H42*X)+
     +          H42*DS42*COS(H42*X)
        DUCO =  G42*AC42*EXP(G42*X)-
```

```
     +          G42*BC42*EXP(-G42*X)-
     +          H42*CC42*SIN(H42*X)+
     +          H42*DC42*COS(H42*X)
        BONDSTR = XC1+XK2*(UCO-UST)
        FRICSTR = XK3*UCO
         ELSE
      UST =  AS41*EXP(GY1)+
     +       BS41*EXP(-GY1)+
     +       CS41*EXP(HY1)+
     +       DS41*EXP(-HY1)
      UCO =   AC41*EXP(GY1)+
     +        BC41*EXP(-GY1)+
     +        CC41*EXP(HY1)+
     +        DC41*EXP(-HY1)
      DUST =  G41*AS41*EXP(GY1)-
     +        G41*BS41*EXP(-GY1)+
     +        H41*CS41*EXP(HY1)-
     +        H41*DS41*EXP(-HY1)
      DUCO =  G41*AC41*EXP(GY1)-
     +         G41*BC41*EXP(-GY1)+
     +         H41*CC41*EXP(HY1)-
     +         H41*DC41*EXP(-HY1)
      BONDSTR = XK1*(UCO-UST)
      FRICSTR = XK3*UCO
       ENDIF
       ENDIF
C
      POTEN = EC*(DUCDX+DUCO)
      WRITE(6,50) X,UST,UCO,BONDSTR,FRICSTR,
     +            ES*(DUST+STRN),POTEN
 50   FORMAT(1X,F9.2,2X,F8.6,2X,F8.6,2X,F10.3,2X,F10.3,2X,
     +       F10.2,2X,F10.2)
      IF (X.EQ.UO) X = 0.0
      X = X + DELX
      IF (X.LE.S) GOTO 10
      RETURN
      END

C=========================================================================C
C                                                                         C
C  When the solution is in Case 5, this routine is called to help print   C
C  out the output.                                                        C
C                                                                         C
C                                                                         C
      SUBROUTINE OUT5(TRIP)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
```

```
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_5A/ G51,H51,G52,H52,G53,G54
      COMMON /CASE_5B/ AC51,BC51,CC51,DC51,AS51,BS51,CS51,DS51
      COMMON /CASE_5C/ AC52,BC52,CC52,DC52,AS52,BS52,CS52,DS52
      COMMON /CASE_5D/ AC53,BC53,AC54,BC54,UCP54,UCP52,USP52
      X = U0
      DELX = S/TRIP
10    GY1 = G51*X
      HY1 = H51*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL3) THEN
        UCO = AC54*COS(G54*X)+
     +        BC54*SIN(G54*X)+UCP54
        DUCO = -G54*AC54*SIN(G54*X)+
     +          G54*BC54*COS(G54*X)
        DUST =  G52*AS52*EXP(G52*VL2)-
     +          G52*BS52*EXP(-G52*VL2)-
     +          H52*CS52*SIN(H52*VL2)+
     +          H52*DS52*COS(H52*VL2)
        UST = DUST*X
        BONDSTR = 0.0
        FRICSTR = XC2+XK4*UCO
        ELSE
        IF (X.LE.VL2) THEN
          UCO = AC53*EXP(G53*X)+
     +          BC53*EXP(-G53*X)
        DUCO =  G53*AC53*EXP(G53*X)-
     +          G53*BC53*EXP(-G53*X)
        DUST =  G52*AS52*EXP(G52*VL2)-
     +          G52*BS52*EXP(-G52*VL2)-
     +          H52*CS52*SIN(H52*VL2)+
     +          H52*DS52*COS(H52*VL2)
        UST = DUST*X
        BONDSTR = 0.0
        FRICSTR = XK3*UCO
         ELSE
         IF (X.LE.VL1) THEN
          UCO = AC52*EXP(G52*X)+
     +          BC52*EXP(-G52*X)+
     +          CC52*COS(H52*X)+
     +          DC52*SIN(H52*X)+UCP52
          UST = AS52*EXP(G52*X)+
     +          BS52*EXP(-G52*X)+
     +          CS52*COS(H52*X)+
     +          DS52*SIN(H52*X)+USP52
        DUST =  G52*AS52*EXP(G52*X)-
     +          G52*BS52*EXP(-G52*X)-
     +          H52*CS52*SIN(H52*X)+
     +          H52*DS52*COS(H52*X)
        DUCO =  G52*AC52*EXP(G52*X)-
     +          G52*BC52*EXP(-G52*X)-
     +          H52*CC52*SIN(H52*X)+
     +          H52*DC52*COS(H52*X)
```

```
          BONDSTR = XC1+XK2*(UCO-UST)
          FRICSTR = XK3*UCO
           ELSE
      UST =  AS51*EXP(GY1)+
     +       BS51*EXP(-GY1)+
     +       CS51*EXP(HY1)+
     +       DS51*EXP(-HY1)
      UCO =   AC51*EXP(GY1)+
     +        BC51*EXP(-GY1)+
     +        CC51*EXP(HY1)+
     +        DC51*EXP(-HY1)
      DUST =  G51*AS51*EXP(GY1)-
     +        G51*BS51*EXP(-GY1)+
     +        H51*CS51*EXP(HY1)-
     +        H51*DS51*EXP(-HY1)
      DUCO =  G51*AC51*EXP(GY1)-
     +         G51*BC51*EXP(-GY1)+
     +         H51*CC51*EXP(HY1)-
     +         H51*DC51*EXP(-HY1)
      BONDSTR = XK1*(UCO-UST)
      FRICSTR = XK3*UCO
       ENDIF
       ENDIF
       ENDIF
C
      POTEN = EC*(DUCDX+DUCO)
      WRITE(6,50)  X,UST,UCO,BONDSTR,FRICSTR,
     +             ES*(DUST+STRN),POTEN
 50   FORMAT(1X,F9.2,2X,F8.6,2X,F8.6,2X,F10.3,2X,F10.3,2X,
     +       F10.2,2X,F10.2)
      IF (X.EQ.UO) X = 0.0
      X = X + DELX
      IF (X.LE.S) GOTO 10
      RETURN
      END

C=========================================================================C
C                                                                         C
C  When the solution is in Case 6, this routine is called to help print   C
C  out the output.                                                        C
C                                                                         C
C                                                                         C
      SUBROUTINE OUT6(TRIP)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_6A/ G61,H61,G62,H62,G63,H63,G64
```

```
      COMMON /CASE_6B/ AC61,BC61,CC61,DC61,AS61,BS61,CS61,DS61
      COMMON /CASE_6C/ AC62,BC62,CC62,DC62,AS62,BS62,CS62,DS62
      COMMON /CASE_6D/ AC63,BC63,CC63,DC63,AS63,BS63,CS63,DS63
      COMMON /CASE_6E/ AC64,BC64,UCP64,UCP63,USP63,UCP62,USP62
      X = UO
      DELX =S/TRIP
10    GY1 = G61*X
      HY1 = H61*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL2) THEN
        UCO = AC64*COS(G64*X)+
     +        BC64*SIN(G64*X)+UCP64
        DUCO = -G64*AC64*SIN(G64*X)+
     +          G64*BC64*COS(G64*X)
        DUST = -G63*AS63*SIN(G63*VL2)+
     +          G63*BS63*COS(G63*VL2)-
     +          H63*CS63*SIN(H63*VL2)+
     +          H63*DS63*COS(H63*VL2)
        UST = DUST*X
        BONDSTR = 0.0
        FRICSTR = XC2+XK4*UCO
        ELSE
        IF (X.LE.VL3) THEN
          UCO = AC63*COS(G63*X)+
     +          BC63*SIN(G63*X)+
     +          CC63*COS(H63*X)+
     +          DC63*SIN(H63*X)+UCP63
          UST = AS63*COS(G63*X)+
     +          BS63*SIN(G63*X)+
     +          CS63*COS(H63*X)+
     +          DS63*SIN(H63*X)+USP63
        DUST = -G63*AS63*SIN(G63*X)+
     +          G63*BS63*COS(G63*X)-
     +          H63*CS63*SIN(H63*X)+
     +          H63*DS63*COS(H63*X)
        DUCO = -G63*AC63*SIN(G63*X)+
     +          G63*BC63*COS(G63*X)-
     +          H63*CC63*SIN(H63*X)+
     +          H63*DC63*COS(H63*X)
        BONDSTR = XC1+XK2*(UCO-UST)
        FRICSTR = XC2+XK4*UCO
         ELSE
         IF (X.LE.VL1) THEN
          UCO = AC62*EXP(G62*X)+
     +          BC62*EXP(-G62*X)+
     +          CC62*COS(H62*X)+
     +          DC62*SIN(H62*X)+UCP62
          UST = AS62*EXP(G62*X)+
     +          BS62*EXP(-G62*X)+
     +          CS62*COS(H62*X)+
     +          DS62*SIN(H62*X)+USP62
        DUST =  G62*AS62*EXP(G62*X)-
     +          G62*BS62*EXP(-G62*X)-
```

```
     +          H62*CS62*SIN(H62*X)+
     +          H62*DS62*COS(H62*X)
        DUCO =  G62*AC62*EXP(G62*X)-
     +          G62*BC62*EXP(-G62*X)-
     +          H62*CC62*SIN(H62*X)+
     +          H62*DC62*COS(H62*X)
        BONDSTR = XC1+XK2*(UCO-UST)
        FRICSTR = XK3*UCO
         ELSE
      UST =  AS61*EXP(GY1)+
     +       BS61*EXP(-GY1)+
     +       CS61*EXP(HY1)+
     +       DS61*EXP(-HY1)
      UCO =   AC61*EXP(GY1)+
     +        BC61*EXP(-GY1)+
     +        CC61*EXP(HY1)+
     +        DC61*EXP(-HY1)
      DUST =  G61*AS61*EXP(GY1)-
     +        G61*BS61*EXP(-GY1)+
     +        H61*CS61*EXP(HY1)-
     +        H61*DS61*EXP(-HY1)
      DUCO =  G61*AC61*EXP(GY1)-
     +         G61*BC61*EXP(-GY1)+
     +         H61*CC61*EXP(HY1)-
     +         H61*DC61*EXP(-HY1)
      BONDSTR = XK1*(UCO-UST)
      FRICSTR = XK3*UCO
       ENDIF
       ENDIF
       ENDIF
C
      POTEN = EC*(DUCDX+DUCO)
      WRITE(6,50) X,UST,UCO,BONDSTR,FRICSTR,
     +            ES*(DUST+STRN),POTEN
 50   FORMAT(1X,F9.2,2X,F8.6,2X,F8.6,2X,F10.3,2X,F10.3,2X,
     +       F10.2,2X,F10.2)
      IF (X.EQ.UO) X = 0.0
      X = X + DELX
      IF (X.LE.S) GOTO 10
      RETURN
      END

C======================================================================C
C                                                                      C
C  When the solution is in Case 7, this routine is called to help print C
C  out the output.                                                     C
C                                                                      C
C                                                                      C
      SUBROUTINE OUT7(TRIP)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
```

```
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_7A/ G71,H71,G72,H72,UCP72,USP72
      COMMON /CASE_7B/ AC71,BC71,CC71,DC71,AS71,BS71,CS71,DS71
      COMMON /CASE_7C/ AC72,BC72,CC72,DC72,AS72,BS72,CS72,DS72
      X = UO
      DELX = S/TRIP
10    GY1 = G71*X
      HY1 = H71*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
         IF (X.LE.VL3) THEN
          UCO = AC72*EXP(G72*X)+
     +            BC72*EXP(-G72*X)+
     +            CC72*COS(H72*X)+
     +            DC72*SIN(H72*X)+UCP72
          UST = AS72*EXP(G72*X)+
     +            BS72*EXP(-G72*X)+
     +            CS72*COS(H72*X)+
     +            DS72*SIN(H72*X)+USP72
        DUST =    G72*AS72*EXP(G72*X)-
     +            G72*BS72*EXP(-G72*X)-
     +            H72*CS72*SIN(H72*X)+
     +            H72*DS72*COS(H72*X)
        DUCO =    G72*AC72*EXP(G72*X)-
     +            G72*BC72*EXP(-G72*X)-
     +            H72*CC72*SIN(H72*X)+
     +            H72*DC72*COS(H72*X)
        BONDSTR = XK1*(UCO-UST)
        FRICSTR = XC2+XK4*UCO
         ELSE
      UST =    AS71*EXP(GY1)+
     +         BS71*EXP(-GY1)+
     +         CS71*EXP(HY1)+
     +         DS71*EXP(-HY1)
      UCO =     AC71*EXP(GY1)+
     +          BC71*EXP(-GY1)+
     +          CC71*EXP(HY1)+
     +          DC71*EXP(-HY1)
      DUST =    G71*AS71*EXP(GY1)-
     +          G71*BS71*EXP(-GY1)+
     +          H71*CS71*EXP(HY1)-
     +          H71*DS71*EXP(-HY1)
      DUCO =    G71*AC71*EXP(GY1)-
     +           G71*BC71*EXP(-GY1)+
     +           H71*CC71*EXP(HY1)-
     +           H71*DC71*EXP(-HY1)
      BONDSTR = XK1*(UCO-UST)
      FRICSTR = XK3*UCO
       ENDIF
C
```

```
      POTEN = EC*(DUCDX+DUCO)
      WRITE(6,50) X,UST,UCO,BONDSTR,FRICSTR,
     +            ES*(DUST+STRN),POTEN
 50   FORMAT(1X,F9.2,2X,F8.6,2X,F8.6,2X,F10.3,2X,F10.3,2X,
     +       F10.2,2X,F10.2)
      IF (X.EQ.UO) X = 0.0
      X = X + DELX
      IF (X.LE.S) GOTO 10
      RETURN
      END

C=========================================================================C
C                                                                         C
C  When the solution is in Case 8, this routine is called to help print   C
C  out the output.                                                        C
C                                                                         C
C                                                                         C
      SUBROUTINE OUT8(TRIP)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_8A/ G81,H81,G82,H82,G83,H83,UCP83,USP83,UCP82,USP82
      COMMON /CASE_8B/ AC81,BC81,CC81,DC81,AS81,BS81,CS81,DS81
      COMMON /CASE_8C/ AC82,BC82,CC82,DC82,AS82,BS82,CS82,DS82
      COMMON /CASE_8D/ AC83,BC83,CC83,DC83,AS83,BS83,CS83,DS83
      X = UO
      DELX = S/TRIP
 10   GY1 = G81*X
      HY1 = H81*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
        IF (X.LE.VL1) THEN
          UCO = AC83*COS(G83*X)+
     +          BC83*SIN(G83*X)+
     +          CC83*COS(H83*X)+
     +          DC83*SIN(H83*X)+UCP83
          UST = AS83*COS(G83*X)+
     +          BS83*SIN(G83*X)+
     +          CS83*COS(H83*X)+
     +          DS83*SIN(H83*X)+USP83
        DUST = -G83*AS83*SIN(G83*X)+
     +          G83*BS83*COS(G83*X)-
     +          H83*CS83*SIN(H83*X)+
     +          H83*DS83*COS(H83*X)
        DUCO = -G83*AC83*SIN(G83*X)+
     +          G83*BC83*COS(G83*X)-
     +          H83*CC83*SIN(H83*X)+
```

```
     +          H83*DC83*COS(H83*X)
        BONDSTR = XC1+XK2*(UCO-UST)
        FRICSTR = XC2+XK4*UCO
         ELSE
         IF (X.LE.VL3) THEN
          UCO = AC82*EXP(G82*X)+
     +          BC82*EXP(-G82*X)+
     +          CC82*COS(H82*X)+
     +          DC82*SIN(H82*X)+UCP82
          UST = AS82*EXP(G82*X)+
     +          BS82*EXP(-G82*X)+
     +          CS82*COS(H82*X)+
     +          DS82*SIN(H82*X)+USP82
        DUST =  G82*AS82*EXP(G82*X)-
     +          G82*BS82*EXP(-G82*X)-
     +          H82*CS82*SIN(H82*X)+
     +          H82*DS82*COS(H82*X)
        DUCO =  G82*AC82*EXP(G82*X)-
     +          G82*BC82*EXP(-G82*X)-
     +          H82*CC82*SIN(H82*X)+
     +          H82*DC82*COS(H82*X)
        BONDSTR = XK1*(UCO-UST)
        FRICSTR = XC2+XK4*UCO
         ELSE
      UST =  AS81*EXP(GY1)+
     +       BS81*EXP(-GY1)+
     +       CS81*EXP(HY1)+
     +       DS81*EXP(-HY1)
      UCO =   AC81*EXP(GY1)+
     +        BC81*EXP(-GY1)+
     +        CC81*EXP(HY1)+
     +        DC81*EXP(-HY1)
      DUST =  G81*AS81*EXP(GY1)-
     +        G81*BS81*EXP(-GY1)+
     +        H81*CS81*EXP(HY1)-
     +        H81*DS81*EXP(-HY1)
      DUCO =  G81*AC81*EXP(GY1)-
     +         G81*BC81*EXP(-GY1)+
     +         H81*CC81*EXP(HY1)-
     +         H81*DC81*EXP(-HY1)
      BONDSTR = XK1*(UCO-UST)
      FRICSTR = XK3*UCO
       ENDIF
       ENDIF
C
      POTEN = EC*(DUCDX+DUCO)
      WRITE(6,50) X,UST,UCO,BONDSTR,FRICSTR,
     +             ES*(DUST+STRN),POTEN
 50   FORMAT(1X,F9.2,2X,F8.6,2X,F8.6,2X,F10.3,2X,F10.3,2X,
     +       F10.2,2X,F10.2)
      IF (X.EQ.UO) X = 0.0
      X = X + DELX
      IF (X.LE.S) GOTO 10
      RETURN
```

```
      END
C======================================================================C
C                                                                      C
C  When the solution is in Case 9, this routine is called to help print  C
C  out the output.                                                     C
C                                                                      C
C                                                                      C
      SUBROUTINE OUT9(TRIP)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_9A/ G91,H91,G92,H92,G93,H93,G94
      COMMON /CASE_9B/ AC91,BC91,CC91,DC91,AS91,BS91,CS91,DS91
      COMMON /CASE_9C/ AC92,BC92,CC92,DC92,AS92,BS92,CS92,DS92
      COMMON /CASE_9D/ AC93,BC93,CC93,DC93,AS93,BS93,CS93,DS93
      COMMON /CASE_9E/ AC94,BC94,UCP94,UCP93,USP93,UCP92,USP92
      X = UO
      DELX =S/TRIP
 10   GY1 = G91*X
      HY1 = H91*X
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
      IF (X.LE.VL2) THEN
        UCO = AC94*COS(G94*X)+
     +        BC94*SIN(G94*X)+UCP94
        DUCO = -G94*AC94*SIN(G94*X)+
     +          G94*BC94*COS(G94*X)
        DUST = -G93*AS93*SIN(G93*VL2)+
     +          G93*BS93*COS(G93*VL2)-
     +          H93*CS93*SIN(H93*VL2)+
     +          H93*DS93*COS(H93*VL2)
        UST = DUST*X
        BONDSTR = 0.0
        FRICSTR = XC2+XK4*UCO
        ELSE
        IF (X.LE.VL1) THEN
          UCO = AC93*COS(G93*X)+
     +          BC93*SIN(G93*X)+
     +          CC93*COS(H93*X)+
     +          DC93*SIN(H93*X)+UCP93
          UST = AS93*COS(G93*X)+
     +          BS93*SIN(G93*X)+
     +          CS93*COS(H93*X)+
     +          DS93*SIN(H93*X)+USP93
        DUST = -G93*AS93*SIN(G93*X)+
     +          G93*BS93*COS(G93*X)-
```

```
     +          H93*CS93*SIN(H93*X)+
     +          H93*DS93*COS(H93*X)
        DUCO = -G93*AC93*SIN(G93*X)+
     +          G93*BC93*COS(G93*X)-
     +          H93*CC93*SIN(H93*X)+
     +          H93*DC93*COS(H93*X)
        BONDSTR = XC1+XK2*(UCO-UST)
        FRICSTR = XC2+XK4*UCO
         ELSE
         IF (X.LE.VL3) THEN
          UCO = AC92*EXP(G92*X)+
     +          BC92*EXP(-G92*X)+
     +          CC92*COS(H92*X)+
     +          DC92*SIN(H92*X)+UCP92
          UST = AS92*EXP(G92*X)+
     +          BS92*EXP(-G92*X)+
     +          CS92*COS(H92*X)+
     +          DS92*SIN(H92*X)+USP92
        DUST =  G92*AS92*EXP(G92*X)-
     +          G92*BS92*EXP(-G92*X)-
     +          H92*CS92*SIN(H92*X)+
     +          H92*DS92*COS(H92*X)
        DUCO =  G92*AC92*EXP(G92*X)-
     +          G92*BC92*EXP(-G92*X)-
     +          H92*CC92*SIN(H92*X)+
     +          H92*DC92*COS(H92*X)
        BONDSTR = XK1*(UCO-UST)
        FRICSTR = XC2+XK4*UCO
         ELSE
      UST =  AS91*EXP(GY1)+
     +       BS91*EXP(-GY1)+
     +       CS91*EXP(HY1)+
     +       DS91*EXP(-HY1)
      UCO =   AC91*EXP(GY1)+
     +        BC91*EXP(-GY1)+
     +        CC91*EXP(HY1)+
     +        DC91*EXP(-HY1)
      DUST =  G91*AS91*EXP(GY1)-
     +        G91*BS91*EXP(-GY1)+
     +        H91*CS91*EXP(HY1)-
     +        H91*DS91*EXP(-HY1)
      DUCO =  G91*AC91*EXP(GY1)-
     +         G91*BC91*EXP(-GY1)+
     +         H91*CC91*EXP(HY1)-
     +         H91*DC91*EXP(-HY1)
      BONDSTR = XK1*(UCO-UST)
      FRICSTR = XK3*UCO
       ENDIF
       ENDIF
       ENDIF
C
      POTEN = EC*(DUCDX+DUCO)
      WRITE(6,50) X,UST,UCO,BONDSTR,FRICSTR,
     +             ES*(DUST+STRN),POTEN
```

```
 50   FORMAT(1X,F9.2,2X,F8.6,2X,F8.6,2X,F10.3,2X,F10.3,2X,
     +        F10.2,2X,F10.2)
      IF (X.EQ.UO) X = 0.0
      X = X + DELX
      IF (X.LE.S) GOTO 10
      RETURN
      END

C=========================================================================C
C                                                                         C
C  The VAL subroutines calculate the coefficients that are needed to      C
C  obtain the exact solution to the differential equation system that     C
C  each case has.  First, the necessary supporting values are calculated, C
C  then the "A" matrix is set up, then inverted, and then multiplied by   C
C  the "D" matrix to get the coefficients for the concrete displacement.  C
C  The coefficients of the steel displacement are obtained from the       C
C  concrete's coefficients.                                               C
C                                                                         C
C  VAL1 solves the Case 1 Equation System.                                C
C                                                                         C
C                                                                         C
      SUBROUTINE VAL1(UO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_1A/ G11,H11,AS11,BS11,CS11,DS11
      COMMON /CASE_1B/ AC11,BC11,CC11,DC11
      DIMENSION A(20,20),D(20),E(20)
C
C  CALCULATION OF ALL OF THE IMPORTANT STUFF!
C
       NR = 4
       D(1) = UO
       D(2) = UO*STSTRAIN
       D(3) = 0.0
       D(4) = 0.0
C
       A1 = (XK1*PI*STLDIA)/(ES*ASTL)
       C1 = (XK1*PI*STLDIA+XK3*WIDT)/(EC*ACON)
       D1 = (XK1*PI*STLDIA)/(EC*ACON)
C
       G11 = (1.0-4.0*(A1*C1-A1*D1)/(A1+C1)**2.0)**0.5
       H11 = ((A1+C1)/2.0-(A1+C1)*G11/2.0)**0.5
       G11 = ((A1+C1)/2.0+(A1+C1)*G11/2.0)**0.5
C
       R1 = A1/(A1-G11**2.0)
       S1 = A1/(A1-H11**2.0)
```

```
C
      GY1 = G11*S
      HY1 = H11*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
C INITIALIZING THE "A" MATRIX FOR CASE 1
C
       A(1,1) = EXP(G11*UO)
       A(1,2) = EXP(-G11*UO)
       A(1,3) = EXP(H11*UO)
       A(1,4) = EXP(-H11*UO)
C
       A(2,1) = R1*EXP(G11*UO)
       A(2,2) = R1*EXP(-G11*UO)
       A(2,3) = S1*EXP(H11*UO)
       A(2,4) = S1*EXP(-H11*UO)
C
        A(3,1) = EXP(GY1)
        A(3,2) = EXP(-GY1)
        A(3,3) = EXP(HY1)
        A(3,4) = EXP(-HY1)
C
        A(4,1) = R1*EXP(GY1)
        A(4,2) = R1*EXP(-GY1)
        A(4,3) = S1*EXP(HY1)
        A(4,4) = S1*EXP(-HY1)
C
       CALL INVERT(A,NR)
       CALL MULTI(A,D,E,NR)
C
       AC11 = E(1)
       BC11 = E(2)
       CC11 = E(3)
       DC11 = E(4)
        AS11 = R1*AC11
        BS11 = R1*BC11
        CS11 = S1*CC11
        DS11 = S1*DC11
C
      RETURN
      END

C=========================================================================C
C                                                                         C
C  VAL2 solves the Case 2 Equation System.                                C
C                                                                         C
C                                                                         C
      SUBROUTINE VAL2(UO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
```

```
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_2A/ G21,H21,G22,H22,UCP22,USP22
      COMMON /CASE_2B/ AC21,BC21,CC21,DC21,AS21,BS21,CS21,DS21
      COMMON /CASE_2C/ AC22,BC22,CC22,DC22,AS22,BS22,CS22,DS22
      DIMENSION A(20,20),D(20),E(20)
C
      NR = 8
C
      A1 = (XK1*PI*STLDIA)/(ES*ASTL)
      C1 = (XK1*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D1 = (XK1*PI*STLDIA)/(EC*ACON)
C
      A2 = (XK2*PI*STLDIA)/(ES*ASTL)
      B2 = (-XC1*PI*STLDIA)/(ES*ASTL)
      C2 = (XK2*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D2 = (XK2*PI*STLDIA)/(EC*ACON)
      E2 = (XC1*PI*STLDIA)/(EC*ACON)
C
      G21 = (1.0-4.0*(A1*C1-A1*D1)/(A1+C1)**2.0)**0.5
      H21 = ((A1+C1)/2.0 - (A1+C1)*G21/2.0)**0.5
      G21 = ((A1+C1)/2.0 + (A1+C1)*G21/2.0)**0.5
C
      G22 = (1.0-4.0*(A2*C2-A2*D2)/(A2+C2)**2.0)**0.5
      H22 = ((A2+C2)/-2.0 - (A2+C2)*G22/2.0)**0.5
      G22 = ((A2+C2)/2.0 - (A2+C2)*G22/2.0)**0.5
C
      R1 = A1/(A1-G21**2.0)
      S1 = A1/(A1-H21**2.0)
C
      R2 = A2/(A2-G22**2.0)
      S2 = A2/(A2+H22**2.0)
C
      UCP22 = (B2*D2+A2*E2)/(A2*(D2-C2))
      USP22 = (A2*E2+B2*C2)/(A2*(D2-C2))
C
      GY1 = G21*S
      HY1 = H21*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      D(1) = 0.0
      D(2) = 0.0
      D(3) = UCP22
      D(4) = USP22
      D(5) = 0.0
      D(6) = 0.0
      D(7) = UO-UCP22
      D(8) = UO*STSTRAIN-USP22
C
      A(1,1) = EXP(GY1)
```

```
      A(1,2)  = EXP(-GY1)
      A(1,3)  = EXP(HY1)
      A(1,4)  = EXP(-HY1)
      A(1,5)  = 0.0
      A(1,6)  = 0.0
      A(1,7)  = 0.0
      A(1,8)  = 0.0
C
      A(2,1)  = R1*EXP(GY1)
      A(2,2)  = R1*EXP(-GY1)
      A(2,3)  = S1*EXP(HY1)
      A(2,4)  = S1*EXP(-HY1)
      A(2,5)  = 0.0
      A(2,6)  = 0.0
      A(2,7)  = 0.0
      A(2,8)  = 0.0
C
      A(3,1)  = EXP(G21*VL1)
      A(3,2)  = EXP(-G21*VL1)
      A(3,3)  = EXP(H21*VL1)
      A(3,4)  = EXP(-H21*VL1)
      A(3,5)  = -EXP(G22*VL1)
      A(3,6)  = -EXP(-G22*VL1)
      A(3,7)  = -COS(H22*VL1)
      A(3,8)  = -SIN(H22*VL1)
C
      A(4,1)  = R1*EXP(G21*VL1)
      A(4,2)  = R1*EXP(-G21*VL1)
      A(4,3)  = S1*EXP(H21*VL1)
      A(4,4)  = S1*EXP(-H21*VL1)
      A(4,5)  = -R2*EXP(G22*VL1)
      A(4,6)  = -R2*EXP(-G22*VL1)
      A(4,7)  = -S2*COS(H22*VL1)
      A(4,8)  = -S2*SIN(H22*VL1)
C
      A(5,1)  = G21*EXP(G21*VL1)
      A(5,2)  = -G21*EXP(-G21*VL1)
      A(5,3)  = H21*EXP(H21*VL1)
      A(5,4)  = -H21*EXP(-H21*VL1)
      A(5,5)  = -G22*EXP(G22*VL1)
      A(5,6)  = G22*EXP(-G22*VL1)
      A(5,7)  = H22*SIN(H22*VL1)
      A(5,8)  = -H22*COS(H22*VL1)
C
      A(6,1)  = R1*G21*EXP(G21*VL1)
      A(6,2)  = -R1*G21*EXP(-G21*VL1)
      A(6,3)  = S1*H21*EXP(H21*VL1)
      A(6,4)  = -S1*H21*EXP(-H21*VL1)
      A(6,5)  = -R2*G22*EXP(G22*VL1)
      A(6,6)  = R2*G22*EXP(-G22*VL1)
      A(6,7)  = S2*H22*SIN(H22*VL1)
      A(6,8)  = -S2*H22*COS(H22*VL1)
C
      A(7,1)  = 0.0
```

```
      A(7,2) = 0.0
      A(7,3) = 0.0
      A(7,4) = 0.0
      A(7,5) = EXP(G22*UO)
      A(7,6) = EXP(-G22*UO)
      A(7,7) = COS(H22*UO)
      A(7,8) = SIN(H22*UO)
C
      A(8,1) = 0.0
      A(8,2) = 0.0
      A(8,3) = 0.0
      A(8,4) = 0.0
      A(8,5) = R2*EXP(G22*UO)
      A(8,6) = R2*EXP(-G22*UO)
      A(8,7) = S2*COS(H22*UO)
      A(8,8) = S2*SIN(H22*UO)
C
      CALL INVERT(A,NR)
      CALL MULTI(A,D,E,NR)
C
      AC21 = E(1)
      BC21 = E(2)
      CC21 = E(3)
      DC21 = E(4)
      AC22 = E(5)
      BC22 = E(6)
      CC22 = E(7)
      DC22 = E(8)
C
      AS21 = R1*AC21
      BS21 = R1*BC21
      CS21 = S1*CC21
      DS21 = S1*DC21
      AS22 = R2*AC22
      BS22 = R2*BC22
      CS22 = S2*CC22
      DS22 = S2*DC22
C
      RETURN
      END

C=====================================================================C
C                                                                     C
C  VAL3 solves the Case 3 Equation System.                            C
C                                                                     C
C                                                                     C
      SUBROUTINE VAL3(UO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
```

```
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_3A/ G31,H31,G32,H32,G33,UCP32,USP32
      COMMON /CASE_3B/ AC31,BC31,CC31,DC31,AS31,BS31,CS31,DS31
      COMMON /CASE_3C/ AC32,BC32,CC32,DC32,AS32,BS32,CS32,DS32
      COMMON /CASE_3D/ AC33,BC33
      DIMENSION A(20,20),D(20),E(20)
C
      NR = 10
C
      A1 = (XK1*PI*STLDIA)/(ES*ASTL)
      C1 = (XK1*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D1 = (XK1*PI*STLDIA)/(EC*ACON)
C
      A2 = (XK2*PI*STLDIA)/(ES*ASTL)
      B2 = (-XC1*PI*STLDIA)/(ES*ASTL)
      C2 = (XK2*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D2 = (XK2*PI*STLDIA)/(EC*ACON)
      E2 = (XC1*PI*STLDIA)/(EC*ACON)
C
      A3 = (XK3*WIDT)/(EC*ACON)
C
      G31 = (1.0-4.0*(A1*C1-A1*D1)/(A1+C1)**2.0)**0.5
      H31 = ((A1+C1)/2.0 - (A1+C1)*G31/2.0)**0.5
      G31 = ((A1+C1)/2.0 + (A1+C1)*G31/2.0)**0.5
C
      G32 = (1.0-4.0*(A2*C2-A2*D2)/(A2+C2)**2.0)**0.5
      H32 = ((A2+C2)/-2.0 - (A2+C2)*G32/2.0)**0.5
      G32 = ((A2+C2)/2.0 - (A2+C2)*G32/2.0)**0.5
C
      G33 = A3**0.5
C
      R1 = A1/(A1-G31**2.0)
      S1 = A1/(A1-H31**2.0)
C
      R2 = A2/(A2-G32**2.0)
      S2 = A2/(A2+H32**2.0)
C
      UCP32 = (B2*D2+A2*E2)/(A2*(D2-C2))
      USP32 = (A2*E2+B2*C2)/(A2*(D2-C2))
C
      GY1 = G31*S
      HY1 = H31*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      D(1) = 0.0
      D(2) = 0.0
      D(3) = UCP32
      D(4) = USP32
      D(5) = 0.0
      D(6) = 0.0
      D(7) = VL2*STSTRAIN-USP32
```

```
      D(8) = -UCP32
      D(9) = 0.0
      D(10) = U0
C
      A(1,1) = EXP(GY1)
      A(1,2) = EXP(-GY1)
      A(1,3) = EXP(HY1)
      A(1,4) = EXP(-HY1)
      A(1,5) = 0.0
      A(1,6) = 0.0
      A(1,7) = 0.0
      A(1,8) = 0.0
      A(1,9) = 0.0
      A(1,10) = 0.0
C
      A(2,1) = R1*EXP(GY1)
      A(2,2) = R1*EXP(-GY1)
      A(2,3) = S1*EXP(HY1)
      A(2,4) = S1*EXP(-HY1)
      A(2,5) = 0.0
      A(2,6) = 0.0
      A(2,7) = 0.0
      A(2,8) = 0.0
      A(2,9) = 0.0
      A(2,10) = 0.0
C
      A(3,1) = EXP(G31*VL1)
      A(3,2) = EXP(-G31*VL1)
      A(3,3) = EXP(H31*VL1)
      A(3,4) = EXP(-H31*VL1)
      A(3,5) = -EXP(G32*VL1)
      A(3,6) = -EXP(-G32*VL1)
      A(3,7) = -COS(H32*VL1)
      A(3,8) = -SIN(H32*VL1)
      A(3,9) = 0.0
      A(3,10) = 0.0
C
      A(4,1) = R1*EXP(G31*VL1)
      A(4,2) = R1*EXP(-G31*VL1)
      A(4,3) = S1*EXP(H31*VL1)
      A(4,4) = S1*EXP(-H31*VL1)
      A(4,5) = -R2*EXP(G32*VL1)
      A(4,6) = -R2*EXP(-G32*VL1)
      A(4,7) = -S2*COS(H32*VL1)
      A(4,8) = -S2*SIN(H32*VL1)
      A(4,9) = 0.0
      A(4,10) = 0.0
C
      A(5,1) = G31*EXP(G31*VL1)
      A(5,2) = -G31*EXP(-G31*VL1)
      A(5,3) = H31*EXP(H31*VL1)
      A(5,4) = -H31*EXP(-H31*VL1)
      A(5,5) = -G32*EXP(G32*VL1)
      A(5,6) = G32*EXP(-G32*VL1)
```

```
      A(5,7)  = H32*SIN(H32*VL1)
      A(5,8)  = -H32*COS(H32*VL1)
      A(5,9)  = 0.0
      A(5,10)  = 0.0
C
      A(6,1)  = R1*G31*EXP(G31*VL1)
      A(6,2)  = -R1*G31*EXP(-G31*VL1)
      A(6,3)  = S1*H31*EXP(H31*VL1)
      A(6,4)  = -S1*H31*EXP(-H31*VL1)
      A(6,5)  = -R2*G32*EXP(G32*VL1)
      A(6,6)  = R2*G32*EXP(-G32*VL1)
      A(6,7)  = S2*H32*SIN(H32*VL1)
      A(6,8)  = -S2*H32*COS(H32*VL1)
      A(6,9)  = 0.0
      A(6,10)  = 0.0
C
      A(7,1)  = 0.0
      A(7,2)  = 0.0
      A(7,3)  = 0.0
      A(7,4)  = 0.0
      A(7,5)  = R2*EXP(G32*VL2)
      A(7,6)  = R2*EXP(-G32*VL2)
      A(7,7)  = S2*COS(H32*VL2)
      A(7,8)  = S2*SIN(H32*VL2)
      A(7,9)  = 0.0
      A(7,10)  = 0.0
C
      A(8,1)  = 0.0
      A(8,2)  = 0.0
      A(8,3)  = 0.0
      A(8,4)  = 0.0
      A(8,5)  = EXP(G32*VL2)
      A(8,6)  = EXP(-G32*VL2)
      A(8,7)  = COS(H32*VL2)
      A(8,8)  = SIN(H32*VL2)
      A(8,9)  = -EXP(G33*VL2)
      A(8,10)  = -EXP(-G33*VL2)
C
      A(9,1)  = 0.0
      A(9,2)  = 0.0
      A(9,3)  = 0.0
      A(9,4)  = 0.0
      A(9,5)  = G32*EXP(G32*VL2)
      A(9,6)  = -G32*EXP(-G32*VL2)
      A(9,7)  = -H32*SIN(H32*VL2)
      A(9,8)  = H32*COS(H32*VL2)
      A(9,9)  = -G33*EXP(G33*VL2)
      A(9,10)  = G33*EXP(-G33*VL2)
C
      A(10,1)  = 0.0
      A(10,2)  = 0.0
      A(10,3)  = 0.0
      A(10,4)  = 0.0
      A(10,5)  = 0.0
```

```
      A(10,6) = 0.0
      A(10,7) = 0.0
      A(10,8) = 0.0
      A(10,9) = EXP(G33*UO)
      A(10,10) = EXP(-G33*UO)
C
      CALL INVERT(A,NR)
      CALL MULTI(A,D,E,NR)
C
      AC31 = E(1)
      BC31 = E(2)
      CC31 = E(3)
      DC31 = E(4)
      AC32 = E(5)
      BC32 = E(6)
      CC32 = E(7)
      DC32 = E(8)
      AC33 = E(9)
      BC33 = E(10)
C
      AS31 = R1*AC31
      BS31 = R1*BC31
      CS31 = S1*CC31
      DS31 = S1*DC31
      AS32 = R2*AC32
      BS32 = R2*BC32
      CS32 = S2*CC32
      DS32 = S2*DC32
C
      RETURN
      END
C=========================================================================C
C                                                                         C
C  VAL4 solves the Case 4 Equation System.                                C
C                                                                         C
C                                                                         C
      SUBROUTINE VAL4(UO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_4A/ G41,H41,G42,H42,G43,H43,UCP43,USP43,UCP42,USP42
      COMMON /CASE_4B/ AC41,BC41,CC41,DC41,AS41,BS41,CS41,DS41
      COMMON /CASE_4C/ AC42,BC42,CC42,DC42,AS42,BS42,CS42,DS42
      COMMON /CASE_4D/ AC43,BC43,CC43,DC43,AS43,BS43,CS43,DS43
      DIMENSION A(20,20),D(20),E(20)
C
```

```
      NR = 12
C
      A1 = (XK1*PI*STLDIA)/(ES*ASTL)
      C1 = (XK1*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D1 = (XK1*PI*STLDIA)/(EC*ACON)
C
      A2 = (XK2*PI*STLDIA)/(ES*ASTL)
      B2 = (-XC1*PI*STLDIA)/(ES*ASTL)
      C2 = (XK2*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D2 = (XK2*PI*STLDIA)/(EC*ACON)
      E2 = (XC1*PI*STLDIA)/(EC*ACON)
C
      A3 = (XK2*PI*STLDIA)/(ES*ASTL)
      B3 = (-XC1*PI*STLDIA)/(ES*ASTL)
      C3 = (XK2*PI*STLDIA + XK4*WIDT)/(EC*ACON)
      D3 = (XK2*PI*STLDIA)/(EC*ACON)
      E3 = (XC1*PI*STLDIA + XC2*WIDT)/(EC*ACON)
C
      G41 = (1.0-4.0*(A1*C1-A1*D1)/(A1+C1)**2.0)**0.5
      H41 = ((A1+C1)/2.0 - (A1+C1)*G41/2.0)**0.5
      G41 = ((A1+C1)/2.0 + (A1+C1)*G41/2.0)**0.5
C
      G42 = (1.0-4.0*(A2*C2-A2*D2)/(A2+C2)**2.0)**0.5
      H42 = ((A2+C2)/-2.0 - (A2+C2)*G42/2.0)**0.5
      G42 = ((A2+C2)/2.0 - (A2+C2)*G42/2.0)**0.5
C
      G43 = (1.0-4.0*(A3*C3-A3*D3)/(A3+C3)**2.0)**0.5
      H43 = ((A3+C3)/-2.0 - (A3+C3)*G43/2.0)**0.5
      G43 = ((A3+C3)/-2.0 + (A3+C3)*G43/2.0)**0.5
C
      R1 = A1/(A1-G41**2.0)
      S1 = A1/(A1-H41**2.0)
C
      R2 = A2/(A2-G42**2.0)
      S2 = A2/(A2+H42**2.0)
C
      R3 = A3/(A3+G43**2.0)
      S3 = A3/(A3+H43**2.0)
C
      UCP42 = (B2*D2+A2*E2)/(A2*(D2-C2))
      USP42 = (A2*E2+B2*C2)/(A2*(D2-C2))
C
      UCP43 = (B3*D3+A3*E3)/(A3*(D3-C3))
      USP43 = (A3*E3+B3*C3)/(A3*(D3-C3))
C
      GY1 = G41*S
      HY1 = H41*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      D(1) = 0.0
      D(2) = 0.0
      D(3) = UCP42
      D(4) = USP42
```

```
      D(5) = 0.0
      D(6) = 0.0
      D(7) = UCP43-UCP42
      D(8) = USP43-USP42
      D(9) = 0.0
      D(10) = 0.0
      D(11) = UO-UCP43
      D(12) = UO*STSTRAIN-USP43
C
      A(1,1) = EXP(GY1)
      A(1,2) = EXP(-GY1)
      A(1,3) = EXP(HY1)
      A(1,4) = EXP(-HY1)
      A(1,5) = 0.0
      A(1,6) = 0.0
      A(1,7) = 0.0
      A(1,8) = 0.0
      A(1,9) = 0.0
      A(1,10) = 0.0
      A(1,11) = 0.0
      A(1,12) = 0.0
C
      A(2,1) = R1*EXP(GY1)
      A(2,2) = R1*EXP(-GY1)
      A(2,3) = S1*EXP(HY1)
      A(2,4) = S1*EXP(-HY1)
      A(2,5) = 0.0
      A(2,6) = 0.0
      A(2,7) = 0.0
      A(2,8) = 0.0
      A(2,9) = 0.0
      A(2,10) = 0.0
      A(2,11) = 0.0
      A(2,12) = 0.0
C
      A(3,1) = EXP(G41*VL1)
      A(3,2) = EXP(-G41*VL1)
      A(3,3) = EXP(H41*VL1)
      A(3,4) = EXP(-H41*VL1)
      A(3,5) = -EXP(G42*VL1)
      A(3,6) = -EXP(-G42*VL1)
      A(3,7) = -COS(H42*VL1)
      A(3,8) = -SIN(H42*VL1)
      A(3,9) = 0.0
      A(3,10) = 0.0
      A(3,11) = 0.0
      A(3,12) = 0.0
C
      A(4,1) = R1*EXP(G41*VL1)
      A(4,2) = R1*EXP(-G41*VL1)
      A(4,3) = S1*EXP(H41*VL1)
      A(4,4) = S1*EXP(-H41*VL1)
      A(4,5) = -R2*EXP(G42*VL1)
      A(4,6) = -R2*EXP(-G42*VL1)
```

```
      A(4,7)  =  -S2*COS(H42*VL1)
      A(4,8)  =  -S2*SIN(H42*VL1)
      A(4,9)  =  0.0
      A(4,10)  =  0.0
      A(4,11)  =  0.0
      A(4,12)  =  0.0
C
      A(5,1)  =  G41*EXP(G41*VL1)
      A(5,2)  =  -G41*EXP(-G41*VL1)
      A(5,3)  =  H41*EXP(H41*VL1)
      A(5,4)  =  -H41*EXP(-H41*VL1)
      A(5,5)  =  -G42*EXP(G42*VL1)
      A(5,6)  =  G42*EXP(-G42*VL1)
      A(5,7)  =  H42*SIN(H42*VL1)
      A(5,8)  =  -H42*COS(H42*VL1)
      A(5,9)  =  0.0
      A(5,10)  =  0.0
      A(5,11)  =  0.0
      A(5,12)  =  0.0
C
      A(6,1)  =  R1*G41*EXP(G41*VL1)
      A(6,2)  =  -R1*G41*EXP(-G41*VL1)
      A(6,3)  =  S1*H41*EXP(H41*VL1)
      A(6,4)  =  -S1*H41*EXP(-H41*VL1)
      A(6,5)  =  -R2*G42*EXP(G42*VL1)
      A(6,6)  =  R2*G42*EXP(-G42*VL1)
      A(6,7)  =  S2*H42*SIN(H42*VL1)
      A(6,8)  =  -S2*H42*COS(H42*VL1)
      A(6,9)  =  0.0
      A(6,10)  =  0.0
      A(6,11)  =  0.0
      A(6,12)  =  0.0
C
      A(7,1)  =  0.0
      A(7,2)  =  0.0
      A(7,3)  =  0.0
      A(7,4)  =  0.0
      A(7,5)  =  EXP(G42*VL3)
      A(7,6)  =  EXP(-G42*VL3)
      A(7,7)  =  COS(H42*VL3)
      A(7,8)  =  SIN(H42*VL3)
      A(7,9)  =  -COS(G43*VL3)
      A(7,10)  =  -SIN(G43*VL3)
      A(7,11)  =  -COS(H43*VL3)
      A(7,12)  =  -SIN(H43*VL3)
C
      A(8,1)  =  0.0
      A(8,2)  =  0.0
      A(8,3)  =  0.0
      A(8,4)  =  0.0
      A(8,5)  =  R2*EXP(G42*VL3)
      A(8,6)  =  R2*EXP(-G42*VL3)
      A(8,7)  =  S2*COS(H42*VL3)
      A(8,8)  =  S2*SIN(H42*VL3)
```

```
      A(8,9) = -R3*COS(G43*VL3)
      A(8,10) = -R3*SIN(G43*VL3)
      A(8,11) = -S3*COS(H43*VL3)
      A(8,12) = -S3*SIN(H43*VL3)
C
      A(9,1) = 0.0
      A(9,2) = 0.0
      A(9,3) = 0.0
      A(9,4) = 0.0
      A(9,5) = G42*EXP(G42*VL3)
      A(9,6) = -G42*EXP(-G42*VL3)
      A(9,7) = -H42*SIN(H42*VL3)
      A(9,8) = H42*COS(H42*VL3)
      A(9,9) = G43*SIN(G43*VL3)
      A(9,10) = -G43*COS(G43*VL3)
      A(9,11) = H43*SIN(H43*VL3)
      A(9,12) = -H43*COS(H43*VL3)
C
      A(10,1) = 0.0
      A(10,2) = 0.0
      A(10,3) = 0.0
      A(10,4) = 0.0
      A(10,5) = R2*G42*EXP(G42*VL3)
      A(10,6) = -R2*G42*EXP(-G42*VL3)
      A(10,7) = -S2*H42*SIN(H42*VL3)
      A(10,8) = S2*H42*COS(H42*VL3)
      A(10,9) = R3*G43*SIN(G43*VL3)
      A(10,10) = -R3*G43*COS(G43*VL3)
      A(10,11) = S3*H43*SIN(H43*VL3)
      A(10,12) = -S3*H43*COS(H43*VL3)
C
      A(11,1) = 0.0
      A(11,2) = 0.0
      A(11,3) = 0.0
      A(11,4) = 0.0
      A(11,5) = 0.0
      A(11,6) = 0.0
      A(11,7) = 0.0
      A(11,8) = 0.0
      A(11,9) = COS(G43*UO)
      A(11,10) = SIN(G43*UO)
      A(11,11) = COS(H43*UO)
      A(11,12) = SIN(G43*UO)
C
      A(12,1) = 0.0
      A(12,2) = 0.0
      A(12,3) = 0.0
      A(12,4) = 0.0
      A(12,5) = 0.0
      A(12,6) = 0.0
      A(12,7) = 0.0
      A(12,8) = 0.0
      A(12,9) = R3*COS(G43*UO)
      A(12,10) = R3*SIN(G43*UO)
```

```
      A(12,11) = S3*COS(H43*UO)
      A(12,12) = S3*SIN(H43*UO)
C
      CALL INVERT(A,NR)
      CALL MULTI(A,D,E,NR)
C
      AC41 = E(1)
      BC41 = E(2)
      CC41 = E(3)
      DC41 = E(4)
      AC42 = E(5)
      BC42 = E(6)
      CC42 = E(7)
      DC42 = E(8)
      AC43 = E(9)
      BC43 = E(10)
      CC43 = E(11)
      DC43 = E(12)
C
      AS41 = R1*AC41
      BS41 = R1*BC41
      CS41 = S1*CC41
      DS41 = S1*DC41
      AS42 = R2*AC42
      BS42 = R2*BC42
      CS42 = S2*CC42
      DS42 = S2*DC42
      AS43 = R3*AC43
      BS43 = R3*BC43
      CS43 = S3*CC43
      DS43 = S3*DC43
C
      RETURN
      END
C=======================================================================C
C                                                                       C
C  VAL5 solves the Case 5 Equation System.                              C
C                                                                       C
C                                                                       C
      SUBROUTINE VAL5(UO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_5A/ G51,H51,G52,H52,G53,G54
      COMMON /CASE_5B/ AC51,BC51,CC51,DC51,AS51,BS51,CS51,DS51
      COMMON /CASE_5C/ AC52,BC52,CC52,DC52,AS52,BS52,CS52,DS52
```

```
      COMMON /CASE_5D/ AC53,BC53,AC54,BC54,UCP54,UCP52,USP52
      DIMENSION A(20,20),D(20),E(20)
C
      NR = 12
C
      A1 = (XK1*PI*STLDIA)/(ES*ASTL)
      C1 = (XK1*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D1 = (XK1*PI*STLDIA)/(EC*ACON)
C
      A2 = (XK2*PI*STLDIA)/(ES*ASTL)
      B2 = (-XC1*PI*STLDIA)/(ES*ASTL)
      C2 = (XK2*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D2 = (XK2*PI*STLDIA)/(EC*ACON)
      E2 = (XC1*PI*STLDIA)/(EC*ACON)
C
      A3 = (XK3*WIDT)/(EC*ACON)
C
      A4 = (-XK4*WIDT)/(EC*ACON)
      B4 = (XC2*WIDT)/(EC*ACON)
C
      G51 = (1.0-4.0*(A1*C1-A1*D1)/(A1+C1)**2.0)**0.5
      H51 = ((A1+C1)/2.0 - (A1+C1)*G51/2.0)**0.5
      G51 = ((A1+C1)/2.0 + (A1+C1)*G51/2.0)**0.5
C
      G52 = (1.0-4.0*(A2*C2-A2*D2)/(A2+C2)**2.0)**0.5
      H52 = ((A2+C2)/-2.0 - (A2+C2)*G52/2.0)**0.5
      G52 = ((A2+C2)/2.0 - (A2+C2)*G52/2.0)**0.5
C
      G53 = A3**0.5
C
      G54 = A4**0.5
C
      R1 = A1/(A1-G51**2.0)
      S1 = A1/(A1-H51**2.0)
C
      R2 = A2/(A2-G52**2.0)
      S2 = A2/(A2+H52**2.0)
C
      UCP52 = (B2*D2+A2*E2)/(A2*(D2-C2))
      USP52 = (A2*E2+B2*C2)/(A2*(D2-C2))
C
      UCP54 = B4/A4
C
      GY1 = G51*S
      HY1 = H51*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      D(1) = 0.0
      D(2) = 0.0
      D(3) = UCP52
      D(4) = USP52
      D(5) = 0.0
      D(6) = 0.0
```

```
      D(7) = VL2*STSTRAIN-USP52
      D(8) = -UCP52
      D(9) = 0.0
      D(10) = UCP54
      D(11) = 0.0
      D(12) = U0-UCP54
C
      A(1,1) = EXP(GY1)
      A(1,2) = EXP(-GY1)
      A(1,3) = EXP(HY1)
      A(1,4) = EXP(-HY1)
      A(1,5) = 0.0
      A(1,6) = 0.0
      A(1,7) = 0.0
      A(1,8) = 0.0
      A(1,9) = 0.0
      A(1,10) = 0.0
      A(1,11) = 0.0
      A(1,12) = 0.0
C
      A(2,1) = R1*EXP(GY1)
      A(2,2) = R1*EXP(-GY1)
      A(2,3) = S1*EXP(HY1)
      A(2,4) = S1*EXP(-HY1)
      A(2,5) = 0.0
      A(2,6) = 0.0
      A(2,7) = 0.0
      A(2,8) = 0.0
      A(2,9) = 0.0
      A(2,10) = 0.0
      A(2,11) = 0.0
      A(2,12) = 0.0
C
      A(3,1) = EXP(G51*VL1)
      A(3,2) = EXP(-G51*VL1)
      A(3,3) = EXP(H51*VL1)
      A(3,4) = EXP(-H51*VL1)
      A(3,5) = -EXP(G52*VL1)
      A(3,6) = -EXP(-G52*VL1)
      A(3,7) = -COS(H52*VL1)
      A(3,8) = -SIN(H52*VL1)
      A(3,9) = 0.0
      A(3,10) = 0.0
      A(3,11) = 0.0
      A(3,12) = 0.0
C
      A(4,1) = R1*EXP(G51*VL1)
      A(4,2) = R1*EXP(-G51*VL1)
      A(4,3) = S1*EXP(H51*VL1)
      A(4,4) = S1*EXP(-H51*VL1)
      A(4,5) = -R2*EXP(G52*VL1)
      A(4,6) = -R2*EXP(-G52*VL1)
      A(4,7) = -S2*COS(H52*VL1)
      A(4,8) = -S2*SIN(H52*VL1)
```

```
      A(4,9) = 0.0
      A(4,10) = 0.0
      A(4,11) = 0.0
      A(4,12) = 0.0
C
      A(5,1) = G51*EXP(G51*VL1)
      A(5,2) = -G51*EXP(-G51*VL1)
      A(5,3) = H51*EXP(H51*VL1)
      A(5,4) = -H51*EXP(-H51*VL1)
      A(5,5) = -G52*EXP(G52*VL1)
      A(5,6) = G52*EXP(-G52*VL1)
      A(5,7) = H52*SIN(H52*VL1)
      A(5,8) = -H52*COS(H52*VL1)
      A(5,9) = 0.0
      A(5,10) = 0.0
      A(5,11) = 0.0
      A(5,12) = 0.0
C
      A(6,1) = R1*G51*EXP(G51*VL1)
      A(6,2) = -R1*G51*EXP(-G51*VL1)
      A(6,3) = S1*H51*EXP(H51*VL1)
      A(6,4) = -S1*H51*EXP(-H51*VL1)
      A(6,5) = -R2*G52*EXP(G52*VL1)
      A(6,6) = R2*G52*EXP(-G52*VL1)
      A(6,7) = S2*H52*SIN(H52*VL1)
      A(6,8) = -S2*H52*COS(H52*VL1)
      A(6,9) = 0.0
      A(6,10) = 0.0
      A(6,11) = 0.0
      A(6,12) = 0.0
C
      A(7,1) = 0.0
      A(7,2) = 0.0
      A(7,3) = 0.0
      A(7,4) = 0.0
      A(7,5) = R2*EXP(G52*VL2)
      A(7,6) = R2*EXP(-G52*VL2)
      A(7,7) = S2*COS(H52*VL2)
      A(7,8) = S2*SIN(H52*VL2)
      A(7,9) = 0.0
      A(7,10) = 0.0
      A(7,11) = 0.0
      A(7,12) = 0.0
C
      A(8,1) = 0.0
      A(8,2) = 0.0
      A(8,3) = 0.0
      A(8,4) = 0.0
      A(8,5) = EXP(G52*VL2)
      A(8,6) = EXP(-G52*VL2)
      A(8,7) = COS(H52*VL2)
      A(8,8) = SIN(H52*VL2)
      A(8,9) = -EXP(G53*VL2)
      A(8,10) = -EXP(-G53*VL2)
```

```
      A(8,11) = 0.0
      A(8,12) = 0.0
C
      A(9,1) = 0.0
      A(9,2) = 0.0
      A(9,3) = 0.0
      A(9,4) = 0.0
      A(9,5) = G52*EXP(G52*VL2)
      A(9,6) = -G52*EXP(-G52*VL2)
      A(9,7) = -H52*SIN(H52*VL2)
      A(9,8) = H52*COS(H52*VL2)
      A(9,9) = -G53*EXP(G53*VL2)
      A(9,10) = G53*EXP(-G53*VL2)
      A(9,11) = 0.0
      A(9,12) = 0.0
C
      A(10,1) = 0.0
      A(10,2) = 0.0
      A(10,3) = 0.0
      A(10,4) = 0.0
      A(10,5) = 0.0
      A(10,6) = 0.0
      A(10,7) = 0.0
      A(10,8) = 0.0
      A(10,9) = EXP(G53*VL3)
      A(10,10) = EXP(-G53*VL3)
      A(10,11) = -COS(G54*VL3)
      A(10,12) = -SIN(G54*VL3)
C
      A(11,1) = 0.0
      A(11,2) = 0.0
      A(11,3) = 0.0
      A(11,4) = 0.0
      A(11,5) = 0.0
      A(11,6) = 0.0
      A(11,7) = 0.0
      A(11,8) = 0.0
      A(11,9) = G53*EXP(G53*VL3)
      A(11,10) = -G53*EXP(-G53*VL3)
      A(11,11) = G54*SIN(G54*VL3)
      A(11,12) = -G54*COS(G54*VL3)
C
      A(12,1) = 0.0
      A(12,2) = 0.0
      A(12,3) = 0.0
      A(12,4) = 0.0
      A(12,5) = 0.0
      A(12,6) = 0.0
      A(12,7) = 0.0
      A(12,8) = 0.0
      A(12,9) = 0.0
      A(12,10) = 0.0
      A(12,11) = COS(G54*UO)
      A(12,12) = SIN(G54*UO)
```

```
C
      CALL INVERT(A,NR)
      CALL MULTI(A,D,E,NR)
C
      AC51 = E(1)
      BC51 = E(2)
      CC51 = E(3)
      DC51 = E(4)
      AC52 = E(5)
      BC52 = E(6)
      CC52 = E(7)
      DC52 = E(8)
      AC53 = E(9)
      BC53 = E(10)
      AC54 = E(11)
      BC54 = E(12)
C
      AS51 = R1*AC51
      BS51 = R1*BC51
      CS51 = S1*CC51
      DS51 = S1*DC51
      AS52 = R2*AC52
      BS52 = R2*BC52
      CS52 = S2*CC52
      DS52 = S2*DC52
C
      RETURN
      END

C=====================================================================C
C                                                                     C
C  VAL6 solves the Case 6 Equation System.                            C
C                                                                     C
C                                                                     C
      SUBROUTINE VAL6(UO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_6A/ G61,H61,G62,H62,G63,H63,G64
      COMMON /CASE_6B/ AC61,BC61,CC61,DC61,AS61,BS61,CS61,DS61
      COMMON /CASE_6C/ AC62,BC62,CC62,DC62,AS62,BS62,CS62,DS62
      COMMON /CASE_6D/ AC63,BC63,CC63,DC63,AS63,BS63,CS63,DS63
      COMMON /CASE_6E/ AC64,BC64,UCP64,UCP63,USP63,UCP62,USP62
      DIMENSION A(20,20),D(20),E(20)
C
      NR = 14
C
```

```
      A1 = (XK1*PI*STLDIA)/(ES*ASTL)
      C1 = (XK1*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D1 = (XK1*PI*STLDIA)/(EC*ACON)
C
      A2 = (XK2*PI*STLDIA)/(ES*ASTL)
      B2 = (-XC1*PI*STLDIA)/(ES*ASTL)
      C2 = (XK2*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D2 = (XK2*PI*STLDIA)/(EC*ACON)
      E2 = (XC1*PI*STLDIA)/(EC*ACON)
C
      A3 = (XK2*PI*STLDIA)/(ES*ASTL)
      B3 = (-XC1*PI*STLDIA)/(ES*ASTL)
      C3 = (XK2*PI*STLDIA + XK4*WIDT)/(EC*ACON)
      D3 = (XK2*PI*STLDIA)/(EC*ACON)
      E3 = (XC1*PI*STLDIA + XC2*WIDT)/(EC*ACON)
C
      A4 = (-XK4*WIDT)/(EC*ACON)
      B4 = (XC2*WIDT)/(EC*ACON)
C
      G61 = (1.0-4.0*(A1*C1-A1*D1)/(A1+C1)**2.0)**0.5
      H61 = ((A1+C1)/2.0 - (A1+C1)*G61/2.0)**0.5
      G61 = ((A1+C1)/2.0 + (A1+C1)*G61/2.0)**0.5
C
      G62 = (1.0-4.0*(A2*C2-A2*D2)/(A2+C2)**2.0)**0.5
      H62 = ((A2+C2)/-2.0 - (A2+C2)*G62/2.0)**0.5
      G62 = ((A2+C2)/2.0 - (A2+C2)*G62/2.0)**0.5
C
      G63 = (1.0-4.0*(A3*C3-A3*D3)/(A3+C3)**2.0)**0.5
      H63 = ((A3+C3)/-2.0 - (A3+C3)*G63/2.0)**0.5
      G63 = ((A3+C3)/-2.0 + (A3+C3)*G63/2.0)**0.5
C
      G64 = A4**0.5
C
      R1 = A1/(A1-G61**2.0)
      S1 = A1/(A1-H61**2.0)
C
      R2 = A2/(A2-G62**2.0)
      S2 = A2/(A2+H62**2.0)
C
      R3 = A3/(A3+G63**2.0)
      S3 = A3/(A3+H63**2.0)
C
      UCP62 = (B2*D2+A2*E2)/(A2*(D2-C2))
      USP62 = (A2*E2+B2*C2)/(A2*(D2-C2))
C
      UCP63 = (B3*D3+A3*E3)/(A3*(D3-C3))
      USP63 = (A3*E3+B3*C3)/(A3*(D3-C3))
C
      UCP64 = B4/A4
C
      GY1 = G61*S
      HY1 = H61*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
```

```
C
      D(1) = 0.0
      D(2) = 0.0
      D(3) = UCP62
      D(4) = USP62
      D(5) = 0.0
      D(6) = 0.0
      D(7) = UCP63-UCP62
      D(8) = USP63-USP62
      D(9) = 0.0
      D(10) = 0.0
      D(11) = UCP64-UCP63
      D(12) = 0.0
      D(13) = VL2*STSTRAIN-USP63
      D(14) = UO-UCP64
C
      A(1,1) = EXP(GY1)
      A(1,2) = EXP(-GY1)
      A(1,3) = EXP(HY1)
      A(1,4) = EXP(-HY1)
      A(1,5) = 0.0
      A(1,6) = 0.0
      A(1,7) = 0.0
      A(1,8) = 0.0
      A(1,9) = 0.0
      A(1,10) = 0.0
      A(1,11) = 0.0
      A(1,12) = 0.0
      A(1,13) = 0.0
      A(1,14) = 0.0
C
      A(2,1) = R1*EXP(GY1)
      A(2,2) = R1*EXP(-GY1)
      A(2,3) = S1*EXP(HY1)
      A(2,4) = S1*EXP(-HY1)
      A(2,5) = 0.0
      A(2,6) = 0.0
      A(2,7) = 0.0
      A(2,8) = 0.0
      A(2,9) = 0.0
      A(2,10) = 0.0
      A(2,11) = 0.0
      A(2,12) = 0.0
      A(2,13) = 0.0
      A(2,14) = 0.0
C
      A(3,1) = EXP(G61*VL1)
      A(3,2) = EXP(-G61*VL1)
      A(3,3) = EXP(H61*VL1)
      A(3,4) = EXP(-H61*VL1)
      A(3,5) = -EXP(G62*VL1)
      A(3,6) = -EXP(-G62*VL1)
      A(3,7) = -COS(H62*VL1)
      A(3,8) = -SIN(H62*VL1)
```

```
      A(3,9)  = 0.0
      A(3,10)  = 0.0
      A(3,11)  = 0.0
      A(3,12)  = 0.0
      A(3,13)  = 0.0
      A(3,14)  = 0.0
C
      A(4,1)  = R1*EXP(G61*VL1)
      A(4,2)  = R1*EXP(-G61*VL1)
      A(4,3)  = S1*EXP(H61*VL1)
      A(4,4)  = S1*EXP(-H61*VL1)
      A(4,5)  = -R2*EXP(G62*VL1)
      A(4,6)  = -R2*EXP(-G62*VL1)
      A(4,7)  = -S2*COS(H62*VL1)
      A(4,8)  = -S2*SIN(H62*VL1)
      A(4,9)  = 0.0
      A(4,10)  = 0.0
      A(4,11)  = 0.0
      A(4,12)  = 0.0
      A(4,13)  = 0.0
      A(4,14)  = 0.0
C
      A(5,1)  = G61*EXP(G61*VL1)
      A(5,2)  = -G61*EXP(-G61*VL1)
      A(5,3)  = H61*EXP(H61*VL1)
      A(5,4)  = -H61*EXP(-H61*VL1)
      A(5,5)  = -G62*EXP(G62*VL1)
      A(5,6)  = G62*EXP(-G62*VL1)
      A(5,7)  = H62*SIN(H62*VL1)
      A(5,8)  = -H62*COS(H62*VL1)
      A(5,9)  = 0.0
      A(5,10)  = 0.0
      A(5,11)  = 0.0
      A(5,12)  = 0.0
      A(5,13)  = 0.0
      A(5,14)  = 0.0
C
      A(6,1)  = R1*G61*EXP(G61*VL1)
      A(6,2)  = -R1*G61*EXP(-G61*VL1)
      A(6,3)  = S1*H61*EXP(H61*VL1)
      A(6,4)  = -S1*H61*EXP(-H61*VL1)
      A(6,5)  = -R2*G62*EXP(G62*VL1)
      A(6,6)  = R2*G62*EXP(-G62*VL1)
      A(6,7)  = S2*H62*SIN(H62*VL1)
      A(6,8)  = -S2*H62*COS(H62*VL1)
      A(6,9)  = 0.0
      A(6,10)  = 0.0
      A(6,11)  = 0.0
      A(6,12)  = 0.0
      A(6,13)  = 0.0
      A(6,14)  = 0.0
C
      A(7,1)  = 0.0
      A(7,2)  = 0.0
```

```
      A(7,3)  = 0.0
      A(7,4)  = 0.0
      A(7,5)  = EXP(G62*VL3)
      A(7,6)  = EXP(-G62*VL3)
      A(7,7)  = COS(H62*VL3)
      A(7,8)  = SIN(H62*VL3)
      A(7,9)  = -COS(G63*VL3)
      A(7,10)  = -SIN(G63*VL3)
      A(7,11)  = -COS(H63*VL3)
      A(7,12)  = -SIN(H63*VL3)
      A(7,13)  = 0.0
      A(7,14)  = 0.0
C
      A(8,1)  = 0.0
      A(8,2)  = 0.0
      A(8,3)  = 0.0
      A(8,4)  = 0.0
      A(8,5)  = R2*EXP(G62*VL3)
      A(8,6)  = R2*EXP(-G62*VL3)
      A(8,7)  = S2*COS(H62*VL3)
      A(8,8)  = S2*SIN(H62*VL3)
      A(8,9)  = -R3*COS(G63*VL3)
      A(8,10)  = -R3*SIN(G63*VL3)
      A(8,11)  = -S3*COS(H63*VL3)
      A(8,12)  = -S3*SIN(H63*VL3)
      A(8,13)  = 0.0
      A(8,14)  = 0.0
C
      A(9,1)  = 0.0
      A(9,2)  = 0.0
      A(9,3)  = 0.0
      A(9,4)  = 0.0
      A(9,5)  = G62*EXP(G62*VL3)
      A(9,6)  = -G62*EXP(-G62*VL3)
      A(9,7)  = -H62*SIN(H62*VL3)
      A(9,8)  = H62*COS(H62*VL3)
      A(9,9)  = G63*SIN(G63*VL3)
      A(9,10)  = -G63*COS(G63*VL3)
      A(9,11)  = H63*SIN(H63*VL3)
      A(9,12)  = -H63*COS(H63*VL3)
      A(9,13)  = 0.0
      A(9,14)  = 0.0
C
      A(10,1)  = 0.0
      A(10,2)  = 0.0
      A(10,3)  = 0.0
      A(10,4)  = 0.0
      A(10,5)  = R2*G62*EXP(G62*VL3)
      A(10,6)  = -R2*G62*EXP(-G62*VL3)
      A(10,7)  = -S2*H62*SIN(H62*VL3)
      A(10,8)  = S2*H62*COS(H62*VL3)
      A(10,9)  = R3*G63*SIN(G63*VL3)
      A(10,10)  = -R3*G63*COS(G63*VL3)
      A(10,11)  = S3*H63*SIN(H63*VL3)
```

```
      A(10,12) = -S3*H63*COS(H63*VL3)
      A(10,13) = 0.0
      A(10,14) = 0.0
C
      A(11,1) = 0.0
      A(11,2) = 0.0
      A(11,3) = 0.0
      A(11,4) = 0.0
      A(11,5) = 0.0
      A(11,6) = 0.0
      A(11,7) = 0.0
      A(11,8) = 0.0
      A(11,9) = COS(G63*VL2)
      A(11,10) = SIN(G63*VL2)
      A(11,11) = COS(H63*VL2)
      A(11,12) = SIN(G63*VL2)
      A(11,13) = -COS(G64*VL2)
      A(11,14) = -SIN(G64*VL2)
C
      A(12,1) = 0.0
      A(12,2) = 0.0
      A(12,3) = 0.0
      A(12,4) = 0.0
      A(12,5) = 0.0
      A(12,6) = 0.0
      A(12,7) = 0.0
      A(12,8) = 0.0
      A(12,9) = -G63*SIN(G63*VL2)
      A(12,10) = G63*COS(G63*VL2)
      A(12,11) = -H63*SIN(H63*VL2)
      A(12,12) = H63*COS(H63*VL2)
      A(12,13) = G64*SIN(G64*VL2)
      A(12,14) = -G64*COS(G64*VL2)
C
      A(13,1) = 0.0
      A(13,2) = 0.0
      A(13,3) = 0.0
      A(13,4) = 0.0
      A(13,5) = 0.0
      A(13,6) = 0.0
      A(13,7) = 0.0
      A(13,8) = 0.0
      A(13,9) = R3*COS(G63*VL2)
      A(13,10) = R3*SIN(G63*VL2)
      A(13,11) = S3*COS(H63*VL2)
      A(13,12) = S3*SIN(H63*VL2)
      A(13,13) = 0.0
      A(13,14) = 0.0
C
      A(14,1) = 0.0
      A(14,2) = 0.0
      A(14,3) = 0.0
      A(14,4) = 0.0
      A(14,5) = 0.0
```

```
      A(14,6) = 0.0
      A(14,7) = 0.0
      A(14,8) = 0.0
      A(14,9) = 0.0
      A(14,10) = 0.0
      A(14,11) = 0.0
      A(14,12) = 0.0
      A(14,13) = COS(G64*UO)
      A(14,14) = SIN(G64*UO)
C
      CALL INVERT(A,NR)
      CALL MULTI(A,D,E,NR)
C
      AC61 = E(1)
      BC61 = E(2)
      CC61 = E(3)
      DC61 = E(4)
      AC62 = E(5)
      BC62 = E(6)
      CC62 = E(7)
      DC62 = E(8)
      AC63 = E(9)
      BC63 = E(10)
      CC63 = E(11)
      DC63 = E(12)
      AC64 = E(13)
      BC64 = E(14)
C
      AS61 = R1*AC61
      BS61 = R1*BC61
      CS61 = S1*CC61
      DS61 = S1*DC61
      AS62 = R2*AC62
      BS62 = R2*BC62
      CS62 = S2*CC62
      DS62 = S2*DC62
      AS63 = R3*AC63
      BS63 = R3*BC63
      CS63 = S3*CC63
      DS63 = S3*DC63
C
      RETURN
      END
C=========================================================================C
C                                                                         C
C  VAL7 solves the Case 7 Equation System.                                C
C                                                                         C
C                                                                         C
      SUBROUTINE VAL7(UO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
```

```
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_7A/ G71,H71,G72,H72,UCP72,USP72
      COMMON /CASE_7B/ AC71,BC71,CC71,DC71,AS71,BS71,CS71,DS71
      COMMON /CASE_7C/ AC72,BC72,CC72,DC72,AS72,BS72,CS72,DS72
      DIMENSION A(20,20),D(20),E(20)
C
      NR = 8
C
      A1 = (XK1*PI*STLDIA)/(ES*ASTL)
      C1 = (XK1*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D1 = (XK1*PI*STLDIA)/(EC*ACON)
C
      A2 = (XK1*PI*STLDIA)/(ES*ASTL)
      C2 = (XK1*PI*STLDIA + XK4*WIDT)/(EC*ACON)
      D2 = (XK1*PI*STLDIA)/(EC*ACON)
      E2 = (XC2*WIDT)/(EC*ACON)
C
      G71 = (1.0-4.0*(A1*C1-A1*D1)/(A1+C1)**2.0)**0.5
      H71 = ((A1+C1)/2.0 - (A1+C1)*G71/2.0)**0.5
      G71 = ((A1+C1)/2.0 + (A1+C1)*G71/2.0)**0.5
C
      G72 = (1.0-4.0*(A2*C2-A2*D2)/(A2+C2)**2.0)**0.5
      H72 = ((A2+C2)/-2.0 + (A2+C2)*G72/2.0)**0.5
      G72 = ((A2+C2)/2.0 + (A2+C2)*G72/2.0)**0.5
C
      R1 = A1/(A1-G71**2.0)
      S1 = A1/(A1-H71**2.0)
C
      R2 = A2/(A2-G72**2.0)
      S2 = A2/(A2+H72**2.0)
C
      UCP72 = E2/(D2-C2)
      USP72 = E2/(D2-C2)
C
      GY1 = G71*S
      HY1 = H71*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      D(1) = 0.0
      D(2) = 0.0
      D(3) = UCP72
      D(4) = USP72
      D(5) = 0.0
      D(6) = 0.0
      D(7) = UO-UCP72
      D(8) = UO*STSTRAIN-USP72
C
      A(1,1) = EXP(GY1)
```

```
      A(1,2)  = EXP(-GY1)
      A(1,3)  = EXP(HY1)
      A(1,4)  = EXP(-HY1)
      A(1,5)  = 0.0
      A(1,6)  = 0.0
      A(1,7)  = 0.0
      A(1,8)  = 0.0
C
      A(2,1)  = R1*EXP(GY1)
      A(2,2)  = R1*EXP(-GY1)
      A(2,3)  = S1*EXP(HY1)
      A(2,4)  = S1*EXP(-HY1)
      A(2,5)  = 0.0
      A(2,6)  = 0.0
      A(2,7)  = 0.0
      A(2,8)  = 0.0
C
      A(3,1)  = EXP(G71*VL3)
      A(3,2)  = EXP(-G71*VL3)
      A(3,3)  = EXP(H71*VL3)
      A(3,4)  = EXP(-H71*VL3)
      A(3,5)  = -EXP(G72*VL3)
      A(3,6)  = -EXP(-G72*VL3)
      A(3,7)  = -COS(H72*VL3)
      A(3,8)  = -SIN(H72*VL3)
C
      A(4,1)  = R1*EXP(G71*VL3)
      A(4,2)  = R1*EXP(-G71*VL3)
      A(4,3)  = S1*EXP(H71*VL3)
      A(4,4)  = S1*EXP(-H71*VL3)
      A(4,5)  = -R2*EXP(G72*VL3)
      A(4,6)  = -R2*EXP(-G72*VL3)
      A(4,7)  = -S2*COS(H72*VL3)
      A(4,8)  = -S2*SIN(H72*VL3)
C
      A(5,1)  = G71*EXP(G71*VL3)
      A(5,2)  = -G71*EXP(-G71*VL3)
      A(5,3)  = H71*EXP(H71*VL3)
      A(5,4)  = -H71*EXP(-H71*VL3)
      A(5,5)  = -G72*EXP(G72*VL3)
      A(5,6)  = G72*EXP(-G72*VL3)
      A(5,7)  = H72*SIN(H72*VL3)
      A(5,8)  = -H72*COS(H72*VL3)
C
      A(6,1)  = R1*G71*EXP(G71*VL3)
      A(6,2)  = -R1*G71*EXP(-G71*VL3)
      A(6,3)  = S1*H71*EXP(H71*VL3)
      A(6,4)  = -S1*H71*EXP(-H71*VL3)
      A(6,5)  = -R2*G72*EXP(G72*VL3)
      A(6,6)  = R2*G72*EXP(-G72*VL3)
      A(6,7)  = S2*H72*SIN(H72*VL3)
      A(6,8)  = -S2*H72*COS(H72*VL3)
C
      A(7,1)  = 0.0
```

```
      A(7,2) = 0.0
      A(7,3) = 0.0
      A(7,4) = 0.0
      A(7,5) = EXP(G72*UO)
      A(7,6) = EXP(-G72*UO)
      A(7,7) = COS(H72*UO)
      A(7,8) = SIN(H72*UO)
C
      A(8,1) = 0.0
      A(8,2) = 0.0
      A(8,3) = 0.0
      A(8,4) = 0.0
      A(8,5) = R2*EXP(G72*UO)
      A(8,6) = R2*EXP(-G72*UO)
      A(8,7) = S2*COS(H72*UO)
      A(8,8) = S2*SIN(H72*UO)
C
      CALL INVERT(A,NR)
      CALL MULTI(A,D,E,NR)
C
      AC71 = E(1)
      BC71 = E(2)
      CC71 = E(3)
      DC71 = E(4)
      AC72 = E(5)
      BC72 = E(6)
      CC72 = E(7)
      DC72 = E(8)
C
      AS71 = R1*AC71
      BS71 = R1*BC71
      CS71 = S1*CC71
      DS71 = S1*DC71
      AS72 = R2*AC72
      BS72 = R2*BC72
      CS72 = S2*CC72
      DS72 = S2*DC72
C
      RETURN
      END
C=========================================================================C
C                                                                         C
C  VAL8 solves the Case 8 Equation System.                                C
C                                                                         C
C                                                                         C
      SUBROUTINE VAL8(UO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
```

```
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_8A/ G81,H81,G82,H82,G83,H83,UCP83,USP83,UCP82,USP82
      COMMON /CASE_8B/ AC81,BC81,CC81,DC81,AS81,BS81,CS81,DS81
      COMMON /CASE_8C/ AC82,BC82,CC82,DC82,AS82,BS82,CS82,DS82
      COMMON /CASE_8D/ AC83,BC83,CC83,DC83,AS83,BS83,CS83,DS83
      DIMENSION A(20,20),D(20),E(20)
C
      NR = 12
C
      A1 = (XK1*PI*STLDIA)/(ES*ASTL)
      C1 = (XK1*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D1 = (XK1*PI*STLDIA)/(EC*ACON)
C
      A2 = (XK1*PI*STLDIA)/(ES*ASTL)
      C2 = (XK1*PI*STLDIA + XK4*WIDT)/(EC*ACON)
      D2 = (XK1*PI*STLDIA)/(EC*ACON)
      E2 = (XC2*WIDT)/(EC*ACON)
C
      A3 = (XK2*PI*STLDIA)/(ES*ASTL)
      B3 = (-XC1*PI*STLDIA)/(ES*ASTL)
      C3 = (XK2*PI*STLDIA + XK4*WIDT)/(EC*ACON)
      D3 = (XK2*PI*STLDIA)/(EC*ACON)
      E3 = (XC1*PI*STLDIA + XC2*WIDT)/(EC*ACON)
C
      G81 = (1.0-4.0*(A1*C1-A1*D1)/(A1+C1)**2.0)**0.5
      H81 = ((A1+C1)/2.0 - (A1+C1)*G81/2.0)**0.5
      G81 = ((A1+C1)/2.0 + (A1+C1)*G81/2.0)**0.5
C
      G82 = (1.0-4.0*(A2*C2-A2*D2)/(A2+C2)**2.0)**0.5
      H82 = ((A2+C2)/-2.0 + (A2+C2)*G82/2.0)**0.5
      G82 = ((A2+C2)/2.0 + (A2+C2)*G82/2.0)**0.5
C
      G83 = (1.0-4.0*(A3*C3-A3*D3)/(A3+C3)**2.0)**0.5
      H83 = ((A3+C3)/-2.0 - (A3+C3)*G83/2.0)**0.5
      G83 = ((A3+C3)/-2.0 + (A3+C3)*G83/2.0)**0.5
C
      R1 = A1/(A1-G81**2.0)
      S1 = A1/(A1-H81**2.0)
C
      R2 = A2/(A2-G82**2.0)
      S2 = A2/(A2+H82**2.0)
C
      R3 = A3/(A3+G83**2.0)
      S3 = A3/(A3+H83**2.0)
C
      UCP82 = E2/(D2-C2)
      USP82 = E2/(D2-C2)
C
      UCP83 = (B3*D3+A3*E3)/(A3*(D3-C3))
      USP83 = (A3*E3+B3*C3)/(A3*(D3-C3))
C
      GY1 = G81*S
```

```
      HY1 = H81*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      D(1) = 0.0
      D(2) = 0.0
      D(3) = UCP82
      D(4) = USP82
      D(5) = 0.0
      D(6) = 0.0
      D(7) = UCP83-UCP82
      D(8) = USP83-USP82
      D(9) = 0.0
      D(10) = 0.0
      D(11) = UO-UCP83
      D(12) = UO*STSTRAIN-USP83
C
      A(1,1) = EXP(GY1)
      A(1,2) = EXP(-GY1)
      A(1,3) = EXP(HY1)
      A(1,4) = EXP(-HY1)
      A(1,5) = 0.0
      A(1,6) = 0.0
      A(1,7) = 0.0
      A(1,8) = 0.0
      A(1,9) = 0.0
      A(1,10) = 0.0
      A(1,11) = 0.0
      A(1,12) = 0.0
C
      A(2,1) = R1*EXP(GY1)
      A(2,2) = R1*EXP(-GY1)
      A(2,3) = S1*EXP(HY1)
      A(2,4) = S1*EXP(-HY1)
      A(2,5) = 0.0
      A(2,6) = 0.0
      A(2,7) = 0.0
      A(2,8) = 0.0
      A(2,9) = 0.0
      A(2,10) = 0.0
      A(2,11) = 0.0
      A(2,12) = 0.0
C
      A(3,1) = EXP(G81*VL3)
      A(3,2) = EXP(-G81*VL3)
      A(3,3) = EXP(H81*VL3)
      A(3,4) = EXP(-H81*VL3)
      A(3,5) = -EXP(G82*VL3)
      A(3,6) = -EXP(-G82*VL3)
      A(3,7) = -COS(H82*VL3)
      A(3,8) = -SIN(H82*VL3)
      A(3,9) = 0.0
      A(3,10) = 0.0
      A(3,11) = 0.0
```

```
      A(3,12) = 0.0
C
      A(4,1) = R1*EXP(G81*VL3)
      A(4,2) = R1*EXP(-G81*VL3)
      A(4,3) = S1*EXP(H81*VL3)
      A(4,4) = S1*EXP(-H81*VL3)
      A(4,5) = -R2*EXP(G82*VL3)
      A(4,6) = -R2*EXP(-G82*VL3)
      A(4,7) = -S2*COS(H82*VL3)
      A(4,8) = -S2*SIN(H82*VL3)
      A(4,9) = 0.0
      A(4,10) = 0.0
      A(4,11) = 0.0
      A(4,12) = 0.0
C
      A(5,1) = G81*EXP(G81*VL3)
      A(5,2) = -G81*EXP(-G81*VL3)
      A(5,3) = H81*EXP(H81*VL3)
      A(5,4) = -H81*EXP(-H81*VL3)
      A(5,5) = -G82*EXP(G82*VL3)
      A(5,6) = G82*EXP(-G82*VL3)
      A(5,7) = H82*SIN(H82*VL3)
      A(5,8) = -H82*COS(H82*VL3)
      A(5,9) = 0.0
      A(5,10) = 0.0
      A(5,11) = 0.0
      A(5,12) = 0.0
C
      A(6,1) = R1*G81*EXP(G81*VL3)
      A(6,2) = -R1*G81*EXP(-G81*VL3)
      A(6,3) = S1*H81*EXP(H81*VL3)
      A(6,4) = -S1*H81*EXP(-H81*VL3)
      A(6,5) = -R2*G82*EXP(G82*VL3)
      A(6,6) = R2*G82*EXP(-G82*VL3)
      A(6,7) = S2*H82*SIN(H82*VL3)
      A(6,8) = -S2*H82*COS(H82*VL3)
      A(6,9) = 0.0
      A(6,10) = 0.0
      A(6,11) = 0.0
      A(6,12) = 0.0
C
      A(7,1) = 0.0
      A(7,2) = 0.0
      A(7,3) = 0.0
      A(7,4) = 0.0
      A(7,5) = EXP(G82*VL1)
      A(7,6) = EXP(-G82*VL1)
      A(7,7) = COS(H82*VL1)
      A(7,8) = SIN(H82*VL1)
      A(7,9) = -COS(G83*VL1)
      A(7,10) = -SIN(G83*VL1)
      A(7,11) = -COS(H83*VL1)
      A(7,12) = -SIN(H83*VL1)
C
```

```
      A(8,1) = 0.0
      A(8,2) = 0.0
      A(8,3) = 0.0
      A(8,4) = 0.0
      A(8,5) = R2*EXP(G82*VL1)
      A(8,6) = R2*EXP(-G82*VL1)
      A(8,7) = S2*COS(H82*VL1)
      A(8,8) = S2*SIN(H82*VL1)
      A(8,9) = -R3*COS(G83*VL1)
      A(8,10) = -R3*SIN(G83*VL1)
      A(8,11) = -S3*COS(H83*VL1)
      A(8,12) = -S3*SIN(H83*VL1)
C
      A(9,1) = 0.0
      A(9,2) = 0.0
      A(9,3) = 0.0
      A(9,4) = 0.0
      A(9,5) = G82*EXP(G82*VL1)
      A(9,6) = -G82*EXP(-G82*VL1)
      A(9,7) = -H82*SIN(H82*VL1)
      A(9,8) = H82*COS(H82*VL1)
      A(9,9) = G83*SIN(G83*VL1)
      A(9,10) = -G83*COS(G83*VL1)
      A(9,11) = H83*SIN(H83*VL1)
      A(9,12) = -H83*COS(H83*VL1)
C
      A(10,1) = 0.0
      A(10,2) = 0.0
      A(10,3) = 0.0
      A(10,4) = 0.0
      A(10,5) = R2*G82*EXP(G82*VL1)
      A(10,6) = -R2*G82*EXP(-G82*VL1)
      A(10,7) = -S2*H82*SIN(H82*VL1)
      A(10,8) = S2*H82*COS(H82*VL1)
      A(10,9) = R3*G83*SIN(G83*VL1)
      A(10,10) = -R3*G83*COS(G83*VL1)
      A(10,11) = S3*H83*SIN(H83*VL1)
      A(10,12) = -S3*H83*COS(H83*VL1)
C
      A(11,1) = 0.0
      A(11,2) = 0.0
      A(11,3) = 0.0
      A(11,4) = 0.0
      A(11,5) = 0.0
      A(11,6) = 0.0
      A(11,7) = 0.0
      A(11,8) = 0.0
      A(11,9) = COS(G83*UO)
      A(11,10) = SIN(G83*UO)
      A(11,11) = COS(H83*UO)
      A(11,12) = SIN(G83*UO)
C
      A(12,1) = 0.0
      A(12,2) = 0.0
```

```
      A(12,3) = 0.0
      A(12,4) = 0.0
      A(12,5) = 0.0
      A(12,6) = 0.0
      A(12,7) = 0.0
      A(12,8) = 0.0
      A(12,9) = R3*COS(G83*UO)
      A(12,10) = R3*SIN(G83*UO)
      A(12,11) = S3*COS(H83*UO)
      A(12,12) = S3*SIN(H83*UO)
C
      CALL INVERT(A,NR)
      CALL MULTI(A,D,E,NR)
C
      AC81 = E(1)
      BC81 = E(2)
      CC81 = E(3)
      DC81 = E(4)
      AC82 = E(5)
      BC82 = E(6)
      CC82 = E(7)
      DC82 = E(8)
      AC83 = E(9)
      BC83 = E(10)
      CC83 = E(11)
      DC83 = E(12)
C
      AS81 = R1*AC81
      BS81 = R1*BC81
      CS81 = S1*CC81
      DS81 = S1*DC81
      AS82 = R2*AC82
      BS82 = R2*BC82
      CS82 = S2*CC82
      DS82 = S2*DC82
      AS83 = R3*AC83
      BS83 = R3*BC83
      CS83 = S3*CC83
      DS83 = S3*DC83
C
      RETURN
      END

C=====================================================================C
C                                                                     C
C  VAL9 solves the Case 9 Equation System.                            C
C                                                                     C
C                                                                     C
      SUBROUTINE VAL9(UO)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
```

```
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      COMMON /CASE_9A/ G91,H91,G92,H92,G93,H93,G94
      COMMON /CASE_9B/ AC91,BC91,CC91,DC91,AS91,BS91,CS91,DS91
      COMMON /CASE_9C/ AC92,BC92,CC92,DC92,AS92,BS92,CS92,DS92
      COMMON /CASE_9D/ AC93,BC93,CC93,DC93,AS93,BS93,CS93,DS93
      COMMON /CASE_9E/ AC94,BC94,UCP94,UCP93,USP93,UCP92,USP92
      DIMENSION A(20,20),D(20),E(20)
C
      NR = 14
C
      A1 = (XK1*PI*STLDIA)/(ES*ASTL)
      C1 = (XK1*PI*STLDIA + XK3*WIDT)/(EC*ACON)
      D1 = (XK1*PI*STLDIA)/(EC*ACON)
C
      A2 = (XK1*PI*STLDIA)/(ES*ASTL)
      C2 = (XK1*PI*STLDIA + XK4*WIDT)/(EC*ACON)
      D2 = (XK1*PI*STLDIA)/(EC*ACON)
      E2 = (XC2*WIDT)/(EC*ACON)
C
      A3 = (XK2*PI*STLDIA)/(ES*ASTL)
      B3 = (-XC1*PI*STLDIA)/(ES*ASTL)
      C3 = (XK2*PI*STLDIA + XK4*WIDT)/(EC*ACON)
      D3 = (XK2*PI*STLDIA)/(EC*ACON)
      E3 = (XC1*PI*STLDIA + XC2*WIDT)/(EC*ACON)
C
      A4 = (-XK4*WIDT)/(EC*ACON)
      B4 = (XC2*WIDT)/(EC*ACON)
C
      G91 = (1.0-4.0*(A1*C1-A1*D1)/(A1+C1)**2.0)**0.5
      H91 = ((A1+C1)/2.0 - (A1+C1)*G91/2.0)**0.5
      G91 = ((A1+C1)/2.0 + (A1+C1)*G91/2.0)**0.5
C
      G92 = (1.0-4.0*(A2*C2-A2*D2)/(A2+C2)**2.0)**0.5
      H92 = ((A2+C2)/-2.0 + (A2+C2)*G92/2.0)**0.5
      G92 = ((A2+C2)/2.0 + (A2+C2)*G92/2.0)**0.5
C
      G93 = (1.0-4.0*(A3*C3-A3*D3)/(A3+C3)**2.0)**0.5
      H93 = ((A3+C3)/-2.0 - (A3+C3)*G93/2.0)**0.5
      G93 = ((A3+C3)/-2.0 + (A3+C3)*G93/2.0)**0.5
C
      G94 = A4**0.5
C
      R1 = A1/(A1-G91**2.0)
      S1 = A1/(A1-H91**2.0)
C
      R2 = A2/(A2-G92**2.0)
      S2 = A2/(A2+H92**2.0)
C
      R3 = A3/(A3+G93**2.0)
      S3 = A3/(A3+H93**2.0)
```

```
C
      UCP92 = E2/(D2-C2)
      USP92 = E2/(D2-C2)
C
      UCP93 = (B3*D3+A3*E3)/(A3*(D3-C3))
      USP93 = (A3*E3+B3*C3)/(A3*(D3-C3))
C
      UCP94 = B4/A4
C
      GY1 = G91*S
      HY1 = H91*S
      IF (GY1.GE.700.0) GY1 = 700.0
      IF (HY1.GE.700.0) HY1 = 700.0
C
      D(1) = 0.0
      D(2) = 0.0
      D(3) = UCP92
      D(4) = USP92
      D(5) = 0.0
      D(6) = 0.0
      D(7) = UCP93-UCP92
      D(8) = USP93-USP92
      D(9) = 0.0
      D(10) = 0.0
      D(11) = UCP94-UCP93
      D(12) = 0.0
      D(13) = VL2*STSTRAIN-USP93
      D(14) = UO-UCP94
C
      A(1,1) = EXP(GY1)
      A(1,2) = EXP(-GY1)
      A(1,3) = EXP(HY1)
      A(1,4) = EXP(-HY1)
      A(1,5) = 0.0
      A(1,6) = 0.0
      A(1,7) = 0.0
      A(1,8) = 0.0
      A(1,9) = 0.0
      A(1,10) = 0.0
      A(1,11) = 0.0
      A(1,12) = 0.0
      A(1,13) = 0.0
      A(1,14) = 0.0
C
      A(2,1) = R1*EXP(GY1)
      A(2,2) = R1*EXP(-GY1)
      A(2,3) = S1*EXP(HY1)
      A(2,4) = S1*EXP(-HY1)
      A(2,5) = 0.0
      A(2,6) = 0.0
      A(2,7) = 0.0
      A(2,8) = 0.0
      A(2,9) = 0.0
      A(2,10) = 0.0
```

```
      A(2,11) = 0.0
      A(2,12) = 0.0
      A(2,13) = 0.0
      A(2,14) = 0.0
C
      A(3,1) = EXP(G91*VL3)
      A(3,2) = EXP(-G91*VL3)
      A(3,3) = EXP(H91*VL3)
      A(3,4) = EXP(-H91*VL3)
      A(3,5) = -EXP(G92*VL3)
      A(3,6) = -EXP(-G92*VL3)
      A(3,7) = -COS(H92*VL3)
      A(3,8) = -SIN(H92*VL3)
      A(3,9) = 0.0
      A(3,10) = 0.0
      A(3,11) = 0.0
      A(3,12) = 0.0
      A(3,13) = 0.0
      A(3,14) = 0.0
C
      A(4,1) = R1*EXP(G91*VL3)
      A(4,2) = R1*EXP(-G91*VL3)
      A(4,3) = S1*EXP(H91*VL3)
      A(4,4) = S1*EXP(-H91*VL3)
      A(4,5) = -R2*EXP(G92*VL3)
      A(4,6) = -R2*EXP(-G92*VL3)
      A(4,7) = -S2*COS(H92*VL3)
      A(4,8) = -S2*SIN(H92*VL3)
      A(4,9) = 0.0
      A(4,10) = 0.0
      A(4,11) = 0.0
      A(4,12) = 0.0
      A(4,13) = 0.0
      A(4,14) = 0.0
C
      A(5,1) = G91*EXP(G91*VL3)
      A(5,2) = -G91*EXP(-G91*VL3)
      A(5,3) = H91*EXP(H91*VL3)
      A(5,4) = -H91*EXP(-H91*VL3)
      A(5,5) = -G92*EXP(G92*VL3)
      A(5,6) = G92*EXP(-G92*VL3)
      A(5,7) = H92*SIN(H92*VL3)
      A(5,8) = -H92*COS(H92*VL3)
      A(5,9) = 0.0
      A(5,10) = 0.0
      A(5,11) = 0.0
      A(5,12) = 0.0
      A(5,13) = 0.0
      A(5,14) = 0.0
C
      A(6,1) = R1*G91*EXP(G91*VL3)
      A(6,2) = -R1*G91*EXP(-G91*VL3)
      A(6,3) = S1*H91*EXP(H91*VL3)
      A(6,4) = -S1*H91*EXP(-H91*VL3)
```

```
      A(6,5)  = -R2*G92*EXP(G92*VL3)
      A(6,6)  = R2*G92*EXP(-G92*VL3)
      A(6,7)  = S2*H92*SIN(H92*VL3)
      A(6,8)  = -S2*H92*COS(H92*VL3)
      A(6,9)  = 0.0
      A(6,10)  = 0.0
      A(6,11)  = 0.0
      A(6,12)  = 0.0
      A(6,13)  = 0.0
      A(6,14)  = 0.0
C
      A(7,1)  = 0.0
      A(7,2)  = 0.0
      A(7,3)  = 0.0
      A(7,4)  = 0.0
      A(7,5)  = EXP(G92*VL1)
      A(7,6)  = EXP(-G92*VL1)
      A(7,7)  = COS(H92*VL1)
      A(7,8)  = SIN(H92*VL1)
      A(7,9)  = -COS(G93*VL1)
      A(7,10)  = -SIN(G93*VL1)
      A(7,11)  = -COS(H93*VL1)
      A(7,12)  = -SIN(H93*VL1)
      A(7,13)  = 0.0
      A(7,14)  = 0.0
C
      A(8,1)  = 0.0
      A(8,2)  = 0.0
      A(8,3)  = 0.0
      A(8,4)  = 0.0
      A(8,5)  = R2*EXP(G92*VL1)
      A(8,6)  = R2*EXP(-G92*VL1)
      A(8,7)  = S2*COS(H92*VL1)
      A(8,8)  = S2*SIN(H92*VL1)
      A(8,9)  = -R3*COS(G93*VL1)
      A(8,10)  = -R3*SIN(G93*VL1)
      A(8,11)  = -S3*COS(H93*VL1)
      A(8,12)  = -S3*SIN(H93*VL1)
      A(8,13)  = 0.0
      A(8,14)  = 0.0
C
      A(9,1)  = 0.0
      A(9,2)  = 0.0
      A(9,3)  = 0.0
      A(9,4)  = 0.0
      A(9,5)  = G92*EXP(G92*VL1)
      A(9,6)  = -G92*EXP(-G92*VL1)
      A(9,7)  = -H92*SIN(H92*VL1)
      A(9,8)  = H92*COS(H92*VL1)
      A(9,9)  = G93*SIN(G93*VL1)
      A(9,10)  = -G93*COS(G93*VL1)
      A(9,11)  = H93*SIN(H93*VL1)
      A(9,12)  = -H93*COS(H93*VL1)
      A(9,13)  = 0.0
```

```
      A(9,14) = 0.0
C
      A(10,1) = 0.0
      A(10,2) = 0.0
      A(10,3) = 0.0
      A(10,4) = 0.0
      A(10,5) = R2*G92*EXP(G92*VL1)
      A(10,6) = -R2*G92*EXP(-G92*VL1)
      A(10,7) = -S2*H92*SIN(H92*VL1)
      A(10,8) = S2*H92*COS(H92*VL1)
      A(10,9) = R3*G93*SIN(G93*VL1)
      A(10,10) = -R3*G93*COS(G93*VL1)
      A(10,11) = S3*H93*SIN(H93*VL1)
      A(10,12) = -S3*H93*COS(H93*VL1)
      A(10,13) = 0.0
      A(10,14) = 0.0
C
      A(11,1) = 0.0
      A(11,2) = 0.0
      A(11,3) = 0.0
      A(11,4) = 0.0
      A(11,5) = 0.0
      A(11,6) = 0.0
      A(11,7) = 0.0
      A(11,8) = 0.0
      A(11,9) = COS(G93*VL2)
      A(11,10) = SIN(G93*VL2)
      A(11,11) = COS(H93*VL2)
      A(11,12) = SIN(G93*VL2)
      A(11,13) = -COS(G94*VL2)
      A(11,14) = -SIN(G94*VL2)
C
      A(12,1) = 0.0
      A(12,2) = 0.0
      A(12,3) = 0.0
      A(12,4) = 0.0
      A(12,5) = 0.0
      A(12,6) = 0.0
      A(12,7) = 0.0
      A(12,8) = 0.0
      A(12,9) = -G93*SIN(G93*VL2)
      A(12,10) = G93*COS(G93*VL2)
      A(12,11) = -H93*SIN(H93*VL2)
      A(12,12) = H93*COS(H93*VL2)
      A(12,13) = G94*SIN(G94*VL2)
      A(12,14) = -G94*COS(G94*VL2)
C
      A(13,1) = 0.0
      A(13,2) = 0.0
      A(13,3) = 0.0
      A(13,4) = 0.0
      A(13,5) = 0.0
      A(13,6) = 0.0
      A(13,7) = 0.0
```

```
      A(13,8) = 0.0
      A(13,9) = R3*COS(G93*VL2)
      A(13,10) = R3*SIN(G93*VL2)
      A(13,11) = S3*COS(H93*VL2)
      A(13,12) = S3*SIN(H93*VL2)
      A(13,13) = 0.0
      A(13,14) = 0.0
C
      A(14,1) = 0.0
      A(14,2) = 0.0
      A(14,3) = 0.0
      A(14,4) = 0.0
      A(14,5) = 0.0
      A(14,6) = 0.0
      A(14,7) = 0.0
      A(14,8) = 0.0
      A(14,9) = 0.0
      A(14,10) = 0.0
      A(14,11) = 0.0
      A(14,12) = 0.0
      A(14,13) = COS(G94*UO)
      A(14,14) = SIN(G94*UO)
C
      CALL INVERT(A,NR)
      CALL MULTI(A,D,E,NR)
C
      AC91 = E(1)
      BC91 = E(2)
      CC91 = E(3)
      DC91 = E(4)
      AC92 = E(5)
      BC92 = E(6)
      CC92 = E(7)
      DC92 = E(8)
      AC93 = E(9)
      BC93 = E(10)
      CC93 = E(11)
      DC93 = E(12)
      AC94 = E(13)
      BC94 = E(14)
C
      AS91 = R1*AC91
      BS91 = R1*BC91
      CS91 = S1*CC91
      DS91 = S1*DC91
      AS92 = R2*AC92
      BS92 = R2*BC92
      CS92 = S2*CC92
      DS92 = S2*DC92
      AS93 = R3*AC93
      BS93 = R3*BC93
      CS93 = S3*CC93
      DS93 = S3*DC93
C
```

```
      RETURN
      END

C==========================================================================C
C                                                                          C
C  INPUT does exactly what you guessed it does...it reads in the data!     C
C                                                                          C
C                                                                          C
      SUBROUTINE INPUT(MTIME)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
C
C
C  THE FIRST STEP IS TO INPUT THE NON-TIME DEPENDENT PARAMETERS
C
C
      READ(5,*) SPACE,STLDIA,LAYER,ES,ALPHAS,WIDT,THICK,ULTSHR,CURTEMP
      PI = 3.14159266
      STLDIA = STLDIA*LAYER
      ASTL = (PI*(STLDIA/2.0)**2.0)/LAYER
      ACON = (WIDT*THICK-ASTL)
      S = SPACE/2.0
C
C
C  AND NOW . . .  THE TIME DEPENDENT PARAMETERS !
C
C
      READ(5,*) FLG1
C
C  When FLG1 = 1.0, only 1 day's data is input.
C
      IF (FLG1.EQ.1.0) THEN
        MTIME = 1
        READ(5,*) FPRMC(1),FPRMT(1),ELASCON(1),ALPHACON(1)
        READ(5,*) BONSLP(1,1),BONSLP(2,1),BONSLP(3,1),BONSLP(4,1)
        READ(5,*) STLSLIP(1),CONSLIP(1),AMAGLOD,SHRINKAGE,DAYTEMP(1)
        READ(5,*) RADLOD,BASEK
        RELHUM(1) = 100.0
        KDAYLOBEG = 1
        GOTO 200
        ENDIF
C
C  MTIME is the length of the analysis period in days.
C
      READ(5,*) MTIME
      READ(5,*) FLG2
```

```
C
C  When FLG2 = 1.0, the compressive strength of the concrete is read in
C  for each day.  Otherwise, the computer will generate strength data from
C  the 7 and 28 day compressive strengths.
C
      IF (FLG2.EQ.1.0) THEN
        DO 10 I = 1,MTIME
        READ(5,*) FPRMC(I)
 10     CONTINUE
      ELSE
        READ(5,*) STREN7,STREN28
        ENDIF
      READ(5,*) FLG3
C
C  When FLG3 = 1.0, the tensile strength of the concrete is read in for
C  each day.  Otherwise, the tensile strength is assumed to be a function
C  of the compressive strength (see below).
C
      IF (FLG3.EQ.1.0) THEN
        DO 30 I = 1,MTIME
         READ(5,*) FPRMT(I)
 30     CONTINUE
      ELSE
        DO 40 I = 1,MTIME
          FPRMT(I) = 7.5*FPRMC(I)**0.5
 40     CONTINUE
      ENDIF
      READ(5,*) FLG4
C
C  When FLG4 = 1.0, the elastic modulus of the concrete is read in for each
C  day.  Otherwise, it is assumed to be a function of the compressive
C  strength.
C
      IF (FLG4.EQ.1.0) THEN
        DO 50 I = 1,MTIME
          READ(5,*) ELASCON(I)
 50     CONTINUE
      ELSE
        DO 60 I = 1,MTIME
          ELASCON(I) = 57000.0*FPRMC(I)**0.5
 60     CONTINUE
      ENDIF
      READ(5,*) FLG5
C
C  When FLG5 = 1.0 the concrete's coefficient of thermal expansion is read
C  for each day.  If FLG5 = 2.0, it is assumed to be constant.  Otherwise,
C  it is assumed to be a function of the maturity of the concrete and the
C  aggregate type. (This has not yet been implemented.)
C
      IF (FLG5.EQ.1.0) THEN
        DO 70 I = 1,MTIME
          READ(5,*) ALPHACON(I)
 70     CONTINUE
      ELSE
```

```
        IF (FLG5.EQ.2.0) THEN
          READ(5,*) ALPHA
          DO 80 I = 1,MTIME
            ALPHACON(I) = ALPHA
 80       CONTINUE
        ELSE
          DO 90 I = 1,MTIME
C           ALPHACON(I) = [ SOME FUNCTION ]
 90       CONTINUE
        ENDIF
      ENDIF
      READ(5,*) FLG6
C
C  When FLG6 = 1.0, the bond/slip moduli are read in for each day.
C  Otherwise, they are assumed to be a function of the compressive
C  strength. (The functions have not been substantiated in the least.)
C
      IF (FLG6.EQ.1.0) THEN
        DO 100 I = 1,MTIME
          READ(5,*) BONSLP(1,I),BONSLP(2,I),BONSLP(3,I),BONSLP(4,I)
 100    CONTINUE
      ELSE
        DO 110 I = 1,MTIME
          BONSLP(1,I) = 5500.0*FPRMC(I)**0.5
          BONSLP(2,I) = -BONSLP(1,I)/2.0
          BONSLP(3,I) = 2000.0 + 40.0*FPRMC(I)**0.5
          BONSLP(4,I) = -BONSLP(3,I)/10.0
 110    CONTINUE
      ENDIF
      READ(5,*) FLG7
C
C  When FLG7 = 1.0, the peak bond stress/slip value is read in for each
C  day.  Otherwise, it is assumed to be a function of the bond/slip
C  moduli.
C
      IF (FLG7.EQ.1.0) THEN
        DO 120 I = 1,MTIME
          READ(5,*) STLSLIP(I),CONSLIP(I)
 120    CONTINUE
      ELSE
        DO 130 I = 1,MTIME
          STLSLIP(I) = 0.0040 - 5.0E-9*BONSLP(1,I)
          CONSLIP(I) = 0.0266 - 3.3E-6*BONSLP(3,I)
 130    CONTINUE
      ENDIF
      READ(5,*) FLG8
C
C  When FLG8 = 1.0, the minimum temperatures are read in for each day.
C  Otherwise, a linear relationship is assumed, with a initial
C  temperature (STARTEM), and a final temperature (STOPEM).
C
      IF (FLG8.EQ.1.0) THEN
        DO 140 I = 1,MTIME
          READ(5,*) DAYTEMP(I)
```

```
 140    CONTINUE
      ELSE
        READ(5,*) STARTEM,STOPEM
        DTEM = (STOPEM-STARTEM)/MTIME
        DO 150 I = 1,MTIME
          DAYTEMP(I) = STARTEM + DTEM*I
 150    CONTINUE
      ENDIF
      READ(5,*) FLG9
C
C  When FLG9 = 1.0, the relative humidity of the concrete is read in for
C  each day.  Otherwise it is assumed to be constant.
C
      IF (FLG9.EQ.1.0) THEN
        DO 160 I = 1,MTIME
          READ(5,*) RELHUM(I)
 160    CONTINUE
      ELSE
        READ(5,*) RELAHU
        DO 170 I = 1,MTIME
          RELHUM(I) = RELAHU
 170    CONTINUE
      ENDIF
      SHRINKAGE = 0.0
      READ(5,*) FLG10
C
C  When FLG10 = 1.0, wheel loading of the pavement will be taken into
C  consideration.  Otherwise, it is assumed that no external loading
C  of the pavement occurs.
C
      IF (FLG10.EQ.1.0) THEN
        READ(5,*) KDAYLOBEG,AMAGLOD,RADLOD,BASEK
      ELSE
        KDAYLOBEG = MTIME+1
        AMAGLOD = 0.0
        RADLOD = 0.0
        BASEK = 1.0
      ENDIF
 200  CALL OUTINPUT(MTIME)
      RETURN
      END

C=========================================================================C
C                                                                         C
C  And now, after all the input data has been read, it is time to print   C
C  it all back out just to make sure that it was all read in correctly.   C
C                                                                         C
C                                                                         C
      SUBROUTINE OUTINPUT(MTIME)
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON /BLK1/ XK1,XK2,XK3,XK4,ES,EC,ALPHA,PI,SPACE,THICK
      COMMON /BLK2/ NC,WIDT,STLDIA,ASTL,ACON,DELTEM,S,KASE,TAE
      COMMON /BLK3/ GF,DIF,STRN,VL1,VL2,VL3,STSTRAIN,DUCDX
      COMMON /BLK4/ FRICTION,STREEN,POTENTIAL,FORCE
```

```fortran
      COMMON /BLK5/ XC1,XC2,STLSLP,STLLIM,CONSLP,CONLIM
      COMMON /BLK6/ ELASCON(100),FPRMC(100),FPRMT(100),BONSLP(4,100)
      COMMON /BLK7/ ALPHACON(100),DAYTEMP(100),RELHUM(100),STLSLIP(100)
      COMMON /BLK8/ CONSLIP(100),ALPHAS,KDAYLOBEG,AMAGLOD,STRESS,ULTSHR
      COMMON /BLK9/ SHRINKAGE,CURTEMP,LAYER,CENTERSTR,RADLOD,BASEK
      WRITE(6,1)
1     FORMAT('1','===================================================',
     +           '=======================')
      WRITE(6,*)
      WRITE(6,*) 'THE INPUT PARAMETERS ARE:'
      WRITE(6,*)
      WRITE(6,11) MTIME
11    FORMAT(1X,'Length of Analysis Period = ',I3,' day(s)')
      WRITE(6,2) SPACE
2     FORMAT(1X,'Initial Crack Spacing = ',F7.1,' inches')
      WRITE(6,3) THICK
3     FORMAT(1X,'Slab Thickness = ',F4.1,' inches')
      WRITE(6,*)
      WRITE(6,4) WIDT
4     FORMAT(1X,'Steel Bar Spacing = ',F4.1,' inches')
      WRITE(6,5) STLDIA/LAYER
5     FORMAT(1X,'Steel Bar Diameter = ',F5.3,' inches')
      WRITE(6,6) LAYER
6     FORMAT(1X,'Layers of Steel =',I2)
      WRITE(6,58) ASTL
58    FORMAT('0','Cross-Sectional Area of the Steel = ',F5.3,
     +           ' sq. inches')
      WRITE(6,59) ACON
59    FORMAT(1X,'Cross-Sectional Area of the Concrete = ',F6.2,
     +           ' sq. inches')
      WRITE(6,89) (ASTL/(ACON+ASTL))*100.0
89    FORMAT(1X,'Steel Percentage = ',F4.2,'%')
      WRITE(6,33) ES
33    FORMAT('0','Steel Modulus of Elasticity = ',F10.1,' psi')
      WRITE(6,7) ALPHAS
7     FORMAT(1X,'Steel Coefficient of Thermal Expansion = ',F9.7,
     +           ' inches/inch-degree F')
      WRITE(6,*)
      WRITE(6,8) CURTEMP
8     FORMAT(1X,'Concrete Curing Temperature = ',F5.1,' degrees F')
      WRITE(6,22) SHRINKAGE
22    FORMAT(1X,'Initial Shrinkage of Concrete  = ',F9.7,
     +           ' inches/inch')
      WRITE(6,9) ULTSHR
9     FORMAT(1X,'Ultimate Shrinkage of Concrete = ',F9.7,
     +           ' inches/inch')
      WRITE(6,*)
      WRITE(6,20) KDAYLOBEG
20    FORMAT(1X,'Wheel Loading of the Pavement Begins on Day ',I3)
      WRITE(6,21) AMAGLOD
21    FORMAT(1X,'Magnitude of Wheel Load = ',F7.1,' lbs')
      WRITE(6,25) RADLOD
25    FORMAT(1X,'Radius of Tire Contact Area = ',F4.1,' inches')
      WRITE(6,26) BASEK
```

```
26    FORMAT(1X,'Modulus of Subgrade Reaction = ',F5.1,' lb/in≠2/in')
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,10)
      WRITE(6,*)
10    FORMAT(1X,'Concrete Properties and Environmental Conditions')
      WRITE(6,*)
      WRITE(6,12)
12    FORMAT(1X,'Day       Ec',9X,'f¢c       f¢t       Alpha     Rel. ',
     +           'Hum.   Min. Temp.')
      WRITE(6,13)
13    FORMAT(1X,'___   _________   ______   ______   _________    ',
     +           '_________   __________')
      DO 100 I = 1,MTIME
      WRITE(6,14)I,ELASCON(I),FPRMC(I),FPRMT(I),ALPHACON(I),RELHUM(I),
     +           DAYTEMP(I)
14    FORMAT(1X,I3,3X,F9.1,3X,F6.1,3X,F6.1,3X,F9.7,5X,F5.1,7X,F6.2)
100   CONTINUE
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,15)
      WRITE(6,*)
15    FORMAT(1X,'Local Bond Stress / Slip Coefficients')
      WRITE(6,*)
      WRITE(6,17)
17    FORMAT(1X,'Day',7X,'K1',11X,'K2',7X,'STLSLP',7X,'K3',11X,
     +           'K4',7X,'CONSLP')
      WRITE(6,18)
18    FORMAT(1X,'---   ----------   ----------   ------   ----------',
     +           '   ----------   ------')
      DO 110 I = 1,MTIME
      WRITE(6,19) I,BONSLP(1,I),BONSLP(2,I),STLSLIP(I),BONSLP(3,I),
     +            BONSLP(4,I),CONSLIP(I)
19    FORMAT(1X,I3,3X,F10.1,3X,F10.1,3X,F6.4,3X,F10.1,3X,F10.1,3X,F6.4)
110   CONTINUE
      WRITE(6,*)
      WRITE(6,30)
30    FORMAT(' ','==================================================',
     +           '========================')
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)
      RETURN
      END

C=====================================================================C
C                                                                     C
C  This routine is the workhorse behind the whole program.  It inverts C
C  the "A" matrix and enables the solution to those nasty differential C
C  equations to be found without too much sweat loss.                 C
C                                                                     C
C                                                                     C
      SUBROUTINE INVERT(A,N)
      IMPLICIT REAL*8 (A-H, O-Z)
```

```
      DIMENSION INDEX(20,2), A(20,20), B(20,20)
      DO 20 I=1,N
      DO 10 J=1,N
      B(I,J)=A(I,J)
10    CONTINUE
20    CONTINUE
      DO 30 I = 1,N
      INDEX(I,1) = 0
30    CONTINUE
      II = 0
32    AMAX = -1.0
      DO 50 I = 1,N
      IF(INDEX(I,1)) 50,35,50
35    DO 40 J = 1,N
      IF(INDEX(J,1)) 40,37,40
37    TEMP = ABS(A(I,J))
      IF(TEMP-AMAX) 40,40,38
38    IROW = I
      ICOL = J
      AMAX = TEMP
40    CONTINUE
50    CONTINUE
      IF (AMAX) 102,103,52
52    INDEX(ICOL,1) = IROW
      IF (IROW-ICOL) 54,63,54
54    DO 60 J =1,N
      TEMP = A(IROW,J)
      A(IROW,J) = A(ICOL,J)
      A(ICOL,J) = TEMP
60    CONTINUE
      II =II+1
      INDEX(II,2) = ICOL
63    PIVOT = A(ICOL,ICOL)
      A(ICOL,ICOL) = 1.0
      PIVOT = 1.0/PIVOT
      DO 70 J = 1,N
      A(ICOL,J) = A(ICOL,J)*PIVOT
70    CONTINUE
      DO 90 I = 1,N
      IF (I-ICOL) 72,90,72
72    TEMP = A(I,ICOL)
      A(I,ICOL) = 0.0
      DO 80 J = 1,N
      A(I,J) = A(I,J)-A(ICOL,J)*TEMP
80    CONTINUE
90    CONTINUE
      GOTO 32
92    ICOL = INDEX(II,2)
      IROW = INDEX(ICOL,1)
      DO 100 I = 1,N
      TEMP = A(I,IROW)
      A(I,IROW) = A(I,ICOL)
      A( I,ICOL) = TEMP
100   CONTINUE
```

```
      II = II-1
 102  IF (II) 92,111,92
 103  WRITE(6,110)
 110  FORMAT(1X,'THIS MATRIX HAS A ZERO PIVOT')
 111     RETURN
      END

C==========================================================================C
C                                                                          C
C  This is just your basic matrix multiplier subroutine...an average,      C
C  run-of-the-mill, no frills special.  Whoop-de-doos and advanced         C
C  coding techniques can be found someplace else.                          C
C                                                                          C
C                                                                          C
      SUBROUTINE MULTI(A,D,E,N)
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION A(20,20), D(20), E(20)
      DO 10 I = 1,N
        E(I) = 0.0
        DO 5 J = 1,N
          E(I) = E(I) + A(I,J)*D(J)
 5      CONTINUE
 10   CONTINUE
      RETURN
      END
```