Analysis for Creep, Shrinkage and Temperature Effects on Expansion Joint Movements in Composite Prestressed Concrete Bridges.

Keith Joseph Rebello

Louisiana State University and Agricultural & Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_disstheses

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_disstheses/5019
INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

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

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

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

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Analysis for creep, shrinkage and temperature effects on expansion joint movements in composite prestressed concrete bridges

Rebello, Keith Joseph, Ph.D.
The Louisiana State University and Agricultural and Mechanical Col., 1990
Analysis for Creep, Shrinkage and Temperature Effects on Expansion Joint Movements in Composite Prestressed Concrete Bridges

A Dissertation

Submitted to the Graduate Faculty of the Louisiana State University and Agricultural and Mechanical College in partial fulfillment of the requirements for the degree of Doctor of Philosophy

in

The Department of Civil Engineering

by

Keith J. Rebello
B.Tech., Indian Institute of Technology, Bombay, 1983
M.S., Louisiana State University, 1986
August 1990
Acknowledgements

My sincere thanks are due to Dr. Vijaya K.A. Gopu for the continuous guidance and support he has given me throughout this dissertation. He showed a special interest in this research investigation and provided information, useful suggestions, criticisms and encouragement.

I would also like to thank the members of my examining committee for the help they have given me. In particular, I thank Dr. R. Richard Avent for his guidance as principal investigator in the research project of which this dissertation is a part. I thank Dr. Fariborz Barzegar for the invaluable criticisms and directions he has given me. Dr. George Z. Voyiadjis is sincerely appreciated for taking special interest in my academic career at L.S.U. and for always having the time to advise me. I would also like to thank Dr. David E. Thompson who graciously accepted the responsibility of serving on my graduate committee at very short notice despite a busy schedule.

The Department of Civil Engineering at L.S.U. and in particular, Dr. Roger K. Seals, have my appreciation for providing me with the opportunity to pursue part of my degree by way of a teaching assistantship. Funding for this research was provided by the Louisiana Department of Transportation and Development, and the Louisiana Transportation Research Center through a grant from the Federal Highway Administration. I am grateful for the assistance they have provided.

This work is dedicated to my parents, Mr. and Mrs. Venci and Teresa Rebello, my sister, Lorraine Rebello, and my fiancée, Julie Cavanaugh, for all their prayers, support and encouragement.
# Contents

Acknowledgements .......................................................... ii
List of Tables ................................................................. vi
List of Figures ................................................................. vii
Notation ............................................................................. ix
Abstract .............................................................................. xiv

1. Introduction ........................................................................ 1
   1.1 General Remarks .......................................................... 1
   1.2 Previous Work ............................................................ 2
      1.2.1 Experimental Investigations ..................................... 2
      1.2.2 Analytical Procedures ............................................ 3
      1.2.3 Temperature Effects ................................................ 3
      1.2.4 Creep and Shrinkage Studies .................................... 4
      1.2.5 Finite Element Analyses ........................................... 5
   1.3 Objectives and Scope .................................................... 6

2. Finite Element Modelling Of Bridges ................................. 8
   2.1 Finite Element Displacement Formulation ....................... 8
   2.2 Choice of Element ....................................................... 10
      2.2.1 Numerical Integration ............................................. 17
      2.2.2 Stress Extrapolation ............................................... 19

3. Time-Dependent Behavior of Structural Concrete ............... 22
   3.1 General Remarks ....................................................... 22
   3.2 Components of Deformation in Concrete ....................... 22
      3.2.1 Instantaneous Elastic Strain .................................... 22
      3.2.2 Creep Strain ....................................................... 24
      3.2.3 Shrinkage strain .................................................. 26
   3.3 Prediction of Material Properties .................................... 27
      3.3.1 ACI Committee 209 Recommendations ..................... 28
      3.3.2 CEB-FIP Recommendations .................................... 33
      3.3.3 Bažant-Panula Model II (BP2) ................................ 40
List of Tables

3.1 Member Size Correction Factor ........................................... 30
3.2 Thickness Correction Factor for Shrinkage ............................. 32
3.3 Constants a and b for use in equation (3.22) .......................... 33
3.4 Coefficients of creep for use in equations (3.29) and (3.30) .... 35
3.5 Time delay factor for creep flow ....................................... 35
3.6 Shrinkage coefficient $\varepsilon_{sh,1}$ ........................................ 40
6.1 Comparisons with Sinno-Furr Experiment ............................ 97
6.2 Midspan Deflections - Heuristic Girder-Slab System ............... 107
7.1 Expressions for $C_{J M}$ for Type III Girder-Slab Systems ........ 131
7.2 Expressions for $C_{J M}$ for Type IV Girder-Slab Systems .......... 132
7.3 Joint Openings due to Creep and Shrinkage (Span Length = 70') 142
7.4 Joint Openings for Systems with Type IV Girders (Span Length = 95') .......................................................... 143
7.5 Temperatures from Profiles P1 and P2 ............................... 146
7.6 Comparison of Movements using Profile P2 and the LaDOTD Procedure .................................................. 149
A.1 Data Files for PCBRIDGE .................................................. 168
B.1 Data Files for MESHGEN ............................................... 274
List of Figures

2.1 Typical Bridge Cross Section .......................................................... 12
2.2 Configuration of Elements and Nodes .................................................. 13
2.3 20-Node Isoparametric Element .......................................................... 14
2.4 Sampling Points for Rule 15b ............................................................... 18
3.1 Components of Deformation in Concrete .............................................. 23
3.2 Creep at Different Ages of Loading ..................................................... 25
3.3 Definitions of the Creep Function and Creep Coefficient ...................... 29
3.4 Coefficient $\phi_f$ for the flow component of creep .............................. 36
3.5 Coefficient $\epsilon_{sh2}$ for shrinkage .................................................... 38
3.6 Development function for shrinkage (CEB-FIP Recommendations) .......... 39
3.7 Time-shift Principle for Temperature Variation ..................................... 57
3.8 Assumed Temperature Profile on Composite Section ............................ 63
4.1 Position of Tendon Segment in an Element ......................................... 67
4.2 Definition of Tendon Profile ............................................................... 68
4.3 Typical Stress-Strain Curve for Prestressing Steel ............................... 72
4.4 Stress Relaxation under Varying Prestress ......................................... 79
5.1 Concrete Prism at Release of Prestress ................................................. 90
6.1 Sinno-Furr Girder: Elevation, Cross section and Mesh Configuration ...... 96
6.2 Midspan Camber versus Time for the Sinno-Furr Girder ....................... 100
6.3 Prestress Loss at Midspan versus Time for the Sinno-Furr Girder .......... 101
6.4 Creep coefficients for Sinno-Furr Girder Concrete for a loading age of 1 day ........................................................................................................... 102
6.5 Shrinkage strains for Sinno-Furr Girder Concrete .................................. 103
6.6 Cross section of Heuristic Girder-Slab System ..................................... 105
6.7 Midspan Deflections for the Heuristic Girder-Slab System .................... 108
6.8 Creep Coefficients for Girder Concrete ............................................... 109
6.9 Creep coefficients for Slab Concrete .................................................... 110
6.10 Shrinkage strains for Girder Concrete ................................................ 111
6.11 Shrinkage strains for Slab Concrete .................................................... 112
6.12 Elevation of East Approachway of the Krotz Springs Bridge ............... 115
6.13 Analytical and Experimental Results for Expansion Joint 1 ................. 116
6.14 Analytical and Experimental Results for Expansion Joint 2 ................. 117
7.1 $C_{JM}$ for Single-span Type III Girder and Slab Systems ....................... 123
7.2 $C_{JM}$ for Two-span Type III Girder and Slab Systems .......................... 124
7.3 $C_{JM}$ for Three-span Type III Girder and Slab Systems ......................... 125

vii
7.4 $C_{JM}$ for Four-span Type III Girder and Slab Systems .......... 126
7.5 $C_{JM}$ for Single-span Type IV Girder and Slab Systems .......... 127
7.6 $C_{JM}$ for Two-span Type IV Girder and Slab Systems .......... 128
7.7 $C_{JM}$ for Three-span Type IV Girder and Slab Systems ......... 129
7.8 $C_{JM}$ for Four-span Type IV Girder and Slab Systems .......... 130
7.9 $C_{JM}$ v/s No. of Continuous Spans for Systems with Type IV Girders
          using the ACI-209 Model .............................................. 134
7.10 $C_{JM}$ v/s No. of Continuous Spans for Systems with Type IV Girders
              using the BP2 Model ................................................ 135
7.11 $C_{JM}$ v/s No. of Continuous Spans for Systems with Type IV Girders
              using the CEB-FIP Model ........................................... 136
7.12 $C_{JM}$ for Systems with Medium Strength Concrete Type IV Girders
              using the ACI-209 Model ............................................ 137
7.13 $C_{JM}$ for Systems with Medium Strength Concrete Type IV Girders
              using the BP2 Model ................................................ 138
7.14 $C_{JM}$ for Systems with Medium Strength Concrete Type IV Girders
              using the CEB-FIP Model ............................................ 139
7.15 Temperatures for Profile P2 ............................................. 145
7.16 Effect of Temperature Profiles P1 and P2 on Movement at Joints .. 147
A.1 Flow Diagram for Program P ChrisBridge ............................ 169
B.1 Finite Element Mesh and Tendon Generation Example .............. 264
Notation

The following symbols are used in the text:

- $A_c = \text{cross sectional area of a concrete member}$;
- $A_s = \text{cross sectional area of prestressing steel}$;
- $a_i(t') = \text{aging coefficient in a Dirichlet series}$;
- $[B] = \text{strain displacement matrix for an element}$;
- $< B_s > = \text{strain displacement matrix for a prestressing strand segment}$;
- $C_{JM} = \text{coefficient of joint movement}$;
- $C(t, t') = \text{specific creep function}$;
- $C_b(t, t') = \text{specific creep function for basic creep in the BP2 recommendations}$;
- $C_d(t, t', t_{sh.o}) = \text{specific creep function for drying creep in the BP2 recommendations}$;
- $[D], [D_n], [D_o] = \text{three-dimensional elasticity matrix}$;
- $[EX] = \text{stress extrapolation matrix}$;
- $E, E(t), E_c, E_n = \text{concrete modulus of elasticity}$;
- $E'_s = \text{difference in modulii of elasticity of prestressing steel and concrete}$;
- $E(28) = \text{modulus of elasticity of concrete at an age of 28 days}$;
- $ER = \text{least squares error}$;
- $\varepsilon_i^n = \text{hidden state variables for the accumulation of the stress history up to time } t_n$;
- $\{F\} = \text{vector of body forces in an element}$;
- $\{F_p\} = \text{nodal body force vector}$;
- $\{F_y\} = \text{nodal surface traction force vector}$;
- $\{F_{c_i}\} = \text{nodal force vector due to initial strains}$;
- $\{F_{\sigma_o}\} = \text{nodal force vector due to initial stresses}$;
\{f\} = \text{vector of surface traction forces on an element;}

\[ f'_{28} = 28\text{-day concrete compressive strength;} \]

\[ f'(t) = \text{concrete compressive strength at time } t; \]

\[ f_i = \text{initial stress in prestressing steel;} \]

\[ f_s = \text{stress in prestressing steel;} \]

\[ f_y = 0.1\% \text{ offset yield stress for prestressing steel;} \]

\([J]\) = \text{Jacobian matrix for the 3-D 20-node isoparametric element;}

\[ J^* = \text{Jacobian of transformation from the } S_e\text{-coordinate system to the } \xi\text{-coordinate system;} \]

\[ J(t,t') = \text{creep function or creep compliance;} \]

\([K]\) = \text{structure stiffness matrix;}

\([K_s]\) = \text{prestressing strand stiffness matrix contribution to the parent element;}

\[ L = \text{span length of girders in feet;} \]

\[ l,m,n = \text{direction cosines of the tangent to a point on a prestressing strand segment;} \]

\[ l_c = \text{time length of concrete curing;} \]

\([N]\) = \text{matrix of shape functions for the 3-D isoparametric element;}

\{P\} = \text{load vector for external forces;}

\{Q_s\} = \text{vector of nodal loads on the parent element due to stress in a prestressing strand segment;}

\{R\} = \text{external load vector;}

\{r\} = \text{nodal displacement vector;}

\((r,s,t) = \text{normalized local coordinates of a point in a 3-D isoparametric element;} \]

\((r^*,s^*,t^*) = \text{normalized local coordinates of a point on a prestressing strand segment;} \]

\[ S = \text{element surface;} \]
$S$ = curvilinear coordinate along a prestressing tendon segment in the global reference frame;

$T$ = temperature;

$t$ = time or age of concrete;

$t'$ = loading age of concrete;

$t_e$ = effective age of concrete;

$t_h$ = time at which the concrete slab hardens;

$t_n$ = $n^{th}$ time step;

$t_{sh.o}$ = time at which drying of concrete commences;

$U_a$ = activation energy of concrete creep;

$U_h$ = activation energy of hydration;

$\{u\}$ = displacement vector at a point in a 3-D isoparametric element;

$u, v, w$ = displacements in the $x$-, $y$- and $z$- directions;

$V$ = element volume;

$(x, y, z)$ = global coordinates of a point in a 3-D isoparametric element;

$(x^*, y^*, z^*)$ = global coordinates of a point on a prestressing strand segment;

$\alpha$ = coefficient of thermal expansion of concrete;

$\beta_a(t')$ = rapid initial strain component of creep in the CEB-FIP recommendations;

$\beta_d(t - t')$ = delayed elastic strain component of creep in the CEB-FIP recommendations;

$\beta_f(t)$ = flow component of creep strain in the CEB-FIP recommendations;

$\gamma_{xy}, \gamma_{yz}, \gamma_{xz}$ = shear strains on the $xy$-, $yz$- and $xz$-planes;

$\Delta$ = prefix denoting an increment;

$\Delta_{cst}$ = joint movement due to creep, shrinkage and temperature;

$\Delta_i$ = initial girder midspan deflection;
\( \Delta J_M = \) joint movement 2 years after prestress release;

\( \Delta \varepsilon_n^A = \) correction strain to account for discrete jumps in the variation of the modulus of elasticity of concrete;

\( \Delta \sigma_d = \) change in stress in a prestressing strand segment due to deformation;

\( \Delta \sigma_s = \) change in stress in a prestressing strand segment due to steel relaxation;

\( \Delta \sigma_a = \) change in stress in a prestressing strand segment;

\( \delta = \) prefix denoting a virtual quantity;

\( \{ \varepsilon \} = \) strain vector;

\( \{ \varepsilon_0 \} = \) initial strain vector;

\( \{ \varepsilon_n \} = \) total creep strain vector at time \( t_n \);

\( \{ \varepsilon'_n \} = \) total pseudo-inelastic strain vector at time \( t_n \);

\( \varepsilon_c = \) creep strain;

\( \varepsilon_E, \varepsilon_e = \) instantaneous elastic strain;

\( \varepsilon^o = \) stress-independent inelastic strain;

\( \varepsilon_s = \) tangential strain at a point on a prestressing strand segment;

\( \varepsilon_{sh} = \) shrinkage strain;

\( \varepsilon_{sh\infty} = \) ultimate shrinkage strain;

\( \varepsilon_T, \varepsilon_T = \) thermal strain;

\( \varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz} = \) normal strains in the \( x-, y-, \) and \( z- \) directions;

\( \varepsilon_\sigma = \) stress-produced strain;

\( \lambda_i = \) inverse retardation times in a Dirichlet series;

\( \nu = \) Poisson's ratio;

\( \xi = \) normalized coordinate along a prestressing strand segment;

\( \Sigma = \) summation symbol;

\( \{ \sigma \} = \) stress vector;
\{ \sigma_0 \} = \text{initial stress vector;}

\sigma_{xx}, \sigma_{yy}, \sigma_{zz} = \text{normal stresses in the x-, y- and z-directions;}

\tau_{xy}, \tau_{yz}, \tau_{xz} = \text{shear stresses on the xy-, yz- and xz-planes;}

\Phi(T) = \text{temperature time-shift function;}

\phi(t, t') = \text{creep coefficient;}

\phi_b(t, t') = \text{creep coefficient for basic creep in the BP2 recommendations;}

\phi_d(t, t', t_{sh, o}) = \text{creep coefficient for drying creep in the BP2 recommendations;}

\phi_{28}(t, t') = \text{creep coefficient defined at a loading age of 28 days;}

\phi_\infty = \text{ultimate creep coefficient;}

< \psi > = \text{row vector of shape functions for nodes on a prestressing strand segment;}

\{ \} = \text{column vector;}

<> = \text{row vector;}

[ ] = \text{matrix;}

[ ]^{-1} = \text{inverse of a matrix; and}

[ ]^T = \text{transpose of a matrix.}
Abstract

A rigorous and efficient analytical model to predict the long-term deformation behavior of bridges with multiple, precast, pretensioned, prestressed concrete girders supporting cast-in-place concrete deck slabs, was developed. The analytical procedure uses the finite element method with three-dimensional 20-node isoparametric elements to realistically model bridge geometry. Time dependent effects due to load and temperature history, creep, shrinkage and aging of concrete are included in the analysis. Creep and shrinkage strains are evaluated at different times using the more commonly used procedures, namely, the ACI-209, Bažant-Panula II and CEB-FIP procedures. Temperature strains are calculated from an assumed typical bridge temperature distribution based on the average temperature occurring during any time period. The effect of temperature on creep is also accounted for. Prestressing tendons are modelled as being embedded in concrete and as contributing to girder stiffness. Position continuity in tendon profiles is maintained. Losses in prestress due to steel relaxation and geometry changes are calculated in the analysis. The analytical model is capable of simulating typical construction schedules to predict deformations at any stage during the service life of a bridge.

A parametric study was conducted to quantify the influence of key geometric and material properties of the bridge on the long-term expansion joint movements. Bridge systems representing a wide range of key parameters were analyzed to develop formulas to estimate creep and shrinkage movements with a certain degree of confidence. These formulas formed the basis of a rational procedure for calculating the long-term bridge deck movements. The recommended procedure accounts for the effects of bridge geometry and material properties on joint movements. These effects are ignored in current highway bridge deck joint design methodology. The use of the recommended procedure permits the designer to determine span lengths and the maximum number of continuous spans between expansion joints in bridge decks, if the limit of movement that can be accommodated by the chosen joint sealing system is known.
The analytical model has been coded into a FORTRAN program which can be used to evaluate the long-term behavior of bridges with or without expansion joints, and with different support conditions.
Chapter I
Introduction

1.1 General Remarks

Bridges with cast-in-place concrete decks supported by precast pretensioned girders are commonly encountered in the highway systems in the United States. These bridges have been overlooked for a full-range time-dependent analysis. A direct consequence of this apparent neglect is that these bridges are often plagued with the poor performance of deck expansion joint sealing systems. These performance problems could be rectified if accurate predictions of bridge deformations are made throughout the service life of the structure.

An analytical determination of bridge deformations is complicated by the time-dependent phenomena of creep and shrinkage which, along with temperature variations, are responsible for longitudinal movements. Added to this, the usual construction process results in the concrete deck slab shrinking at a rate that is different from that of the girders. This causes differential shrinkage between the two components which gives rise to longitudinal stresses and movements.

Engineers in the past have been relying mainly on empirical formulas to ensure that their bridge designs satisfy serviceability requirements. The present availability of high-speed computation facilities and modern numerical analysis techniques like the finite element method, will hopefully mitigate this reliance on experiments. In this study, an analytical model is developed for estimating long-term movements for the bridge type mentioned. Emphasis is placed on using this procedure to make proper and rational bridge deck joint designs.
1.2 Previous Work

In this section a brief review of previous studies pertaining to the different aspects of bridge analysis is listed. The main areas of research significant to the present analysis could be classified broadly as experimental investigations, simplified analytical procedures, temperature effects, creep and shrinkage studies and time-dependent finite element analyses.

1.2.1 Experimental Investigations

Many experimental studies on actual bridge movements have been conducted. One research record by Moulton (1983) lists data on bridge movements from 314 highway bridges in the U.S. and Canada. He concluded that horizontal movements are generally more damaging to bridge superstructures than vertical movements and that relatively small horizontal movements can cause significant damage to expansion joints. A study of movements in concrete bridges in the U.K. has been performed by Emerson (1979). In that study measurements of longitudinal movements due to temperature changes were made. Temperatures in the bridge superstructure were measured using thermocouples for various types of bridges. Results discussed include measured values of daily and annual ranges of movements, the seasonal effect on the measured movements, and short and long term values of the coefficient of thermal expansion of the deck. Also shown is how the knowledge of deck temperatures, which are derived from ambient temperatures, can be used to predict daily and annual ranges of movement. The main causes of longitudinal movement were identified as temperature changes, creep and shrinkage. It was concluded that an estimate of the extreme range of movement likely to occur during the life of a bridge could be arrived at if the extreme values of the shade temperature are known.
1.2.2 Analytical Procedures

Various analytical procedures have been developed to calculate deflections in prestressed concrete members due to time-dependent creep and shrinkage and due to short-term loading. These procedures are too numerous to describe here and are too cumbersome to apply to bridge structures which have more than one span. Recently Moustafa (1986) presented an iterative procedure for analyzing composite sections for time-dependent effects. This method is based on the residual strain concept. Strain compatibility and an iterative solution of equilibrium equations are used to compute strains and stresses at discrete cross sections along a structural member. Time-dependent strains resulting from creep, shrinkage and prestressing steel relaxation are reflected in the analysis as residual strains. This procedure requires the use of a computer as it is based on iteration. An extensive overview of analytical methods for stresses and deformations is available in a book by Ghali and Favre (1986).

1.2.3 Temperature Effects

The effects of varying temperatures on bridge movements are most apparent. Besides the well-known temperature effects of expansion and contraction, temperatures have an effect on creep strains. The problem of predicting temperature distributions over bridge cross sections is a formidable one. Nonlinear variations of temperature over a cross section result in complex stress patterns. In a paper by Elbadry and Ghali (1983) a method is presented to predict the temperature distribution over bridge cross sections from data related to their geometry, location, orientation, material and climatological conditions. This method uses the finite element procedure to solve the heat flow equation to determine the temperature variation at any time. The use of this procedure, however, becomes extremely time-consuming in an analysis that spans the life of a bridge. At the cost of loss of accuracy, it is more viable to assume a temperature distribution based on ambient temperatures.
A report by the National Cooperative Research Program (1985) contains the findings of a comprehensive study of thermally induced stresses in reinforced and prestressed concrete bridge superstructures. The temperature profiles on bridge cross sections suggested by various international design codes are presented. The profile proposed by Houdshell et al. (1972) is recommended in the report for U.S. bridges based on the fact that the profile was determined from the experimental investigation of one bridge. In the present study, a profile proposed by the Committee on Loads and Forces (1981) is employed as it is based on a wider sampling of experimental data.

1.2.4 Creep and Shrinkage Studies

Creep and shrinkage strains play a major role in the time-dependent analysis of concrete structures. In prestressed concrete girders, prestress losses are significantly affected by creep and shrinkage. Much work has been done in the last two decades to investigate the nature of creep and shrinkage in concrete. Neville et al. (1983) discuss the various theories of creep and review them against the background of observed influences and factors. The mathematical modelling of creep and shrinkage is described extensively by Bažant (1982). In a time-dependent analysis of a bridge, it becomes necessary to know the magnitudes of creep and shrinkage strains at any age in the life of the structure. One way to obtain this knowledge is through actual creep and shrinkage tests. Another way is to predict these strains based on the composition of concrete and one area of research is focussed on the development of models to achieve this. The most widely used models in practice are the ACI 209 Method (American Concrete Institute, Committee 209, (1982)), the CEB-FIP Method (CEB-FIP, (1978)) and the Bažant-Panula II Model (Bažant and Panula, (1980)). A description of these three models will be given in Chapter 3.
1.2.5 Finite Element Analyses

The finite element method has been applied to time-dependent analyses for a wide variety of structures. Scordelis (1984) has reviewed these analyses wherein planar or three-dimensional rigid frames, panels or slabs, thin shells and three-dimensional solids are considered. A survey of available software for precast and prestressed concrete structures is listed in a paper by Nasser (1987). Researchers in the field of Nuclear Engineering have accounted for many of the earliest time-dependent finite element analyses of concrete structures. Zienkiewicz et al. (1971) first analyzed prestressed concrete reactor vessels for the effects of creep and shrinkage. Three-dimensional isoparametric elements were used to model the concrete and prestressing strands were modelled by linear bar elements. The initial strain method was employed for the time-step analysis and creep functions were expressed as exponential functions based on the work done by Zienkiewicz and Watson (1966).

Kang (1977) used layered plane-stress elements to analyze prestressed concrete frames. The creep strain at any time was evaluated using an age and temperature dependent integral formulation of the specific creep function which was first developed by Kabir (1976). Prestressing steel tendons were approximated by a series of steel segments. Each segment was assumed straight and as having a constant force along its length. Strains in the segments were evaluated from changes in the segment length. Van Zyl (1978) extended this analysis to curved segmentally erected prestressed concrete box girder bridges. He employed a finite element developed by Bazant and ElNimeiri (1974) to represent the girder cross section. Both pretensioning and post-tensioning for the prestressing steel were considered. Van Greunen and Scordelis (1983) further used the analysis for prestressed concrete slabs employing 15 degree-of-freedom flat triangular shell elements. The three analyses just described used the ACI 209 Method to evaluate creep and shrinkage strains.

El-Shafey et al. (1982) performed the analysis of post-tensioned precast single tee girders supporting cast-in-place concrete slabs and evaluated deflections at various times. Creep and shrinkage strains were predicted using the ACI 209 and
CEB-FIP Methods. The creep strain formulation assumed a constant rate of creep flow, thus simplifying the age dependency of creep. Only single tee beams were analyzed after determining an effective flange width for the slab-girder composite section. Eight-node quadrilateral elements were used to model the girder and the slab. A conclusion was reached that a comparison of deflections from the analysis with actual measured deflections was adequate.

References to other works are noted as the development of this report dictates.

1.3 Objectives and Scope

This research project has three main objectives. The first objective is to define a finite element analysis exclusive to bridge structures consisting of cast-in-place concrete decks supported by multiple precast pretensioned concrete girders. This analysis takes into account time-dependent effects, prestressing forces and losses, temperature effects and construction schedules. The aim is to model the bridge prototype as closely as possible with respect to material properties, response to environmental conditions and elastic response to imposed loads.

The second objective is to present this analysis in a usable format to aid in analysis and design applications as well as for future research. This objective is realized through the writing and testing of a FORTRAN program. The use of this program permits the designer to obtain reasonable estimates of deformations of the bridge he is considering.

The third objective is the utilization of the analytical model to perform parametric studies that result in design aids for predicting deck joint movements. These aids provide the designer with realistic values for bridge movements based on bridge cross sections, material properties, span lengths and support conditions.

The scope is limited to the bridge type mentioned above. Cross sectional dimensions can vary along the span. The bridge can be arbitrarily curved or straight and can be supported in any manner. The structure is treated as three-dimensional.
The bridge can be analyzed for prestressing loads. Only pretensioning is considered. Losses in prestress can be calculated at any time. The time-dependent effects of creep and shrinkage are included and their strains are derived from the ACI 209, CEB-FIP and Bažant-Panula II models. Temperature loading is considered on a long-term basis. Diurnal fluctuations of temperature are neglected. The effect of temperature on creep is accounted for.

The completed bridge structure is analyzed for its response to loads already imposed during construction stages, due to environmental conditions or as live loads. Loads can be applied either at nodes or as distributed loads on elements in the model. All through the analysis, the emphasis is placed on overall bridge behavior and local behavior at expansion joint lines.
The finite element method is chosen to perform the analysis of bridge structures consisting of cast-in-place concrete decks supported by multiple precast pretensioned concrete girders. In the finite element method a continuum with an infinite number of unknowns is approximated as an assemblage of elements having a finite number of unknowns. Literature abounds in texts written on the finite element method. Examples of these include Zienkiewicz (1977) and Cook (1981). A brief description of the finite element displacement formulation is given in the next section and is the approach used in this study.

2.1 Finite Element Displacement Formulation

The displacement formulation of the finite element method can be broken into steps as follows:

1. The elastic continuum is discretized by a number of elements. The geometry of these elements is characterized by nodes at which displacements are sought. The displacement vector of an element contains the displacements at all nodes in the element and is denoted by \( \{r\} \).

2. A matrix \([N]\) of interpolation functions (also known as shape functions) approximates the displacement vector \(\{u\}\) at any point within an element. The shape functions relate \(\{u\}\) to the nodal displacement vector \(\{r\}\) as

\[
\{u\} = [N]\{r\} \tag{2.1}
\]

3. Compatibility relationships are used to define the strain vector \(\{\epsilon\}\) at any point within an element. \(\{\epsilon\}\) is related to \(\{u\}\) by

\[
\{\epsilon\} = [B]\{u\} \tag{2.2}
\]
where \([B]\) is the strain-displacement matrix.

4. The constitutive relationship for an element is expressed as

\[
\{\sigma\} = [D](\{\epsilon\} - \{\epsilon_0\}) + \{\sigma_0\}
\]

where \([D]\) = elasticity matrix

\(\{\epsilon_0\}\) = initial strain vector

\(\{\sigma\}\) = stress vector

\(\{\sigma_0\}\) = initial stress vector

5. The principle of virtual work is applied to the discretized continuum to obtain the equilibrium equation as

\[
\{\delta \epsilon\}^T (\{P\} + \sum_e \int_V \{\delta u\}^T \{F\} \, dV + \sum_e \int_S \{\delta u\}^T \{f\} \, dS) = \sum_e \int_V \{\delta \epsilon\}^T \{\sigma\} \, dV
\]

where \(\{P\}\) = load vector for external forces

\(\{F\}\) = body force vector

\(\{f\}\) = surface traction vector

\(V\) = element volume

\(S\) = element surface

\(\sum_e\) = summation over all elements

\(\{\delta \epsilon\}\) = virtual nodal displacement vector

\(\{\delta u\}\) = virtual element displacement vector

\(\{\delta \epsilon\}\) = virtual element strain vector

Substitution of equations (2.1), (2.2) and (2.3) into equation (2.4) results in

\[
\{\delta \epsilon\}^T \{\sigma\} - \sum_e \int_V [B]^T \{\delta \epsilon\} \, dV + \sum_e \int_V [B]^T \{\epsilon\} \, dV
\]

Since \(\{\delta \epsilon\}\) is arbitrary, equation (2.5) can be rewritten as

\[
\{P\} + \sum_e \int_V [N]^T \{F\} \, dV + \sum_e \int_S [N]^T \{f\} \, dS
\]

\[
= \sum_e \int_V [B]^T [D][B] \{\epsilon\} \, dV - \sum_e \int_V [B]^T [D][\epsilon_0] \, dV
\]

\[
+ \sum_e \int_V [B]^T \{\sigma_0\} \, dV
\]
An external load vector \( \{ R \} \) can be defined as

\[
\{ R \} = \{ P \} + \{ F_P \} + \{ F_f \} + \{ F_{e_0} \} + \{ F_{\sigma_0} \} \tag{2.7}
\]

where

\[
\{ F_P \} = \sum_e \int_V [N]^T \{ F \} dV = \text{Nodal Body Force Vector} \tag{2.7a}
\]

\[
\{ F_f \} = \sum_e \int_S [N]^T \{ f \} dS = \text{Nodal Surface Traction Force Vector} \tag{2.7b}
\]

\[
\{ F_{e_0} \} = \sum_e \int_V [B]^T [D] \{ \epsilon_0 \} dV = \text{Nodal Force Vector due to Initial Strains} \tag{2.7c}
\]

\[
\{ F_{\sigma_0} \} = - \sum_e \int_V [B]^T \{ \sigma_0 \} dV = \text{Nodal Force Vector due to Initial Stresses} \tag{2.7d}
\]

The structural stiffness matrix is given by

\[
[K] = \sum_e \int_V [B]^T [D] [B] dV \tag{2.8}
\]

Using equations (2.7) and (2.8), equation (2.6) can be rewritten as the classical force-displacement relationship (stiffness form) as follows:

\[
\{ R \} = [K] \{ r \} \tag{2.9}
\]

6. Equation (2.9) is solved to yield the unknown nodal displacement vector \( \{ r \} \). Strains and stresses are then computed in any element from equations (2.2) and (2.3). Thus the solution of the elastic analysis problem is completed.

### 2.2 Choice of Element

A typical cross section of bridges with precast, pretensioned concrete girders supporting cast-in-place concrete deck slabs is shown in Figure 2.1. These bridges
pose a problem as far as their finite element representation is concerned. A true reproduction of their complex geometry can only be achieved with the use of three-dimensional elements; especially so for the case of curved superstructures. In the present analysis, the use of three-dimensional quadratic isoparametric elements is made to model both girders and the slab. The choice of such elements allows for a realistic simulation of the interaction between slab and girder and the representation of curved geometries without tedious geometric transformations. Figure 2.2 shows a viable configuration of these elements for a single girder and slab structure.

A brief description of the 20-node 3-dimensional isoparametric finite element is given here. Figure 2.3 shows such an element. Points within the element are described in terms of a normalized set of curvilinear coordinates \( r, s, t \). The element shape functions are described in terms of these coordinates and are as follows (Bathe and Wilson, (1976)) for each node.

\[
\begin{align*}
N_1 & = g_1 - (g_0 + g_{12} + g_{17})/2 \\
N_2 & = g_2 - (g_0 + g_{10} + g_{18})/2 \\
N_3 & = g_3 - (g_{10} + g_{11} + g_{19})/2 \\
N_4 & = g_4 - (g_{11} + g_{12} + g_{20})/2 \\
N_5 & = g_5 - (g_{13} + g_{16} + g_{17})/2 \\
N_6 & = g_6 - (g_{13} + g_{14} + g_{18})/2 \\
N_7 & = g_7 - (g_{14} + g_{15} + g_{19})/2 \\
N_8 & = g_8 - (g_{15} + g_{16} + g_{20})/2 \\
N_j & = g_j
\end{align*}
\]

and \( g_i = G(r, r_i)G(s, s_i)g(t, t_i); \ i = 1, \ldots, 20 \)

where \( G(h, h_i) = \frac{1}{2}(1 + hh_i); \) for \( h_i = \pm 1; \ h = r, s, t \)

\( G(h, h_i) = (1 - h^2); \) for \( h_i = 0 \)

The global coordinates \( (x, y \text{ and } z) \) of any point within the element are related
Figure 2.1: Typical Bridge Cross Section
Figure 2.2: Configuration of Elements and Nodes
Figure 2.3: 20-Node Isoparametric Element
to the global coordinates of the nodes by

\[ x = \sum_{i=1}^{20} N_i x_i \]
\[ y = \sum_{i=1}^{20} N_i y_i \]
\[ z = \sum_{i=1}^{20} N_i z_i \]  

(2.11)

where \( i \) is the node number. In an isoparametric formulation, the displacement vector is related to nodal displacements in the same manner as the geometry and hence

\[ \{u\} = \begin{\{u \ v \ w \} \end{\} \]  

(2.12)

where

\[ u = \sum_{i=1}^{20} N_i u_i \]
\[ v = \sum_{i=1}^{20} N_i v_i \]  

(2.12a)
\[ w = \sum_{i=1}^{20} N_i w_i \]

\( u, v, w \) are displacements in the \( x, y \) and \( z \) directions respectively and \( u_i, v_i \) and \( w_i \) are the corresponding displacements of node \( i \).

Equation (2.2) is the strain-displacement relationship and relates strains at any point in the element to the displacement vector \( \{u\} \) at that point. The strain vector for the 3-D element is given as

\[ \{\epsilon\}^T = \begin{\{\epsilon_{xx} \ \epsilon_{yy} \ \epsilon_{zz} \ \gamma_{xy} \ \gamma_{yz} \ \gamma_{zx} \} \end{\} \]  

(2.13)

where \( \epsilon \) denotes normal strains and \( \gamma \) are shearing strains. These strains are related to displacements as follows:

\[ \epsilon_{xx} = u_x \]
\[ \epsilon_{yy} = v_y \]
\( \varepsilon_{zz} = w_{sz} \)
\( \gamma_{xy} = u_{y} + v_{x} \)
\( \gamma_{yz} = v_{z} + w_{y} \)
\( \gamma_{xz} = u_{z} + w_{x} \)

(2.14)

where \( u_{sz} \) represents the partial derivative of \( u \) with respect to \( x \) and so on.

The strain-displacement matrix \([B]\) is set up using equation (2.14). It can be seen that strains are related to displacements which are in turn related to nodal displacements through the use of shape functions. However, the shape functions are defined in terms of the normalized coordinates \( r, s \) and \( t \) and from equations (2.12) and (2.14) it is apparent that the derivatives of the shape functions are needed with respect to the \( x, y, z \) coordinate system. To achieve this, a \( 3 \times 3 \) matrix called the Jacobian matrix is defined to perform the transformation from the \( x, y, z \) system to the normalized \( r, s, t \) system of coordinates. The inverse of this transformation is used to evaluate the strain displacement matrix \([B]\). The Jacobian matrix is defined as

\[
[J] = \begin{bmatrix}
x_{rr} & y_{rr} & z_{rr} \\
x_{rs} & y_{rs} & z_{rs} \\
x_{rt} & y_{rt} & z_{rt}
\end{bmatrix}
\]

(2.15)

where as is evident from equation (2.11), \( x, y \) and \( z \) are functions of \( r, s \) and \( t \).

In three-dimensional elasticity, the elastic matrix \([D]\) in equation (2.3), assuming isotropy, is given as

\[
[D] = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix}
1 & \frac{\nu}{(1-\nu)} & 0 & 0 & 0 \\
\frac{\nu}{(1-\nu)} & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & \frac{(1-2\nu)}{2(1-\nu)} & 0 \\
0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2(1-\nu)} \\
\end{bmatrix}
\]

(2.16)

where \( \nu \) is Poisson's ratio and \( E \) is the modulus of elasticity. The stress vector is written as
\( \{ \sigma \}^T = \{ \sigma_{xx} \sigma_{yy} \sigma_{zz} \tau_{xy} \tau_{yz} \tau_{xz} \} \) \hspace{1cm} (2.17)

which corresponds to the definition of the strain vector in equation (2.13).

2.2.1 Numerical Integration

The structural stiffness matrix \([K]\) defined in equation (2.8) involves an integration over the volume of every element. This integration is performed numerically. Let \(|J|\) represent the determinant of the Jacobian matrix defined in equation (2.15). The integrand of equation (2.8) consists of functions of the normalized coordinates \(\rho, s\) and \(t\). Therefore, the volume integration extends over the normalized coordinate volume, and the volume differential must be written in terms of the normalized coordinates. In general

\[
dV = dx dy dz = |J|dr ds dt
\]

Equation (2.8) now becomes

\[
[K] = \sum_e \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} [B]^T[D][B]|J|dr ds dt
\]

The integration in equation (2.19) has to be performed numerically for each element and this can be achieved by using one of a variety of numerical quadrature rules. A listing of these rules is given in a paper by Irons (1971). The rule adopted in this study is identified as Rule 15b in the paper mentioned above. The 15 sampling points used for the integration are distributed symmetrically over the element as depicted in Figure 2.3. Rule 15b is given as

\[
\int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} f(x, y, z)dx dy dz = Af(0, 0, 0) + B\{ f(-b, 0, 0) + f(b, 0, 0) + \ldots 6 \ terms \} + C\{ f(-c, -c, -c) + f(c, -c, -c) + \ldots 8 \ terms \}
\]

\hspace{1cm} (2.20)
Figure 2.4: Sampling Points for Rule 15b.
where \( A = 0.712137436; \) \( b = 0.848418011; \)
\( B = 0.686227234; \) \( c = 0.727662442; \)
\( C = 0.396312395 \)

This particular rule is chosen because it requires less computational time than the usual 3×3×3 Gauss rule. Also, it has been observed in this study that the 20-node element is excessively stiff if used with the 3×3×3 Gauss rule. The rule 15b scheme of integration has been employed successfully for 3-D concrete structures and thick shells by Sarne (1975), Buyukozturk and Shareef (1985) and by Cervera et al. (1986). Studies performed by Cervera (1986) on plates using 3-D 20-node elements along with the 15-point integration rule, indicate accurate results even for element aspect ratios as high as 25.

2.2.2 Stress Extrapolation

The element stresses described in equations (2.8) and (2.17) are evaluated at the sampling points for numerical integration. The least accurate points for stress evaluation are at the element nodes. Unfortunately, the nodes are the points at which stresses are most desired as output. Therefore, stresses need to be extrapolated to the nodes. A slightly modified version of the "local discrete smoothing of stresses" procedure given by Hinton et al. (1975) is used in this analysis.

Let \( \sigma(r, s, t) \) represent the stresses at the sampling points within the element. A function \( g(r, s, t) \) is sought such that it is an exact least squares fit to the selected values of \( \sigma(r, s, t) \). Let \( \bar{\sigma}_1, \bar{\sigma}_2, \ldots, \bar{\sigma}_8 \) be the extrapolated stresses at nodes 1 through 8 (Figure 2.2). The function \( g(r, s, t) \) can be defined as

\[
g(r, s, t) = \sum_{i=1}^{8} \bar{N}_i \bar{\sigma}_i \tag{2.21}
\]

Since stresses in a quadratic (parabolic) element have a linear distribution over the element, \( \bar{N}_i \) in equation (2.21) can be expressed as trilinear functions of \( r, s \) and \( t. \)
Consequently,

\[
\begin{align*}
\bar{N}_1 &= \frac{1}{8}(1 + r)(1 + s)(1 + t) \\
\bar{N}_2 &= \frac{1}{8}(1 - r)(1 + s)(1 + t) \\
\bar{N}_3 &= \frac{1}{8}(1 - r)(1 - s)(1 + t) \\
\bar{N}_4 &= \frac{1}{8}(1 + r)(1 - s)(1 + t) \\
\bar{N}_5 &= \frac{1}{8}(1 + r)(1 + s)(1 - t) \\
\bar{N}_6 &= \frac{1}{8}(1 - r)(1 + s)(1 - t) \\
\bar{N}_7 &= \frac{1}{8}(1 - r)(1 - s)(1 - t) \\
\bar{N}_8 &= \frac{1}{8}(1 + r)(1 - s)(1 - t) \\
\end{align*}
\] (2.22)

\(g(r, s, t)\) can now be found by minimizing the function \(\Psi\) defined as

\[
\Psi = \sum_{k=1}^{15} [\sigma(r_k, s_k, t_k) - g(r_k, s_k, t_k)]^2
\]

\[
= \sum_{k=1}^{15} [\sigma(r_k, s_k, t_k) - \sum_{i=1}^{8} \bar{N}_i(r_k, s_k, t_k)\bar{\sigma}_i]^2
\] (2.23)

To minimize \(\Psi\), set \(\frac{\partial \Psi}{\partial \bar{\sigma}_i} = 0\) for \(i = 1, \ldots, 8\). This yields the normal equations

\[
\sum_{k=1}^{15} \left[ \sum_{j=1}^{8} \sum_{i=1}^{8} \bar{N}_j \bar{\sigma}_i \bar{N}_i \right] = \sum_{k=1}^{15} \bar{N}_j \sigma_k
\] (2.24)

The \(j^{th}\) normal equation is

\[
\sum_{i=1}^{8} \left[ \sum_{k=1}^{15} \bar{N}_j(r_k, s_k, t_k)\bar{N}_i(r_k, s_k, t_k)\bar{\sigma}_i \right] = \sum_{k=1}^{15} \bar{N}_j(r_k, s_k, t_k)\sigma_k
\] (2.25)

Equation (2.25) can be written in matrix form for every \(j = 1, \ldots, 8\) as

\[
[S]\{\bar{\sigma}\} = [G]\{\sigma\}
\] (2.26)

where

\[
[S] = \begin{bmatrix}
\sum_{i=1}^{15} \bar{N}_i; \bar{N}_i & \cdots & \sum_{i=1}^{15} \bar{N}_i; \bar{N}_8i \\
\vdots & \ddots & \vdots \\
\sum_{i=1}^{15} \bar{N}_8i; \bar{N}_i & \cdots & \sum_{i=1}^{15} \bar{N}_8i; \bar{N}_8i
\end{bmatrix}
\] (2.27)
and $N_{1i} = N_1(r_i, s_i, t_i)$ with $i$ being the sampling point number.

$$\{\sigma\} = \left\{ \begin{array}{c} \sigma_1 \\ \vdots \\ \sigma_8 \end{array} \right\}$$  \hspace{1cm} (2.28)

$$[G] = \begin{bmatrix} N_1(r_1, s_1, t_1) & \cdots & N_1(r_{15}, s_{15}, t_{15}) \\ \vdots & \ddots & \vdots \\ N_8(r_1, s_1, t_1) & \cdots & N_8(r_{15}, s_{15}, t_{15}) \end{bmatrix}$$  \hspace{1cm} (2.29)

$$\{\sigma\}^T = \left\{ \sigma_1 \sigma_2 \cdots \sigma_{16} \right\}$$  \hspace{1cm} (2.30)

From equation (2.26), the stresses at the nodes can be calculated as

$$\{\sigma\} = [S]^{-1}[G]{\{\sigma\}}$$  \hspace{1cm} (2.31)

The matrices $[S]$ and $[G]$ consist of terms evaluated in the normalized $r$, $s$, $t$ system and hence are identical for every element. Thus, a constant stress extrapolation matrix $[EX]$ can be defined as

$$[EX] = [S]^{-1}[G]$$  \hspace{1cm} (2.32)

Premultiplying the stresses $\{\sigma\}$ at the sampling points by $[EX]$ yields nodal stresses irrespective of element shape. The stresses at the edge nodes (nodes 9 through 20) are obtained as averages of the stresses at the corner nodes of the particular edge.

Final stresses at a node are evaluated by taking the average of the stresses obtained from each element containing the node.
Chapter III
Time-Dependent Behavior of Structural Concrete

3.1 General Remarks

Concrete bridges exhibit time-dependent behavior which may significantly affect their serviceability. This behavior is due to the interaction of concrete with its environment, resulting in complex physical and chemical changes with time. It is therefore essential to investigate the long-term deformation behavior of concrete bridges and to attempt to ensure the satisfaction of serviceability criteria during the design life of the structure.

Among the phenomena that affect the time-dependent behavior of concrete bridges, the three most important are creep, shrinkage and temperature. These phenomena will now be defined. Creep and shrinkage, along with the instantaneous elastic strain on loading, form the three components of deformation of concrete. These components are assumed to be independent of each other in that they are additive. In this chapter the components of deformation and temperature induced strains will be discussed in detail.

3.2 Components of Deformation in Concrete

The three components of deformation in concrete for a specimen loaded at time \( t_o \) change with time. This is shown qualitatively in Figure 3.1.

3.2.1 Instantaneous Elastic Strain

The instantaneous elastic strain \( \varepsilon_E \) is the strain that occurs immediately upon load application. Its value depends on the value of the modulus of elasticity \( E \)
Figure 3.1: Components of Deformation in Concrete
of concrete, which is a function of the age of concrete. The modulus of elasticity increases with time and hence the elastic strain decreases as shown in Figure 3.1. The determination of the concrete modulus of elasticity and its change with time are generally made from the expressions proposed by the ACI Committee 209 (1982). These expressions are given in subsection 3.3.1.

### 3.2.2 Creep Strain

Creep is defined as the increase in strain with time under, and induced by, a constant sustained stress. The nature of creep is illustrated in Figure 3.1(a) which indicates that it increases at a decreasing rate. The creep strain could be broken up into two components: basic creep and drying creep. Basic creep is identified as the creep occurring when concrete is in hygral equilibrium (no moisture exchange) with its environment. Drying creep is the excess strain that occurs under conditions of drying.

The creep strain in concrete is influenced by a variety of factors. Some of the important factors are discussed below:

1. **Age at loading**: Creep strain is inversely proportional to the age of concrete at the time of application of loading. This is shown in Figure 3.2 in which typical specific compliance curves for different ages of loading are depicted. The specific compliance is the total stress-produced strain due to a unit sustained stress and is comprised of the instantaneous elastic strain and the creep strain. The decrease in instantaneous elastic strain and creep strain is attributed to the increase in the degree of hydration in the cement-water reaction with time.

2. **Stress/strength ratio**: For stress/strength ratios up to about 0.5 and for constant mix proportions and the same type of aggregate, creep is proportional to the applied stress and inversely proportional to the strength at the time of application of load. This has been verified in many experiments as reported
Figure 3.2: Creep at Different Ages of Loading
by Neville (1959). Above a stress/strength ratio of 0.6, severe internal microcracking occurs in concrete and creep is accelerated.

3. **Type of aggregate:** Aggregates influence creep by the restraining effect they impose on the free creep of cement paste. Studies by Davis and Davis (1931) indicate that creep decreases with the increase of the volume and the modulus of elasticity of the aggregate.

4. **Size and shape of specimen:** In general, creep decreases with an increase in the size of the specimen, but when the specimen thickness exceeds about 3 feet, no further effect is apparent. Hanson and Mattock (1966) performed tests which indicate that creep decreases with an increase in the volume/surface ratio of the specimen. The size effect is therefore an indirect one, involving the surface of the specimen.

5. **Ambient humidity:** Creep is generally inversely proportional to the ambient relative humidity. However creep is not affected by relative humidity if concrete has reached hygral equilibrium prior to loading. This was first observed by Troxell et al. (1958).

6. **Temperature:** A temperature increase has two mutually competing effects on creep. It increases the creep rate and also accelerates the hydration of cement. The effects of temperature on creep will be discussed in more detail in section 3.7.

### 3.2.3 Shrinkage strain

Volume changes that occur in concrete independently of externally imposed stresses and of temperature changes are termed as shrinkage. The primary cause of shrinkage is the loss of water from the concrete during drying. The inverse process of swelling is of little significance in practice. The shrinkage process starts at the surface of a concrete specimen and gradually penetrates into the center. This results in a nonuniform distribution of shrinkage known as differential shrinkage. In
concrete design and analysis, shrinkage is usually considered to be uniform. The most important factors that influence shrinkage are:

1. **Aggregate**: The larger the aggregate content in a concrete mix, the greater is the tendency to restrain cement paste from shrinking. Thus shrinkage is inversely proportional to the aggregate content.

2. **Water/cement ratio**: An increase in the water/cement ratio is equivalent to an increase in the potential for moisture loss. Shrinkage therefore increases with increasing water/cement ratios.

3. **Volume/surface ratio**: A larger surface area of a concrete specimen implies a greater drying potential. Hence shrinkage decreases with increasing volume/surface ratios.

4. **Ambient humidity**: Shrinkage decreases with an increase in the relative humidity of the ambient medium. This again is due to the fact that for high relative humidity, the drying potential is decreased.

### 3.3 Prediction of Material Properties

The performance of a time-dependent analysis requires the knowledge of creep and shrinkage strains at any time during the lifetime of the structure. The best source of these strains are from creep and shrinkage tests performed on the concrete used. As the availability of long-term creep and shrinkage data is rarely guaranteed in bridge projects, fair estimates of the properties needed for the analysis can be made using approximate procedures. Three reliable sources of material properties are the ACI Committee 209 (1982) recommendations, the CEB-FIP (1978) recommendations and the simplified model developed by Bažant and Panula (1980). The last model will be referred to as the BP2 model.

The above mentioned procedures will be discussed briefly in the following subsections. Prior to this, certain definitions must be made. The sum of the in-
stantaneous elastic strain and creep strain at observation time \( t \) caused by a unit sustained stress applied at age \( t' \) is called the creep function or creep compliance and is denoted by \( J(t, t') \). It is written as

\[
J(t, t') = \frac{1}{E(t')} + C(t, t')
\]

where \( E(t') = \) modulus of elasticity at age \( t' \)
\( C(t, t') = \) specific creep (creep strain/unit stress)

The creep coefficient is then defined as the creep strain/elastic strain ratio and is given by

\[
\phi(t, t') = C(t, t')E(t')
\]

Using the creep coefficient, equation (3.2) becomes

\[
J(t, t') = \frac{1}{E(t')} [1 + \phi(t, t')]
\]

Figure 3.3 depicts the above definitions.

### 3.3.1 ACI Committee 209 Recommendations

The American Concrete Institute, Committee 209, Subcommittee II (1982) recommends expressions for the prediction of creep and shrinkage strains, based on studies by Branson and Christiason (1971).

**Creep:** The creep coefficient of equation (3.2) is obtained by representing experimental creep curves of concrete as products of age and duration functions. The creep coefficient is given as

\[
\phi(t, t') = \frac{(t - t')^{0.8}}{10 + (t - t')^{0.8}} \phi_\infty(t')
\]

where \( \phi_\infty \) is the ultimate creep coefficient and is calculated from
Figure 3.3: Definitions of the Creep Function and Creep Coefficient

(a) Creep function.
(b) Creep coefficient.
\[ \phi_{\infty} = 2.35k_c^t k_h^c k_f^c k_s^c k_A^c k_{AC} \tag{3.5} \]

For ages of application of load greater than 7 days for moist curing, or greater than one to three days for steam curing, the correction factor \( k_a^c \) for age at application of load is estimated from

\[ k_a^c = \begin{cases} 
1.25t^{-0.118} & \text{for moist curing} \\
1.13t^{-0.094} & \text{for steam curing} 
\end{cases} \tag{3.6} \]

\( k_h^c \) is a correction factor that accounts for the size of the member. For members with average thickness less than 6”, values of \( k_h^c \) are given in Table 3.1.

Table 3.1: Member Size Correction Factor

<table>
<thead>
<tr>
<th>Average thickness (inches)</th>
<th>( k_h^c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.30</td>
</tr>
<tr>
<td>3</td>
<td>1.17</td>
</tr>
<tr>
<td>4</td>
<td>1.11</td>
</tr>
<tr>
<td>5</td>
<td>1.04</td>
</tr>
</tbody>
</table>

For values of average thickness greater than 6” and less than 12”,

\[ k_h^c = 1.14 - 0.023t \tag{3.7} \]

The recommendations suggest the use of the volume/surface ratio \((v/s)\) for members with average thicknesses exceeding 12" to calculate \( k_h^c \) as

\[ k_h^c = \frac{2}{3} \left[ 1 + 1.13e^{-0.64(v/s)} \right] \tag{3.8} \]

To account for the ambient relative humidity, the correction factor \( k_H^c \) is employed and is given by

\[ k_H^c = 1.27 - 0.0067H \tag{3.9} \]

where \( H \) is the percentage relative humidity.

To account for the composition of concrete, three factors \( k_c^c, k_f^c \) and \( k_{AC}^c \) are given.
\[ k^* = 0.82 + 0.067s \]  
(3.10)

where \( s \) is the slump of fresh concrete in inches.

\( k'_j \) accounts for the ratio of fine aggregates to total aggregates by weight (percent) and is calculated from

\[ k'_j = 0.88 + 0.0024f \]  
(3.11)

The air content is taken into consideration by \( k_{AC}^* \) as

\[ k_{AC}^* = 0.46 + 0.094AC \geq 1 \]  
(3.12)

where \( AC \) is the air content expressed as a percentage.

**Shrinkage:** Analogous to creep, the ACI Committee 209 estimates shrinkage from

\[ \varepsilon_{sh}(t) = \frac{(t - t_{sh,0})}{\alpha + (t - t_{sh,0})} \varepsilon_{sh\infty} \]  
(3.13)

where \( t_{sh,0} = \) curing time  
\( \alpha = \) constant

For moist curing \( \alpha = 35 \) and \( t_{sh,0} = 7 \) days.

For steam curing \( \alpha = 55 \) and \( t_{sh,0} = 3 \) days.

The ultimate shrinkage strain, \( \varepsilon_{sh\infty} \) is calculated as

\[ \varepsilon_{sh\infty} = -780 \times 10^{-6} \times k_h^* k_H^* k_s^* k_f^* k_{AC}^* k_{CC}^* \]  
(3.14)

\( k_h^* \) is a correction factor that accounts for member size and its values for members with average thicknesses less than 6" are given in Table 3.2.

For average thickness between 6" and 12",\n
\[ k_h^* = 1.23 - 0.03t \]  
(3.15)

and for thickness greater than 12",
32

Table 3.2: T h ic k n e s s C o r r e c tio n F a c to r fo r S h rin k a g e
Average thickness
(inches)

kKh*

2

1.35
1.25
1.17
1.08

3
4
5
k ha = i.2 e - 0ll2(w/*>

(3.16)

To account for th e am bient relative humidity,
3 _ ( 1.4 - O.Oltf
H
\ 3.0 - 0.03H

for 40 < H < 80%
for 80 < H < 100%

(
(

’

T he slum p correction factor &® is calculated as
fc® = 0.89 + 0.041s

(3.18)

and th e fine aggregate correction factor is given by

f

J 0.3 +
0.0 14/ for / <
\ 0.9 + 0.0 0 2 / for / > 50%

50%
^

'

k*AC is th e air content correction factor and
k*AC = 0.95 +

0.0084(7

(3.20)

The recom m endations neglect th e effect of the cem ent content on th e creep coeffi­
cient, b u t include a correction factor for shrinkage.
k c‘ c = 0.75 +

0.00036c

(3.21)

where c is the cem ent content in pounds per cubic yard.
M o d u lu s o f e la s tic ity : T he m odulus of elasticity of concrete at tim e t is based on
th e compressive stren g th /'(< ). T he compressive strength is related to th e 28-day
compressive strength f'c2B as


\[ f'_c(t) = \frac{t}{a + bt} f'_{28} \]  
\text{(3.22)}

\( a \) and \( b \) are constants dependent on the type of cement and the type of curing and are given in Table 3.3.

**Table 3.3: Constants \( a \) and \( b \) for use in equation (3.22)**

<table>
<thead>
<tr>
<th>Cement</th>
<th>Curing type</th>
<th>( a )</th>
<th>( b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type I</td>
<td>Moist</td>
<td>4.0</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>Steam</td>
<td>1.0</td>
<td>0.95</td>
</tr>
<tr>
<td>Type III</td>
<td>Moist</td>
<td>2.3</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>Steam</td>
<td>0.7</td>
<td>0.98</td>
</tr>
</tbody>
</table>

The modulus of elasticity (psi) is then calculated from

\[
E_C(t) = 33w^{1.5} \sqrt{f'_c}
\]  
\text{(3.23)}

where \( w \) is the unit weight of concrete in pounds per cubic foot.

### 3.3.2 CEB-FIP Recommendations

The CEB-FIP (1978) recommendations estimate creep via a creep coefficient defined at 28 days, namely \( \phi_{28}(t, t') \). The creep coefficient is related to the modulus of elasticity at 28 days by

\[
\phi_{28}(t, t') = C(t, t') E(28)
\]  
\text{(3.24)}

A comparison of equations (3.2) and (3.24) yields

\[
\phi(t, t') = \frac{E(t')}{E(28)} \phi_{28}(t, t')
\]  
\text{(3.25)}
The CEB-FIP recommendations are based on the creep model developed by Rüscher and Jungwirth (1976).

**Creep:** The recommendations are unique in that creep strain is considered to be composed of three components: a rapid strain, a flow component and a delayed elastic component. The creep coefficient is given as

\[ \phi_{26}(t, t') = \beta_a(t') + \phi_f [\beta_f(t) - \beta_f(t')] + \phi_d\beta_d(t - t') \]  \hspace{1cm} (3.26)

The first term \( \beta_a(t') \) in equation (3.26) represents the rapid initial strain and is irreversible. It depends on the age at loading of concrete and is expressed as

\[ \beta_a(t') = 0.8 \left[ 1 - \frac{1}{1.276} \left( \frac{t'}{4.2 + 0.85t'} \right)^{1.276} \right] \]  \hspace{1cm} (3.27)

The rapid initial strain is assumed to occur within the first few days of loading.

The second term in equation (3.26) is the flow component of creep and is irreversible. The flow component depends upon the age of application of load, the ambient relative humidity and the size of the member. It is sensitive to temperature. This component is given as

\[ \phi_f [\beta_f(t) - \beta_f(t')] = \phi_f \left[ \left( \frac{t}{t + H_f} \right)^{\frac{1}{2}} - \left( \frac{t'}{t' + H_f} \right)^{\frac{1}{2}} \right] \]  \hspace{1cm} (3.28)

where \( \phi_f \) = flow coefficient

\( H_f \) = time delay factor (Table 3.5)

The flow coefficient \( \phi_f \) is the product of factors that account for the ambient humidity and the theoretical thickness of the member and is evaluated from

\[ \phi_f = \phi_{f1} \times \phi_{f2} \]  \hspace{1cm} (3.29)

where \( \phi_{f1} \) is the ambient humidity coefficient as given in Table 3.4 and \( \phi_{f2} \) is the theoretical thickness coefficient which depends on the notional thickness \( h_o \) defined as
\[ h_o = \lambda \frac{A_c}{u} \quad (3.30) \]

where \( A_c \) = cross-sectional area of the member (\( mm^2 \))

\( u \) = perimeter exposed to drying (\( mm \))

\( \lambda \) = coefficient dependent on ambient humidity (Table 3.4)

Table 3.4: Coefficients of creep for use in equations (3.29) and (3.30)

<table>
<thead>
<tr>
<th>Relative Humidity (percent)</th>
<th>( \phi_{f1} )</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.8</td>
<td>30</td>
</tr>
<tr>
<td>90</td>
<td>1.0</td>
<td>5</td>
</tr>
<tr>
<td>70</td>
<td>2.0</td>
<td>1.5</td>
</tr>
<tr>
<td>40</td>
<td>3.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

\( \phi_{f2} \) is evaluated from the curve shown in Figure 3.4. The value of the time delay factor \( H_f \) in equation (3.28) depends on the theoretical thickness \( h_o \) and can be interpolated from Table 3.5.

Table 3.5: Time delay factor for creep flow

<table>
<thead>
<tr>
<th>Theoretical thickness ( h_o ) (mm.)</th>
<th>( H_f ) (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>330</td>
</tr>
<tr>
<td>100</td>
<td>425</td>
</tr>
<tr>
<td>200</td>
<td>570</td>
</tr>
<tr>
<td>400</td>
<td>870</td>
</tr>
<tr>
<td>800</td>
<td>1500</td>
</tr>
<tr>
<td>1600</td>
<td>2500</td>
</tr>
</tbody>
</table>

The final component of creep is the delayed elastic strain which is considered to be entirely recoverable on load reversal. It is temperature independent and is a function of load duration only. It is given by

\[ \phi_{d,\beta}(t - t') = 0.4 \left\{ 0.73 \left[ 1 - e^{-0.01(t-t')} \right] + 0.27 \right\} \quad (3.31) \]

To account for the ambient temperature during curing, and for cement type, the age of concrete is adjusted using the equation
Figure 3.4: Coefficient $\phi_{f2}$ for the flow component of creep
\[ t_e = \frac{k}{30} \sum_{0}^{t_M} \{[T^o + 10] \Delta t \} \]  

(3.32)

where \( t_M \) = age at end of curing

\[ \Delta t = \text{the number of days during which concrete} \]

\[ \text{curing takes place at } T^o \text{C} \]

The coefficient \( k \) accounts for the cement type and takes on values of 1 for normal and slow-hardening cements, 2 for rapid-hardening cements and 3 for rapid-hardening high-strength cements. The creep compliance can now be evaluated as

\[ J(t, t') = \frac{1}{E(t')} + \frac{\phi_{28}(t, t')}{E(28)} \]  

(3.33)

The modulus of elasticity at any time \( t \) is calculated from the modulus at 28 days by

\[ E(t') = E(28) \left( \frac{t'}{4.2 + 0.85t'} \right)^\frac{1}{2} \]  

(3.34)

**Shrinkage:** The strain due to shrinkage that occurs in an interval of time \( (t - t_{sh,o}) \) is given by

\[ \epsilon_{sh}(t, t_{sh,o}) = \epsilon_{sh,o} \left[ \beta_{sh}(t) - \beta_{sh}(t_{sh,o}) \right] \]  

(3.35)

where \( \epsilon_{sh,o} \) = basic shrinkage coefficient

\( \beta_{sh} \) = development of shrinkage with time function

\( t \) = age of concrete

\( t_{sh,o} \) = age at which drying starts

The basic shrinkage coefficient is the product of 2 coefficients

\[ \epsilon_{sh,o} = \epsilon_{sh,1} \times \epsilon_{sh,2} \]  

(3.36)
Figure 3.5: Coefficient $\epsilon_{sh2}$ for shrinkage.
Figure 3.6: Development function for shrinkage (CEB-FIP Recommendations).
ε_{sh,1} depends on the ambient humidity and its value is to be interpolated from Table 3.6.

**Table 3.6: Shrinkage coefficient ε_{sh,1}**

<table>
<thead>
<tr>
<th>Relative humidity (percent)</th>
<th>ε_{sh,1} \times 10^{-6}</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>+100</td>
</tr>
<tr>
<td>90</td>
<td>-130</td>
</tr>
<tr>
<td>70</td>
<td>-320</td>
</tr>
<tr>
<td>40</td>
<td>-520</td>
</tr>
</tbody>
</table>

ε_{sh,2} depends on the theoretical thickness \( h_o \) and its values are given in Figure 3.5.

The function \( \beta_{sh} \), which represents the development of shrinkage with time, depends on the theoretical thickness \( h_o \) and its variation with time is shown in Figure 3.6. Ages \( t \) and \( t_{sh,o} \) must be adjusted for different curing temperatures using equation (3.32).

### 3.3.3 Bažant-Panula Model II (BP2)

Bažant and Panula (1980) proposed a simplified model for estimating basic creep, total creep and shrinkage based on their more complex model (See Bažant and Panula (1978, 1979)). Their model differs substantially from the ACI 209 and CEB-FIP recommendations in that creep is now broken up into basic and drying creep components. The drying creep is a function of shrinkage and it is therefore preferable to present the equations for shrinkage first.

**Shrinkage:** The shrinkage strain in concrete at any time \( t \) measured from the time at the start of drying, \( t_{sh,o} \), is given by

\[
\varepsilon_{sh} = \varepsilon_{shoo} k_h S \left( \frac{t}{\tau_{sh}} \right) \tag{3.37}
\]
where $e_{sh\infty}$ = ultimate shrinkage strain

$k_h$ = humidity coefficient

$S$ = function of shrinkage development

$t$ = duration of drying

$\tau_{sh}$ = shrinkage square half-time

The function of shrinkage development is given as a hyperbolic variation in time:

$$S \left( \frac{\dot{t}}{\tau_{sh}} \right) = \sqrt{\frac{\dot{t}}{\tau_{sh} + \dot{t}}}$$  \hspace{1cm} (3.38)

The ultimate shrinkage strain is calculated from the following expression:

$$e_{sh\infty} = (1330 - 970y) \times 10^{-8}$$  \hspace{1cm} (3.39)

where $y$ depends on the composition of concrete and is defined as

$$y = \left( \frac{390}{z^4 + 1} + 1 \right)^{-1}$$  \hspace{1cm} (3.40)

and

$$z = \sqrt{f'_c} \left[ 1.25 \sqrt{\frac{a}{c}} + 0.5 \left( \frac{g}{s} \right)^2 \right] \left( \frac{1 + s/c}{w/c} \right)^2 - 12$$  \hspace{1cm} (3.41)

where $f'_c$ = standard 28-day cylinder strength (ksi)

$w/c$ = water/cement ratio (by weight)

$a/c$ = aggregate/cement ratio (by weight)

$g/s$ = gravel/sand ratio (by weight)

$s/c$ = sand/cement ratio (by weight)

If $z < 0$, $z$ must be set equal to zero.

The humidity coefficient $k_h$ is to be calculated as
\[ k_h = \begin{cases} 
1 - h^3 & \text{for } h \leq 0.98 \\
-0.2 & \text{for } h = 1.0 
\end{cases} \quad (3.42) \]

The shrinkage duration function \( S \) takes into account the effects of size, age and diffusivity of concrete through \( \tau_{sh} \), which is calculated from

\[ \tau_{sh} = \left( \frac{k_s D}{C_1(t_{sh,o})} \right)^2 \quad (3.43) \]

In equation (3.43), \( C_1(t_{sh,o}) \) represents the diffusivity at the start of drying and is given as

\[ C_1(t_{sh,o}) = 2.4 + \frac{120}{\sqrt{t_{sh,o}}} \quad (3.44) \]

\( D \) is the effective cross-section thickness in mm and is set to be twice the volume/surface ratio in mm. \( k_s \) is a shape factor which equals 1.0 for a slab, 1.15 for a cylinder, 1.25 for a square prism, 1.30 for a sphere and 1.55 for a cube.

**Basic creep:** The basic creep coefficient is based on the double power law written as

\[ \phi_b(t, t') = E_0 C_b(t, t') = \psi_1 (t'^{-m} + \alpha)(t - t')^n \quad (3.45) \]

where \( E_0 \) is the asymptotic modulus. This modulus is obtained by evaluating the asymptote of a plot of basic creep versus the logarithm of short times (say 1 day) under load. It is given by

\[ \frac{1}{E_0} = 0.1 + 0.5 (f_{c28}')^{-2} \times 10^{-6} / \text{psi} \quad (3.46) \]

where \( f_{c28}' \) is in ksi and is the 28-day cylinder strength. The parameters \( \psi_1, m \) and \( n \) are all functions of \( f_{c28}' \) and are approximated by

\[ \psi_1 = 0.3 + 15 (f_{c28}')^{-1.2} \quad (3.47) \]

\[ m = 0.28 + (f_{c28}')^{-2} \quad (3.48) \]

\[ n = 0.115 + 0.0002 (f_{c28}')^3 \quad (3.49) \]
\( \alpha \) is a constant equal to 0.05.

The basic creep function can then be calculated from

\[
J_b(t, t') = \frac{1}{E_0} \left[ 1 + \phi_b(t, t') \right]
\]

(3.50)

The conventional elastic modulus at time \( t' \) is \( E(t') \) and is obtained from \( E(t') = 1/J_b \) for \( t - t' = 1 \) day. Therefore

\[
\frac{1}{E(t')} = \frac{1}{E_0} \left[ 1 + \psi_1(t'^{-m} + \alpha) \right]
\]

(3.51)

**Drying creep:** The creep coefficient for drying creep is assumed to be dependent on shrinkage and is expressed as

\[
\phi_d(t, t', t_{sh,0}) = E_0 C_d(t, t', t_{sh,0}) = \tilde{\psi}_d k_h' t'^{-m/2} S_d(t, t')
\]

(3.52)

\( k_h' \) is a coefficient for humidity \( h \) and is to be taken as

\[
k_h' = 1.0 - h^{1.5}
\]

(3.53)

\( \tilde{\psi}_d \) represents the effect of age at loading and is calculated as

\[
\tilde{\psi}_d = \left( 1 + \frac{t' - t_{sh,0}}{10 \tau_{sh}} \right)^{-1/2} \psi_d(10^6 \varepsilon_{sh,\infty})
\]

(3.54)

\( \psi_d \) introduces the concrete composition effects and is calculated as follows:

\[
r = \left( \frac{f_{c28}}{f_c} \right)^{0.3} \left( \frac{g}{d} \right)^{1.3} \left( \frac{0.00161 w/c}{\varepsilon_{sh,\infty}} \right)^{1.5} - 0.85
\]

(3.55)

and

\[
\psi_d = \begin{cases} 
0.0056 + 0.0169 \\ 0.7r^{-1.4} \\ 0.0056 
\end{cases} \quad \text{for } r > 0 \\
\begin{cases} 
0.0056 + 0.0169 \\ 0.7r^{-1.4} \\ 0.0056 
\end{cases} \quad \text{for } r \leq 0
\]

(3.56)

\( S_d(t, t') \) is the evolution of shrinkage function and is given as

\[
S_d(t, t') = \left( 1 + \frac{3 \tau_{sh}}{t - t'} \right)^{-0.35}
\]

(3.57)
The total compliance is now

\[ J(t, t') = \frac{1}{E_o} [1 + \phi_b(t, t') + \phi_d(t, t')] \]  

(3.58)

and the total creep coefficient based on the modulus of elasticity at the age of loading is

\[ \phi(t, t') = E(t')J(t, t') - 1. \]  

(3.59)

### 3.4 Elementary Material Characteristics

Bridges with precast pretensioned concrete girders and cast-in-place concrete deck slabs are designed to be stressed within their service range. It is therefore a safe and verifiable assumption that the concrete remains within the linear range on the stress-strain curve. The modulus of elasticity may be considered to vary with time. Expressions for the change of the modulus of elasticity with time are given in the creep models described in the previous section.

Concrete exhibits both elastic and viscous behavior. An elastic solid is defined by a constant and stable shape which stores deformation energy when deformed under loading and releases all the energy upon load removal to return to its initial state. On the other hand, a viscous fluid flows into any shape on loading without storing deformation energy and its deformation is irrecoverable. Concrete is both elastic and viscous, forming a viscoelastic material that responds in a time-dependent manner under constant applied stress. Within the service range (i.e. being stressed less than about 50% of its strength), concrete is commonly accepted to be a linear viscoelastic material.

The total strain of a concrete specimen uniaxially loaded at time \( t \) may be subdivided as

\[ \epsilon(t) = \epsilon_c(t) + \epsilon_e(t) + \epsilon_{\text{ah}}(t) + \epsilon_T(t) \]

\[ = \epsilon_c(t) + \epsilon'(t) \]
\[ = \epsilon_e(t) + \epsilon_c(t) + \epsilon^o(t) \]  
\[ = \epsilon_e(t) + \epsilon^o(t) \]  

wherein \( \epsilon_e(t) \) is the instantaneous elastic strain, \( \epsilon_c(t) \) is the creep strain, \( \epsilon_{sh}(t) \) is the shrinkage strain, \( \epsilon_T(t) \) is the thermal strain, \( \epsilon'(t) \) is the inelastic strain, \( \epsilon^o(t) \) is the stress-independent inelastic strain and \( \epsilon_o(t) \) is the stress-produced strain.

For a linear viscoelastic material, stress is proportional to strain. Therefore, from equations (3.1) and (3.60), for a constant stress \( \sigma \) that has been acting since age \( t' \), the total strain is

\[ \epsilon(t) = \sigma J(t, t') + \epsilon^o(t) \]  

(3.61)

To extend equation (3.61) for three-dimensional stress states, the assumption is made that concrete is isotropic and that the Poisson’s ratios for creep and elastic strains are equal. Then \( J(t, t') \) can be written as

\[ [J(t, t')] = J(t, t') [D] \]  

(3.62)

where \([D]\) is the elasticity matrix defined in equation (2.16). Equation (3.61) can now be written in matrix form as

\[ \{\epsilon(t)\} = \{\sigma\}^T [J(t, t')] + \{\epsilon^o(t)\} \]  

(3.63)

and used for a three-dimensional analysis.

The equations for strain in the remaining part of this report will be written for the one-dimensional case, unless specifically noted.

### 3.5 The Principle of Superposition

A linear relation is assumed to exist between internal stresses and total strains in the linear theory of viscoelasticity. The principle of superposition was first introduced by Boltzmann (1874) for non-aging materials with time-dependent behavior.
McHenry (1943) extended the principle to include aging materials like concrete provided that stresses were kept within a limit below the ultimate strength. For concrete stressed within the service range, the validity of this principle has been commonly accepted.

The principle of superposition states that the response to a sum of two or more stress histories is the sum of the responses to each of them taken separately. Therefore, for a continuously varying stress, equation (3.61) can be rewritten as

\[ \varepsilon(t) = \int_0^t J(t, t') \frac{d\sigma(t')}{dt'} dt' + \varepsilon^o(t) \]

(3.64)

Since \( J(t, t') \) is continuous and is independent of \( \sigma(t') \), the Reimann integral in equation (3.64) can be converted into a Stieltjes integral (refer to Flügge (1975) and Creus (1986)) to take the form

\[ \varepsilon(t) = \int_0^t J(t, t')d\sigma(t') + \varepsilon^o(t) \]

(3.65)

The advantage of the Stieltjes integral is that it is applicable to even discontinuous stress histories.

The use of the principle of superposition requires several creep tests to be performed on virgin concrete specimens at different ages so that strains can be predicted under varying stress. Ross (1958) measured the total creep of concrete under various histories of stress and compared the observed values with those predicted using the principle of superposition. His results indicated that the creep recovery on removal of stress predicted by the principle was greater than the actual values. He also observed an overestimation in strains calculated even under increasing stress. These discrepancies can be explained by the fact that by considering the creep of virgin concrete, the effect of the previous stress history on the creep coefficient is neglected. However, at present, the principle of superposition is the most efficient and accurate method for the creep analysis of structures (Bažant (1982) and Neville et al. (1983)).
3.6 Formulation of the Creep Strain Increment

The constitutive relation (equation (3.65)) for concrete is expressed in terms of a superposition integral with the creep compliance representing a so-called memory function. The memory function evaluates the response of the material to unit step function inputs. The solution of equation (3.65) is performed using varying time steps and is dependent upon the form of the creep compliance $J(t, t')$.

Several expressions are available in literature that formulate the creep compliance in various ways so as to aid in the evaluation of the integral. Since the creep compliance is the sum of the instantaneous strain and the creep per unit stress, the creep strain at any time $t$ can be calculated from the integral

$$e_c(t) = \int_0^t C(t,t')d\sigma(t')$$  \hspace{1cm} (3.66)

It remains now to express $C(t, t')$ in a suitable manner. Any expression for $C(t, t')$ must be such that it can fit experimental data when available. It should have undetermined coefficients that are easy to evaluate and it should permit numerical computations without excessive need for computer storage.

In a paper that introduced the use of the principle of superposition for the creep analysis of concrete, McHenry (1943) suggested:

$$C(t, t') = \alpha \left[ 1 - e^{-\nu(t-t')} \right] + \beta e^{-pt} \left[ 1 - e^{-m(t-t')} \right]$$  \hspace{1cm} (3.67)

where $\alpha$, $\beta$, $\nu$, $m$ and $p$ are constants to be determined from experimental data.

Arutyunyan (1966) suggested another expression that can be fit to experimental data. This expression is

$$C(t, t') = \left( a + \frac{b}{t'} \right) \sum_{k=0}^{m} \beta_k e^{-\nu_k(t-t')}$$  \hspace{1cm} (3.68)

where $a$, $b$, $\beta_k$, $\nu_k$ and $m$ are the parameters to be determined.

Selna (1967) performed the analysis of reinforced concrete frames and chose the specific creep to be
\[ C(t, t') = \sum_{i=1}^{3} \sum_{j=1}^{4} \alpha_i a_j t'^{-0.14 i-j} \left[ 1 - e^{-k_i (t-t')} \right] \]

\( \alpha_i, k_i \) and \( a_j \) are constants determined from experiments. This expression enabled the calculation of the total strain at any time from stresses computed and stored in the previous two time intervals, instead of the entire history.

Mukaddam and Bresler (1972) included the effects of the age of concrete and the temperature by assuming concrete to be a thermorheologically simple material. They suggested:

\[ C(t, t', T) = \sum_{i=1}^{n} a_i e^{-\lambda_i \Phi(T) \psi(t') (t-t')} \]

in which \( a_i \) and \( \lambda_i \) are coefficients determined from experiments. \( \Phi(T) \) and \( \psi(t') \) are empirical temperature and age shift functions. The drawback of their proposed expression lies in the fact that the evaluation of current strains requires the storage of the entire stress history.

Zienkiewicz and Watson (1966) proposed a specific creep function of the form:

\[ C(t, t', T) = c(t', T) \left( 1 - e^{-p(t-t')} \right) \]

c\( (t', T) \) is an age and temperature dependent function and \( p \) is a constant determined from experimental data. It was shown how, for equal time intervals, the above form of the specific creep could be used to evaluate total strains from the knowledge of stress increments occurring in only the previous time interval. However, no suggestion was made as to the form of function \( c(t', T) \).

Bažant and Wu (1973) first expressed the creep compliance in the form of a Dirichlet series and developed a method to identify the series from known data. The creep compliance was given by

\[ J(t, t') = \frac{1}{E(t')} + \sum_{\mu=1}^{n} \frac{1}{E_{\mu}(t')} \left[ 1 - e^{-(t-t')/\tau_{\mu}} \right] \]
where \( \tau_\mu \) = retardation times

\( \dot{E}_\mu \) = aging coefficients

\( E(t') \) = instantaneous elastic modulus

Bazant and Wu also showed how each term of the Dirichlet series represents one component of \( n \) Kelvin models in series, thus giving physical significance to the retardation times \( \tau_\mu \).

Kabir (1976) extended the work done by Zienkiewicz and Watson (1966), Bazant and Wu (1973) and Mukaddam and Bresler (1972) and expressed the specific creep in Dirichlet series form as

\[
C(t, t') = \sum_{i=1}^{m} a_i(t') \left[ 1 - e^{-\lambda_i \phi(T)(t-t')} \right]
\]  

(3.73)

where \( a_i(t') \) = aging coefficients

\( \Phi(T) \) = temperature-shift function

\( m_i, \lambda_i \) = constants

The stress history is contained in a set of hidden state variables and the computation of the strain requires the incremental stresses in the previous time interval.

In the present analysis the basic form used by Kabir will be adopted and temperature effects will be included as described in section 3.6.3. Since creep strains and the modulus of elasticity in this analysis are evaluated from empirical procedures, it is advantageous to express the creep coefficient in Dirichlet series form and to divide by the elastic modulus to get the specific creep. Therefore

\[
C(t, t') = \frac{\phi(t, t')}{E(t')} = \frac{1}{E(t')} \sum_{i=1}^{m} a_i(t') \left[ 1 - e^{-\lambda_i(t-t')} \right]
\]  

(3.74)

Extending equation (3.74) to the three-dimensional case, we have

\[
[C(t, t')] = [D(t')]^{-1} \phi(t, t') = [D(t')]^{-1} \sum_{i=1}^{m} a_i(t') \left[ 1 - e^{-\lambda_i(t-t')} \right]
\]  

(3.75)
3.6.1 Evaluation of the Creep Strain Increment

The initial strain method (see Zienkiewicz et al. (1971)) is used here for the time-dependent analysis of bridges, wherein the time under consideration is broken into time steps and inelastic strains occurring during a time interval are applied as initial strains at the next time step. Let the time over which the analysis is to be performed be divided into steps \( t_n; n = 1, 2, \ldots, r \) where \( t_r \) is the final time. The \( n^{th} \) time interval would be \( \Delta t_n = t_n - t_{n-1} \) and the stress and creep strain increments during \( \Delta t_n \) can be written as \( \{\Delta \sigma_n\} = \{\sigma_n\} - \{\sigma_{n-1}\} \) and \( \{\Delta \epsilon^c_n\} = \{\epsilon^c_n\} - \{\epsilon^c_{n-1}\} \). \( \{\sigma_n\} \) and \( \{\epsilon^c_n\} \) are the stress and strain vectors acting at time \( t_n \). The elasticity matrix from equation (2.16) can be written at time \( t_n \) by replacing \( E \) by \( E(t_n) \) or \( E_n \). The constitutive relationship is then given by

\[
\{\sigma_n\} = [D_n] \{\{\epsilon_n\} - \{\epsilon^c_n\}\} = E_n [D_o] \{\{\epsilon_n\} - \{\epsilon^c_n\}\} \tag{3.76}
\]

where \( \epsilon^c_n \) is the sum of all inelastic strain increments (creep + shrinkage + temperature) occurring up to time \( t_{n-1} \).

If the integral in equation (3.66), written for three-dimensions, is approximated using the trapezoidal rule, the creep strain at time \( t_n \) is given by

\[
\{\epsilon^c_n\} = [D_1]^{-1} \{\Delta \sigma_1\} \phi(t_n, t_1) + [D_2]^{-1} \{\Delta \sigma_2\} \phi(t_n, t_2) + \cdots + [D_{n-1}]^{-1} \{\Delta \sigma_{n-1}\} \phi(t_n, t_{n-1}) \tag{3.77}
\]

Substituting for \( \phi(t, t') \) from equation (3.75) and noting that \( E_n[D_o] = [D_n] \) implies that \( [D_n]^{-1} = 1/E_n[D_o]^{-1} \);

\[
[D_o] \{\epsilon^c_n\} = \{\Delta \sigma_1\} \frac{1}{E(t_1)} \sum_{i=1}^{m} a_i(t_1) \left[ 1 - e^{-\lambda_i(\Delta t_2 + \Delta t_3 + \cdots + \Delta t_n)} \right] \\
+ \{\Delta \sigma_2\} \frac{1}{E(t_2)} \sum_{i=1}^{m} a_i(t_2) \left[ 1 - e^{-\lambda_i(\Delta t_3 + \cdots + \Delta t_n)} \right] \\
+ \cdots + \{\Delta \sigma_{n-2}\} \frac{1}{E(t_{n-2})} \sum_{i=1}^{m} a_i(t_{n-2}) \left[ 1 - e^{-\lambda_i(\Delta t_{n-1} + \Delta t_n)} \right] \\
+ \{\Delta \sigma_{n-1}\} \frac{1}{E(t_{n-1})} \sum_{i=1}^{m} a_i(t_{n-1}) \left[ 1 - e^{-\lambda_i(\Delta t_n)} \right] \tag{3.78}
\]
Similarly at time step $t_{n-1}$:

$$[D_o]\{e_{n-1}\} = \{\Delta \sigma_1\} \frac{1}{E(t_1)} \sum_{i=1}^{m} a_i(t_1) \left[ 1 - e^{-\lambda_i(\Delta t_2+\Delta t_3+\cdots+\Delta t_{n-1})} \right]$$
$$+ \{\Delta \sigma_2\} \frac{1}{E(t_2)} \sum_{i=1}^{m} a_i(t_2) \left[ 1 - e^{-\lambda_i(\Delta t_3+\cdots+\Delta t_{n-1})} \right]$$
$$+ \cdots + \{\Delta \sigma_{n-2}\} \frac{1}{E(t_{n-2})} \sum_{i=1}^{m} a_i(t_{n-2}) \left[ 1 - e^{-\lambda_i(\Delta t_{n-1})} \right]$$

$$[D_o]\{\Delta e_{n}^c\} = [D_o]\left[\{e_{n}\} - \{e_{n-1}\}\right]$$
$$= \{\Delta \sigma_1\} \frac{1}{E(t_1)} \sum_{i=1}^{m} a_i(t_1) \left[ 1 - e^{-\lambda_i(\Delta t_n)} \right] e^{-\lambda_i(\Delta t_2+\Delta t_3+\cdots+\Delta t_{n-1})}$$
$$+ \{\Delta \sigma_2\} \frac{1}{E(t_2)} \sum_{i=1}^{m} a_i(t_2) \left[ 1 - e^{-\lambda_i(\Delta t_n)} \right] e^{-\lambda_i(\Delta t_3+\cdots+\Delta t_{n-1})}$$
$$+ \cdots + \{\Delta \sigma_{n-1}\} \frac{1}{E(t_{n-1})} \sum_{i=1}^{m} a_i(t_{n-1}) \left[ 1 - e^{-\lambda_i(\Delta t_n)} \right] \quad (3.79)$$

The creep strain increment occurring during $\Delta t_n$ can then be calculated as

$$[D_o]\{\Delta e_{n}^c\} = [D_o]\left[\{e_{n}\} - \{e_{n-1}\}\right]$$
$$= \{\Delta \sigma_1\} \frac{1}{E(t_1)} \sum_{i=1}^{m} a_i(t_1) \left[ 1 - e^{-\lambda_i(\Delta t_n)} \right]$$
$$+ \{\Delta \sigma_2\} \frac{1}{E(t_2)} \sum_{i=1}^{m} a_i(t_2) \left[ 1 - e^{-\lambda_i(\Delta t_n)} \right]$$
$$+ \cdots + \{\Delta \sigma_{n-1}\} \frac{1}{E(t_{n-1})} \sum_{i=1}^{m} a_i(t_{n-1}) \left[ 1 - e^{-\lambda_i(\Delta t_n)} \right] \quad (3.80)$$

Equation (3.80) can be rearranged as

$$[D_o]\{\Delta e_{n}^c\} = \sum_{i=1}^{m} \left[ 1 - e^{-\lambda_i(\Delta t_n)} \right] \left\{ \{\Delta \sigma_1\} \frac{1}{E(t_1)} a_i(t_1) e^{-\lambda_i(\Delta t_2+\cdots+\Delta t_{n-1})} + \{\Delta \sigma_2\} \frac{1}{E(t_2)} a_i(t_2) e^{-\lambda_i(\Delta t_3+\cdots+\Delta t_{n-1})} + \cdots + \{\Delta \sigma_{n-1}\} \frac{1}{E(t_{n-1})} a_i(t_{n-1}) \right\} \quad (3.81)$$

In concise form, equation (3.81) can be written as

$$[D_o]\{\Delta e_{n}^c\} = \sum_{i=1}^{m} \left[ 1 - e^{-\lambda_i(\Delta t_n)} \right] e_i^*\left\{ e_i^* \frac{1}{E(t_1)} a_i(t_1) e^{-\lambda_i(\Delta t_2+\cdots+\Delta t_{n-1})} + \{\Delta \sigma_2\} \frac{1}{E(t_2)} a_i(t_2) e^{-\lambda_i(\Delta t_3+\cdots+\Delta t_{n-1})} + \cdots + \{\Delta \sigma_{n-1}\} \frac{1}{E(t_{n-1})} a_i(t_{n-1}) \right\} \quad (3.82)$$

$e_i^*$ are called hidden state variables and represent the accumulation of the stress history. A comparison of equations (3.81) and (3.82) yields

$$e_i^* = \{\Delta \sigma_1\} \frac{1}{E(t_1)} a_i(t_1) e^{-\lambda_i(\Delta t_2+\Delta t_3+\cdots+\Delta t_{n-1})}$$
$$+ \cdots + \{\Delta \sigma_{n-2}\} \frac{1}{E(t_{n-2})} a_i(t_{n-2}) e^{-\lambda_i(\Delta t_{n-1})}$$
$$+ \{\Delta \sigma_{n-1}\} \frac{1}{E(t_{n-1})} a_i(t_{n-1}) \quad (3.83)$$
Similarly, an expression can be written for the hidden state variables at time $t_{n-1}$ as

$$e_{i,n-1}^* = \left\{ \Delta \sigma_1 \right\} \frac{1}{E(t_1)} a_i(t_1) e^{-\lambda_i(\Delta t_2 + \Delta t_3 + \cdots + \Delta t_{n-2})}$$

$$+ \cdots + \left\{ \Delta \sigma_{n-2} \right\} \frac{1}{E(t_{n-2})} a_i(t_{n-2})$$

(3.84)

Multiplying equation (3.84) by $e^{-\lambda_i(\Delta t_{n-1})}$ and comparing the result with equation (3.83) we have

$$e^{-\lambda_i(\Delta t_{n-1})} e_{i,n-1}^* = \left\{ \Delta \sigma_1 \right\} \frac{1}{E(t_1)} a_i(t_1) e^{-\lambda_i(\Delta t_2 + \Delta t_3 + \cdots + \Delta t_{n-2})}$$

$$+ \cdots + \left\{ \Delta \sigma_{n-2} \right\} \frac{1}{E(t_{n-2})} a_i(t_{n-2}) e^{-\lambda_i(\Delta t_{n-1})}$$

$$= e_{i,n}^* - \left\{ \Delta t_{n-1} \right\} \frac{1}{E(t_{n-1})} a_i(t_{n-1})$$

(3.85)

Therefore, the hidden state variables at time $t_n$ can be evaluated from the variables at $t_{n-1}$ by

$$e_{i,n}^* = e_{i,n-1}^* e^{-\lambda_i(\Delta t_{n-1})} + \left\{ \Delta \sigma_{n-1} \right\} \frac{1}{E(t_{n-1})} a_i(t_{n-1}) \; \text{for } n \geq 3$$

(3.86)

and

$$e_{i,2}^* = \left\{ \Delta \sigma_1 \right\} \frac{1}{E(t_1)} a_i(t_1)$$

(3.87)

$\left\{ \Delta \sigma_1 \right\}$ is the instantaneous stress applied during the first time interval $\Delta t_1$ which is taken to be of zero length.

Equations (3.82), (3.86) and (3.87) indicate that the creep strain increment during any time interval is evaluated from the stress increments from the previous time interval by updating the hidden state variables.

### 3.6.2 Determination of the Aging Coefficients

The aging coefficients $a_i(t')$ in equation (3.74) have to be determined by fitting the Dirichlet series expression for the creep coefficient to the creep coefficients predicted by the three empirical procedures. The constants $m$ and $\lambda_i$ have also to be chosen
and the Dirichlet series fit can be made arbitrarily accurate depending on these choices. If creep coefficients are to be estimated using the ACI 209 procedure, sufficiently accurate results are obtained with \( m = 3 \) and \( \lambda_i = 0.1, 0.01 \) and 0.001.

To calculate the aging coefficients the following procedure is adhered to:

1. Choose \( m \) and \( \lambda_i \).

2. Choose a number of ages at loading \( t' \).

3. Choose a number \( (n) \) of observation times \( t \) for each age at loading.

4. For each age at loading and each observation time, calculate \( \phi(t, t') \) from one of the empirical procedures and set up the equations (in matrix form):

\[
\begin{bmatrix}
1 - e^{-\lambda_1(t_1-t')} & \cdots & 1 - e^{-\lambda_m(t_1-t')}
1 - e^{-\lambda_1(t_2-t')} & \cdots & 1 - e^{-\lambda_m(t_2-t')}
\vdots & \ddots & \vdots
1 - e^{-\lambda_1(t_n-t')} & \cdots & 1 - e^{-\lambda_m(t_n-t')}
\end{bmatrix}_{n \times m}
\begin{bmatrix}
\alpha_1(t') \\
\vdots \\
\alpha_m(t')
\end{bmatrix}_{m \times 1}
\]

\[
= \begin{bmatrix}
\phi(t_1, t') \\
\phi(t_2, t') \\
\vdots \\
\phi(t_n, t')
\end{bmatrix}_{n \times 1} \tag{3.88}
\]

This can be written as

\[
[S]_{n \times m} \{\alpha\}_{m \times 1} = \{\phi\}_{n \times 1} \tag{3.89}
\]

5. Solve for \( \{\alpha\} \) using the least squares procedure, as outlined by the equations given below. From equation (3.89)

\[
[S]^T [S] \{\alpha\} = [S]^T \{\phi\} \tag{3.90}
\]

and

\[
\{\alpha\} = ([S]^T [S])^{-1} [S]^T \{\phi\} \tag{3.91}
\]

To evaluate the accuracy of the Dirichlet series fit, the aging coefficients obtained must be used to calculate the least squares error:
\[ ER = \sum_{j=1}^{n} \left\{ \phi(t_j, t') - \sum_{i=1}^{m} a_i(t') \left( 1 - e^{-\lambda_i(t_j-t')} \right) \right\}^2 \] (3.92)

The least squares error \( ER \) must then be added to the errors for every age at loading and the values of \( m \) and \( \lambda_i \) which gives the least total error must be chosen. However, this procedure can be extremely time-consuming, especially since the evaluation of \( \phi(t, t') \) depends on a large variety of factors. It is more expedient to choose \( m \) and \( \lambda_i \) once and for all based on the creep coefficients predicted by the empirical procedures for typical concrete compositions.

When the CEB-FIP recommendations are used, a Dirichlet series fit is made only for the flow component of creep. Sufficiently accurate values (within 1%) are \( m = 3 \) and \( \lambda_i = 0.15, 0.015 \) and 0.0015. It should be noted that from equation (3.31), if the constant part of the delayed elastic strain \((0.4 \times 0.27)\) is lumped with the rapid initial flow component, the expression for the delayed elastic strain is already in Dirichlet series form, without any age dependence and with \( \lambda_i = 0.01 \).

The BP2 method poses a problem in that the creep coefficient is a result of both basic and drying creep. The drying creep term reaches a limiting value while the basic creep does not. One way of dealing with this problem is to have separate aging coefficients for each of the creep components. This would entail the doubling of the computer memory space allocated to the hidden state variables. The other alternative, which is chosen here, is to adjust the values of \( \lambda_i \) so as to allow for the development of creep strain even after long times and to use only one set of aging coefficients. Increasing the values of \( \lambda_i \) produce the desired effect and 0.2, 0.02 and 0.002 represent an adequate and accurate choice for \( i = 1, 2, 3 \).

### 3.7 Temperature Effects on Creep

Temperature is a factor of primary importance in the time-dependent analysis of concrete structures as it affects the properties of concrete. Temperature has a two-fold effect on creep. It directly influences the deformation rate and it aids in the progress of cement hydration. Experimental research on the effects of tem-
perature on creep has been limited in amount, possibly because the time length of creep tests makes the control of temperature difficult. A large portion of the experimentation has concentrated on temperatures greater than 80°C (common in prestressed concrete nuclear vessels) and is of no significance in bridge analyses.

It is widely accepted that the effect on creep of temperatures lower than 20°C is negligible. A study by Johansen and Best (1962) concentrated on creep at temperatures down to −20°C. The study showed that the presence of ice led to high initial rates of creep strain but these quickly dropped to zero. Ice-free concrete however did experience slightly lower initial rates of creep as compared to concrete at 20°C. Hansen (1960) carried out creep tests on specimens loaded at temperatures ranging from −15°C to 60°C, which showed that the creep rate increased with increasing temperature. His tests, however, lasted only 24 days. Tests by Arthanari and Yu (1967) on sealed concrete under biaxial compression subjected to a temperature range of 20°C to 80°C qualitatively indicated that creep is linearly related to temperature. This linear behavior is lost above 80°C as shown by tests on sealed concrete by Nasser and Neville (1962) and Ross and England (1962). Tests on unsealed concrete indicates results that are similar to the sealed concrete case. Arthanari and Yu (1967) compared the creep of unsealed concrete slabs with the creep of sealed specimens under 40 days of load. They found that while basic creep at 80°C was approximately twice the basic creep at 20°C, the total creep was three times as much as the total creep at 20°C. This indicates that both basic and drying creep rates increase with increasing temperature, the latter component being affected by an acceleration of moisture loss.

Tests on concrete subjected to varying temperature fields indicate some interesting results. Fahmi et al. (1973), subjected micro-concrete to temperature cycles between 23°C and 60°C in compression and in torsion. Illston and Sanders (1973) published test results on cylindrical specimens subjected to torsion and temperature variations. Both the above tests indicated an acceleration in creep with an increase in temperature. However, while a rise in temperature caused an increase
in creep, an equal drop in temperature hardly had any effect. Also indicated by
the test results was that only the first increase in temperature to a certain value
caused an increase in creep, while further cycling to the same temperature had
no additional effect. A further increase in temperature to a previously unattained
value caused an increase in creep. Illston and Sanders (1973) described these phe­
nomena in terms of a 'transitional thermal creep' component which is independent
of concrete maturity and zero when temperature decreases or is increased to a given
level for a second or subsequent time. The above experiments also indicated that
the increase in creep strain due to temperature was irrecoverable.

The question that has to be addressed now is how to quantitatively include
the effects of temperature in the expressions for the creep coefficients given in
equations (3.74) and (3.75). Mukaddam and Bresler (1972) assumed concrete to
be a thermorheologically simple material which obeys the time-shift principle for
temperature variations. This principle is represented graphically in Figure 3.7. In
the Figure, $T_o$ is a fixed reference temperature and $T$ is a constant temperature
greater than $T_o$. The specific creep at temperatures $T_o$ and $T$ are plotted versus
the logarithm of time. They are identical in shape but separated horizontally by a
distance $\psi(T)$. The distance $\psi(T)$ represents an acceleration of the creep rate and
the relationship between the specific creep $C_{T_o}$ and $C_T$ at temperatures $T_o$ and $T$
can be written as

$$C_T(\ln t) = C_{T_o}(\ln t + \psi(T))$$

(3.93)

Since $C_T$ and $C_{T_o}$ represent the same curve, from equation (3.93) we have

$$C_T(e^{\ln t}) = C_{T_o}[e^{(\ln t + \psi(T))}]$$

(3.94)

Noting that $e^{\ln t} = t$, equation (3.94) becomes

$$C_T(t) = C_{T_o}(te^{\psi(T)})$$

(3.95)

or
Figure 3.7: Time-shift Principle for Temperature Variation
\[ CT(t) = C_{T_o}(t\Phi(T)) \]  

(3.96)

with

\[ \Phi(T) = e^{\psi(T)} \]  

(3.97)

\( \Phi(T) \) is the temperature time-shift function which multiplies the time at which creep is observed to obtain a 'reduced time'. The increment in reduced time \((t - t')\), can now be obtained by

\[ (t - t')_r = \Phi(T)(t - t') \]  

(3.98)

and the use of the reduced time in equation (3.75) as

\[ C(t, t') = [D(t')]^{-1} \sum_{i=1}^{m} \alpha_i(t') \left[ 1 - e^{-\lambda_i\Phi(T)(t-t')} \right] \]  

(3.99)

results in an acceleration of the creep rate during the time interval from \( t' \) to \( t \).

Mukaddam and Bresler (1972) demonstrated the validity of the time-shift principle to concrete using temperature dependent creep data obtained by England and Ross (1965) and Browne (1967) and advanced a cubic expression for \( \Phi(T) \). Another cubic expression for \( \Phi(T) \) was proposed by Fahmi et al. (1973).

Bazant and Wu (1974) argued that the creep mechanism consists in the breakage and reformation of bonds which represent thermally activated processes on the molecular scale. Therefore the coefficient \( \Phi(T) \) must be related to the activation energy of the thermal processes and based on various fits with experimental data, they suggested:

\[ \Phi(T) = e^{\frac{U_a}{R}(\frac{T}{T_o} - \frac{1}{T})} \]  

(3.100)

where \( U_a \) is the activation energy of concrete creep and \( R \) is the gas constant. The value of \( U_a/R \) was given as 5000°K and is adopted in this analysis. The value of \( \Phi(T) \) is dependent only on the current temperature since activation energy in any chemical reaction is independent of temperature history. The reference temperature \( T_o \) is arbitrarily chosen as 293°K.
The other effect of temperature on creep to be considered is that of the advancement of concrete maturity with increases in temperature. As stated earlier, the hydration of cement is accelerated at higher temperatures. The aging coefficients $a_i(t')$ in equation (3.99) are a measure of the maturity of the concrete and must be adjusted for varying temperatures. To achieve this, the age at loading $t'$ is replaced by an effective age $t_e$ (or maturity). $t_e$ represents the hydration period at temperature $T$ for which the same degree of hydration is reached as that during an actual time period $t$ at reference temperature $T_0$. Since hydration is a chemical reaction, its rate depends only on the current temperature and for a constant temperature over the time interval $\Delta t_n = t_n - t_{n-1}$, we have at time $t_n$,

$$t_e = t_{n-1,e} + \int_{t_{n-1}}^{t_n} \beta_T dt$$  \hspace{1cm} (3.101)

Bazant and Wu (1974) derived an expression for $\beta_T$ from activation energy concepts as

$$\beta_T = e^{U_h/T_0} \left( \frac{1}{T_0} - \frac{1}{T} \right)$$  \hspace{1cm} (3.102)

where $U_h$ is the activation energy of hydration. Based on experimental data, Copeland et al. (1960) found $U_h$ to be approximately constant between $0^\circ C$ and $100^\circ C$ and set $U_h/R$ to be equal to $2700^\circ K$.

To account for the effects of temperature, equation (3.99) is now modified as

$$C(t, t') = [D(t')]^{-1} \sum_{i=1}^{m} a_i(t_e) \left[ 1 - e^{-\lambda_i \Phi(T)(t-t')} \right]$$  \hspace{1cm} (3.103)

Equation (3.82) can be re-derived for varying temperature and can be written as

$$[D_0] \{ \Delta \epsilon^*_{i,n} \} = \sum_{i=1}^{m} \epsilon^*_{i,n} \left[ 1 - e^{-\lambda_i \Phi(T)(\Delta t_n)} \right]$$  \hspace{1cm} (3.104)

where $T_n$ is the temperature present during $\Delta t_n$. Equation (3.86) now takes on the form:

$$\epsilon^*_{i,n} = \epsilon^*_{i,n-1} e^{-\lambda_i \Phi(T_{n-1})(\Delta t_n)} + \{ \Delta \sigma_{n-1} \} \frac{1}{E(t_{n-1})} a_i(t_{n-1,e}) \text{ for } n \geq 3$$  \hspace{1cm} (3.105)
and equation (3.87) remains unchanged.

It should be noted that the effect of the temperature on the modulus of elasticity is neglected in the analysis. The modulus of elasticity does drop appreciably at high temperatures but is accepted as nearly constant up to about 100°C.

In view of the results of the experimental work referenced in this section, the temperature effects on the creep strain increment are evaluated based on the following assumptions:

1. Only temperature increases affect the creep rate and the values of the aging coefficients.

2. Effects of temperatures below 20°C are negligible; ie. 20°C is the reference temperature for equations (3.100) and (3.102).

3. The first attainment of a particular temperature affects creep and subsequent cycling to that temperature causes no further increase in creep. This is valid for both the creep rate and the aging of concrete.

Of special note in the evaluation of the temperature effects on creep is that while the forgoing analysis is applicable to the total creep in the ACI-209 and BP2 procedures, temperatures affect only the flow component of creep in the CEB-FIP procedure.

### 3.8 The Shrinkage Strain Increment

For the shrinkage strain increment, it is conceivable to apply a different strain increment at every integration point in the finite element mesh chosen. However, in practice, a member is assumed to shrink uniformly and the shrinkage strain increment as a function of time is calculated from one of the 3 empirical procedures described earlier.

The definition of shrinkage prescribes the application of initial strains due to
shrinkage only to the volumetric components of the strain vector. Hence, from equation (2.13), the shrinkage strain increment during $\Delta t_n$ is given as

$$
\{\Delta \epsilon_n^s\} = \begin{bmatrix}
\Delta \epsilon_{sh,n} \\
\Delta \epsilon_{sh,n} \\
\Delta \epsilon_{sh,n} \\
0 \\
0 \\
0
\end{bmatrix}
$$

(3.106)

where $\Delta \epsilon_{sh,n}$ is the shrinkage strain predicted (for the uniaxial case) to occur during the time interval $\Delta t_n = t_n - t_{n-1}$.

### 3.9 The Temperature Strain Increment

The temperature strain increment during the time interval $\Delta t_n$ depends on the temperatures $T_n$ and $T_{n-1}$ present at times $t_n$ and $t_{n-1}$ and the coefficient of thermal expansion $\alpha$. The coefficient $\alpha$ is assumed to remain constant and is taken to be equal to $1.0 \times 10^{-5}/^\circ C$ ($5.6 \times 10^{-6}/^\circ F$).

The temperature strain increment $\Delta \epsilon_n^T$ is easily calculated from

$$
\Delta \epsilon_n^T = \alpha(T_n - T_{n-1})
$$

(3.107)

Again, this increment of strain causes only volume changes and hence the incremental inelastic strain vector for temperature changes is

$$
\{\Delta \epsilon_n^T\} = \begin{bmatrix}
\Delta \epsilon_n^T \\
\Delta \epsilon_n^T \\
\Delta \epsilon_n^T \\
0 \\
0 \\
0
\end{bmatrix}
$$

(3.108)

The problem of adopting a temperature distribution on the bridge cross section is addressed as follows:

1. Temperature is assumed uniform across the girder cross section prior to pouring of the slab.
2. After the slab is poured, the temperature distribution takes on the profile suggested by the Committee on Loads and Forces on Bridges (1981). This profile is shown based on a reference ambient temperature in Figure 3.8.

### 3.10 Correction Strain for Change in Elastic Modulus

There is one component of the total strain increment which is absent in the expression for total strain given in equation (3.60). The reason for this exclusion is that this strain increment arises from the application of the principle of superposition in the analysis and not due to any material changes. Consider the time interval \( \Delta t_n = t_n - t_{n-1} \). The modulus of elasticity will have values \( E_{n-1} \) and \( E_n \) at the beginning and end of the interval. The stress \( \sigma_{n-1} \) is assumed constant during the time interval. If \( \{\epsilon_n\} \) is the net strain (total strain - initial strain) at \( t_n \) and \( \{\epsilon_{n-1}\} \) is the net strain at \( t_{n-1} \), the constant stress acting during \( \Delta t_n \) is calculated from (refer to equation (3.76))

\[
\{\sigma_{n-1}\} = E_n[D_0]\{\epsilon_n\}
\]

and

\[
\{\sigma_{n-1}\} = E_{n-1}[D_0]\{\epsilon_{n-1}\}
\]

From equations (3.109) and (3.110) it is seen that the concrete strain changes at the end of the time interval by the amount:

\[
\{\Delta \epsilon_n^4\} = \{\epsilon_n\} - \{\epsilon_{n-1}\} = [D_0]^{-1}\left(\frac{1}{E_n} - \frac{1}{E_{n-1}}\right)\{\sigma_{n-1}\}
\]

This increment of strain \( \{\Delta \epsilon_n^4\} \) is therefore a correction strain that has to be applied to account for the discrete changes in the elastic modulus at each time. It should not contribute to the element load vectors but should be included in the global constitutive equations as a part of the incremental inelastic strain vector \( \{\Delta \epsilon_n''\} \). At time \( t_n \), the total initial inelastic (pseudo inelastic) strain vector is given by
Figure 3.8: Assumed Temperature Profile on Composite Section
\{\varepsilon''_n\} = \{\varepsilon''_{n-1}\} + \{\Delta \varepsilon''_n\} \quad (3.112)

and

\{\Delta \varepsilon''_n\} = \{\Delta \varepsilon''_n\} + \{\Delta \varepsilon'_n\} + \{\Delta \varepsilon''_n\} + \{\Delta \varepsilon^4_n\} \quad (3.113)

All the components of strain that affect the time-dependent behavior of the bridge have now been described.
Chapter IV
Analysis For Prestressing

Pretensioning in the bridge type under consideration is an important factor in the overall time-dependent behavior, because this behavior is largely dependent on the amount of prestress acting at any particular time. The prestress force in prestressing strands changes continuously. In pretensioned structures, prestress loss occurs prior to the time of transfer due to relaxation of prestressing steel and shrinkage of concrete. At transfer, further losses occur due to the elastic shortening of concrete. After transfer, prestress forces change because of concrete creep and shrinkage, steel relaxation and the effects of loading and temperature history. In the present analysis all the above factors that contribute to the behavior of prestressing steel are considered and the scope is limited to pretensioning.

4.1 Assumptions Regarding Tendon Profile and Behavior

The following assumptions are made with regard to the profile and behavior of the prestressing steel:

1. Prior to transfer, each tendon has a given profile, initial tensioning force and constant cross sectional area along its length. The initial profile can be either straight or harped. The analysis for parabolic tendons can be included with minimal changes to the pre-processing unit of the computer code written for the analysis.

2. The cross sectional area of the tendon does not change with time.

3. Complete bond exists between the tendon steel and the surrounding concrete and hence any effects due to transfer length and slippage are neglected.

4. The stiffness of the tendons contributes to the stiffness of the structure.
5. The overall behavior of the tendon is approximated by the behavior of tendon segments, each lying within an element in the finite element mesh of the structure.

6. Temperature strains in the prestressing steel are neglected.

4.2 Definition of Tendon Segment Profile

Figure 4.1 shows the profile of a tendon segment in an element. Each segment is considered to have three nodes, each of whose coordinates are \((x^*, y^*, z^*)\) in the global reference frame and \((r^*, s^*, t^*)\) in normalized coordinates.

Consider now Figure 4.2, in which a single tendon segment is shown in both the global and local normalized coordinate systems. \(S\) is a curvilinear coordinate along the tendon segment in global coordinates and \(\xi\) is the corresponding normalized coordinate. The procedure that follows for the calculation of tendon position and strain is adapted for three dimensions from the work done by Elwi and Hrudey (1989).

As stated earlier, three nodes (strand nodes) are chosen along the segment. Two of these nodes lie on the boundaries of the element (parent element) containing the strand segment to ensure appropriate interelement continuity. The third strand node can be anywhere between the first two. The normalized coordinate \(\xi\) is centered at this node and the node is named node no. 2 as shown in Figure 4.2. Shape function \(\psi\) can now be selected such that

\[
\begin{align*}
\psi_1 &= \frac{1}{2}(\xi^2 - \xi) \\
\psi_2 &= 1 - \xi^2 \\
\psi_3 &= \frac{1}{2}(\xi^2 + \xi)
\end{align*}
\]

(4.1)

The global coordinates at any point on the strand segment can be obtained from the coordinates of the nodes by setting up the equation:
Figure 4.1: Position of Tendon Segment in an Element
Figure 4.2: Definition of Tendon Profile
\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = \begin{bmatrix}
  <\psi> & 0 & 0 \\
  0 & <\psi> & 0 \\
  0 & 0 & <\psi>
\end{bmatrix} \begin{bmatrix}
  \{x^*\} \\
  \{y^*\} \\
  \{z^*\}
\end{bmatrix}
\]

(4.2)

where \(<\psi> = <\psi_1, \psi_2, \psi_3>\) and \(\{x^*\}, \{y^*\}\) and \(\{z^*\}\) are vectors containing the global coordinates of the strand nodes.

### 4.3 Evaluation of Strain in Strand Segment

To calculate the prestress force within a strand segment, it is necessary to know the distribution of strain within the segment. Only the strain component along the strand is of any consequence. Let \(l, m\) and \(n\) be the direction cosines of the tangent at any point \(P\) on the strand. From Figure 4.2 and the definition of direction cosines we have

\[
l = \cos \alpha = \frac{dx}{dS_s} \\
m = \cos \beta = \frac{dy}{dS_s} \\
n = \cos \gamma = \frac{dz}{dS_s}
\]

(4.3)

Noting that \(l^2 + m^2 + n^2 = 1\), equation (4.3) yields

\[
\left(\frac{dx}{dS_s}\right)^2 + \left(\frac{dy}{dS_s}\right)^2 + \left(\frac{dz}{dS_s}\right)^2 = 1
\]

(4.4)

Multiplying through by \((dS/d\xi)^2\), equation (4.4) becomes

\[
\frac{dS}{d\xi} = J^* = \sqrt{\left(\frac{dx}{d\xi}\right)^2 + \left(\frac{dy}{d\xi}\right)^2 + \left(\frac{dz}{d\xi}\right)^2}
\]

(4.5)

\(J^*\) is the Jacobian of the transformation from the \(S_s\)-coordinate system to the normalized \(\xi\)-coordinate system. The three terms under the square root sign in the above equation can be evaluated from equation (4.2) as

\[
\frac{dx}{d\xi} = <\frac{d\psi}{d\xi} > \{x^*\} \\
\frac{dy}{d\xi} = <\frac{d\psi}{d\xi} > \{y^*\} \\
\frac{dz}{d\xi} = <\frac{d\psi}{d\xi} > \{z^*\}
\]

(4.6)
and from equation (4.1),

\[ < d\psi/d\xi > = < \xi - 1/2, -2\xi, \xi + 1/2 > \]  

(4.7)

The value of \( J^* \) can now be easily calculated at any point along the strand. The direction cosines of the tangent are therefore written as

\[
\begin{align*}
  l &= \frac{dx}{d\xi} = \frac{1}{J^* \frac{d\xi}{dS}} \\
  m &= \frac{dy}{d\xi} = \frac{1}{J^* \frac{d\xi}{dS}} \\
  n &= \frac{dz}{d\xi} = \frac{1}{J^* \frac{d\xi}{dS}}
\end{align*}
\]  

(4.8)

and are now known.

Assume that the strain increment in the parent element at any point on the strand segment is known. A general expression for the tangential strain can be written in terms of the direction cosines as given by Timoshenko and Goodier (1982):

\[
\Delta \epsilon_\tau = \Delta \epsilon_x l^2 + \Delta \epsilon_y m^2 + \Delta \epsilon_z n^2 + \Delta \gamma_{xy}lm + \Delta \gamma_{yz}mn + \Delta \gamma_{xz}ln
\]  

(4.9)

where \( \Delta \epsilon_\tau \) represents the change in tangential strain at the arbitrary chosen point \( P \). The incremental strains on the right-hand side of equation (4.9) are strain increments in the parent element.

To aid in computations, equation (4.9) can be written in matrix form as

\[
\Delta \epsilon_\tau = \left( \begin{array}{ccc} l^2 & m^2 & n^2 \\ lm & mn & ln \end{array} \right) \left( \begin{array}{c} \Delta \epsilon_x \\ \Delta \epsilon_y \\ \Delta \epsilon_z \\ \Delta \gamma_{xy} \\ \Delta \gamma_{yz} \\ \Delta \gamma_{xz} \end{array} \right)
\]  

(4.10)

The incremental strain vector can be expressed in terms of the incremental displacements \( \{u\} \) within the parent element via equation (2.2). In that equation \( [B] \) is the strain-displacement matrix and equation (4.10) becomes
\[
\Delta \epsilon_s = \left( \begin{array}{cccccc}
I^2 & m^2 & n^2 & lm & mn & ln
\end{array} \right) [B] \{ \Delta y \}
\]  
(4.11)

or

\[
\Delta \epsilon_s = \langle B_s \rangle \{ \Delta y \}
\]  
(4.12)

In the above equation, the row vector \(\langle B_s \rangle\) represents the strain-displacement matrix for strand strains and has to be evaluated at points along the strand. Matrix \([B]\) in equation (4.11) has to be evaluated at these same points. The best points for these calculations are the Gauss points given by the 2-point Gauss quadrature rule. It is apparent that the points chosen have to be identified in the normalized \(r, s, t\)-coordinate system. A procedure to achieve this is fully derived in section 4.5.

### 4.4 Consistent Nodal Loads

In the time-dependent analysis of the bridge structure, the stress in the prestressing steel changes during each time interval. The changes in stress have to be applied to the structure at the end of the time interval, by way of nodal loads on the finite element mesh. The total change in stress during a given time interval can be considered to have two components and is given by

\[
\Delta \sigma_s = \Delta \sigma_r + \Delta \sigma_d
\]  
(4.13)

where

\[
\Delta \sigma_s = \text{change in strand stress}
\]
\[
\Delta \sigma_r = \text{change in stress due to steel relaxation}
\]
\[
\Delta \sigma_d = \text{change in stress due to deformation}
\]

A method for evaluating the change in stress due to steel relaxation is described in section 4.6. The change in stress due to deformation is calculated from \(\Delta \epsilon_s\) using a typical stress-strain curve shown for prestressing steel in Figure 4.3. Unloading and reloading to previously attained stresses on the curve are considered to occur parallel to the initial elastic modulus.
Figure 4.3: Typical Stress-Strain Curve for Prestressing Steel
Consider a strand segment with cross sectional area $A_s$. Since changes in stress can be considered to be initial stresses at the beginning of the next time interval, analogous to equation (2.7d), we can write

$$\{Q_s\} = - \int_\xi \{B_s\} \Delta \sigma_s A_s \frac{dS}{d\xi} d\xi$$

(4.14)

where $\{Q_s\}$ is the vector of nodal loads on the parent element. Noting that $dS/d\xi = J^*$, equation (4.14) is rewritten as

$$\{Q_s\} = - \int_\xi \{B_s\} \Delta \sigma_s J^* A_s d\xi$$

(4.15)

The evaluation of the above integral is performed numerically using the 2-point Gauss quadrature rule.

The contribution to the stiffness of the parent element from the strand segment it contains must be evaluated. To achieve this an expression, similar to equation (2.8) for a single element, based on the principle of virtual work can be written. Let $[K_s]$ be the strand segment stiffness contribution to the parent element. Then

$$[K_s] = \int_\xi \{B_s\} E'_s < B_s > \frac{dS}{d\xi} A_s d\xi = \int_\xi \{B_s\} E'_s < B_s > J^* A_s d\xi$$

(4.16)

In the above equation, the modulus $E'_s$ is equal to the modulus of elasticity of the prestressing steel minus that of the concrete in the parent element, i.e.

$$E'_s = E_s - E_c$$

(4.17)

Thus, the area of the concrete replaced by steel does not contribute to the stiffness of the parent element.

A few points need to be made here. If we are considering the time interval $\Delta t_n = t_n - t_{n-1}$, then the change in stress at $t_n$ is the sum of the relaxation occurring during $\Delta t_n$ and the change in stress due to deformation at $t_{n-1}$. Therefore, the deformation stress is calculated from the change in strain occurring during the previous time interval, which should be calculated from the geometry of the strand.
prior to $t_{n-1}$. However, after $t_{n-1}$, the structure has a new geometry and this implies a new geometry for the strand at time $t_n$. The computed nodal forces have to be applied on to this new geometry and the stiffness contributions must be based on this geometry. To do this, the positions of the strand nodes at $t_n$ can be calculated from those at $t_{n-1}$ by

$$
\begin{bmatrix}
  x^* \\
  y^* \\
  z^*
\end{bmatrix}_{t_n} = \begin{bmatrix}
  x^* \\
  y^* \\
  z^*
\end{bmatrix}_{t_{n-1}} + [N]\{\Delta u\}_{t_{n-1}}
$$

(4.18)

where $[N]$ is the shape function matrix (of size $3 \times 60$) for the parent element as defined in equation (2.1). $\{\Delta u\}_{t_{n-1}}$ is the vector of incremental nodal displacements occurring at $t_{n-1}$. The matrix $[N]$ is evaluated at the $(r, s, t)$ coordinates of point $(x^*, y^*, z^*)$ and $J^*$ for use in equation (4.15) and (4.16) is computed from the current coordinates at time $t_n$.

### 4.5 Determination of Normalized Coordinates of a Point on the Strand

Let the global coordinates of a point $P$ on a strand segment be given by $(x_p, y_p, z_p)$. The corresponding $(r, s, t)$ coordinates of point $P$ are the roots of the equation:

$$
F(r, s, t) = [N]\{x\} - \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
$$

(4.19)

$x$ in the above equation is a column vector defined as

$$
\begin{bmatrix}
  r \\
  s \\
  t
\end{bmatrix}
$$

(4.20)

To find the $(r, s, t)$ coordinates of $P$, the fixed-point problem

$$
\{G(x)\} = x
$$

(4.21)

is set up by defining
\( \{ G(x) \} = x - [A(x)]^{-1}\{F(x)\} \) \hspace{1cm} (4.22)

The matrix \([A(x)]^{-1}\) is chosen to obtain quadratic convergence to the solution \(p\) of the fixed-point problem where

\[
p = \begin{pmatrix}
    r_p \\
    s_p \\
    t_p
\end{pmatrix}
\] \hspace{1cm} (4.23)

Let \([A(x)]^{-1}\) have entries \(b_{ij}(r,s,t)\) with \(i = 1,2,3\) and \(j = 1,2,3\). \(\{F(x)\}\) and \(\{G(x)\}\) can now be represented as

\[
\{F(x)\} = \begin{pmatrix}
    f_1(r,s,t) \\
    f_2(r,s,t) \\
    f_3(r,s,t)
\end{pmatrix}
\] \hspace{1cm} (4.24)

and

\[
\{G(x)\} = \begin{pmatrix}
    g_1(r,s,t) \\
    g_2(r,s,t) \\
    g_3(r,s,t)
\end{pmatrix}
\] \hspace{1cm} (4.25)

It can be proved (Burden and Faires (1985)) that if \(p\) is the solution to the fixed-point problem \(\{G(x)\} = x\), quadratic convergence is achieved if:

(i) \(\frac{\partial g_i}{\partial x_j}\) is continuous on \(M_\delta = \{x \mid \|x - p\| < \delta\}\) for each \(i = 1,2,3\) and \(j = 1,2,3\).

(ii) \(\frac{\partial^2 g_i}{\partial x_j \partial x_k}\) is continuous near \(p\) for each \(i = 1,2,3; j = 1,2,3; k = 1,2,3\).

(iii) \(\frac{\partial g_i}{\partial x_j}\) = 0 for each \(i = 1,2,3\) and \(j = 1,2,3\).

Using equations (4.22) and (4.23) and expressing \([A(x)]^{-1}\) in terms of its components, the fixed-point problem in equation (4.22) is

\[
g_i(x) = x_i - \sum_{j=1}^{3} b_{ij}(x)f_i(x)
\] \hspace{1cm} (4.26)

Taking partial derivatives with respect to each component of \(x\), we have
\[
\frac{\partial g_i(x)}{\partial x_k} = \begin{cases} 
1 - \sum_{j=1}^{3} \left( b_{ij}(x) \frac{\partial f_j(x)}{\partial x_k} + \frac{\partial b_{ij}(x)}{\partial x_k} f_j(x) \right); & i = k \\
- \sum_{j=1}^{3} \left( b_{ij}(x) \frac{\partial f_j(x)}{\partial x_k} + \frac{\partial b_{ij}(x)}{\partial x_k} f_j(x) \right); & i \neq k
\end{cases}
\] (4.27)

To satisfy condition (iii), for \( i = k \);

\[
\sum_{j=1}^{3} b_{ij}(p) \frac{\partial f_j(p)}{\partial x_i} = 1
\] (4.28)

and for \( i \neq k \);

\[
\sum_{j=1}^{3} b_{ij}(p) \frac{\partial f_j(p)}{\partial x_k} = 0
\] (4.29)

Combining equations (4.28) and (4.29) and writing them in matrix form yields

\[
\begin{bmatrix}
b_{11}(p) & b_{12}(p) & b_{13}(p) \\
b_{21}(p) & b_{22}(p) & b_{23}(p) \\
b_{31}(p) & b_{32}(p) & b_{33}(p)
\end{bmatrix}
\begin{bmatrix}
\frac{\partial f_1}{\partial r} & \frac{\partial f_1}{\partial s} & \frac{\partial f_1}{\partial t} \\
\frac{\partial f_2}{\partial r} & \frac{\partial f_2}{\partial s} & \frac{\partial f_2}{\partial t} \\
\frac{\partial f_3}{\partial r} & \frac{\partial f_3}{\partial s} & \frac{\partial f_3}{\partial t}
\end{bmatrix}
= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\] (4.30)

or

\[
[A(x)]^{-1} [B(x)] = [I]
\] (4.30a)

From the definition of the function \( F(r,s,t) \), we see that the matrix \([B(x)]\) in equation (4.30) is the transpose of the Jacobian matrix for the transformation of coordinates \((x,y,z)\) to \((r,s,t)\). Therefore \([B(x)] = [J]^T\), and from equation (4.30a) it follows that \([A(x)] = [B(x)]\). The coefficient matrix \([A(x)]^{-1}\) is now fully defined as

\[
[A(x)]^{-1} = [J^{-1}]^T
\] (4.31)

The fixed-point problem is reduced to finding the solution \(x\) for

\[
\{G(x)\} = x - [J^{-1}]^T \{F(x)\}
\] (4.32)

Fixed-point problems can be solved by setting up a functional iteration procedure. Set up \(x^{(0)}\) as a starting value and generate \(x^{(k)}\) for \( k \geq 1 \) as
\[ x^{(k)} = \{ G(x^{(k-1)}) \} = x^{(k-1)} - [J^{-1}_{(k-1)}]^T \{ F(x^{(k-1)}) \} \] (4.33)

or, more explicitly as
\[
\begin{bmatrix}
  r \\
  s \\
  t 
\end{bmatrix}^{(k)} = \begin{bmatrix}
  r \\
  s \\
  t 
\end{bmatrix}^{(k-1)} - [J^{-1}_{(k-1)}]^T \left( [N] \{ x^{(k-1)} \} - \begin{bmatrix}
  x_p \\
  y_p \\
  z_p 
\end{bmatrix} \right) \] (4.34)

The Euclidean norm can be used as the convergence criterion for the iteration. It is apparent that conditions (i) and (ii) are satisfied because \([N]\) and \([J]\) consist of shape functions and derivatives of shape functions of the parent element which are continuous over the element.

The iterative procedure just outlined can be time-consuming, especially for a large mesh with many tendons. Nevertheless, the normalized coordinates of a point never change in the analysis and have to be determined only at the start.

### 4.6 Stress Relaxation in Prestressing Steel

The phenomenon of relaxation is defined as the loss of stress over time of a material held at constant strain. Relaxation is another manifestation of the creep phenomenon and in fact for a given material, the creep compliance and the relaxation function can be quantitatively related. This is a basic law of linear viscoelasticity. In prestressed concrete structures, the geometry (length) of tendons fluctuates due to superimposed loads, creep, shrinkage and temperature variations. Nevertheless, the freedom of the deformation of tendons is curtailed and hence the stress in them can change due to relaxation. The second component of the change in stress in prestressing steel in equation (4.13) is attributed to the phenomenon of relaxation and its quantification is necessary.

Magura, Sozen and Siess (1964) carried out 57 tests on stress relaxation of prestressing steel by vibrating wires of constant length at different times and relating the tension in the wires to the frequency of vibration observed. They used data from these tests and 447 tests performed by others to obtain an empirical expres-
sion for the value of stress in the wire at different times after initial tensioning. The stress was related to the initial stress and the 0.1% offset yield stress as:

\[
\frac{f_s}{f_i} = 1 - \frac{\log t}{10} \left( \frac{f_i}{f_y} - 0.55 \right) ; \quad \frac{f_i}{f_y} \geq 0.55
\]

(4.35)

where \( f_s \) = stress at time \( t \)

\( f_i \) = initial stress immediately after tensioning

\( f_y \) = 0.1% offset yield stress

\( t \) = time in hours after tensioning.

The logarithm in the above equation is taken to the base 10.

In the case of pretensioned structures, the loss in the steel occurring before release must be subtracted from the total loss predicted for the stress at release. Therefore, for a wire tensioned at time zero and released at time \( t_r \), equation (4.35) is modified as:

\[
\frac{f_s}{f_i} = 1 - \left( \frac{\log t - \log t_r}{10} \right) \left( \frac{f_i}{f_y} - 0.55 \right)
\]

(4.36)

The above equations are valid only for the condition where strain is kept constant and \( f_i \) is the applied stress. In bridge girders, however, the stress keeps changing with time and the relaxation losses are required at the end of every time interval in the analysis. The use of equation (4.35) or (4.36) is therefore modified according to a procedure developed by Hernandez and Gamble (1975). Figure 4.4 graphically illustrates the procedure.

Referring to Figure 4.4, assume that at time \( t_0 = 0 \) the stress in the steel is the initial stress \( f_{i0} \). From time \( t_0 \) to \( t_1 \) the prestress drops by an amount \( \Delta f_{r1} \) due to relaxation. The quantity \( \Delta f_{r1} \) is equal to \( f_s - f_{i0} \) and can be calculated from equation (4.35). At time \( t_1 \), the stress drops additionally to \( f_{s1} \) due to causes other than relaxation. \( \Delta f_{r2} \), which is the relaxation occurring during the time interval from \( t_1 \) to \( t_2 \) is calculated as follows:
Figure 4.4: Stress Relaxation under Varying Prestress
1. Based on $f_{s1}$, calculate a fictitious initial stress $f_{i1}$ at time $t_0$ such that $f_{s1}$ applied at $t_0$ would result in a relaxation to $f_{s1}$ at time $t_1$. This can be done using equation (4.35).

2. From equation (4.35), calculate $\Delta f_{-2}$ occurring during $t_2 - t_1$ with $f_{i1}$ as the initial prestress.

The above procedure can be repeated for every time step.
Chapter V
Analysis of the Bridge System

The previous three chapters described three different aspects pertaining to the analysis of bridge superstructures. Chapter 2 highlighted the stiffness method for the finite element of structures using three-dimensional 20-node isoparametric elements. In chapter 3, the mathematical formulations of the various strains contributing to the time-dependent behavior of concrete have been presented. The finite element representation of prestressing strands and the calculation of changes in stress and strain within the strands were described in chapter 4. In this chapter, the overall algorithm for use with the stiffness method in performing the time-dependent step-by-step analysis is outlined. The accuracy of the algorithm is dependent on the time intervals into which the total time frame of the analysis is divided. A rational choice of time intervals is also presented in this chapter. Finally, inherent in the analysis of bridges with cast-in-place concrete deck slabs supported by precast, pretensioned concrete girders, are special considerations which are explained at the end of the chapter.

5.1 Step-by-Step Time-Dependent Analysis Algorithm

The overall analysis is now described in the form of an algorithm. This is being done in two parts. In the first part, the operations required for the first time step of the analysis are listed. The second part deals with the procedure during an arbitrary time interval $\Delta t_n = t_n - t_{n-1}$.

At the start of the analysis (i.e. time $t_0$), the procedure is summarized as follows:

1. For each type of concrete, set up the following material properties:
$l_c$ - time length of curing in days

$t_{in}$ - time at which the concrete type is added
to the structure and loaded

$f'_{c28}$ - 28-day compressive strength of concrete

$\nu$ - Poisson's ratio

$\alpha$ - coefficient of thermal expansion.

2. Based upon the creep model chosen, set up the inverse retardation time $\lambda_i$, and evaluate the aging coefficients $a_i(t')$ for different ages of loading.

3. At each integration point within each element, initialize the following variables:

$e_i^*$ - hidden state variables

$\{\sigma\}$ - stresses

$\{\epsilon\}$ - strains

$\{\epsilon''\}$ - pseudo inelastic strains.

4. Compute the elastic modulus $E_0$ at time $t_0$ for each concrete type that exists on the structure. In this analysis, the concrete type at the start of the analysis corresponds to girder concrete. The calculation of the initial elastic modulus is made using the equations given by the creep procedure being used.

5. Apply the external loads to the structure. In the present analysis, the only loads acting on the girder arise from its own weight.

6. Calculate the loads applied to the girders by the pretensioning strands. The geometry of each prestressing strand segment (section 4.2) must be given as input to the analysis and this can be achieved using the finite element mesh generator. The analysis for prestressing at time $t_0$ requires the evaluation
of the normalized coordinates of each strand node as explained in section 4.5. Then the existing prestress in each strand segment contained in an element must be converted into consistent nodal loads as described in section 4.4 (equation 4.15). The existing value of the prestress is not the initial prestressing as will be explained in subsection 5.3.1.

7. Calculate and include the stiffness contributions of prestressing strand segments in the overall stiffness matrix as explained in section 4.5.

8. Apply nodal loads due to initial shrinkage strains as described in subsection 5.3.1 and as calculated from equation (2.7c).

9. Set up the global equilibrium equations

\[ [K]\{r\} = \{R\} \]  

and solve them to obtain the global node displacement vector \{r\}. Based on the nodal displacements, calculate the strains and stresses due to the first loading.

The algorithm that is applied during an arbitrary time interval \( \Delta t_n = t_n - t_{n-1} \) is now summarized. At time \( t_{n-1} \) it is assumed that the displacements \{\Delta r\} for each element at the nodal points are known. The following steps are performed for each element:

**Step 1:** For each integration point compute:

(a). The strain increment that occurred during the previous time interval \( \Delta t_{n-1} = t_{n-1} - t_{n-2} \) from

\[ \{\Delta \varepsilon_{n-1}\} = [B]\{\Delta y\} \]  

where \{\Delta y\} is related to \{\Delta r\} via the element shape functions.

(b). The total strain:

\[ \{\varepsilon_{n-1}\} = \{\varepsilon_{n-2}\} + \{\Delta \varepsilon_{n-1}\} \]
(c). The total stress:

\[ \{\sigma_{n-1}\} = E_n[D_0][\{\varepsilon_{n-1}\} - \{\varepsilon''_{n-1}\}] \]  \hspace{1cm} (5.4)

where \( \{\varepsilon''_{n-1}\} \) are the pseudo inelastic strains that occurred prior to \( t_{n-1} \).

(d). The stress increment in time interval \( \Delta t_{n-1} \):

\[ \{\Delta\sigma_{n-1}\} = \{\sigma_{n-1}\} - \{\sigma_{n-2}\} \]  \hspace{1cm} (5.5)

**Step 2:** Add the global displacements \( \{\Delta r\} \) to the nodal coordinates to obtain the new geometry of the structure.

**Step 3:** In this step, the following operations must be performed on the prestress strand segments within their parent elements:

(a). Calculate the strains \( \Delta\varepsilon_s \) occurring within each strand segment as elucidated in section 4.3.

(b). Redefine the geometry of each strand segment at time \( t_{n-1} \) from its position at \( t_{n-2} \) using equation (4.18).

(c). Based on the strain change \( \Delta\varepsilon_s \) calculate the change in stress \( \Delta\sigma_d \) for each segment from the stress-strain curve for prestressing steel.

(d). Based on the strain-displacement vector calculated in performing step (a), evaluate the contribution to the element stiffness from each strand segment.

**Step 4:** Give the next time step \( t_n \) as input along with the ambient temperature occurring at \( t_n \). Compute \( \Delta t_n = t_n - t_{n-1} \).

**Step 5:** Redefine the temperature profile on the depth of the cross section and calculate the temperature \( T_n \) at each integration point within an element.

**Step 6:** Calculate \( E(t_n) \) at the end of the current time interval.

**Step 7:** Calculate the creep strain increment occurring during \( \Delta t_n \). This is achieved by implementing the following sub-steps for each integration point:
(a). Update the hidden state variables $\varepsilon_{i,n}$ according to equation (3.105). In that equation $T_{n-1}$ is the temperature at the integration point at time $t_{n-1}$. $a_i(t_{n-1})$ are the aging coefficients chosen at $t_{n-1}$ based on the reduced or effective age of concrete at that time. The effective age is computed using equation (3.101) over the previous time interval.

(b). Calculate the creep strain increment $\{\Delta \varepsilon_n^c\}$ that will occur during $\Delta t_n$ from equation (3.104).

**Step 8:** Evaluate the incremental shrinkage strain vector $\{\Delta \varepsilon_n^s\}$ as given by equation (3.106), wherein the value of $\Delta \varepsilon_{sh,n}$ depends on the creep and shrinkage recommendations chosen and on times $t_{n-1}$ and $t_n$.

**Step 9:** Compute the temperature strain increment vector using equation (3.108).

**Step 10:** As per equation (3.111), calculate the strain increment $\{\Delta \varepsilon_n^d\}$ to account for the change in elastic modulus that occurs over $\Delta t_n$.

**Step 11:** Calculate the loss of stress due to relaxation for each strand segment using equation (4.35) along with the procedure of section 4.6. In the use of that procedure, the current stress in the strand segment is given by

\[ \sigma_{s,n} = \sigma_{s,n-1} + \Delta \sigma_d \]  

(5.6)

where $\Delta \sigma_d$ is as obtained in Step 3. Once the relaxation loss $\Delta \sigma_r$ is known, the current strand stress is

\[ \sigma_{s,n} = \sigma_{s,n} + \Delta \sigma_r \]  

(5.7)

and the total change in strand stress during the interval from $t_{n-1}$ to $t_n$ is

\[ \Delta \sigma_s = \Delta \sigma_d + \Delta \sigma_r \]  

(5.8)

**Step 12:** Based on $\Delta \sigma_s$ calculated for each strand segment, set up the global nodal incremental load vector by assembling the element nodal load vectors evaluated using equation (4.15). Let this vector be denoted by $\{\Delta R_n\}$. 
Step 13: Add any external loads applied at $t_n$ to the incremental nodal load vector $\{\Delta R_n\}$.

Step 14: Set up the incremental pseudo inelastic strain vector as

$$\{\Delta \epsilon_n''\} = \{\Delta \epsilon_n^e\} + \{\Delta \epsilon_n^a\} + \{\Delta \epsilon_n^T\} \quad (5.9)$$

and calculate the incremental nodal load vector as

$$\{\Delta R_n\} = \{\Delta R_n\} + \sum_e \int_V [B]^T E_n [D_o] \{\Delta \epsilon_n''\}_e dV \quad (5.10)$$

where the summation is performed over all elements.

Step 15: Add the incremental strain vector calculated in Step 10 to get the total incremental pseudo inelastic strain vector as

$$\{\Delta \epsilon_n''\} = \{\Delta \epsilon_n''\} + \{\Delta \epsilon_n^A\} \quad (5.11)$$

Step 16: Assemble the structure stiffness matrix from the element stiffness matrices using the equation:

$$[K_n] = \sum_e \int_V [B]^T E_n [D_o] [B] dV + \sum_{e^*} \int_{\xi} \{B_s\} E'_s \lt B_s \gt \ast A_s d\xi \quad (5.12)$$

where $\sum_{e^*}$ represents a summation over all elements containing prestressing strand segments.

Step 17: Solve for the incremental displacement vector $\{\Delta r\}$ from

$$[K_n] \{\Delta r\} = \{\Delta R_n\} \quad (5.13)$$

The analysis from time $t_{n-1}$ to $t_n$ is thus complete.

For the final time interval, after performing the 17 steps listed above, steps 1 through 3 must be repeated to obtain deflections and stresses at the end of the analysis.
5.2 Choice of Time Intervals

The length of any time interval in the analysis is an important factor in the outcome of the evaluated behavior of the structure. A large time interval implies a large incremental pseudo inelastic strain vector resulting in a large incremental nodal load vector. This in turn results in large non-linearities that should be avoided in a step-by-step linear approximation to nonlinear behavior. A wise choice of the time intervals is one that minimizes the loss in accuracy due to non-linearities.

The two strain increments that are most profoundly affected by the length of a time interval are those of creep and shrinkage. An inspection of typical creep and shrinkage curves indicates that if the lengths of time intervals are chosen so that they are small when there are large increases in strain, the accuracy of the analysis would improve. Also, large increases in applied loads cause large increases in creep strains and time intervals should be small whenever external loads are applied.

In light of the fact that the creep coefficient is being approximated by a Dirichlet series which is basically the sum of inverse exponential terms, a time interval distribution that is constant in the logarithmic scale seems to be an appropriate choice. Such a distribution implies short time intervals soon after loading when the rates of change of creep and shrinkage strains are high. Noting that the integral in equation (3.66) is approximated by the trapezoidal rule which has an error of the order of the length of the time interval, a sufficient number of intervals is required. Twenty time intervals per decade appear to be sufficient. To obtain the elastic response immediately after application of a load, a time interval of zero length is needed. Then the length of the time intervals can be increased up to the time of the next external loading. This choice of time intervals will not result in any convergence problems regarding the jumps in creep strains at each time step. Except for time steps at which external loads are applied, the change in stress during any time interval is mainly due to prestress losses and hence very small. This causes small increases in creep strains over the next time interval. Also, the exponential term relating hidden state variables in equation (3.86) is always less than 1 and hence
numerical instabilities are never magnified from one time interval to the next, as long as changes in stress are small.

5.3 Special Analysis Considerations

The algorithm described in section 5.1 is applicable for every time interval. However, certain extra considerations must be given to the time steps just prior to the release of prestress (the starting time of the analysis) and at the times of slab casting and hardening. These considerations are now dealt with.

5.3.1 Analysis at Transfer of Prestress

The age of concrete in the girders at the time of the release of prestress plays an important part in the behavior of the bridge superstructure. Nowadays, most prestressed concrete girders are cured using the steam curing process and the release of prestress is done from about 18 hours to 3 days after casting. From the time of casting to the time of release of prestress two phenomena occur. The prestressing steel experiences a loss of stress due to relaxation and the girder concrete experiences shrinkage. As a result, the earlier the age at transfer, the higher will be the value of the stress in the steel strands. This higher stress in turn causes larger creep strains in concrete after release resulting in higher subsequent losses of prestress. It is therefore necessary to correctly estimate the state of stress in concrete immediately after release.

The analysis at the time of transfer of prestress is performed by applying the following loads to the girder:

1. Nodal loads due to the prestressing force. The prestressing force equals the initial force minus the loss in prestress due to relaxation up to the time of release. The calculation of the loss due to relaxation is described in section 4.6 and the calculation of nodal loads due to the net force is described in section 4.4.
2. Nodal loads due to shrinkage of concrete upto transfer by treating the shrinkage strain as a pseudo inelastic strain.

3. Dead weight of the girder.

The girder can be considered to be simply supported because it will camber due to the effect of the eccentric prestressing.

A certain approximation is inherent in the application of the shrinkage strain in concrete prior to release as an initial pseudo inelastic strain at release. To gain an understanding of this approximation, consider a concrete specimen that is concentrically prestressed, as shown in Figure 5.1(a). The specimen is depicted just prior to release. Let $\varepsilon_{sh}$ be the shrinkage strain that occurs in concrete from the time of its casting to the time of prestress release. The equivalent load acting on the concrete due to this shrinkage strain can be calculated as

$$ P_{sh} = A_c E_c \varepsilon_{sh} $$

(5.14)

where $A_c = \text{area of the concrete cross section}$

$E_c = \text{elastic modulus of concrete at the age of release}$

When the load $P_{sh}$ is applied to the composite concrete and steel prism, a change in strain $\Delta\varepsilon_{sh}$ is experienced by the prism which causes a decrease in the prestressing force by an amount:

$$ \Delta P_s = -A_s E_s \Delta\varepsilon_{sh} $$

(5.15)

where $A_s = \text{area of prestressing steel}$

$E_s = \text{elastic modulus of the steel}$

The force in the prestressing steel between the prism and the abutment increases by the amount $\Delta P_s$ as shown in the free body diagram of Figure 5.1(b). In that figure, $P_r$ is the prestress force existing after the steel relaxation prior to release has
(a) Concentrally Loaded Pre-tensioned Concrete Prism

(b) Equivalent Load Due to Shrinkage and Free Body Diagram

Figure 5.1: Concrete Prism at Release of Prestress
occurred. The change in strain $\Delta \varepsilon_{sh}$ in the composite prism can now be calculated by simultaneously applying $P_{sh}$ and $\Delta P$ to the prism to yield

$$\Delta \varepsilon_{sh} = \frac{P_{sh} + \Delta P}{E_c A_c + E_s A_s} = \frac{A_c E_c \varepsilon_{sh}^c - A_s E_s \Delta \varepsilon_{sh}}{E_c A_c + E_s A_s} \tag{5.16}$$

Using equation (5.16), $\Delta \varepsilon_{sh}$ can be solved for as

$$\Delta \varepsilon_{sh} = \frac{\varepsilon_{sh}^c}{1 + 2m} \tag{5.17}$$

with $m = A_s E_s / A_c E_c$.

The axial force $P_c$ in the concrete prism can now be calculated using $\Delta \varepsilon_{sh}$ as the total strain and $\varepsilon_{sh}^c$ as the initial pseudo inelastic strain due to shrinkage as

$$P_c = A_c E_c (\Delta \varepsilon_{sh} - \varepsilon_{sh}^c) \tag{5.18}$$

Using equations (5.15) and (5.17), equation (5.18) can be manipulated to yield

$$P_c = 2 \Delta P \tag{5.19}$$

which, as shown in Figure 5.1(b), satisfies static equilibrium.

However, in the present analysis, the calculation of $\Delta \varepsilon_{sh}$ does not include the contribution due to the decrease in prestress given in equation (5.15). Instead, equation (5.16) is simply restated as

$$\Delta \varepsilon_{sh}^a = \frac{A_c E_c \varepsilon_{sh}^c}{E_c A_c + E_s A_s} = \frac{\varepsilon_{sh}^c}{1 + m} \tag{5.20}$$

where $\Delta \varepsilon_{sh}^a$ is the approximate total strain calculated. Therefore, an error $ERR$ is introduced which, when evaluated as a percentage of the approximate total strain, is given by

$$ERR = \left( \frac{\Delta \varepsilon_{sh} - \Delta \varepsilon_{sh}^a}{\Delta \varepsilon_{sh}^a} \right) \times 100 = \left[ 1 - \left( \frac{1 + m}{1 + 2m} \right) \right] \times 100 \tag{5.21}$$

Noting that the Maclaurin series expansion of $(1 + m)/(1 + 2m)$ is
the percentage error at most equals 100m for \( m \leq 1 \). In girders used in bridges, the value of \( m \) does not exceed 0.06 which implies a 6% error in the calculation of the total strain due to shrinkage. Consider a concrete prism with \( m = 0.06 \), \( A_p = 1.0 \text{ sq.in.} \) and \( E_p = 30 \times 10^6 \text{ psi} \), subjected to an initial prestressing force of 180000 lbs. and experiencing an initial shrinkage strain equal to \(-2 \times 10^{-4}\). The percentage prestress loss calculated using equations (5.15) and (5.18) equals 2.98%. If the approximate total strain due to shrinkage from equation (5.20) is used, the prestress loss calculated is 3.14%. Thus the error involved in using the approximate total strain is only 0.16% of the initial prestress force and it can be concluded that the approximate solution is acceptable.

5.3.2 Analysis at the Times of Slab Casting and Hardening

At the time of casting of the deck slab, the prestressed concrete girders immediately experience the dead weight of the slab. This results in a large reduction of the concrete stress at the level of the prestress resulting in a gain in the prestressing force. This gain represents a large change in the prestress as compared with time intervals prior to slab casting, causing large increases or decreases in creep strains. To accommodate this, a very short time interval must be used immediately after the slab is cast. The analysis allows for the use of a time interval of zero length for this purpose. The elements that comprise the slab are assumed not to contribute to the stiffness of the overall bridge section.

After the slab has been poured and cast, the concrete in it hardens. There is no specific time that can be identified at which true composite action of the girders and the deck slab commences. This is especially so for unshored construction. A reasonable estimate of the time at which composite action can be assumed to occur would be when the curing of the slab concrete ends and the formwork is removed. For moist cured concrete, sufficient strength is developed after 7 days from the time of casting. It is assumed in the analysis that the amount of shrinkage occurring in
the slab concrete during the curing period is negligible.

In the present analysis, the changes in stress occurring during a time interval are applied as initial loads at the start of the next time interval. This means that during a particular interval, the girders experience a newly applied load corresponding to a prestress loss or gain that occurred during the previous time interval and the total stress at any point (from equation (5.4)) due to prestressing is the sum of stress due to the initial prestress and due to subsequent changes in prestress. If the time interval prior to assumed composite action is small (less than 1 day), the slab concrete will experience a stress distribution caused only by superimposed live loads, when the section becomes composite. The following steps must be taken at the time \( t_h \) of assumed slab hardening:

1. Store the geometry of the girders at \( t_h \) in terms of the nodal point coordinates.

2. Using the geometry determined in step 1, allow the slab to take on a value of the modulus of elasticity based on the age of its concrete, so that a composite section is the result.

3. Apply any loads that are known to occur at time \( t_h \).

4. Impose the temperature distribution on the cross section. The slab temperatures are treated as initial temperatures at time \( t_h \).

5. Initialize the hidden state variables at the integration points in the slab elements, as well as the shrinkage strain increment.

6. Perform the stiffness analysis on the composite section and determine the new geometry and new state of stress.

Live loads may be applied to the composite section whenever they are assumed to occur.
Chapter VI
Application and Verification of Analytical Model

The analytical procedure developed and explained in the previous chapters has been coded into a FORTRAN program named PCBIDGE for use on the IBM 3090 computer. A flow diagram of the program is given in the Appendix A. Three numerical studies using PCBIDGE were carried out to:

1. Verify the validity and accuracy of the analytical procedure developed in this study to predict the long-term behavior of prestressed concrete bridges.

2. Gain an understanding of how the three standard creep and shrinkage models affect predicted bridge behavior.

3. Estimate the joint movements in the Krotz Springs Bridge which was chosen for experimental evaluation.

The first numerical study was performed on prestressed concrete girders for which experimental results are available and the second concentrated on a heuristic girder-slab system. The third analysis was applied to the bridge at Krotz Springs, Louisiana, and comparisons of predicted movements are made with measured values.

6.1 Sinno-Furr Tests

Analyses were performed on simply supported girders to validate the applicability of the model to simple systems. Sinno and Furr (1970) tested a series of precast, pretensioned, simply supported beams and studied the deformations and the prestress loss with time after release. Live loads were not considered in the experiment. For the present numerical study, one of the beams tested – Beam L4-5 – by Sinno and Furr was chosen. The elevation and cross section views of the beam are shown
in Figure 6.1. A finite element mesh which takes advantage of symmetry was employed for the analysis and its configuration is also shown in Figure 6.1. The mesh has 33 sets of elements along half the girder length with each set consisting of 5 elements through the girder depth. The maximum aspect ratio is 4.

Beam L4-5 was fabricated using sand-lightweight concrete (weight = 120 pcft) with an initial elastic modulus of $2.95 \times 10^6$ psi. The 28-day compressive strength was 6166 psi. Prestressing consisted of 270$k$ strands with an elastic modulus of $28.5 \times 10^6$ psi and a yield strength of 230 ksi. The steel area equaled 3.27 sq.in. and strands were draped as shown in Figure 6.1, with an end eccentricity of 7.2" and a midspan eccentricity of 9.6".

In their experiment, Sinno and Furr measured creep and shrinkage strains in test cylinders at different times and analytically fit curves through the observed values. They expressed the creep strain per unit stress for a loading age of 1 day as

$$
\epsilon^c(t) = \frac{0.340t}{5 + t} \times 10^{-6}
$$

and the shrinkage strain as

$$
\epsilon^s(t) = -\frac{175t}{4 + t} \times 10^{-6}
$$

The present numerical study used the mesh shown in Figure 6.1 and creep and shrinkage strains were predicted using (a) equations (6.1) and (6.2); (b) ACI-209 model; (c) BP2 model and (d) CEB-FIP model. Sixteen time steps were chosen as the age at release of prestress and 2.5, 5, 10, 15, 20, 30, 40, 50, 70, 90, 120, 180, 220, 280 and 300 days after release. The experimental and analytical results for the midspan deflections and prestress loss as a percentage of the initial prestress are listed in Table 6.1. In the table, values under the label “Sinno-Furr Expressions” are those obtained using equations (6.1) and (6.2). As indicated by the tabulated values, analytical results are in close agreement with measured values. This example provides a good qualitative understanding of the effects
Figure 6.1: Sinno-Furr Girder: Elevation, Cross section and Mesh Configuration
Table 6.1: Comparisons with Sinno-Furr Experiment

<table>
<thead>
<tr>
<th>Days After Release</th>
<th>Measured</th>
<th>ACI</th>
<th>% Error</th>
<th>BP2</th>
<th>% Error</th>
<th>CEB</th>
<th>% Error</th>
<th>Sinno-Furr Expressions</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.33</td>
<td>1.34</td>
<td>+0.75</td>
<td>1.33</td>
<td>0.0</td>
<td>1.31</td>
<td>-1.50</td>
<td>1.33</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>1.82</td>
<td>1.81</td>
<td>-0.55</td>
<td>1.71</td>
<td>-6.3</td>
<td>1.81</td>
<td>-0.55</td>
<td>1.80</td>
<td>-1.0</td>
</tr>
<tr>
<td>30</td>
<td>1.98</td>
<td>1.93</td>
<td>-2.5</td>
<td>1.90</td>
<td>-4.0</td>
<td>1.98</td>
<td>0.0</td>
<td>2.01</td>
<td>+1.5</td>
</tr>
<tr>
<td>90</td>
<td>2.06</td>
<td>2.00</td>
<td>-2.9</td>
<td>2.12</td>
<td>+2.9</td>
<td>2.01</td>
<td>-2.4</td>
<td>2.21</td>
<td>+7.3</td>
</tr>
<tr>
<td>300</td>
<td>2.11</td>
<td>2.09</td>
<td>-0.95</td>
<td>2.30</td>
<td>+9.0</td>
<td>2.04</td>
<td>-3.3</td>
<td>2.28</td>
<td>+8.0</td>
</tr>
</tbody>
</table>

Prestress Losses (% of initial)

<table>
<thead>
<tr>
<th>Days After Release</th>
<th>Measured</th>
<th>ACI</th>
<th>% Error</th>
<th>BP2</th>
<th>% Error</th>
<th>CEB</th>
<th>% Error</th>
<th>Sinno-Furr Expressions</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11.73</td>
<td>11.85</td>
<td>+1.0</td>
<td>11.77</td>
<td>+0.3</td>
<td>12.54</td>
<td>+6.9</td>
<td>12.50</td>
<td>+6.6</td>
</tr>
<tr>
<td>10</td>
<td>18.57</td>
<td>17.50</td>
<td>-5.8</td>
<td>17.11</td>
<td>-7.8</td>
<td>17.78</td>
<td>-4.25</td>
<td>18.60</td>
<td>+0.2</td>
</tr>
<tr>
<td>30</td>
<td>20.43</td>
<td>20.93</td>
<td>+2.4</td>
<td>22.05</td>
<td>+7.9</td>
<td>20.19</td>
<td>-1.2</td>
<td>21.25</td>
<td>+4.0</td>
</tr>
<tr>
<td>90</td>
<td>21.03</td>
<td>23.90</td>
<td>+13.6</td>
<td>24.83</td>
<td>+18.0</td>
<td>21.27</td>
<td>+1.2</td>
<td>22.46</td>
<td>+6.7</td>
</tr>
<tr>
<td>300</td>
<td>22.65</td>
<td>26.65</td>
<td>+17.7</td>
<td>28.01</td>
<td>+23.7</td>
<td>22.83</td>
<td>+0.8</td>
<td>24.15</td>
<td>+6.6</td>
</tr>
</tbody>
</table>
of creep and shrinkage on the behavior of pretensioned girders. On release of the prestress, the beam experiences eccentric compressive forces on its cross section and its dead weight. Although some of the prestress (about 11.7%) is lost due to elastic shortening and steel relaxation prior to release, the distribution of compressive stresses causes negative moments which are larger than the positive moments due to dead load and the girder deflects upwards. The loss of prestress at midspan is higher than losses at other locations because of larger bending strains.

As the age of concrete increases, the upward deflection increases primarily due to creep. Since the concrete is in compression, compressive creep strains cause shortening in the prestress, thereby increasing prestress losses which in turn decrease the upward deflections (camber). However, the distribution of compressive stress through the depth of the girder is such that the highest stresses are at the bottom and the lowest stresses are at the top. Thus, higher creep strains occur towards the bottom of the girder, resulting in an increase in camber. Shrinkage of concrete has the effect of increasing prestress losses and decreasing camber as does the relaxation of steel. It is obvious that some factors contribute to an increase in camber while others decrease it.

A plot of the midspan camber versus time as obtained in the experiment and from the four analyses is shown in Figure 6.1. A plot of the prestress loss at midspan is given in Figure 6.3. In Figure 6.4, the creep coefficients for a loading age of 1 day are plotted against time and Figure 6.5 depicts the corresponding shrinkage strain curves. A comparison of Figures 6.2 and 6.4 indicates the dominance of the effect of creep on camber. The camber curves are similar to the creep curve for each analysis. For example, the creep curve using the CEB-FIP model indicates that most of the creep occurs within the first 30 days whereas the ACI-209 creep coefficient curve exhibits a large portion of creep subsequent to an age of 30 days. The curves for the camber reflect these trends. The shapes of the prestress loss curves are similar to their corresponding creep curves. The large differences in computed prestress losses with measured losses can be attributed to the large differences in shrinkage
strains predicted by the various models.

An inspection of Table 6.1 suggests that the camber calculated using creep strains from equation (6.1) overestimates the measured camber by as much as 8%. The reason for this observation is that the expression for creep, which is valid only for a loading age of 1 day, yields high strains soon after loading. Therefore, the initial stresses cause large creep strains which are not decreased substantially by subsequent stress decrements.

The marked differences in creep strains evident in Figure 6.4 are to be expected as reported in literature. Müller and Hilsdorf (1982) compared various prediction methods for creep of concrete with experimental data and evaluated them using statistical procedures. They concluded that errors in the prediction of creep functions are large no matter which model is used. In particular, the ACI-209 method is weak for concrete loaded at later ages and the BP2 model is erroneous for concrete loaded at an early age. The CEB-FIP method predicts values that are in good agreement with data for basic creep. However, the method is poor in estimating creep under drying conditions. Figure 6.5 shows the ACI-209 and BP2 methods overestimate shrinkage strains and the CEB-FIP method underestimates them. It should be mentioned here that despite the discrepancies existing between the three creep models, these models are the most widely accepted.

6.2 Application of Model to Heuristic Girder-Slab System

A second numerical study undertaken pertains to an AASHTO Type IV girder supporting a 7.5" slab. The cross section of the system is shown in Figure 6.6. The system was designed as per current AASHTO (1983) specifications. The system considered has a simply supported span of length 86'8". The effective width of the slab was calculated to be 105". Girder prestressing consists of 2 sets of $270k - \frac{1}{2} \phi$ strands. One straight set of 34 strands has its centroid 4.97" above the bottom of the girder. The second set of 8 strands is harped at $\frac{4}{10}$ of the span from each
Figure 6.2: Midspan Camber versus Time for the Sinno-Furr Girder
Figure 6.3: Prestress Loss at Midspan versus Time for the Sinno-Furr Girder
Figure 6.4: Creep coefficients for Sinno-Furr Girder Concrete for a loading age of 1 day
Figure 6.5: Shrinkage strains for Sinno-Furr Girder Concrete
end of the girder. Its centroid is 55.5" and 5.5" above the bottom of the girder at the ends and midspan respectively. Girder concrete has a 28-day compressive strength of 6000 psi and a unit weight of 150 pcf. The corresponding values for slab concrete are 4900 psi and 145 pcf. Girder concrete is fabricated from Type III cement while the slab concrete is made from Type I cement.

Analyses using the three creep models were performed on the system assuming an ambient relative humidity of 70% and a constant ambient temperature of 68°F. A typical construction schedule was assumed and is described as follows:

1. 0 - 3 days: Steam curing of the girder.
2. At 3 days: Release of prestress.
3. At 90 days: Slab is cast.
4. 90-97 days: Slab is moist cured.
5. At 97 days: Slab formwork is removed.

The girder and the slab are assumed to act as a composite section after 97 days.

Several mesh sizes were used in the analysis with a view to deciding upon an accurate and economical solution. A very fine mesh usually ensures sufficient accuracy but at a high cost in computer time. A mesh three times finer than the mesh utilized in this study results in a cost that is approximately 40 times higher. Too coarse a mesh, even within the limits prescribed by Cervera (1986) and reported in subsection 2.2.1, results in a poor representation of prestress tendon profiles as the system deforms. Trial runs for a few time steps resulted in the use of 20 sets of elements along the span length with a maximum aspect ratio of 9.

Analytical values for the deflection at midspan over a period of 700 days are given in Table 6.2. These values are plotted in Figure 6.7. It is evident that midspan deflections predicted by the three creep and shrinkage procedures differ significantly from each other, especially after the slab acts compositely with the
Figure 6.6: Cross section of Heuristic Girder-Slab System
girder. An insight into the overall behavior of the system can be obtained from Figures 6.8 and 6.9 which show plots of creep coefficients for girder and slab concrete and from Figures 6.10 and 6.11 which depict the corresponding shrinkage strains.

Prior to the casting of the slab, the girder deflects upwards and the deflection curves are similar to the creep coefficient curves for girder concrete. At 90 days, the camber predicted by the ACI-209, BP2 and CEB-FIP models are 1.74", 1.88" and 2.03" respectively. The higher values obtained from the CEB-FIP and BP2 methods can be attributed to higher creep coefficients over a 90 day period. As soon as the slab is cast, the girder deflects downwards under the weight of the slab and continues a slightly downward trend for a period of 7 days.

At 97 days the formwork is removed and the girder and the slab act as a composite section. Table 6.2 indicates that all the three analyses predict a downward deflection of approximately 0.76" over the period beginning just prior to slab casting and ending after the onset of composite action. The midspan deflections now become markedly different. These differences are due to the material behavior of the slab concrete. Figure 6.9 brings out the large differences in the creep coefficients while Figure 6.11 shows an even more significant variation in shrinkage strains. It is these shrinkage strains in the slab that dictate the subsequent deformation behavior of the system. The shrinkage strains in the slab increase rapidly while those in the girder increase at a slow rate. This causes differential shrinkage at the girder-slab interface and the slab shortens longitudinally forcing the girder to deflect downwards. Deflections predicted by the BP2 method are the largest while those predicted by the CEB-FIP procedure are the smallest. A levelling off of midspan deflections is evident as soon as the rate of increase of slab shrinkage strains slows down (approximately 6 months after slab casting). The deflection curve obtained using the CEB-FIP recommendations is flatter than those of the other two procedures. This can be attributed to the fact that the CEB-FIP procedure limits the amount of the total creep strain in the girder that is recoverable.
### Table 6.2: Midspan Deflections - Heuristic Girder-Slab System

<table>
<thead>
<tr>
<th>Time in days</th>
<th>ACI strains</th>
<th>BP2 strains</th>
<th>CEB strains</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
<td>1.21</td>
<td>1.21</td>
<td>1.21</td>
</tr>
<tr>
<td>4.7</td>
<td>1.32</td>
<td>1.37</td>
<td>1.40</td>
</tr>
<tr>
<td>7.8</td>
<td>1.44</td>
<td>1.51</td>
<td>1.46</td>
</tr>
<tr>
<td>13.0</td>
<td>1.57</td>
<td>1.62</td>
<td>1.56</td>
</tr>
<tr>
<td>22.0</td>
<td>1.67</td>
<td>1.69</td>
<td>1.66</td>
</tr>
<tr>
<td>36.0</td>
<td>1.72</td>
<td>1.75</td>
<td>1.78</td>
</tr>
<tr>
<td>50.0</td>
<td>1.73</td>
<td>1.80</td>
<td>1.86</td>
</tr>
<tr>
<td>75.0</td>
<td>1.74</td>
<td>1.87</td>
<td>1.98</td>
</tr>
<tr>
<td>89.8</td>
<td>1.74</td>
<td>1.88</td>
<td>2.03</td>
</tr>
<tr>
<td>90.0</td>
<td>0.98</td>
<td>1.13</td>
<td>1.26</td>
</tr>
<tr>
<td>94.0</td>
<td>0.86</td>
<td>0.99</td>
<td>1.18</td>
</tr>
<tr>
<td>96.9</td>
<td>0.81</td>
<td>0.94</td>
<td>1.18</td>
</tr>
<tr>
<td>97.0</td>
<td>0.81</td>
<td>0.94</td>
<td>1.17</td>
</tr>
<tr>
<td>98.0</td>
<td>0.79</td>
<td>0.93</td>
<td>1.16</td>
</tr>
<tr>
<td>105.0</td>
<td>0.68</td>
<td>0.85</td>
<td>1.15</td>
</tr>
<tr>
<td>120.0</td>
<td>0.56</td>
<td>0.77</td>
<td>1.14</td>
</tr>
<tr>
<td>170.0</td>
<td>0.43</td>
<td>0.63</td>
<td>1.15</td>
</tr>
<tr>
<td>250.0</td>
<td>0.38</td>
<td>0.52</td>
<td>1.16</td>
</tr>
<tr>
<td>365.0</td>
<td>0.37</td>
<td>0.46</td>
<td>1.16</td>
</tr>
<tr>
<td>435.0</td>
<td>0.37</td>
<td>0.43</td>
<td>1.17</td>
</tr>
<tr>
<td>510.0</td>
<td>0.38</td>
<td>0.41</td>
<td>1.18</td>
</tr>
<tr>
<td>550.0</td>
<td>0.38</td>
<td>0.4</td>
<td>1.18</td>
</tr>
<tr>
<td>600.0</td>
<td>0.38</td>
<td>0.39</td>
<td>1.18</td>
</tr>
<tr>
<td>650.0</td>
<td>0.39</td>
<td>0.38</td>
<td>1.18</td>
</tr>
<tr>
<td>700.0</td>
<td>0.39</td>
<td>0.38</td>
<td>1.18</td>
</tr>
</tbody>
</table>
Figure 6.7: Midspan Deflections for the Heuristic Girder-Slab System
Figure 6.8: Creep Coefficients for Girder Concrete
Figure 6.9: Creep coefficients for Slab Concrete
Figure 6.10: Shrinkage strains for Girder Concrete
Figure 6.11: Shrinkage strains for Slab Concrete
upon unloading, while the other procedures do not. This behavior has been reported in literature, eg. Smith and Goodyear (1988).

The large variations in predicted shrinkage strains (Figure 6.11) are not uncommon. El-Shafey, Jordaan and Loov (1982) reported large differences in predicted strains in their study on deflections of prestressed concrete members. Comparisons of predicted shrinkage strains with test data were performed by Bążant and Panula (1980) using experimental evidence compiled by Troxell et al. (1958) and Keeton (1965). Their findings indicate that both the ACI-209 and CEB-FIP methods often underestimate shrinkage with the CEB-FIP method being particularly poor in this regard. They attribute this underestimation to the lack of concrete composition parameters in the prediction functions for shrinkage in the CEB-FIP model.

The two analytical studies in this chapter, while verifying the validity of the analytical procedure, clearly indicate that the current creep and shrinkage prediction procedures can be improved. It is up to the engineer to decide which model would be most reliable for the concrete structure he analyzes. Improvements in strain predictions can be realized if limited creep and shrinkage data are used in the formulas presented in the creep models to calculate material parameters. For example, if in using the BP2 method, a value for $\varepsilon_{sh}$ is known at some time $t$, equation (3.37) can be manipulated to yield $\varepsilon_{sh,\infty}$, thereby eliminating the need for all the composition parameters in equation (3.40). The use of curves in the CEB-FIP procedure, however, makes such manipulations difficult.

6.3 Comparison of Analytical Results with Measurements on the Krotz Springs Bridge

A true picture of the effectiveness of the analytical model to simulate actual bridge behavior can be obtained if experimental results for the bridge at Krotz Springs, Louisiana are used as a basis for comparison. Analyses were performed on the first three continuous units of the East approachway between expansion joints 1 and 2. Figure 6.12 shows the elevation of the East approachway of the Krotz
Springs Bridge. A finite element mesh consisting of 10410 nodes and 20 elements per 72’ span was configured with a maximum aspect ratio of 8. The construction schedule was followed and ambient temperatures were assumed based on averages for different seasons. Four days for which field monitoring data are available are October 22, 1987, February 21, 1988, April 15, 1988 and June 10, 1988. Particular attention was given to these days and analytical output was sought every four hours beginning at 8:00 a.m. Ambient temperatures at these times were taken as measured in the field.

Plots of the joint movement for the four days are shown in Figure 6.13 for expansion joint 1 on the North side of the bridge at the location of the LVDT’s. The graphs for each day show the measured movement and the outcome of analyses using the three creep and shrinkage procedures. Negative values indicate a closing of the joint. All movements are presented with respect to the slab position at the joint on October 22, 1987. The ambient temperature variation is also shown in Figure 6.13. Similar plots for expansion joint 2 are shown in Figure 6.14.

An inspection of Figures 6.13 and 6.14 at first indicate fairly close agreement between measurements and analyses for the days of October 22, 1987 and February 21, 1988. The plots for the next two days, however, do not show much agreement. A closer look at measured and analytical results is warranted. At the end of June 10, 1988 a movement of —0.3” at joint 1 and a movement of +0.1” at joint 2 were measured. The corresponding values predicted by the ACI-209 model are —0.1” and —0.1”. Therefore, both the measured values and the values predicted by the ACI-209 analysis imply an expansion of the unit of 0.2” over approximately an 8-month period. The CEB-FIP model underestimates the total expansion, while the BP2 method actually suggests a contraction of the girder and slab units. A couple of observations can be made based on the above results. First of all, the measurements indicate an unequal closing at each joint. It is unlikely that the small longitudinal slope of the approachway is responsible for this. A more plausible explanation is that the bridge experienced an in-plane rotation at the deck level.
Figure 6.12: Elevation of East Approachway of the Krotz Springs Bridge
Figure 6.13: Analytical and Experimental Results for Expansion Joint 1
Figure 6.14: Analytical and Experimental Results for Expansion Joint 2
due to support restraints. Unfortunately, such rotations cannot be accounted for by the analysis. Secondly, the movements predicted by the CEB-FIP and BP2 models, follow the trends predicted by the ACI-209 model. Furthermore, the net relative movement over an entire day of the ends of the structure is close to the measured value, irrespective of the creep model employed. For example, on June 10, 1988, the measured values indicate a net expansion over the day of 0.5" at expansion joint 1 and 0.19" at expansion joint 2, adding up to an elongation of the structure equal to 0.24". Each analytical model predicts equal expansion at each end of approximately 0.095", totalling up to an expansion of 0.19". This is within 20% of the observed value.

The trend of the intra-day predictions and measurements indicate that the bridge responds primarily to temperature variations. Intra-day measurements for April 15, 1988 and June 10, 1988 differ greatly from predicted values. This is particularly the case for expansion joint 1. While a small part of these discrepancies may be due to the apparent in-plane rotation of the unit, the probable reason stems from the theoretical temperature profile adopted for the analysis. The assumed temperature profile changes linearly with changes in the ambient temperature and does not account for larger differences in slab and girder temperatures at higher ambient temperatures. The profiles obtained from field measurements indicate that the assumed temperatures constitute an over-simplification. Better theoretical responses would result if the profiles were based on the expressions for temperature variation compiled in the experimental phase of this project.

The effect of support stiffness (of piers) on joint movements was not considered because the experimental investigation indicated that bents under expansion joints responded to the expansion of the girders and the friction at the bearing pads but did not directly contribute to joint movements. Furthermore, the calculation of bridge deck joint movements based on properties other than those of the bridge girders and deck - such as pile bent properties - will be cumbersome and of little practical value.
Chapter VII
Parametric Studies and Recommendations

7.1 Prediction of Joint Movements

Joint movements in bridges are complex responses involving many elements of the structure and combinations of strains varying across and through the depth of the system. If temperature effects were absent, the joint movements would be affected mainly by the material behavior of the system and by its geometrical configuration. A parameter study that would identify the relative effects of each material and geometric contributing factor would be an impossible task. Therefore, a system response which varies with most of the system’s material and geometric properties and can be used as a reference value for estimating joint movements, was sought.

In the design of bridges with precast, pretensioned concrete girders supporting cast-in-place concrete slabs, girders and prestressing are chosen based on girder spacing, span length, slab thickness, material properties of the slab and girder concrete and the anticipated loading. Once the design of the girder is complete, the designer has in effect determined the behavior of the girder upon release of the prestress. The midspan deflection of the girder immediately after prestress release, therefore, reflects the properties of the entire system and is an appropriate system response for use as a reference value. Knowledge of the tendon profile, the modulus of elasticity of the girder, its cross sectional properties and its dead weight, makes the calculation of this system response an easy task to perform. The present parametric study concentrated on the prediction of joint movements based on initial girder deflection. Since most creep and shrinkage strains occur within the first two years of the life of the structure, the joint movement at the end of 2 years after prestress release, $\Delta_{JM}$, is expressed as follows:

$$\Delta_{JM} = 2 \times C_{JM} \times \Delta_i$$

(7.1)
where $C_{JM} = \text{coefficient of joint movement},$ and \\
$\Delta_i = \text{initial girder deflection}.$

The coefficient of joint movement, $C_{JM},$ depends on several bridge geometric and material parameters. The key parameters are the span length and number of continuous spans, and were considered in the study.

It is pertinent to note that the term $C_{JM}\Delta_i$ represents the joint movement or shortening of the system at each end of a simple span or a continuous span girder-slab system. The temperature effects of expansion and contraction are not accounted for in the coefficient.

### 7.2 Parametric Study to Determine the Coefficient of Joint Movement, $C_{JM}$

An extensive parameter study was conducted to investigate the influence of the two key bridge parameters on the joint movements. Analyses were performed on two types of girder-slab systems representing a wide range of span lengths and numbers of continuous spans. The types chosen were systems with the AASHTO Type III and Type IV girders. The girders were designed as per the AASHTO specifications. The effective slab width was held constant at 96" for every system and slab thicknesses equaled 7". Prestressing was designed and chosen based on the span length, dead loads, girder spacing and the material properties of both the girder and the slab. The girder concrete had a 28-day strength of 6000 psi and is of the type typically used for bridge girders in Louisiana. The 28-day strength of the slab concrete was 4900 psi. The creep and shrinkage characteristics were identical to those of the heuristic system described in section 6.2. Prior to performing the analyses for single-girder slab systems, an entire bridge section was analyzed and movements were compared with the isolated system. Differences in predicted joint movements are negligible and vary slightly only for the exterior girders which are subjected to smaller dead loads.
In practice, Type III AASHTO girders are used for simple span lengths varying from 55' to 75' while Type IV girders are employed for a range of 70' to 95'. For this reason, in the present parametric study, 5 span lengths of 55', 60', 65', 70' and 75' were chosen for systems with Type III girders and 6 span lengths of 70', 75', 80', 85', 90' and 95' for systems with Type IV girders. For each span length and girder type, analyses were performed for systems with a single simple span and with 2, 3 and 4 continuous span configurations. Separate analyses using the ACI-209, BP2 and CEB-FIP creep and shrinkage models were applied in each case bringing the total number of analyses to 132.

The boundary conditions used in the analyses consisted of the modelling of supporting neoprene bearing pads at girder ends as sets of parallel spring elements. The shear stiffness of bearing pads were represented by three parallel springs attached to nodes at the bottom of the girders. For a typical 1" thick neoprene bearing pad, the shear stiffness can be represented by three parallel springs with extensional stiffnesses of 50 psi. To provide for continuity between adjacent spans, the space (approximately 2") is assumed to be filled up with slab concrete at the time of slab casting.

Each analysis resulted in the output of joint movement at the top of the slab at a structure age of 2 years. These movements were divided by the initial girder deflection to produce the coefficient of joint movement, $C_{JM}$. The initial girder deflection is the instantaneous deflection that occurs when the prestress is released. Figures 7.1 through 7.8 show a plot of the coefficient, $C_{JM}$, versus span length for each of the three creep and shrinkage analysis procedures. In each figure, the analytical data points are plotted along with a 'best-fit' quadratic curve obtained by performing regression analyses. A typical regression equation for $C_{JM}$ was obtained as:

$$C_{JM} = A - BL + CL^2$$  \hspace{1cm} (7.2)
where \( A, B, C = \) regression coefficients, and

\[ L = \text{span length in feet}. \]

The application of step-wise regression to each set of data points indicated that no improvement in accuracy would be obtained if a higher order of relationship between \( C_{JM} \) and span length were sought. Figure 7.1 represents single span systems of Type III girders and Figures 7.2, 7.3 and 7.4 represent 2, 3 and 4 continuous spans. The same progression for Type IV girders is maintained in Figures 7.5 through 7.8. The regression equations for \( C_{JM} \) obtained for the systems with Type III and Type IV girders are given in Tables 7.1 and 7.2, respectively. Also shown in the tables are the values of R-square which were calculated in the regression analyses.

The plots in Figures 7.1 through 7.8 indicate that the BP2 model results in values for \( C_{JM} \) that are consistently higher than those predicted using the other two creep procedures. The fact that all values are positive is because creep and shrinkage cause shortening in girder units. The decreasing trend in \( C_{JM} \) with increasing span lengths is a result of the differences in the proportionalities between member stiffness and span length and between creep and shrinkage shortening and span length. The curves in Figures 7.1 through 7.8 point out differences in the magnitudes of joint movement predicted by each creep and shrinkage model. The shortening of girder-slab systems (excluding temperature effects) is mainly due to slab shrinkage. Creep and shrinkage strains in the girder also affect the joint movement, but to a lesser extent. The differences in the values of joint movement can only be attributed to the differences in the mathematical formulation of the creep and shrinkage models.

Figures 7.9, 7.10 and 7.11 show the regression lines obtained when the variation of \( C_{JM} \) with the number of continuous spans was studied. In Figure 7.9, regression lines for the six span lengths for girder-slab systems with Type IV girders, using the ACI-209 model are depicted. Figures 7.10 and 7.11 show similar plots, arrived
Figure 7.1: $C_{JM}$ for Single-span Type III Girder and Slab Systems.
Figure 7.2: $C_{JM}$ for Two-span Type III Girder and Slab Systems.
Figure 7.3: $C_{JM}$ for Three-span Type III Girder and Slab Systems.
Figure 7.4: $C_{JM}$ for Four-span Type III Girder and Slab Systems.
Figure 7.5: $C_{JM}$ for Single-span Type IV Girder and Slab Systems.
Figure 7.6: $C_{JM}$ for Two-span Type IV Girder and Slab Systems.
Figure 7.7: $C_{JM}$ for Three-span Type IV Girder and Slab Systems.
Figure 7.8: $C_{JM}$ for Four-span Type IV Girder and Slab Systems.
Table 7.1: Expressions for $C_{JM}$ for Type III Girder-Slab Systems

<table>
<thead>
<tr>
<th>No. of</th>
<th>Creep</th>
<th>Proposed Expression</th>
<th>R-square</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spans</td>
<td>Model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ACI</td>
<td>$C_{JM} = 1.839 - 0.04505L + 0.0002943L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>BP2</td>
<td>$C_{JM} = 2.681 - 0.06486L + 0.0004200L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CEB</td>
<td>$C_{JM} = 1.149 - 0.02816L + 0.0001857L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>ACI</td>
<td>$C_{JM} = 2.932 - 0.07129L + 0.0004629L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>BP2</td>
<td>$C_{JM} = 4.100 - 0.09861L + 0.0006371L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CEB</td>
<td>$C_{JM} = 2.074 - 0.04930L + 0.0003257L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>ACI</td>
<td>$C_{JM} = 4.196 - 0.10155L + 0.0006571L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>BP2</td>
<td>$C_{JM} = 6.341 - 0.15508L + 0.0010143L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CEB</td>
<td>$C_{JM} = 3.057 - 0.07325L + 0.0004771L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>ACI</td>
<td>$C_{JM} = 5.616 - 0.13623L + 0.0008829L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>BP2</td>
<td>$C_{JM} = 7.297 - 0.17011L + 0.0010686L^2$</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>CEB</td>
<td>$C_{JM} = 4.076 - 0.09777L + 0.0006371L^2$</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Table 7.2: Expressions for $C_{JM}$ for Type IV Girder-Slab Systems

<table>
<thead>
<tr>
<th>No. of Spans</th>
<th>Creep Model</th>
<th>Proposed Expression</th>
<th>R-square</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ACI</td>
<td>$C_{JM} = 2.765 - 0.05377L + 0.000275L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>BP2</td>
<td>$C_{JM} = 4.991 - 0.09558L + 0.0004843L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CEB</td>
<td>$C_{JM} = 1.928 - 0.03753L + 0.0001921L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>ACI</td>
<td>$C_{JM} = 4.151 - 0.07913L + 0.0003971L^2$</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>BP2</td>
<td>$C_{JM} = 6.025 - 0.11574L + 0.0005879L^2$</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CEB</td>
<td>$C_{JM} = 3.331 - 0.06371L + 0.0003236L^2$</td>
<td>0.98</td>
</tr>
<tr>
<td>3</td>
<td>ACI</td>
<td>$C_{JM} = 6.212 - 0.11911L + 0.0005986L^2$</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>BP2</td>
<td>$C_{JM} = 11.06 - 0.22029L + 0.0011450L^2$</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>CEB</td>
<td>$C_{JM} = 4.850 - 0.09231L + 0.0004671L^2$</td>
<td>0.98</td>
</tr>
<tr>
<td>4</td>
<td>ACI</td>
<td>$C_{JM} = 7.485 - 0.14115L + 0.0007014L^2$</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>BP2</td>
<td>$C_{JM} = 12.80 - 0.25129L + 0.0021957L^2$</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>CEB</td>
<td>$C_{JM} = 6.391 - 0.12140L + 0.0006136L^2$</td>
<td>0.99</td>
</tr>
</tbody>
</table>
at using the BP2 and CEB-FIP models. As shown in Figures 7.9, 7.10 and 7.11, the values of $C_{JM}$ lie extremely close to the regression lines. The linear relationship between $C_{JM}$ and the number of continuous spans suggests that linear extrapolation can be used to obtain $C_{JM}$ values for systems with more than 4 continuous spans.

Twenty-four additional analyses were performed on systems with span lengths of 70' and 90' and made of Type IV girders using medium strength concrete. The girder concrete had a 28-day compressive strength of 8230 psi, a cement content of 757 lb./cu.yd. and a water-cement ratio of 0.32. The concrete in the slab was identical to that used in the previous girder-slab systems. Plots of the coefficient of joint movements versus the number of continuous spans are shown, for each of the three creep models, in Figures 7.12, 7.13 and 7.14. The graphs show higher values of $C_{JM}$ when compared with the previous studies. This can be theoretically accounted for because the creep and shrinkage strains in the girder are small due to an increase in concrete strength. This makes the phenomenon of differential shrinkage at the girder-slab interface a critical factor. A considerable number of analyses need to be performed if it is essential to quantify the effect of various concrete strengths on joint movements.

The results of the parametric study form the basis of the design recommendation for estimating bridge deck joint movements. The design recommendation is given in section 7.5.

7.3 Comparison with LaDOTD Specifications: Creep and Shrinkage Effects

The LaDOTD (1987) provides specifications for the sizing of joint openings for bridges built in Louisiana. The specifications consider the effects of temperature, creep and shrinkage. An initial minimum joint dimension is specified as 1" plus the movement caused at the ends of girders spanning into a joint due to a temperature rise of 30°F. The maximum opening of the joint is calculated by adding on the
Figure 7.9: $C_{JM}$ v/s No. of Continuous Spans for Systems with Type IV Girders using the ACI-209 Model
Figure 7.10: $C_{JM}$ v/s No. of Continuous Spans for Systems with Type IV Girders using the BP2 Model
Figure 7.11: $C_{JM}$ v/s No. of Continuous Spans for Systems with Type IV Girders using the CEB-FIP Model
Figure 7.12: $C_{JM}$ for Systems with Medium Strength Concrete Type IV Girders using the ACI-209 Model
Figure 7.13: $C_{JM}$ for Systems with Medium Strength Concrete Type IV Girders using the BP2 Model
Figure 7.14: $C_{JM}$ for Systems with Medium Strength Concrete Type IV Girders using the CEB-FIP Model
movement caused by a 40°F fall in temperature and due to creep and shrinkage.
The maximum opening permitted is 4". In this section, only the effects of creep
and shrinkage on joint movements are considered. Creep and shrinkage shortening
is calculated using a coefficient of $2.08 \times 10^{-4}$ inches per inch of span length by the
LaDOTD.

If the LaDOTD specifications are used for a 70' simple span system, the joint
opening due to creep and shrinkage is computed as:

$$\Delta_{\text{creep + shrinkage}} = 2.08 \times 10^{-4} \times 70 \times 12 = 0.18"$$

For two and three continuous span cases – with 70' span lengths – the joint opening
due to creep and shrinkage calculated by the current LaDOTD procedure is given
as follows:

1. Two continuous spans: $\Delta_{\text{creep + shrinkage}} = 2 \times 0.18 = 0.36"$
2. Three continuous spans: $\Delta_{\text{creep + shrinkage}} = 3 \times 0.18 = 0.54"$

The above calculations neglect the effects of girder type used, the applied prestress,
slab dimensions and concrete composition.

Consider now the use of the coefficient, $C_{JM}$, developed in this study to cal-
culate the joint opening due to creep and shrinkage. For the single span case
discussed above, the joint opening can be estimated (using the ACI-209 model) by
the recommended procedure as follows:

1. $\Delta_i$ for Type III girder (70' span) = 0.96"
2. $C_{JM} = 1.84 - 0.045 \times 70 + 0.00029 \times 70^2 = 0.11$
3. $\Delta_{\text{creep+shrinkage}} = 2 \times C_{JM} \times \Delta_i = 2 \times 0.11 \times 0.96 = 0.21"$

With the CEB-FIP recommendations the corresponding joint opening value is
0.17". If three continuous spans are considered, the ACI-209 model yields a joint
opening due to creep and shrinkage of $2 \times 0.284 \times 0.96 = 0.55"$ which is slightly larger
than the 0.54" that must be designed for as per LaDOTD recommendations. The ACI model applied to a three continuous span system with Type IV girders of 70' yields $C_{JM} = 0.787$, which when multiplied by an initial girder deflection of 0.34", results in a joint opening due to creep and shrinkage equal $2 \times 0.787 \times 0.34 = 0.54"$.

Table 7.3 shows values of joint openings due to creep and shrinkage, calculated using both the LaDOTD recommendations and the coefficient of joint movement, $C_{JM}$, for comparison. The values pertain to systems consisting of individual span lengths of 70'. It can be seen that the openings calculated for systems with Type III girders are, in general, larger than those for systems with Type IV girders. The maximum joint openings for the 5 continuous spans are calculated by linearly extrapolating for values of $C_{JM}$. As a further comparison, the joint openings due to creep and shrinkage were calculated using $C_{JM}$ and the LaDOTD procedure for continuous systems with Type IV girders spanning 95 feet. The results of the computations are shown in Table 7.4. It is clear from the table that the joint movements obtained using $C_{JM}$ are consistently larger than those recommended by the LaDOTD. The brief comparisons made above indicate the following:

1. The LaDOTD specifications do not account for the influence of the girder cross sectional properties on joint movement.

2. The creep and shrinkage coefficient of $2.08 \times 10^{-4}$ inches per inch of span length is empirical and valid for systems with certain girder types only.

3. The use of the coefficient $C_{JM}$ represents a more rational procedure to estimate long-term creep and shrinkage movement at the joints.

4. The values of joint movement calculated using the coefficient $C_{JM}$ depend on the initial girder deflection, thus permitting the estimation of maximum joint openings at the design stage.
Table 7.3: Joint Openings due to Creep and Shrinkage (Span Length = 70')

<table>
<thead>
<tr>
<th>Number of Continuous Spans</th>
<th>System using Girder Type</th>
<th>Initial Girder Deflection (inches)</th>
<th>Joint Opening (inches)</th>
<th>Current Procedure using</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LaDOTD Procedure</td>
<td></td>
<td>ACI Model</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BP2 Model</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CEB Model</td>
</tr>
<tr>
<td>1</td>
<td>III</td>
<td>0.96</td>
<td>0.18</td>
<td>0.21</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td>0.34</td>
<td>0.18</td>
<td>0.20</td>
</tr>
<tr>
<td>2</td>
<td>III</td>
<td>0.96</td>
<td>0.35</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td>0.34</td>
<td>0.35</td>
<td>0.36</td>
</tr>
<tr>
<td>3</td>
<td>III</td>
<td>0.96</td>
<td>0.53</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td>0.34</td>
<td>0.53</td>
<td>0.54</td>
</tr>
<tr>
<td>4</td>
<td>III</td>
<td>0.96</td>
<td>0.70</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td>0.34</td>
<td>0.70</td>
<td>0.71</td>
</tr>
<tr>
<td>5</td>
<td>III</td>
<td>0.96</td>
<td>0.88</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td>0.34</td>
<td>0.88</td>
<td>0.88</td>
</tr>
</tbody>
</table>
Table 7.4: Joint Openings for Systems with Type IV Girders (Span Length = 95')

<table>
<thead>
<tr>
<th>Number of Continuous Spans</th>
<th>LaDOTD Procedure (inches)</th>
<th>Joint Opening (inches)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Current Procedure using</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACI</td>
<td>BP2</td>
</tr>
<tr>
<td>1</td>
<td>0.24</td>
<td>0.35</td>
</tr>
<tr>
<td>2</td>
<td>0.48</td>
<td>0.56</td>
</tr>
<tr>
<td>3</td>
<td>0.72</td>
<td>0.80</td>
</tr>
<tr>
<td>4</td>
<td>0.96</td>
<td>1.04</td>
</tr>
<tr>
<td>5</td>
<td>1.20</td>
<td>1.27</td>
</tr>
</tbody>
</table>

7.4 Comparison with LaDOTD Specifications: Temperature Effects

The LaDOTD recommends that joint movements due to temperature be calculated by applying a coefficient of thermal expansion of $6 \times 10^{-6}/°F$ to the length of single span or continuous span bridge systems, for a temperature change of $70°F$. For example, if 2 sets of continuous spans, each of length 75', span into a joint, the joint movement due to temperature is computed as:

$$Δ_{\text{temp.}} = 6 \times 10^{-6}/°F \times 2 \times 75' \times 12" \times 70°F = 0.756"$$

The implicit assumption here is that the bridge experiences a constant temperature through the depth of its cross section.

In this section, the effects on joint movements due to 2 different temperature profiles acting on a cross section, is first studied. One profile is taken from available literature and the other is obtained from actual bridge measurements. The joint movements due to the second profile are then compared with those specified by the LaDOTD.

In order to gain an understanding of the effects of different temperature profiles on joint movements, a simple study was conducted. A girder-slab system with 2 continuous spans of 85', with Type IV girders were chosen. The material properties
of the girder and slab concrete were the same as those described in section 7.2. Two bridge temperature profiles based on the ambient temperature were considered. The first profile (P1) is shown in Figure 3.8 and is recommended by the Committee on Loads and Forces on Bridges (1981). In that profile and the second one that will shortly be described, let $T_1$, $T_2$ and $T_3$ be the temperatures at the top of the slab, bottom of the slab and mid-depth of the girder respectively. The second profile (P2) used was suggested by Pentas (1990). It was obtained from measured temperatures on the bridge at Krotz Springs, Louisiana. Quadratic equations that relate temperatures $T_1$, $T_2$ and $T_3$ to the ambient temperature, $T_a$, were obtained using a polynomial curve-fitting technique. The profile through the cross section depth was assumed to be linear between $T_1$ and $T_2$ and between $T_2$ and $T_3$. The bottom half of the girder was assumed to be at temperature $T_3$. Figure 7.15 shows a plot of the curves for $T_1$, $T_2$ and $T_3$ versus $T_a$ and the relationships in equation form are as follows:

$$T_1 = 0.095 + 0.832T_a + 0.004T_a^2 \quad (7.3)$$

$$T_2 = 6.630 + 0.648T_a + 0.005T_a^2 \quad (7.4)$$

$$T_3 = 23.88 + 0.206T_a + 0.006T_a^2 \quad (7.5)$$

All temperatures are in degrees Fahrenheit. Profile P2 is more realistic than profile P1 because in P2, cross section temperatures are a non-linear function of the ambient temperature and are obtained from actual measurements. It should be noted that profile P2 is valid for ambient temperatures between $32^\circ F$ ($0^\circ C$) and $92^\circ F$ ($33^\circ C$). Values for $T_1$, $T_2$ and $T_3$ for a few ambient temperatures are listed in Table 7.5.

For this particular analytical study, the ambient temperature was assumed to vary with time as shown in Figure 7.16. Drops and rises in temperatures were made to occur in small decrements and increments over time intervals of 2 hours duration in the analyses. Two sets of three analyses each were performed for each
Figure 7.15: Temperatures for Profile P2

Ambient Temperature $T_a$ (°F)

Bridge Temperatures $T_1, T_2, T_3$ (°F)
Table 7.5: Temperatures from Profiles P1 and P2

<table>
<thead>
<tr>
<th>Amb. Temp. $T_a$ (°C)</th>
<th>$T_a$ (°F)</th>
<th>P1° Temp. (°F)</th>
<th>P2° Temp. (°F)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$T_1$ $T_2$ $T_3$</td>
<td>$T_1$ $T_2$ $T_3$</td>
</tr>
<tr>
<td>0</td>
<td>32</td>
<td>52 42 32</td>
<td>31 32 37</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>70 60 50</td>
<td>52 52 49</td>
</tr>
<tr>
<td>20</td>
<td>68</td>
<td>88 78 68</td>
<td>75 74 66</td>
</tr>
<tr>
<td>22</td>
<td>72</td>
<td>92 82 72</td>
<td>81 79 70</td>
</tr>
<tr>
<td>25</td>
<td>77</td>
<td>97 87 77</td>
<td>88 86 75</td>
</tr>
<tr>
<td>30</td>
<td>86</td>
<td>106 96 86</td>
<td>101 99 86</td>
</tr>
<tr>
<td>33</td>
<td>92</td>
<td>112 102 92</td>
<td>110 109 94</td>
</tr>
</tbody>
</table>

1. P1: Profile proposed by Committee on Loads and Forces on Bridges (1981)
2. P2: Profile proposed by Pentas (1990)

of the two temperature profiles. Each set consisted of analyses using the ACI, BP2 and CEB creep and shrinkage procedures. Composite slab and girder action was assumed to begin at 97 days. The variation with time of the movement, $\Delta t$, at the end of the girder-slab system and at the top of the deck slab was obtained from the analyses and is shown in Figure 7.16.

The use of the two profiles P1 and P2 result in small differences in movements of the system at expansion joints, as is evident in Figure 7.16. The maximum difference in predicted movements occurred at 510 days and equals 0.05" for each of the creep and shrinkage procedures. A closer look at the graphs in Figure 7.16 reveals that the movements due to a given temperature rise or fall is greater when profile P2 is employed. The reason for this becomes evident when Table 7.5 is consulted. For any change in ambient temperature, $\Delta T$, the entire cross section experiences a temperature change of $\Delta T$ when profile P1 is used. This is not the case for profile P2. If $\Delta T = 27°F$ at $T_a = 50°F$, the use of profile P2 results in a change in the slab temperature ($T_1$) equal to $36°F$. Table 7.4 shows that for any change in the ambient temperature, the change in slab temperature from P2 is larger when compared with the corresponding change from P1. Therefore, for design purposes, the use of temperature profile P2 provides for more conservative
P1: Profile proposed by Committee on Loads and Forces on Bridges (1981)
P2: Profile proposed by Pentas (1990)

Figure 7.16: Effect of Temperature Profiles P1 and P2 on Movement at Joints
estimates of joint movements.

A simple comparison of movements due to temperature changes, between those predicted using profile P2 and the LaDOTD recommendations, was made. A coefficient of thermal expansion equal to $5.6 \times 10^{-6}/\degree F$ was used. For an ambient temperature fall from $86 \degree F (30 \degree C)$ to $77 \degree F (25 \degree C)$, the analyses predicted a shortening of the system equal to $0.28"$. The value obtained using the LaDOTD coefficient of thermal expansion is $0.24"$. The analyses indicated a system expansion of $0.4"$ for a temperature rise from $50 \degree F (10 \degree C)$ to $77 \degree F (25 \degree C)$, while the use of the LaDOTD procedure yields a value of $0.33"$. The two simple numerical comparisons show that the LaDOTD procedure results in values of movements due to temperature that are about 15% less than values predicted using profile P2. The maximum joint opening due to temperature is calculated for a temperature increase of $70 \degree F$, in the LaDOTD procedure. The response of the girder-slab system to the above extreme temperature range, assuming the existence of profile P2 on the cross section, was examined. For a temperature fall from $72 \degree F$ to $32 \degree F$, the analyses predicted a system shortening of $0.54"$. An increase in ambient temperature from $32 \degree F$ to $92 \degree F$ resulted in the system expanding by $0.86"$. The application of the LaDOTD procedure gives a shortening of $0.49"$ for the temperature fall and an expansion of $0.73"$ for the temperature rise. The comparisons just described are summarized in Table 7.6. Expansions are listed as positive movements in the table.

From Table 7.6, it is seen that the use of profile P2 results in movements that are consistently higher than those used in design by the LaDOTD by about 15%. Therefore, the use of a coefficient of thermal expansion applied to a bridge, assuming a constant temperature profile, could lead to erroneous and non-conservative estimates of joint movements. A more rigorous investigation of temperature induced movements is required to obtain an improvement over the LaDOTD procedure. Such an investigation requires the performing of a large number of analyses, because the effects of the temperature profile depend on the cross sectional properties of the bridge. Movements due to temperature have to be related to the change
Table 7.6: Comparison of Movements using Profile P2 and the LaDOTD Procedure

<table>
<thead>
<tr>
<th>Initial Temp. (°F)</th>
<th>Final Temp. (°F)</th>
<th>(i) Movements using profile P2 (inches)</th>
<th>(ii) Movements using LaDOTD procedure (inches)</th>
<th>Percentage (ii)/(i)×100</th>
</tr>
</thead>
<tbody>
<tr>
<td>86</td>
<td>77</td>
<td>-0.28</td>
<td>-0.24</td>
<td>86</td>
</tr>
<tr>
<td>50</td>
<td>77</td>
<td>-0.54</td>
<td>-0.48</td>
<td>88</td>
</tr>
<tr>
<td>72</td>
<td>32</td>
<td>+0.40</td>
<td>+0.33</td>
<td>83</td>
</tr>
<tr>
<td>32</td>
<td>92</td>
<td>+0.86</td>
<td>+0.73</td>
<td>85</td>
</tr>
</tbody>
</table>

1. P2: Profile proposed by Pentas (1990)

in ambient temperature, as well as the properties of the cross section. Based on the study described above, a conservative estimate of joint movements due to temperature is obtained by increasing the movement prescribed by the LaDOTD by 15%.

7.5 Recommendations for Estimating Joint Movement

A rational procedure for computing joint openings due to creep and shrinkage can be based on the regression equations for $C_{JM}$, obtained from the parametric study and represented by the regression lines in the plots of Figures 7.1 through 7.8. In the expressions for $C_{JM}$, the span length $L$ is to be given in feet. The thicknesses of continuity diaphragms between continuous spans can be neglected. In Table 7.1, expressions for systems with Type III girders are listed and Table 7.2 contains expressions for systems with Type IV girders. The expressions for $C_{JM}$ are valid for span lengths ranging from 55' to 70' for Type III and 70' to 95' for Type IV girders. The values of $C_{JM}$ can be interpolated for intermediate values of the span length. The 28-day compressive strength of concrete should be about 6000 psi for girders and approximately 4500 psi for slabs. Values of $C_{JM}$ calculated based on the BP2
model will generally be high. This is a consequence of the model predicting higher shrinkage strains. Comparisons of analytical results with experimental observations made for the Krotz Springs Bridge did not indicate a clearly superior model. In light of this ambiguity, an average value of \( C_{JM} \) obtained from the three models can be used.

To evaluate joint movements due to changes in temperature, the LaDOTD recommends that a linear coefficient of thermal expansion equal to \( 6 \times 10^{-6}/°F \) be applied to the entire length of bridge between expansion joints. This is an oversimplification as evidenced by measured temperature profiles and by the studies in the preceding section. The temperature profiles on the Krotz Springs Bridge (see Pentas (1990)) indicate that the temperature in the deck slab varies non-linearly with the ambient temperature. The difference between the slab and ambient temperature is larger at higher ambient temperatures. At ambient temperatures close to \( 32°F \), the slab temperature is lower than the ambient temperature. Temperatures at the bottom of girders are in general equal to the ambient temperature. Consequently, the LaDOTD procedure underestimates joint closings at high ambient temperatures and joint openings at low ambient temperatures. The magnitudes of temperature induced movements are significant as shown by field measurements (Pentas (1990)). Based on the studies of section 7.4, the joint movements due to temperature should be increased by 15% to obtain a conservative estimate.

The following procedure for estimating maximum joint openings is recommended:

1. Calculate the joint opening, \( \Delta_{\text{temp.}} \), caused by a temperature change of \( 70°F \) using the current LaDOTD procedure. This corresponds to an initial joint dimension to accomodate a temperature rise of \( 30°F \) and a joint opening due a temperature fall of \( 40°F \). Increase the movements so obtained by 15%.

2. Calculate an average \( C_{JM} \) corresponding to the three models for the type of girder and number of continuous spans, using the expressions from Tables 7.1 and 7.2.
3. Calculate the initial deflection, \( \Delta_i \), of the girder.

4. Calculate the joint opening due to creep and shrinkage as

\[
\Delta_{\text{creep+shrinkage}} = 2 \times C_{JM} \times \Delta_i \tag{7.6}
\]

5. Determine the joint movement due to creep, shrinkage and temperature as

\[
\Delta_{\text{cst}} = 1.15 \times \Delta_{\text{temp.}} + 2 \times C_{JM} \times \Delta_i \tag{7.7}
\]

The actual maximum joint opening is then \( \Delta_{\text{cst}} \) plus the initial minimum joint dimension at the time of joint installation.

6. Check if the maximum joint opening, \( \Delta_{\text{cst}} + \) minimum joint dimension, is less than 4". The maximum opening that can be accommodated by strip seals used in Louisiana is 4". If the check fails, a new design of the system is required.
Chapter VIII
Summary and Conclusions

8.1 Summary

A rigorous analytical investigation was undertaken to develop an efficient numerical procedure for the long-term analysis of bridges with multiple precast, pretensioned, prestressed concrete girders supporting cast-in-place concrete deck slabs. The analysis concentrated on including the time dependent effects due to load history, temperature history, creep, shrinkage and aging of concrete, and the behavior of the prestress on the movements of deck expansion joints. The capabilities of the analytical method include the prediction of displacements, prestress losses and stresses in the bridge structure throughout the service life as long as the design ensures stress levels below $0.5f'_c$.

The finite element method was used in the quasi-static time dependent analysis which divides the time domain into a discrete number of intervals. At the end of each time interval, equilibrium equations were set up based on the displacement formulation of the finite element method. Implicit in the analysis is the development of the total and incremental form of equilibrium equations, for changing geometric and material properties at the end of any time interval, from the principle of virtual work. Three-dimensional 20-node isoparametric elements coupled with an efficient quadrature scheme are used to represent the bridge geometry.

Emphasis was placed on the evaluation of the effects of changing creep and shrinkage strains on the deformation behavior of the structure. Determination of these strains is made from three prevalent code procedures. The code procedures are also used to predict the aging of concrete. A temperature profile was adopted to include the temperature effects on movement and on creep.

The behavior of prestressing strands was included in the analysis by developing
a three-dimensional embedded tendon model wherein a parabolic prestress strand profile is maintained over the life of the structure. The embedded formulation ensures inter-element continuity in strand profiles. Prestress losses were calculated based on geometry changes in the surrounding concrete by enforcing the compatibility of strains between steel and concrete. The contribution of the steel stiffness to the structure has been accounted for. An empirical method for calculating relaxation losses in prestressing steel has been included in the analysis.

Special consideration to the construction schedule of bridges was given in the analysis. This is reflected in the separate analyses required at the times of prestress transfer, slab casting and the onset of girder and slab composite action.

Boundary conditions imposed by bearing pads were modelled by springs in parallel whose resultant extensional stiffness represented the stiffness of the pad. Continuity for adjacent spans between expansion joints was achieved by using elements with slab concrete material properties in the gaps that were present prior to slab casting.

An efficient numerical analysis Fortran code was written to perform the analysis. Besides the mesh configuration, the only input required are the concrete material properties at 28 days and the times at which analytical output are desired. There are no restrictions on bridge geometry.

Numerical analyses were conducted to investigate the validity and applicability of the analysis procedure by comparison of predicted values by the model with experimental data. A large number of analyses on girder-slab systems were conducted to relate joint movements to bridge properties. Simple procedures for estimating joint movements and which account for the effects of various bridge parameters are recommended. The movements estimated by the recommended procedure are compared with those obtained by the current LaDOTD procedure.
8.2 Conclusions

Based on the analytical study of bridge deck joint movements, the following conclusions were made:

1. The analytical procedure developed in this study predicts accurately the response of bridges with precast, pretensioned girders composite with cast-in-place concrete slabs under both short-term and long-term loads.

2. Prestress losses in pretensioned girders were predicted by incorporating the prestressing steel as being completely embedded in the concrete. The deformations of both concrete and steel were determined at any time in one complete analysis over a time domain and under loading at different stages.

3. Typical construction procedures and schedules can be modelled in the analysis. The analysis accounts for the type of concrete used for girders and slabs, as well as the type and length of concrete curing. The analytical applications indicated that the effects of creep and shrinkage on girder deflection become insignificant after a period of 3 months.

4. The choice of the prediction method for creep and shrinkage strains used in an analysis affects the outcome. In general, extreme joint movements were obtained when the BP2 model was used for the creep and shrinkage analysis. Deformations in bridges, after the slab is cast, are largely affected by the differences in the rates of shrinkage in girder and slab concrete.

5. The comparisons of theoretical joint movements with measured values for the bridge at Krotz Springs, Louisiana, indicate that differences in these values depend on the choice of creep model. The ACI-209 procedure showed the closest agreement. The analytical procedure was not utilized to predict movements caused by support restraints observed in the field and which were not part of the design. The experimental results and the theoretical analyses
clearly indicated that, after a period of 1 year, temperature induced strains dominate the deformation behavior.

6. In the current LaDOTD recommendations, joint movements due to temperature changes are calculated by applying a linear coefficient of thermal expansion to the total span length. These calculations do not account for actual temperature distributions on bridge cross sections. Refined analyses using realistic bridge temperature profiles are likely to produce a method for evaluating joint movements due to temperature changes that would be a significant improvement on the LaDOTD method. A simple study showed that the LaDOTD procedure underestimated movements due to temperature by about 15%.

7. A method for estimating the maximum bridge deck joint movements has been recommended based on the results of an extensive parametric study. The method is easy to apply and takes into account the effects of bridge geometry and material properties. The use of the recommended procedure will permit the designer to determine the span lengths and the maximum number of continuous spans between expansion joints, if the limit of movement that can be accommodated by the joint sealing system chosen is known.

8. The analytical studies on bridge systems indicate that those using medium-strength concrete girders tend to have significantly larger joint movements as compared with systems using normal-strength concrete girders. This is due to the increased differential shrinkage strains between girder and slab. If the magnitudes of joint movements are to be held within limits, higher girder concrete strength should be accompanied by a corresponding increase in slab concrete strength.

9. The analysis program provides the bridge designer with a powerful tool to evaluate the long-term behavior of bridge structures with different support conditions, and with or without joints. The analytical model is also capable
of accounting for support stiffnesses and approach skew.

8.3 Recommendations for Future Research

There are many insights into the deformation and stress behavior of bridges that additional research can provide. Of particular interest are the restraining effects of the substructure on joint movements. An analysis that accounts for the pier stiffnesses and the properties of the various connections is needed. Though there is an abundance of analytical research results on temperature distributions on bridge cross sections, they have not been adequately included in analysis models. The present analytical procedure can be modified to account for substructure behavior and realistic temperature profiles.

Louisiana lags in research on jointless bridges when compared with other states, and still relies on joints to accommodate movements. Jointless bridges will be the norm in the future. Both analytical and experimental studies need to be focussed on such bridges. While there is no doubt about the effectiveness of well-conducted experiments to document and provide an understanding of bridge behavior, the giant strides in the development of the digital computer will make analytical procedures increasingly economical. The analytical procedures can be refined by comparisons with controlled evaluations of scaled-down bridge prototypes in the laboratory.

The primary concern in jointless bridge design is the accommodation of stresses that occur when movements are restrained. High stresses over supports in continuous slabs occur due to negative bending moments and the restraining of free slab shrinkage. The extremely limited number of simplified analyses confirm the existence of tensile stresses due to continuity over supports. However, the experimental investigations performed on actual jointless bridges indicated stress levels that were lower than predicted. An explanation of this phenomenon lies in the fact that creep strains in concrete reduce the high tensile stresses in the support regions. At present, the analytical model is capable of monitoring stress reductions due to creep.
To fully rationalize the elimination of deck expansion joints a study on the state of stress at continuous supports must be performed. To achieve this, the following modifications to the present analytical model are recommended:

1. Model deck slab reinforcement as embedded steel analogous to the prestressing strands and include a constitutive law for the reinforcement.

2. The presence of cracks in the bridge deck and the associated redistribution of stresses is an important aspect of jointless bridge behavior. Hence, include a simple 'smeared' cracking model in the analysis. Assume cracking occurs when the principal tensile stress exceeds a tensile strength criterion for concrete.

3. A simplified method to evaluate creep strains at high stress levels should be included in the analysis. The redistribution of stresses due to cracking in jointless bridges, may cause an increase in compressive stresses in the regions of positive bending moment, which may exceed the \(0.5f'_c\) criterion for assuming linear viscoelastic behavior.

4. Model the stiffness of the pier supports by means of support spring elements. This will permit the evaluation of the extent to which flexible pier supports reduce tensile stresses caused by restraints.

The above modifications to the present analysis procedure will result in an understanding of jointless bridge behavior that has been lacking to date. Furthermore, the evaluation of stresses for different span lengths and number of continuous spans can result in recommendations for jointless bridge design, as well as requirements for the design of the supporting bent structures.
References


Appendix A

Input Instructions
and Listing of Program PCBRIDGE

This appendix contains input instructions, a flow diagram and the listing of program PCBRIDGE. Program PCBRIDGE is written for execution on the IBM 3090 supercomputer.

Input instructions for program PCBRIDGE

Input data is to be given in file unit 9 and input lines follow a sequential order that must be maintained. Units of pounds, inches and °C must be used in the input data. For the three creep and shrinkage models the units of data depend on the model in use. The common blocks in PCBRIDGE which require large memory space use dynamic storage and are specified as parameters in the "@PROCESS DC" statement. Input data are in free format and each type of data occupies one or more input lines as described below. Input variable names or their descriptions (bold letters) occupy separate input lines.

1. TITLE

2. Analysis Code

   IANCOD - '0' implies a mesh check only.

3. Creep model data

   CSF - 'ACI', 'BP2' or 'CEB'
   NOCT - no. of concrete types

For each concrete type:

If CSF = 'ACI':

   FC - initial strength in psi
   ICURTY - concrete cure type ('1' - moist; '2' - steam)
   SLUMP - slump in inches
   FINES - fine/coarse aggregate ratio (%)
   AC - air content (% by weight)
   CC - cement content (lbs./cu.yd.)
   VSR - volume/surface ratio (inches)
   HUMID - percentage relative humidity
If CSF = 'CEB':
- **FC28** - 28-day concrete strength (psi)
- **ICR** - cure type ('1' & '2' - moist; '3' & '4' - steam)
- **HUN** - percentage relative humidity
- **VSR** - volume surface ratio (inches)
- **WCR** - water/cement ratio (by weight)
- **ACR** - aggregate/cement ratio (by weight)
- **GSR** - gravel/sand ratio (by weight)
- **SCR** - sand/cement ratio (by weight)
- **UNITWT** - unit weight of concrete (lbs./cu.ft.)

If CSF = 'CEB':
- **EC28** - 28-day elastic modulus (psi)
- **ICURY** - cure type ('1' - moist; '2' - steam)
- **HUM** - percentage relative humidity
- **AREA** - cross section area (sq.mm.)
- **PER** - perimeter exposed to drying (mm.)

4. Temperature analysis code
- **KTEMP** - '0' implies no temperature analysis

5. Nodal coordinate data
- **NUMNP** - no. of nodes
  
  For each node:
  Node no., x-coordinate, y-coordinate, z-coordinate

6. Material input data
- **NUMMATT** - number of material (concrete) types
  
  For each material:
  - **E, PR, WT-X, WT-Y, WT-Z**
  - **E** - modulus of elasticity at first loading
    (E=0 if material is not loaded at start of analysis)
  - **PR** - Poisson's ratio
  - **WT-X** - unit weight in x-direction
  - **WT-Y** - unit weight in y-direction
  - **WT-Z** - unit weight in z-direction
  - **FPC28, WCONC, TL, ICUR, CURLEN**
FPC28 — 28-day concrete strength
WCONC — unit weight of concrete
TL — time at which concrete type is first loaded
ICUR — concrete cure type:
    '1': Type I cement, moist cured
    '2': Type I cement, steam cured
    '3': Type III cement, moist cured
    '4': Type III cement, steam cured
CURLEN — no. of days of curing

7. Element data

NUMEL — no. of elements
IEL, ILEV
IEL — element type ('3')
ILEV — level of element on cross section
    (elements at the bottom of girders have ILEV = 1)
Element no., Material no., Connectivity
    (see Figure 2.2 for nodal connectivity sequence)

8. Element level data

NOLEV — no. of element levels

For each level:
DEP — depth of elements on the cross section

9. Distributed element load data

NELDL — no. of element distributed loads

For each distributed load:
Element no., Face no., Load
    Face numbers are such that the face at r = +1 is face 1,
at r = -1 is face 2 and so on ...
10. Specified degree of freedom data

\[ \text{NSDF} \quad \text{no. of specified degrees of freedom} \]

For each degree of freedom:
Node no., Direction ('1', '2' & '3' - x, y & z), Displacement

11. Nodal force data

\[ \text{NSBF} \quad \text{no. of loaded nodes} \]

For each loaded node:
Node no., x-force, y-force, z-force

12. Support spring data

\[ \text{NSSP} \quad \text{no. of support springs} \]

For each support spring:
Node no., Direction, Spring constant

13. Prestress data

\[ \text{NOTEN, NOTSG} \]
\[ \text{NOTEN} \quad \text{no. of continuous tendons} \]
\[ \text{NOTSG} \quad \text{no. of tendon segments} \]

For each tendon segment:
Node 1, Node 2, Node 3, Parent element no., Tendon no.

For each segment node:
x-coordinate, y-coordinate, z-coordinate

For each tendon:
Initial stress, Area

For all tendons:
\[ \text{FY} \quad \text{yield stress} \]
14. Time step data

For each time step in the analysis:

**TIME, JCODE, TEMP**

**TIME** – time in days

**JCODE** – time code

‘0’: intermediate time step - no output
‘1’: terminal time step
‘2’: intermediate time step - output
‘3’: apply new loads at this time step
‘4’: time step prior to slab and girder composite action

**TEMP** – ambient temperature in °C

If “JCODE = 3”, input new loading as follows:

**NELDL** – no. of element distributed loads

For each distributed load:

*Element no., Face no., Load*

**NSBF** – no. of loaded nodes

For each loaded node:

*Node no., x-force, y-force, z-force*

For each material specified by NUMMAT:

**KCODE** – code no.

If “KCODE = 0”:

*WT-X, WT-Y, WT-Z*

If “KCODE ≠ 0”:

*E, PR, WT-X, WT-Y, WT-Z*

Once an executable version of program PCB RIDGE is obtained, for execution the following input is required:

**NSETS** – no. of sets of nodes at which displacements and stresses are desired

For each set:

*First node, Last node*
Program Flow Diagram and Listing

The flow diagram and listing of program PCBRIDGE are now given. In the listing, subroutines are arranged in alphabetical order. The execution of PCBRIDGE requires the allocation of the files shown in Table A.1.

Table A.1: Data Files for PCBRIDGE

<table>
<thead>
<tr>
<th>File Unit No.</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Nodal displacement output</td>
</tr>
<tr>
<td>8</td>
<td>Reproduction of mesh data</td>
</tr>
<tr>
<td>9</td>
<td>Input file</td>
</tr>
<tr>
<td>10</td>
<td>Nodal stress output</td>
</tr>
<tr>
<td>11</td>
<td>Prestress loss output</td>
</tr>
</tbody>
</table>
Figure A.1: Flow Diagram for Program PCBRIDGE
CALL RELEASE
Calculate relaxation loss and shrinkage strain prior to release of prestress

CALL PSTRES
Check position of steel segments on the stress-strain curve

T0 ← T1

CALL PSLOAD
Convert prestress into nodal loads

CALL ASSEM
Assemble element stiffness arrays and load vectors into global arrays
Apply boundary conditions

CALL SKYLIN
Solve for global displacements

CALL DEFSTR
Add displacements to nodal coordinates to update geometry

CALL ADDISP
Add displacements to existing displacements

First element

CALL STRESS
Calculate element stresses

Next element

NO

Last element?

YES

Figure A.1: Continued ...
Figure A.1: Continued ...
CALL SETUP
Identify creep model in use

CALL ACI200
or
CALL CEBFIP
or
CALL BAPAN2

CALL ACICRP
or
CALL CEBCRP
or
CALL BP2CRP
Calculate creep strains at different loading ages

CALL DIRICH
Calculate aging coefficients for each loading age

Figure A.1: Continued ...
First element

Retrieve element geometry and material properties

CALL BRQ
Setup up element stiffness and load arrays

CALL DIST20
Impose element face loading

Assemble stiffness array and load vector into global arrays

Last element?

Add nodal loads to global load vector

CALL SUPPOR
Impose nodal spring restraints at supports

CALL BNDY
Impose nodal displacement boundary conditions

Next element

Figure A.1: Continued ...
**LIST OF VARIABLES**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN(I),EN(I)</td>
<td>SHAPE FUNCTION ARRAYS</td>
</tr>
<tr>
<td>B(I,J)</td>
<td>STRAIN-DISPLACEMENT MATRIX</td>
</tr>
<tr>
<td>BOD(I)</td>
<td>ELEMENT BODY FORCE VECTOR</td>
</tr>
<tr>
<td>C(I,J)</td>
<td>ELASTICITY MATRIX</td>
</tr>
<tr>
<td>CC(I,J),XX(I,J)</td>
<td>ELEMENT NODAL COORDINATES</td>
</tr>
<tr>
<td>COOD(I,J)</td>
<td>NODAL COORDINATE ARRAY</td>
</tr>
<tr>
<td>CSCUM(I,J,K)</td>
<td>CUMULATIVE PSEUDO INELASTIC STRAIN MATRIX</td>
</tr>
<tr>
<td>CI DET</td>
<td>DETERMINANT OF JACOBIAN</td>
</tr>
<tr>
<td>CI DE(I),DR(I)</td>
<td>ELEMENT DISPLACEMENT VECTORS</td>
</tr>
<tr>
<td>CI DELCS(I,J,K)</td>
<td>INCREMENTAL PSEUDO INELASTIC</td>
</tr>
<tr>
<td>CI COOD</td>
<td>NODAL COORDINATE ARRAY</td>
</tr>
<tr>
<td>CI ELF(I),RE(I)</td>
<td>ELEMENT LOAD VECTORS</td>
</tr>
<tr>
<td>CI GF(I)</td>
<td>GLOBAL FORCE VECTOR</td>
</tr>
<tr>
<td>CI GNSTR(I,J)</td>
<td>GAUSS POINT STRESS VALUES</td>
</tr>
<tr>
<td>CI GSTIF(I,J)</td>
<td>GLOBAL STIFFNESS MATRIX</td>
</tr>
<tr>
<td>CI HSV,HSV2</td>
<td>HIDDEN STATE VARIABLES</td>
</tr>
<tr>
<td>CI IEL(I)</td>
<td>ELEMENT TYPE ARRAY</td>
</tr>
<tr>
<td>CI IFAC(I,J)</td>
<td>NODAL CONNECTIVITY ON ELEMENT FACE</td>
</tr>
<tr>
<td>CI NEEL(I)</td>
<td>NO. OF ELEMENT FACE LOADS</td>
</tr>
<tr>
<td>CI NEQ</td>
<td>NO. OF EQUATIONS</td>
</tr>
<tr>
<td>CI NHBW</td>
<td>HALF-BANDWIDTH OF</td>
</tr>
<tr>
<td>CI NOLEV</td>
<td>NO. OF LEVELS OF ELEMENTS IN THE BRIDGE</td>
</tr>
<tr>
<td>CI NOTEN</td>
<td>NO. OF PRESTRESS TENDONS</td>
</tr>
<tr>
<td>CI NOTSG</td>
<td>NO. OF TENDON SEGMENTS</td>
</tr>
<tr>
<td>CI NSBF</td>
<td>NO. OF SPECIFIED NODAL FORCES</td>
</tr>
<tr>
<td>CI NSDF</td>
<td>NO. OF SPECIFIED DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>CI NSSP</td>
<td>NO. OF SUPPORT SPRINGS</td>
</tr>
<tr>
<td>CI NUMEL</td>
<td>NO. OF ELEMENTS</td>
</tr>
<tr>
<td>CI NUMMAT</td>
<td>NO. OF MATERIALS</td>
</tr>
<tr>
<td>CI NUMNP</td>
<td>NO. OF NODAL POINTS</td>
</tr>
<tr>
<td>CI P(I,J)</td>
<td>DERIVATIVE OF SHAPE FUNCTION 'J' WITH RESPECT TO DIRECTION 'I'</td>
</tr>
<tr>
<td>CI RNSTR(I,J)</td>
<td>NODAL STRESS ARRAY FOR AN ELEMENT</td>
</tr>
<tr>
<td>CI S(I,J),SP(I,J)</td>
<td>ELEMENT STIFFNESS MATRIX</td>
</tr>
<tr>
<td>CI T(I,J)</td>
<td>STRAIN TRANSFORMATION MATRIX</td>
</tr>
<tr>
<td>CI VBDF(I,J)</td>
<td>VALUES OF SPECIFIED NODAL FORCES</td>
</tr>
<tr>
<td>CI VSDF(I,J)</td>
<td>VALUES OF SPECIFIED D.O.F.</td>
</tr>
<tr>
<td>CI WT(I,J)</td>
<td>ARRAY OF WEIGHTS FOR GAUSS QUADRATURE</td>
</tr>
<tr>
<td>CI XG(I,J)</td>
<td>ARRAY OF GAUSS POINTS</td>
</tr>
</tbody>
</table>
C I XJ(I,J) = JACOBIAN ARRAY
C I XJI(I,J) = INVERSE OF JACOBIAN ARRAY
C I

C I LIST OF FLAGS
C I ---------------
C I
C I
C I IANCOD : (USER DEFINED)
C I 0 ==> CHECK MESH ONLY
C I
C I ICOD :
C I 0 ==> START OF ANALYSIS
C I 1 ==> INTERMEDIATE TIME STEP
C I 2 ==> TIME STEP PRIOR TO SLAB COMPOSITE ACTION
C I 3 ==> TIME STEP AT SLAB COMPOSITE ACTION
C I
C I ISLAB :
C I 1 ==> WEIGHT OF SLAB IMPOSED ON GIRDER
C I 2 ==> SLAB IS COMPOSITE WITH GIRDER
C I
C I ISTRAN : (USER DEFINED)
C I 0 ==> NEGLект ANALYSIS FOR PRESTRESSING
C I
C I JCODE : (USER DEFINED)
C I 0 ==> INTERMEDIATE TIME STEP; NO OUTPUT
C I 1 ==> TERMINAL TIME STEP
C I 2 ==> INTERMEDIATE TIME STEP; OUTPUT REQUIRED
C I 3 ==> TIME STEPS AT WHICH NEW LOADS ARE APPLIED
C I 4 ==> TIME STEP PRIOR TO SLAB COMPOSITE ACTION
C I
C I KTEMP : (USER DEFINED)
C I 0 ==> NEGLект TEMPERATURE ANALYSIS
C I

C I CALLS : ADDISP, ASSEM, CPUTIME, CREEP, DEFSTR,
C I ELAST, HEAT, INIT, NEWLDS, OUTNOD,
C I PRESKY, PRTDIS, PRTLOS, PRTSTR, PSLOAD,
C I PSTRES, REACT, REDATA, RELAX, RELEASE,
C I SETTEXT, SETGPF, SETUP, SKYLIN, STRESS,
C I TEMPDIS, TENEPS, TENPOS, ZEROEP
C I

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION DR(60)
COMMON/STIFF/GSTIF(20000000)
COMMON/GFV/GF(100000)
COMMON/SIZE/NMNP,NMEL,NMNSF,NMNSBF,NMNSBF,NMNBW,NEQ,NMELD
COMMON/MDATA/COOD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBDF(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/QUADR/XG(4,4),WT(4,4)
COMMON/INTEG/PTS(15,3),WTS(3)
COMMON/STREXT/EX(8,15)
COMMON/PSEPS/TEPS(500,2),DELEPS(500,2)
COMMON/STRAND/NOTEN,NOTSG,ISTRAN
COMMON/YIELD/FY
COMMON/ADDRS/JDIAG(40000)
COMMON/ACOEF/S/ST(20,5),AGE(20),NTIMES,MAGES
COMMON/ACOEF2/ST2(20,5)
COMMON/DEPTH/DEP(6),ADTEMP(4,15)
CHARACTER*60 TITLE
CHARACTER*3 CSF
CHARACTER*22 MODEL(4)
DATA MODEL/'ACI-209 METHOD','CEP-FIP MODEL','
*BAZANT-PANULA II MODEL','USER SUPPLIED'/
C
CPU=0.0
CALL CPUTEIM(XCPU1,IRC)
C
CALL SETGPF
READ(9,5) TITLE
5 FORMAT(A60)
WRITE(6,6) TITLE
WRITE(8,6) TITLE
WRITE(10,6) TITLE
WRITE(11,6) TITLE
6 FORMAT(1H1,///,15X,A60,///)
READ(9,* ) IANCOD
C
C-------- CHOOSE CREEP MODEL AND SET UP DIRICLET SERIES
C-------- COEFFICIENTS
C
IA=0
READ(9,7) CSF
7 FORMAT(A5)
IF (CSF.EQ.'ACI') IA=1
IF (CSF.EQ.'CEB') IA=2
IF (CSF.EQ.'BP2') IA=3
IF (CSF.EQ.'USF') IA=4
IF (IA.EQ.0) THEN
WRITE(6,* ) 'INCORRECTLY SPECIFIED CREEP FUNCTION NAME !'
WRITE(6,* ) 'CHOICES ARE: ACI,CEB,BP2 & USF'
GO TO 5000
END IF
WRITE(6,8) MODEL(IA)
WRITE(8,8) MODEL(IA)
WRITE(10,8) MODEL(IA)
WRITE(11,8) MODEL(IA)
8 FORMAT(///T6,'CREEP AND SHRINKAGE STRAINS :',3X,A22,///)
CALL SETUP(IA)
IF (IANCOD.EQ.100) GO TO 5000
READ(9,* ) KTEMP
IF (KTEMP.EQ.0) WRITE(6,601)
IF (KTEMP.NE.0) WRITE(6,602)
ISLAB=0
DO 10 I=1,4
   DO 9 J=1,15
9   ADTEMP(I,J)=0.0
10 CONTINUE
601 FORMAT(/,T6,'NO TEMPERATURE ANALYSIS IS BEING EMPLOYED!',//)
602 FORMAT(/,T6,'TEMPERATURE ANALYSIS EMPLOYED!',//)
   CALL OUTNOD
   C------- READ INPUT DATA
   C
   CALL REDATA(IANCOD)
   C------- INITIALIZE GLOBAL STIFFNESS AND FORCE ARRAYS
   C------ CODE=0 ==> CALCULATES BAND-WIDTH
   C
   CALL PRESKY(20000000)
   LEN=DIAG(NEQ)
   CALL INIT(0,LEN)
   CALL ZEROEP
   C------- FILL UP EXTRAPOLATION ARRAY 'EX' TO EXTRAPOLATE
   C------ GAUSS-POINT STRESSES TO NODES
   C
   CALL SETEXT
   ICOD=0
   LSTEP=0
   PRSTEP=0.0
   C------- READ INITIAL TIME
   C
   READ(9,*) TI,ICODE,DEG
   CALL HEAT(ICODE,DEG,ISLAB,TI)
   IF (KTEMP.NE.0) THEN
      WRITE(6,701) DEG
701   FORMAT(/,T6,'START TEMPERATURE = ',F10.4,
         *2X,'DEGREES CENTIGRADE',//)
   END IF
   C------- SET UP INITIAL ELASTIC MODULUS
   C
   CALL ELAST(TI,IA,ICODE)
   C------- DETERMINE THE CONFIGURATION OF P/S STRANDS AND CALCULATE
   C------ THE NORMALIZED COORDINATES OF SEGMENT NODES
   C
   IF (ISTRAN.EQ.0) GO TO 900
   CALL TENPOS
   CALL RELEASE(TI,IA)
   C------- CHECK POSITION OF P/S SEGMENT ON STRESS-STRAIN CURVE
   C------- ICODE=0 ==> GET STRAIN FROM STRESS
   C------- ICODE>0 ==> GET STRESS FROM STRAIN
   C
CALL PSTRES(ICOD)
900 WRITE(6,1000) TI
   TREL=TI
1000 FORMAT(///,T6,'START TIME = ',T25,F9.3,T40,'DAYS',///)
C
C---------- CONVERT P/S FORCE TO NODAL LOADS
C---------- ICOD=0 ==> USE EXISTING STRESS
C---------- ICOD>0 ==> USE CHANGE IN STRESS OCCURING IN TIME STEP
C
   IF (ISTRAN.EQ.0) GO TO 1020
1010 CALL PSLOAD(ICOD)
C
C-------- ASSEMBLE ELEMENT STIFFNESS ARRAYS, LOAD VECTORS AND
C-------- BUILD UP GLOBAL ARRAYS. APPLY BOUNDARY CONDITIONS
C
1020 CALL ASSEM
C
C---------- SOLVE FOR DISPLACEMENTS
C---------- GLOBAL DISPLACEMENTS ARE HELD IN GF(I)
C
   CALL SKYLIN(GSTIF,GF,J DIAG,NEQ,0)
   DO 15 I=1,NUMMAT
      RMAT(I,3)=0.0
      RMAT(I,4)=0.0
      RMAT(I,5)=0.0
15 CONTINUE

C
C---------- ADD DISPLACEMENTS TO NODAL COORDINATES TO GET
C---------- NEW GEOMETRY
C
   IF (JCODE.EQ.4) GO TO 1030
   CALL DEFSTR(ICOD)
C
C---------- ADD DISPLACEMENTS TO EXISTING DISPLACEMENTS
C
1030 CALL ADDISP(ICOD)
C
C----------CALCULATE STRESSES ------------------------
C
   CALL STRESS(ICOD,TI)
C
   IF (JCODE.GT.0) THEN
      CALL PRTDIS(TI)
      CALL PRTSTR(TI)
   END IF
   IF (JCODE.EQ.4) GO TO 1600
   IF (ISTRAN.EQ.0) GO TO 1500
C
C---------- CALCULATE STRAINS IN TENDON SEGMENTS OCCURING DUE TO
C---------- GEOMETRY CHANGE DURING A TIME STEP AND FIND NEW
C---------- POSITIONS OF TENDON SEGMENT NODES
C
CALL TENEPS(ICOD)
CALL NEWPOS(ICOD)
C
C---------- CALCULATE POSITION OF SEGMENT ON STRESS-STRAIN CURVE
C
   CALL PSTRES(1)
   CALL PRTLOS(TI)
1500 IF (JCODE.EQ.1) GO TO 4000
1600 READ(9,*) TF,JCODE,DEG
   CALL HEAT(ICOD,DEG,ISLAB,TF)
   ICODE=1
   CALL ELAST(TF,IA)
1800 CALL CREEP(TI,TF,IA,PRSTEP)
   LSTEP=LSTEP+1
   PRSTEP=TF-TI
   WRITE(6,2000) LSTEP,TF,TI
2000 FORMAT(///,T6,'TIME STEP NO. : ' ,I2,//,T6,'START TIME = ' ,
     *T25,F9.3,T40,'DAYS',//,T6,'END TIME = ' ,T25,F9.3,
     *T40,'DAYS',///)
   IF (KTEMP.NE.0) THEN
   WRITE(6,2601) DEG
2601 FORMAT(///,T6,'TEMPERATURE @ END OF TIME STEP = ' ,F10.4,
     *2X,'DEGREES CENTIGRADE',//)
   END IF
C
C---------- CALCULATE RELAXATION LOSSES IN SEGMENTS DURING A TIME STEP
C
   IF (TF.EQ.TI) GO TO 3000
   IF ( (TF-TI).LT.1.0) GO TO 3000
   IF (ISTRAN.EQ.0) GO TO 3000
   CALL RELAX(TI,TF,TREL)
3000 TI=TF
   IF (JCODE.EQ.3) THEN
      CALL NEWLDS(TF)
      ISLAB=ISLAB+1
   IF (ISLAB.EQ.1) THEN
      IF (KTEMP.NE.0) CALL TEMPDIS
      END IF
   END IF
   IF (JCODE.EQ.4) ICODE=2
   IF (ISLAB.EQ.2) THEN
      ICODE=3
      ISLAB=ISLAB+1
   END IF
   IF (ISTRAN.EQ.0) GO TO 1020
   GO TO 1010
C
4000 CALL CPUTEIME(XCPU2,IRC)
   IF (IRC.EQ.0) CPU=(XCPU2-XCPU1)*1.D-6
   WRITE(6,411) CPU
411 FORMAT(///,5X,'CPU TIME USED WAS ',F12.6,5X,'SECONDS',//)
C
5000 STOP
END
@PROCESS DC(STIFF,CUMDIS,STDISP,GFV)
C
C================================ ACICRP ======================================
C
SUBROUTINE ACICRP(N,TI,CT,CCR)
C
CALLED BY: DIRICH
CALLS : NONE

USE OF ACI-209 CREEP STRAIN V/S TIME EXPRESSION

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION TI(200),CT(200)

DO 10 I=1,N
   CONST=TI(I)**0.6
   CT(I)=CCR*CONST/(10.+CONST)
10 CONTINUE

RETURN
END

C================================ ACISH ======================================
C
SUBROUTINE ACISH(TI,TF,K,EPISH)
C
CALLED BY: CREEP
CALLS : NONE

USE OF ACI-209 SHRINKAGE STRAIN EXPRESSION

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/MOD/EPR(20),FPC28(20),WCONC(TL(20),ICUR(20),CURLENG(20))
COMMON/ACIS/ALP1,ALP2,SHR1,SHR2

ALPHA=ALP2
IF (ICUR(K).LE.2) ALPHA=ALP1
SHRINK=SHR2
IF (ICUR(K).LE.2) SHRINK=SHR1
TT=CURLENG(K)

SHRINK = ULTIMATE SHRINKAGE STRAIN
CALCULATED IN SUBROUTINE SETUP
C2=TF-TL(K)+TT
C1=TI-TL(K)+TT
S2=C2/(ALPHA+C2)
S1=C1/(ALPHA+C1)
EPISH=SHRINK*(S2-S1)

RETURN
END

C================================ ACI209 ======================================
C
SUBROUTINE ACI209
C CALLED BY: SETUP
C CALLS : DIRICH
C
C READS INPUT FOR THE ACI-209 METHOD AND CALCULATES CREEP
C AND SHRINKAGE COMPOSITION PARAMETERS.

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/ACOEFS/ST(20,5),AGE(20),NTIMES,MAGES
COMMON/AŒF2/ST2(20,5)
COMMON/ACIS/ALP1,ALP2,SHR1,SHR2

IA=1
READ(9,*) NOCT
DO 500 I=1,NOCT
   READ(9,*) FC
   WRITE(6,1001) FC
C ICURTY=1 ==> MOIST CURE; =2 ==> STEAM CURE
   READ(9,*) ICURTY
   WRITE(6,1002) ICURTY
   READ(9,*) SLUMP
   WRITE(6,1003) SLUMP
   READ(9,*) FINES
   WRITE(6,1004) FINES
   READ(9,*) AC
   WRITE(6,1005) AC
   READ(9,*) CC
   WRITE(6,1006) CC
   READ(9,*) VSR
   WRITE(6,1007) VSR
   READ(9,*) HUMID
   WRITE(6,1008) HUMID

C
1001 FORMAT(T5,'INITIAL STRENGTH IN PSI:',T50,F10.4)
1002 FORMAT(T5,'CURE TYPE:',T50,I1,/,*
1003      T5,'1 ==> MOIST CURE; 2 ==> STEAM CURE')
1004 FORMAT(T5,'SLUMP IN INCHES:',T50,F5.2)
1005 FORMAT(T5,'PERCENT AIR CONTENT:',T50,F5.2)
1006 FORMAT(T5,'CEMENT CONTENT (LBS./CUBIC YARD):',T50,F5.2)
1007 FORMAT(T5,'VOLUME/SURFACE RATIO (INCHES):',T50,F5.2)
1008 FORMAT(T5,'RELATIVE HUMIDITY (PERCENT):',T50,F5.2)
C--------- CALCULATE SHRINKAGE ----------
   ALPHA=35.D0
   TSH=7.D0
   IF (ICURTY.EQ.2) THEN
      ALPHA=55.D0
      TSH=3.D0
   END IF
C CALCULATE CREEP AND SHRINKAGE COEFFICIENTS
ZC=1.D0
ZS=1.D0
YCS=1.D0
YSS=1.0D0
IF (SLUMP.EQ.0.0) GO TO 15
YCS=0.82 +0.067*SLUMP
IF (YCS.LE.1.0) YCS=1.D0
YSS=0.89+0.041*SLUMP

15 YCH=1.0D0
YSH=1.0D0
IF (HUMID.EQ.0.0) GO TO 25
YCH=1.27-0.0067*HUMID
IF (HUMID.LT.40.0) GO TO 25
YSH=1.4-0.01*HUMID
IF (HUMID.GT.80.0) YSH=3.0-0.03*HUMID

25 YCF=1.0D0
YSF=1.0D0
IF (FINES.EQ.0.0) GO TO 35
YCF=0.88+0.0024*FINES
YSF=0.3+0.014*FINES
IF (FINES.GT.50.0) YSF=0.9+0.002*FINES

35 YCVS=1.0D0
YSVS=1.0D0
IF (VSR.EQ.0.0) GO TO 45
D=4.*VSR
IND=INT(D)
IF (IND.EQ.2) THEN
  YCVS=1.30D0
  YSVS=1.35D0
END IF
IF (IND.EQ.3) THEN
  YCVS=1.17D0
  YSVS=1.25D0
END IF
IF (IND.EQ.4) THEN
  YCVS=1.11D0
  YSVS=1.17D0
END IF
IF (IND.EQ.5) THEN
  YCVS=1.04D0
  YSVS=1.08D0
END IF
IF ((D.GE.6.0).AND.(D.LE.15.0)) THEN
  YCVS=1.14-0.023*D
  YSVS=1.23-0.038*D
END IF
IF (D.GT.15.0) THEN
  YCVS=2./3.*(1.+1.13*DEXP(-0.54*VSR))
  YSVS=1.2*DEXP(-0.12*VSR)
END IF

45 YCAC=1.0D0
YSAC=1.0D0
IF (AC.EQ.0.0) GO TO 55
YCAC=0.46+0.09*AC
IF (YCAC.LT.1.0) YCAC=1.0D0
YSAC=0.95+0.008*AC

55 YSCC=1.0D0
IF (CC.EQ.0.0) GO TO 65
YSCC=0.75+0.00061*CC
ZC=2.35*YCS*YCH*YCF*YCVS*YCAC
ZS=-YSS*YSH*YSF*YSVS*YSAC*YSCC*780.D-6

WRITE(8,*),'YCS = ',YCS
WRITE(8,*),'YSS = ',YSS
WRITE(8,*),'YCH = ',YCH
WRITE(8,*),'YSH = ',YSH
WRITE(8,*),'YCF = ',YCF
WRITE(8,*),'YSF = ',YSF
WRITE(8,*),'YCAC = ',YCAC
WRITE(8,*),'YSAC = ',YSAC
WRITE(8,*),'YCVS = ',YCVS
WRITE(8,*),'YSVS = ',YSVS
WRITE(8,*),'YSCC = ',YSCC

IF (ICURTY.EQ.1) CALL DIRICH(ZC,ST,AGE,IA,ICURTY)
IF (ICURTY.EQ.2) CALL DIRICH(ZC,ST2,AGE,IA,ICURTY)
IF (ICURTY.EQ.1) THEN
ALP1=ALPHA
SHR1=ZS
END IF
IF (ICURTY.EQ.2) THEN
ALP2=ALPHA
SHR2=ZS
END IF
500 CONTINUE
C
RETURN
END
C
==ADDISP==
C
SUBROUTINE ADDISP(ICODE)
C
CALLED BY: MAIN
CALLS : NONE
C
THIS SUBROUTINE ADDS DISPLACEMENTS TO PREVIOUS DISPLACEMENTS
TO OBTAIN TOTAL DISPLACEMENTS FOR OUTPUT
IN SUBROUTINE PRTDIS
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CUMDIS/DI(1000000)
COMMON/STIFF/GSTIF(20000000)
COMMON/GFV/GF(1000000)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/STDISP/DS(1000000)
C
IF (ICODE.EQ.0) THEN
DO 5 I=1,NEQ
DI(I)=0.0
5 CONTINUE
END IF
C
DO 10 I=1,NEQ
   DI(I)=DI(I)+GF(I)
10 CONTINUE
IF (ICODE.EQ.2) THEN
   DO 20 I=1,NEQ
20   DS(I)=GF(I)
END IF
C
RETURN
END
C

SUBROUTINE AGING(M,N,RLAM,TI,CT,F,IA,TPR)
C
C CALLED BY: DIRICH
C CALLS : GESCP
C
C USE OF LEAST SQUARES METHOD TO CALCULATE DIRICHLET
C SERIES COEFFICIENTS
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RLAM(5),TI(200),CT(200),F(4),A(200,4),B(4,200),S(4,4)
DIMENSION G(4)
C
DO 20 I=1,N
   DO 10 J=1,M
      POW=-RLAM(J)*TI(I)
      A(I,J)=1.0-DEXP(POW)
10 CONTINUE
20 CONTINUE
C
C...............................SET TRANSPOSE...................................
DO 40 I=1,M
   DO 30 J=1,N
      B(I,J)=A(J,I)
30 CONTINUE
40 CONTINUE
C
C.....................PERFORM B*A = S .................
DO 80 I=1,M
   DO 70 J=1,M
      SUM=0.0
      DO 60 K=1,N
         SUM=SUM+B(I,K)*A(K,J)
60 CONTINUE
      S(I,J)=SUM
70 CONTINUE
80 CONTINUE
C.....................PERFORM F = B*C .................
DO 100 I=1,M
   SUM=0.0
   DO 90 J=1,N
SUM=SUM+B(I,J)*CT(J)

CONTINUE
F(I)=SUM

CONTINUE

CALL GESCP(S,F,M)

RETURN
END

@PROCESS DC(MDATA,STIFF,ADDR,GFV)

C
C =*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
\( C(2,2) = R^T \)
\( C(3,3) = R^T \)
\( C(1,2) = F \ast R^T \)
\( C(1,3) = F \ast R^T \)
\( C(2,1) = F \ast R^T \)
\( C(2,3) = F \ast R^T \)
\( C(3,1) = F \ast R^T \)
\( C(3,2) = F \ast R^T \)
\( C(4,4) = 0.5 \ast A \ast R^T \)
\( C(5,5) = 0.5 \ast A \ast R^T \)
\( C(6,6) = 0.5 \ast A \ast R^T \)

\[
\begin{align*}
15 & \text{ DO 20 } J = 1, 3 \\
& \quad J2 = J + 2 \\
& \quad BOD(J) = \text{RMAT}(KM, J2) \\
20 & \text{ CONTINUE} \\
30 & \text{ CONTINUE} \\
40 & \text{ CONTINUE}
\end{align*}
\]

\[
\begin{align*}
\text{C-\text{------ CALL ELEMENT STIFFNESS ASSEMBLAGE ROUTINES ------}} \\
& \text{IF (IEL(I).EQ.3) CALL BRQ(NEL,3,C,BOD,CC,SP,ELF)} \\
\text{C-\text{------- EVALUATE CONSISTENT ELEMENT NODAL LOADS FROM}} \\
& \text{IMPOSED FACE DISTRIBUTED LOADS ------------------------}} \\
& \text{IF (NELDL.NE.0) THEN} \\
& \quad \text{DO 45 } K = 1, \text{NELDL} \\
& \quad \quad N1 = ELDL(K,1) \\
& \quad \quad \text{IF (N1.EQ.NEL) THEN} \\
& \quad \quad \quad \text{ICODE = ELDL(K,2)} \\
& \quad \quad \quad \text{SIG = ELDL(K,3)} \\
& \quad \quad \quad \text{CALL DIST20(NEL,ICODE,3,CC,SIG,ELF)} \\
& \quad \quad \text{END IF} \\
& \quad \text{45 \text{ CONTINUE}} \\
& \text{END IF} \\
\text{C} \\
\text{C-\text{------ ASSEMBLE GLOBAL STIFFNESS AND FORCE MATRICES ------}} \\
& \text{IF (IEL(I).EQ.3) NPE=20} \\
& \quad \text{DO 200 } N = 1, \text{NPE} \\
& \quad \quad N2 = N + 2 \\
& \quad \quad NR = (NELC(I,N2)-1)*3 \\
& \quad \text{DO 190 } J = 1, 3 \\
& \quad \quad NR = NR + 1 \\
& \quad \quad L = (N-1) * 3 + J \\
& \quad \quad GF(NR) = GF(NR) + ELF(L) \\
& \quad \text{DO 180 } K = 1, \text{NPE} \\
& \quad \quad K2 = K + 2 \\
& \quad \quad NNO = NELC(I,K2) \\
& \quad \text{DO 170 } KK = 1, 3 \\
& \quad \quad M = (NNO-1) * 3 + KK \\
\end{align*}
\]
JBJ=3*(K-1)+KK  
KIJ=M-NR  
IF (KIJ.LT.0) GO TO 170  
ND=JDIAG(M)-KIJ  
GSTIF(ND)=GSTIF(ND)+SP(L,JBJ)  
170 CONTINUE  
180 CONTINUE  
190 CONTINUE  
200 CONTINUE  
300 CONTINUE  
C ------- IMPOSE BOUNDARY CONDITIONS -------  
IRES=0  
IF (NSBF.EQ.0) GO TO 800  
DO 710 I=1,NSBF  
   J=VSBF(I,1)  
   DO 705 K=1,3  
      K1=K+1  
      II=(J-1)*3+K  
      GF(II)=VSBF(I,K1)+GF(II)  
705 CONTINUE  
710 CONTINUE  
800 IF (NSSP.EQ.0) GO TO 900  
CALL SUPPOR  
900 IF (NSDF.EQ.0) GO TO 1000  
CALL BNDY(GSTIF,GF,JDIAG)  
1000 RETURN  
END
SUBROUTINE BAPAN2

CALLED BY: SETUP
CALLS : DIRICH

SUBROUTINE TO READ INPUT TO THE BAZANT-PANULA II MODEL AND TO CALCULATE COMPOSITION PARAMETERS.

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/BP2S/FC1,FC2,EP1,EP2,RS1,RS2,TSH1,TSH2,
*RC1,RC2,PS1D1,PSID2,CWT1,CWT2,IC1,IC2
COMMON/ACOEFS/ST(20,5),AGE(20),NTIMES,MAGES
COMMON/ACOEFS2/ST2(20,5)

IA=3
EX=1.D0/3.D0
READ(9,*) NOCT
DO 500 I=1,NOCT
   READ(9,*) FC28
   WRITE(6,1001) FC28
   READ(9,*) ICR
   WRITE(6,1002) ICR
   READ(9,*) HUM
   WRITE(6,1003) HUM
   READ(9,*) VSR
   WRITE(6,1004) VSR
   READ(9,*) WCR
   WRITE(6,1005) WCR
   READ(9,*) ACR
   WRITE(6,1006) ACR
   READ(9,*) GSR
   WRITE(6,1007) GSR
   READ(9,*) SCR
   WRITE(6,1008) SCR
   READ(9,*) UNITWT
   WRITE(6,1009) UNITWT

1001 FORMAT(T5,'28-DAY STRENGTH IN PSI: ',T50,F10.4)
1002 FORMAT(T5,'CURE TYPE: ',T50,I1,/,
   *T5,'1 & 2 ==> MOIST CURE; 3 & 4 ==> STEAM CURE')
1003 FORMAT(T5,'RELATIVE HUMIDITY (PERCENT): ',T50,F5.2)
1004 FORMAT(T5,'VOLUME/SURFACE RATIO (INCHES): ',T50,F5.2)
1005 FORMAT(T5,'WATER/CEMENT RATIO (BY WEIGHT): ',T50,F5.2)
1006 FORMAT(T5,'AGGREGATE/CEMENT RATIO (BY WEIGHT): ',T50,F5.2)
1007 FORMAT(T5,'GRAVEL/SAND RATIO (BY WEIGHT): ',T50,F5.2)
1008 FORMAT(T5,'SAND/CEMENT RATIO (BY WEIGHT): ',T50,F5.2)
1009 FORMAT(T5,'UNIT WEIGHT (LB./CU. FT.): ',T50,F7.3)

ICURTY=1
IF (ICR.GT.2) ICURTY=2
CONST=33.0*(UNITWT**1.5)
FC28=FC28*1.D-3
TO=7.DO
IF (ICURTY.EQ.2) TO=7.DO
C----------------- CALCULATE ULTIMATE SHRINKAGE (EP)
A=DSQRT(ACR)
B=GSR*GSR
C=(1.+SCR)/WCR
D=DSQRT(FC28)
C1=C**EX
Z=D*C1*(1.25*A+0.5*B)-12.
IF (Z.LE.0.0) THEN
  Y=0.DO
  GO TO 10
END IF
Z4=Z*Z*Z*Z
Y=1./((390./Z4+1.)
EP=(1330.-970.*Y)*1.D-6
C----------------- CALCULATE SHRINKAGE HALF-TIME AND HUMIDITY COEFF.
RKSH=-0.2D0
IF (HUM.LE.98.) RKSH=1.-HUM*HUM*HUM*1.D-6
D=50.8*VSR
CF=2.4+120./(DSQRT(T0))
SHPFAC=1.0
IF (ICURTY.EQ.2) SHPFAC=1.55
DD=D*SHPFAC
TSH=DD*DD/CF
C----------------- CALCULATE CREEP COEFFICIENTS
C----------------- CALCULATE (PSID)
A=SCR*FC28/ACR
B=0.00161*WCR/EP
R=(A**0.3)*(GSR**1.3)*(B**1.5)-0.85
IF (R.LE.0.0) THEN
  PSID=0.0056D0
  GO TO 20
END IF
U=-1.4DO
R14=R**U
PSID=0.0056+0.0189/(1.+0.7*R14)
C----------------- CALCULATE HUMIDITY COEFF. (RKCH)
20
HUM=1.D-2*HUM
RKCH=1.-(HUM**1.5)
C
IF (ICURTY.EQ.1) THEN
  FC1=FC28
  EP1=EP
  RS1=RKSH
  TSH1=TSH
  PSID1=PSID
  RC1=RKCH
  IC1=ICR
  CWT1=CONST
END IF
IF (ICURTY.EQ.2) THEN
  FC2=FC28
EP2=EP
RS2=RS2
TSH2=TSH
PSID2=PSID
RC2=RC2
IC2=IC2
CWT2=CWT2
END IF

C IF (ICURY.EQ.1) CALL DIRICH(0.0,ST,AGE,3,ICURY)
IF (ICURY.EQ.2) CALL DIRICH(0.0,ST2,AGE,3,ICURY)
C 500 CONTINUE
C RETURN
END
@PROCESS DC(MDATA)
C
C ================= BNDY =========================================================
C
SUBROUTINE BNDY(S,SL,JDIAG)
C
CALLED BY: ASSEM
CALLS : NONE
C
THIS SUBROUTINE IMPOSES SPECIFIED DEGREES OF FREEDOM
ONTO THE GLOBAL STIFFNESS AND FORCE MATRICES, BEFORE
SOLUTION FOR DISPLACEMENTS.
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION S(1),SL(1),JDIAG(1)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/MDATA/COOD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBDF(2000,3),VSBF(1000,4),ELDL(500,3),COLD(2500,4),ILEV(2500)
C
DO 300 NB=1,NSDF
ID=VBDF(NB,2)
J=VBDF(NB,1)
IDOF=(J-1)*3+ID
SVAL=VBDF(NB,3)
JDIAG=JDIAG(IDOF)
IF (IDOF.EQ.1) GO TO 150
IST=JDIAG(IDOF-1)
ICHT=JDIAG-IST
ITOP=IDOF-ICHT+1
ITM1=IDOF-1
I=1
DO 100 II=ITM1,ITOP,-1
IJ=JDIAG-I
SL(II)=SL(II)-S(IJ)*SVAL
I=I+1
S(IJ)=0.0
100 CONTINUE
150 IDP1=IDOF+1
DO 200 K=IDP1,NEQ
KHT=JDIAG(K)-JDIAG(K-1)
KK=K-KHT
IF (KK.GE.IDOF) GO TO 200
KM=0
KM1=K-1
DO 175 KL=KM1,1,-1
   KM=KM-1
175 IF (KL.EQ.IDOF) GO TO 180
180 KN=JDIAG(K)+KM
   SL(K)=SL(K)-S(KN)*SVAL
   S(KN)=0.0
200 CONTINUE
S(IDIAG)=1.0
SL(IDOF)=SVAL
300 CONTINUE
C
RETURN
END
C================ BP2CRP ===========================
C
SUBROUTINE BP2CRP(N,TI,CT,TPR,ICURTY)
C
CALLED BY: DIRICH
CALLS : NONE
C
CALCULATE CREEP STRAINS FROM THE BAZANT-PANULA II MODEL
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION TI(200),CT(200)
COMMON/BPEL/C1,C2,jEXl,EX2
COMMON/BP2CS/FC1,FC2,EP1,EP2,RS1,RS2,TSH1,TSH2,
   *RC1,RC2,PSID1,PSID2,CWT1,CWT2,IC1,IC2
C
TZERO=7.DO
PSID=PSID1
TSH=TSH1
RK=RC1
FC=FC1
EP=EP1
CW=CWT1
ICR=IC1
IF (ICURTY.EQ.2) THEN
   TZERO=7.DO
   PSID=PSID2
   TSH=TSH2
   RK=RC2
   FC=FC2
   EP=EP2
   CW=CWT2
   ICR=IC2
END IF
FCC=FC*FC
C------------------ CALCULATE EXPONENTS
EXM=-0.28-1./FCC
EXN=0.115+0.0002*FC*FCC
EXNM=-EXN
A=1.2D0
B=-0.35D0
C=-0.5D0
EX=EXM/2.0
C------------------ CALCULATE BASIC AND DRYING CREEP FACTORS
PS=0.3+15.*(FC**A)
FC28=FC*1000.0
E28=CW*DSQRT(FC28)
AA=((28.**EXM)+0.05)
E0=E28*(1.+PS*AA)
COM28=1./E0
BAS=(TPR**EXM)+0.05
ECT=1.0/((1.+BAS*PS)*COM28)
COMP=1./ECT
CB1=PS*BAS

C
DRY=TPR-TZERO
IF (DRY.LT.0.0) DRY=0.D0
SD=1.+DRY/(10.*TSH)
SD=(SD**C)*PSID*EP*1.D6
CD1=SD*RK*(TPR**EX)

C
DO 10 I=1,N
T=TI(I)
CB=CB1*(T**EXN)
SSD=1.+3.*TSH/T
SSD=SSD**B
CD=CD1*SSD
CRP=COM28*(1.+CB+CD)
CT(I)=ECT*CRP-1.0
10 CONTINUE
C

IF (ICURY.EQ.1) THEN
C1=CON
EX1=EXM
END IF
IF (ICURY.EQ.2) THEN
C2=CON
EX2=EXM
END IF
C
RETURN
END
C
C------------------ BP2SH -------------------------------
C
SUBROUTINE BP2SH(TI,TF,K,EPSh)
C
CALLED BY: CREEP
CALLS : NONE
CALCULATE SHRINKAGE STRAINS (BAZANT-PANULA II)

! IMPLICIT REAL*8 (A-H,O-Z)
! COMMON/MOD/EPR(20),FPC28(20),WCONC(20),TL(20),ICUR(20),CURLEN(20)
! COMMON/BP2CS/FC1,FC2,EP1,EP2,RS1,RS2,TSH1,TSH2,
!*RC1,RC2,PSID1,PSID2,CWT1,CWT2,IC1,IC2

RS=RS1
EP=EP1
TSH=TSH1
IF (ICUR(K).GT.2) THEN
  RS=RS2
  EP=EP2
  TSH=TSH2
END IF

TT=CURLEN(K)

T2=TF-TL(K)+TT
T1=TI-TL(K)+TT
S2=DSQRT(T2/(T2+TSH))
S1=DSQRT(T1/(T1+TSH))

EPSH=-EP*RS*(S2-S1)

RETURN
END

@PROCESS DC(STRAIN)

C -------- INITIALIZE LOAD VECTOR --------
DO 25 I=1,60
   RE(I)=0.0
25 CONTINUE

C CALCULATE STIFFNESS
DO 30 I=1,60
DO 30 J=1,60
30 S(I,J)=0.0
SDET=0.0
IST=6
WGT=WTS(1)
DO 100 IP=1,15
RI=PTS(IP,1)
SI=PTS(IP,2)
TI=PTS(IP,3)
IF (IP.GT.1) WGT=WTS(2)
IF (IP.GT.7) WGT=WTS(3)
C
CALL STQB(XX,B,DET,RI,SI,TF,NEL)
SDET=SDET+DET
C
WGT=WGT*DET
C------ ADD CONTRIBUTIONS DUE TO BODY FORCES TO LOAD VECTOR -----
DO 35 J=1,20
 K=J*3
 L=K-1
 M=L-1
 RE(M)=RE(M)+AN(J)*BOD(1)*WGT
 RE(L)=RE(L)+AN(J)*BOD(2)*WGT
 RE(K)=RE(K)+AN(J)*BOD(3)*WGT
35 CONTINUE
C
C ADD CONTRIBUTIONS TO LOAD VECTOR FROM INELASTIC STRAINS
DO 400 I=1,6
 DB(I)=0.0
 DO 380 J=1,6
  DB(I)=DB(I)+C(I,J)*DELCS(NEL,J,IP)
380 CONTINUE
400 CONTINUE
DO 420 I=1,60
 RLOAD=0.0
 DO 410 J=1,6
  RLOAD=RLOAD+B(J,I)*DB(J)
410 CONTINUE
RE(I)=RE(I)+RLOAD*WGT
420 CONTINUE
C
DO 70 J=1,60
DO 45 K=1,IST
 DB(K)=0.0
C------ PERFORM DB = C*DB ------
 DO 40 L=1,IST
 DB(K)=DB(K)+C(K,L)*B(L,J)
40 CONTINUE
45 CONTINUE
DO 60 I=J,60
 STIFF=0.0
C------ STIFFNESS = (B)^T*DB --------------
 DO 50 L=1,IST
STIFF = STIFF + B(L,I) * DB(L)

50 CONTINUE
S(I,J) = S(I,J) + STIFF * WGT

60 CONTINUE
70 CONTINUE
100 CONTINUE

C------- FILL UPPER TRIANGLE OF S ---------

DO 120 J = 1, 60
   J1 = J + 1
   DO 110 I = J1, 60
      S(J,I) = S(I,J)
   110 CONTINUE
120 CONTINUE

C
RETURN
END
SUBROUTINE CEBCRP(N, TI, CT, TPR, ICURTY)

CALLED BY: DIRICH
CALLS : NONE

THIS SUBROUTINE CALCULATES THE CREEP FLOW COMPONENT (CEB-FIP).

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION TI(200), CT(200)
COMMON/CEBCS/SHCEB1, SHCEB2, HF1, HF2, PHIF1, PHIF2,
* THK1, THK2, EC1, EC2

HF=HF1
PHIF=PHIF1
IF (ICURTY.EQ.2) THEN
   HF=HF2
   PHIF=PHIF2
END IF
EX=1.D0/3.D0
B1=TPR/(TPR+HF)
B=B1**EX
DO 10 I=1,N
   TI=TI(I)
   T1=T+TPR
   A=T1/(T1+HF)
   BF=(A**EX)-B
   CT(I)=PHIF*BF
10 CONTINUE
RETURN
END

SUBROUTINE CEBFIP

CALLED BY: SETUP
CALLS : NONE

READS INPUT FOR THE CEB-FIP MODEL AND CALCULATES CREEP
AND SHRINKAGE PARAMETERS.

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION HO(7), HO1(6), HM(3), RLMB(3), HF(6),
* PHI1(3), PHI2(7), SH1(3), SH2(7), TIME(12)
COMMON/AECOF2/ST(20,5), AGE(20), NTIMES, MAGES
COMMON/AECOF2/ST2(20,5)
COMMON/CEBCS/SHCEB1, SHCEB2, HF1, HF2, PHIF1, PHIF2,
* THK1, THK2, EC1, EC2
DATA HO/5.D1, 1.D2, 2.D2, 4.D2, 6.D2, 8.D2, 1.6D3/
DATA HO1/5.D1, 1.D2, 2.D2, 4.D2, 6.D2, 8.D2, 1.6D3/
DATA HM/4.D1,7.D1,9.D1/
DATA RLMB/1.D0,1.5D0,5.D0/
DATA HF/3.3D2,4.25D0,5.7D2,8.7D2,1.5D3,2.5D3/
DATA PHII/3.D0,2.D0,1.D0/
DATA PHII/1.85D0,1.7D0,1.65D0,1.4D0,1.3D0,1.25D0,1.12D0/
DATA SH1/-520.D-6,-320.D-6,-130.D-6/
DATA SH2/1.2D0,1.05D0,0.9D0,0.8D0,0.75D0,0.7D0,0.7D0/
C
IA=2
READ(9,*) NOCT
DO 500 I=1,NOCT
   READ(9,*) EC28
   WRITE(6,1001) EC28
   READ(9,*) ICURTY
   WRITE(6,1002) ICURTY
   READ(9,*) HUM
   WRITE(6,1003) HUM
   READ(9,*) AREA
   WRITE(6,1004) AREA
   READ(9,*) PER
   WRITE(6,1005) PER
C
1001 FORMAT(T5,'28-DAY MODULUS IN PSI:',T50,F15.6)
1002 FORMAT(T5,'CURE TYPE:','T50,11,/
   *T5,'1 ==> MOIST CURE;  2 ==> STEAM CURE')
1003 FORMAT(T5,'RELATIVE HUMIDITY (PERCENT):','T50,F5.2)
1004 FORMAT(T5,'CROSS-SECTIONAL AREA (SQ. MM.):','T50,F7.3)
1005 FORMAT(T5,'PERIMETER EXPOSED TO DRYING (MM.):','T50,F7.3)
C
T=7.D0
IF (ICURTY.EQ.2) T=3.D0
IF (HUM.LT.40.) HUM=40.D0
IF (HUM.GT.90.) HUM=90.D0
DO 10 K=1,3
   IF (HUM.EQ.HM(K)) THEN
      AMBH=RLMB(K)
      P1=PHII(K)
      EP1=SH1(K)
      GO TO 30
   END IF
10 CONTINUE
DO 20 K=2,3
   IF (HUM.LT.HM(K)) THEN
      M1=K-1
      M2=K
      GO TO 25
   END IF
20 CONTINUE
25 CON=(HUM-HM(M1))/(HM(M2)-HM(M1))
   AMBH=RLMB(M1)+CON*(RLMB(M2)-RLMB(M1))
   P1=PHII(M1)+CON*(PHII(M2)-PHII(M1))
   EP1=SH1(M1)+CON*(SH1(M2)-SH1(M1))
THICK=AMB*50.8*AREA/PER
IF (THICK.LT.50.) THICK=50.D0
IF (THICK.GT.1600.) THICK=1600.D0
JC=0
DO 100 K=1,7
   IF (THICK.EQ.HO(K)) THEN
      JC=1
      J1=K
      GO TO 150
   END IF
   IF (THICK.LT.HO(K)) THEN
      JC=2
      J3=K
      J2=K-1
      GO TO 150
   END IF
100 CONTINUE
IF (JC.EQ.1) THEN
   P2=PHI2(J1)
   EP2=SH2(J1)
END IF
IF (JC.EQ.2) THEN
   CON=(THICK-HO(J2))/(HO(J3)-HO(J2))
   P2=PHI2(J2)+CON*(PHI2(J3)-PHI2(J2))
   EP2=SH2(J2)+CON*(SH2(J3)-SH2(J2))
END IF
KC=0
DO 200 K=1,6
   IF (THICK.EQ.HO1(K)) THEN
      KC=1
      K1=K
      GO TO 250
   END IF
   IF (THICK.LT.HO1(K)) THEN
      KC=2
      K3=K
      K2=K-1
      GO TO 250
   END IF
200 CONTINUE
IF (KC.EQ.1) HFAC=HF(K1)
IF (KC.EQ.2) THEN
   CON=(THICK-HO1(K2))/(HO1(K3)-HO1(K2))
   HFAC=HF(K2)+CON*(HF(K3)-HF(K2))
END IF
C-------- CALCULATE CONSTANTS
PHIF=P1*P2
SHRINK=EP1*EP2
IF (ICURTY.EQ.1) THEN
   HF1=HFAC
   PHIF1=PHIF
   SHCEB1=SHRINK
   THK1=THICK
EC1=EC28
END IF
IF (ICURY.EQ.2) THEN
    HF2=HFAC
    PHIF2=PHIF
    SHCEB2=SHRINK
    THK2=THICK
    EC2=EC28
END IF

WRITE(6,*) 'SHRINK = ',SHRINK
WRITE(6,*) 'HFAC = ',HFAC
WRITE(6,*) 'PHIF = ',PHIF
WRITE(6,*) 'THICK = ',THICK
WRITE(6,*) 'EC28 = ',EC28

IF (ICURY.EQ.1) CALL DIRICH(0.0,ST,AGE,2,ICURY)
IF (ICURY.EQ.2) CALL DIRICH(0.0,ST2,AGE,2,ICURY)

500 CONTINUE
RETURN
END

============ CEBSH =============
SUBROUTINE CEBSH(TI,TF,K,EPSH)

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION BSH(12,6),TIME(12),HO1(6)
COMMON/CUB/EP(20),FPC28(20),WCONC(20),TL(20),ICURY(20),CURL(20)
COMMON/CEBSH/SHCEB1,SHCEB2,HF1,HF2,PHIF1,PHIF2,
*THK1,THK2,EC1,EC2
DATA TIME/2.D0,5.D0,1.D1,2.D1,5.D1,1.D2,2.D2,5.D2,1.D3,2.D3,
*5.D3,1.D4/
DATA HO1/5.D1,1.D2,2.D2,4.D2,8.D2,1.6D3/
DATA BSH/0.18D0,0.28D0,0.37D0,0.48D0,0.64D0,0.76D0,
*0.85D0,0.93D0,0.96D0,0.97D0,0.98D0,0.98D0,
*0.99D0,0.16D0,0.24D0,0.34D0,0.49D0,0.62D0,
*0.74D0,0.87D0,0.93D0,0.96D0,0.97D0,0.98D0,
*0.02D0,0.07D0,0.12D0,0.19D0,0.31D0,0.42D0,
*0.53D0,0.72D0,0.86D0,0.93D0,0.96D0,0.97D0,
*0.00D0,0.02D0,0.04D0,0.09D0,0.17D0,0.25D0,
*0.37D0,0.45D0,0.72D0,0.86D0,0.95D0,0.97D0,
*0.00D0,0.00D0,0.01D0,0.02D0,0.06D0,0.10D0,
*0.27D0,0.29D0,0.46D0,0.70D0,0.91D0,0.96D0,
*0.00D0,0.00D0,0.00D0,0.01D0,0.02D0,0.05D0,
*0.08D0,0.14D0,0.24D0,0.41D0,0.75D0,0.90D0/
SH=SHCEB1
IF (ICUR(K).GT.2) SH=SHCEB2
THK=THK1
IF (ICUR(K).GT.2) THK=THK2
IF (THK.GT.1600.) THK=1600.0
IF (THK.LT.50.0) THK=50.0
KC=0
TT=CURLEN(K)
DO 10 I=1,7
   IF (THK.EQ.H01(I)) THEN
      KC=1
      K1=I
      GO TO 20
   END IF
   IF (THK.LT.H01(I)) THEN
      KC=2
      K3=I
      K2=I-1
      GO TO 20
   END IF
10 CONTINUE
DO 100 I=1,2
   T=TF-TL(K)+TT
   IF (I.EQ.2) T=TI-TL(K)+TT
   IF (T.LE.2.) THEN
      BETA=0.0
      GO TO 70
   END IF
   IF (T.GT.10000.) T=10000.D0
   LC=0
   DO 30 J=1,12
      IF (T.EQ.TIME(J)) THEN
         LC=1
         L1=J
         GO TO 35
      END IF
      IF (T.LT.TIME(J)) THEN
         LC=2
         L3=J
         L2=J-1
         GO TO 35
      END IF
30 CONTINUE
35 IF ((LC.EQ.1).AND.(KC.EQ.1)) THEN
   BETA=BSH(L1,K1)
   GO TO 70
END IF
IF ((LC.EQ.1).AND.(KC.EQ.2)) THEN
   CON=(THK-H01(K2))/(H01(K3)-H01(K2))
   BETA=BSH(L1,K2)+CON*(BSH(L1,K3)-BSH(L1,K2))
   GO TO 70
END IF
IF ((LC.EQ.2).AND.(KC.EQ.1)) THEN
   CON=(T-TIME(L2))/(TIME(L3)-TIME(L2))
BETA = BSH(L2,K1) + CON*(BSH(L3,K1) - BSH(L2,K1))
    GO TO 70
END IF
DT = TIME(L3) - TIME(L2)
DH = HO1(K3) - HO1(K2)
DT1 = T - TIME(L2)
DH1 = THK - HO1(K2)
A1 = BSH(L2,K2) + DH1/DH*(BSH(L2,K3) - BSH(L2,K2))
A2 = BSH(L3,K2) + DH1/DH*(BSH(L3,K3) - BSH(L3,K2))
BETA = A1 + DT1/DT*(A2 - A1)
70 IF (1.EQ.1) B1 = BETA
    IF (1.EQ.2) B2 = BETA
100 CONTINUE
C
EPSH = SH*(B1 - B2)
C
RETURN
END
C
C=========================================================================
C
SUBROUTINE CHOOSE(A,RLAM,TP,JK)
C
CALLED BY: CREEP
CALLS : NONE
C
BASED ON THE LOADING AGE, THIS SUBROUTINE CHOOSES THE AGING COEFFICIENTS DEPENDING ON THE MATERIAL TYPE AND THE CREEP AND SHRINKAGE MODEL IN USE.
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(5),RLAM(5),S(20,5)
COMMON/ACOEFS/ST(20,5),AGE(20),NTIMES,MAGES
COMMON/ACOEF2/ST2(20,5)
C
TO = TP
ICODE = 0
IF (TO.LT.AGE(1)) TO = AGE(1)
IF (TO.GT.AGE(NTIMES)) TO = AGE(NTIMES)
DO 100 I = 1, NTIMES
  IF (TO.EQ.AGE(I)) THEN
    ICODE = 1
    I3 = I
    GO TO 150
  END IF
  IF (TO.LT.AGE(I)) THEN
    ICODE = 2
    I2 = I
    I1 = I2 - 1
    GO TO 150
  END IF
100 CONTINUE
C
150 IF (JK.EQ.1) THEN
DO 250 I=1,20
    DO 200 J=1,5
        S(I,J)=ST(I,J)
    200 CONTINUE
250 CONTINUE
END IF
IF (JK.EQ.2) THEN
    DO 350 I=1,20
        DO 300 J=1,5
            S(I,J)=ST2(I,J)
        300 CONTINUE
    350 CONTINUE
END IF
C
IF (ICODE.EQ.1) THEN
    DO 500 K=1,MAGES
        A(K)=S(I3,K)
    500 CONTINUE
GO TO 1000
END IF
IF (ICODE.EQ.2) THEN
    CONST=(TO-AGE(I1))/(AGE(I2)-AGE(I1))
    DO 600 K=1,MAGES
        A(K)=S(I1,K)+(S(I2,K)-S(I1,K))*CONST
    600 CONTINUE
GO TO 1000
END IF
WRITE(6,*)'ERROR OCCURED IN SUBROUTINE CHOOSE !!!'
STOP
1000 RETURN
END
©PROCESS DC(STRAIN,SIGMA,MDATA,STRN2,TEMPS,CEB1)
C
C=====================================================================
C=====================================================================
C
SUBROUTINE CREEP(TI,TF,IA,PRSTEP)
C
C CALLED BY: MAIN
C CALLS : ACISH, BP2SH, CEBCF,
C CEBSH, CHOOSE, TEMSFT
C
C SUBROUTINE TO CALCULATE PSEUDO INELASTIC STRAINS
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(5),RLAM(5),CINV(6,6),DB(6),B(4)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/SIGMA/GPCUM(2500,6,15),DELGP(2500,6,15)
COMMON/STRAIN/CSCUM(2500,6,15),DELC(2500,6,15),HSV(2500,4,6,15)
COMMON/MDATA/COOD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
* VBD(2000,3),VSFB(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/MOD/EPR(20),FPC28(20),WCONC(20),TL(20),ICUR(20),CURLEN(20)
COMMON/CEBCS/SHCEB1,SHCEB2,HF1,HF2,PHIF1,PHIF2,
*THK1,THK2,EC1,EC2
COMMON/STRN2/HSV2(2500,6,15)
COMMON/CEB1/DSTOR(2500,6,15)
COMMON/TEMPS/TEMP(2500,15,3),ITCOD(2500,15)
DATA RLAH/1.0D-1,1.0D-2,1.0D-3,1.0D-4,1.0D-5/

C

IF (IA.EQ.3) THEN
  RLM(1)=2.0D-1
  RLM(2)=2.0D-2
  RLM(3)=2.0D-3
END IF
IF (IA.EQ.2) THEN
  RLM(1)=1.5D-1
  RLM(2)=1.5D-2
  RLM(3)=1.5D-3
END IF

NN=3
C

ALPHA=1.D-5
EFF=2.95D6
EX=1.D0/3.D0
DELTA=TF-TI
K=0
DO 500 N=1,NUMEL
  K1=K
C

  K=NELC(N,2)
  IF (TL(K).GT.TI) GO TO 500
  IF (TL(K).EQ.TI) THEN
    IF ((PRSTEP.GT.0.0).AND.(TI.EQ.TF)) THEN
      GO TO 500
    END IF
  END IF
  IF (K1.EQ.K) GO TO 25
END IF

C------- SET UP INVERSE ELASTICITY MATRIX
E=EPR(K)
E1=EPR(K)
E2=RMAT(K,1)
PR=RMAT(K,2)
DO 7 I=1,6
  DO 6 J=1,6
    CINV(I,J)=0.0
  CONTINUE
6 CONTINUE
7 CONTINUE
DO 8 I=1,3
  CINV(I,I)=1.0
DO 9 I=4,6
9 CINV(I,I)=2.*(1.+PR)
CINV(1,2)=-PR
CINV(1,3)=-PR
CINV(2,1)=-PR
CINV(2,3)=-PR
CINV(3,1)=-PR
CINV(3,2)=-PR
C

TPR=TI-TL(K)+CURLEN(K)


\[ \text{JK} = 2 \]

\[ \text{IF (ICUR(K).LE.2) JK = 1} \]

---

**Pick Aging Coefficients for a Given Age**

```
CALL CHOOSE(A,RLAM,TPR,JK)
```

---

```
\[ \text{IF (IA.EQ.1) CALL ACISH(TI,TF,K,EPSh)} \]
```

```
\[ \text{IF (IA.EQ.2) THEN} \]
\[ \text{CALL CEBSH(TI,TF,K,EPSh)} \]
\[ \text{E} = \text{EC1} \]
\[ \text{IF (ICUR(K).GT.2) E} = \text{EC2} \]
\[ \text{HF} = \text{HF1} \]
\[ \text{IF (ICUR(K).GT.2) HF} = \text{HF2} \]
```

```
\[ \text{END IF} \]
```

```
\[ \text{IF (IA.EQ.3) CALL BP2SH(TI,TF,K,EPSh)} \]
```

```
DO 400 IP=1,15
```

---

**Obtain Temperature-Time Shift Function**

```
CALL TEMSFT(N,IP,F1,F2,TS1,TS2)
```

```
DO 40 I=1,NN
```

```
DO 30 J=1,6
```

```
P=DEXP(-RLAM(I)*PRSTEP*TS1)
Q=DELGP(N,J,IP)*A(I)/E
HSV(N,I,J,IP)=HSV(N,I,J,IP)*P+Q
```

```
IF (IA.EQ.2) THEN
```

```
IF (I.GT.1) GO TO 30
PP=-0.01D0*PRSTEP
P1=DEXP(PP)
Q1=DELGP(N,J,IP)*.292/E
HSV2(N,J,IP)=HSV2(N,J,IP)*P1+Q1
```

```
END IF
```

```
30 CONTINUE
```

```
DO 80 I=1,6
```

```
SUM=0.0
DO 70 J=1,NN
```

```
PQ=-RLAM(J)*DELTA
QR=PQ*TS2
SUM=SUM+HSV(N,J,IP)*(1.-DEXP(QR))
```

```
IF (IA.EQ.2) THEN
```

```
IF (J.GT.1) GO TO 70
PQ1=-0.01D0*DELTA
PQ2=1.-DEXP(PQ1)
SUM=SUM+HSV2(N,I,IP)*PQ2
```

```
END IF
```

```
70 CONTINUE
```

```
DB(I)=SUM
```

```
IF (IA.EQ.2) THEN
```

```
TEF=TPR
```

```
DD=1.276
```

```
IF (ICUR(K).GT.2) THEN
```

```
TEF=TPR+4.0
```

```
DD=1.2
```

```
END IF
CON=TEF/(4.2+0.85*TEF)
CON1=(CON**1.5)/DD
IF (CON1.GT.1.0) CON1=1.0
BA=0.8*(1.-CON1)
IF (TI.EQ.TF) THEN
  DSTOR(N,I,IP)=DSTOR(N,I,IP)+DELG(N,I,IP)
  GO TO 80
END IF
IF (PRSTEP.EQ.0.0) THEN
  DB(I)=DB(I)+0.108*(DELG(N,I,IP)+DSTOR(N,I,IP))/E
  GO TO 80
END IF
DSTOR(N,I,IP)=0.0
DB(I)=DB(I)+0.108*DELG(N,I,IP)/E
END IF

80 CONTINUE
DO 100 I=1,6
  SUM=0.0
  DO 90 J=1,6
    SUM=SUM+CINV(I,J)*DB(J)
  CONTINUE
90 CONTINUE
DELCS(N,I,IP)=SUM
CSCUM(N,I,IP)=CSCUM(N,I,IP)+SUM

C C---------- ADD TEMPERATURE AND SHRINKAGE STRAINS TO PSEUDO INELASTIC STRAIN VECTOR.
TSTRN=ALPHA*(F2-F1)
DO 120 I=1,3
  DELCS(N,I,IP)=DELCS(N,I,IP)+EPSH+TSTRN
  CSCUM(N,I,IP)=CSCUM(N,I,IP)+EPSH+TSTRN
120 CONTINUE

C C------ CALCULATE CORRECTION STRAIN DUE TO CHANGE IN ELASTIC MODULUS.
CONST=1./E1-1./E2
DO 190 I=1,6
  SUM=0.0
  DO 180 J=1,6
    SUM=SUM+CINV(I,J)*GPCUM(N,J,IP)
  CONTINUE
180 CONTINUE
  DB(I)=SUM
190 CONTINUE
DO 200 I=1,6
  CSCUM(N,I,IP)=CSCUM(N,I,IP)+DB(I)*CONST
200 CONTINUE

C C
C
400 CONTINUE
C
500 CONTINUE
C
RETURN
END
SUBROUTINE DCS(DIR,RJST,GP,IS,IG,N)

C CALLED BY: PSLOAD, TENEPS
C CALLS : NONE

C SUBROUTINE TO EVALUATE DIRECTION COSINES AT A POINT
C ON A PRESTRESSING STRAND

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION GP(2),DIR(6),DSI(3),XX(3),YY(3),ZZ(3)
COMMON/PRESTR/NCTEN(500,5),PCOOD(2000,3),FS(500,2),
*CF(500,2),FI(50),AR(50),SEG(500,3,3)

C -- DSI - DERIVATIVES OF STRAND SHAPE FUNCTIONS --
C -- FILL UP XX, YY & ZZ --
DO 20 I=1,3
   IP=NCTEN(IS,I)
   XX(I)=PCOOD(IP,1)
   YY(I)=PCOOD(IP,2)
   ZZ(I)=PCOOD(IP,3)
20 CONTINUE

C -- CALCULATE {DSI} --
   SP=GP(IG)
   DSI(1)=SP-0.5
   DSI(2)=-2.*SP
   DSI(3)=SP+0.5

C -- CALCULATE JACOBIAN OF STRAND - RJST --
   X=0.0
   Y=0.0
   Z=0.0
DO 40 I=1,3
   X=X+DSI(I)*XX(I)
   Y=Y+DSI(I)*YY(I)
   Z=Z+DSI(I)*ZZ(I)
40 CONTINUE
   RJST=DSQRT(X*X+Y*Y+Z*Z)
IF (RJST.EQ.0.0) THEN
   STOP
END IF

C -- CALCULATE DIRECTION COSINES --
   RL=X/RJST
   RM=Y/RJST
   RN=Z/RJST
C -- FILL UP {DIR} --
   DIR(1)=RL*RL
   DIR(2)=RM*RM
   DIR(3)=RN*RN
   DIR(4)=RL*RM
   DIR(5)=RM*RN
   DIR(6)=RL*RN

C
RETURN
END

@PROCESS DC(MDATA,STIFF,STDISP,GFV)
C
C============================================
C
SUBROUTINE DEFSTR(ICODE)
C
CALLED BY:  MAIN
CALLS :  NONE
C
THIS SUBROUTINE ADDS NODAL DISPLACEMENTS TO THE CURRENT
GEOMETRY TO OBTAIN THE NEW GEOMETRY.
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/MDATA/COOD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBD(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/STIFF/GSTIF(20000000)
COMMON/GFV/GF(1000000)
COMMON/STDISP/DS(1000000)
C
DO 100 I=1,NUMNP
   K3=I*3
   K2=K3-1
   K1=K2-1
   COLD(I,2)=COOD(I,2)
   COLD(I,3)=COOD(I,3)
   COLD(I,4)=COOD(I,4)
   COOD(I,2)=COOD(I,2)+GF(K1)
   COOD(I,3)=COOD(I,3)+GF(K2)
   COOD(I,4)=COOD(I,4)+GF(K3)
   IF (ICODE.EQ.3) THEN
      COOD(I,2)=COOD(I,2)+DS(K1)
      COOD(I,3)=COOD(I,3)+DS(K2)
      COOD(I,4)=COOD(I,4)+DS(K3)
   END IF
100 CONTINUE
RETURN
END

C============================================
C
SUBROUTINE DIRICH(CR,A,C,IA,ICURTY)
C
CALLED BY: ACI209, BAPAN2
CALLS : ACICRP, AGING, BP2CRP, CEBCRP
C
SUBROUTINE TO SET UP LEAST-SQUARES PROCEDURE TO CALCULATE
AGING COEFFICIENTS IN THE DIRICHLET SERIES.
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RLM(5),T1(200),A(20,5),C(20),CT(200),T0(19),F(5)
DIMENSION FIP(16,4)
COMMON/ACOEFS/ST(20,5),AGE(20),NTIMES,MAGES
DATA T0/3.D0,7.D0,10.D0,14.D0,20.D0,28.D0,35.D0,60.D0,75.D0,
*90.D0,110.D0,140.D0,180.D0,230.D0,295.D0,365.D0,420.D0,
*600.D0,800.D0/

IF (IA.EQ.3) THEN
   RLAM(1)=2.0D-1
   RLAM(2)=2.0D-2
   RLAM(3)=2.0D-3
END IF
IF (IA.EQ.2) THEN
   RLAM(1)=1.5D-1
   RLAM(2)=1.5D-2
   RLAM(3)=1.5D-3
END IF

N=10
NTIMES=19
MAGES=3
WRITE(6,1001) MAGES
WRITE(6,1002) (RLAM(J),J=1,5)
WRITE(8,1001) MAGES
WRITE(8,1002) (RLAM(J),J=1,5)

1001 FORMAT(/,5X,'N0. OF TERMS = ',I1)
1002 FORMAT(/,5X,'RLAM = ',5(F8.5,2X),//)

TI(1)=30.0
TI(2)=60.0
TI(3)=150.0
TI(4)=300.0
TI(5)=400.0
TI(6)=500.0
TI(7)=600.0
TI(8)=700.0
TI(9)=900.0
TI(10)=1000.0
DO 200 K=1,NTIMES
   AGE(K)=T0(K)
   TPR=T0(K)
   IF (IA.EQ.1) THEN
      W1=-0.118D0
      W2=-0.094D0
      IF (ICURTY.EQ.1) CCR=CR*1.25*(TPR**W1)
      IF (ICURTY.EQ.2) CCR=CR*1.13*(TPR**W2)
      CALL ACICRP(N,TI,CT,CCR)
   END IF
   IF (IA.EQ.2) CALL CEBCRP(N,TI,CT,TPR,ICURTY)
   IF (IA.EQ.3) CALL BP2CRP(N,TI,CT,TPR,ICURTY)
   CALL AGING(MAGES,N,RLAM,TI,CT,F,IA,TPR)
   C(K)=TPR

WRITE(8,*),'AGE =',TPR
WRITE(8,*),' '
WRITE(8,*),'ACTUAL APPROXIMATE %DIFF'
DO 500 I=1,N
   SUM=0.0
DO 400 J=1,MAGES
   SUM=SUM+F(J)*(1.0-DEXP(-RLAM(J)*TI(I)))
400 CONTINUE
DIFF=(SUM-CT(I))/CT(I)*100.0
WRITE(8,*),'TIME = ',TI(I)
WRITE(8,*) CT(I),SUM,DIFF
500 CONTINUE
C
   DO 25 L=1,MAGES
25   A(K,L)=F(L)
C
200 CONTINUE
C
RETURN
END
@PROCESS DC(MDATA)
C
C------------------- DIST20 ------------------------------------------
C
SUBROUTINE DIST20(NEL,ICODE,NINT,XX,SIG,RE)
C
C CALLED BY: ASSEM
C CALLS : JAC20
C
C SUBROUTINE TO CALCULATE EQUIVALENT NODAL LOADS FOR
C GIVEN ELEMENT FACE LOADS.
C
IMPLICIT REAL*8 (A-H,0-Z)
DIMENSION XX(20,3),P(3,20),XJ(3,3),RE(60)
COMMON/FACES/IFACE(6,8)
COMMON/size/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/MDATA/COORD(25000,4),NELC(25000,22),RMAT(20,5),IEL(2500),
*VBDF(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/SHP20/AN(20)
COMMON/QUADR/XG(4,4),WT(4,4)
COMMON/INTEGRAL/PTS(15,3),WTS(3)
C
DO 10 II=1,60
  RE(II)=0.0
10  DO 100 J=1,8
    I=IFACE(ICODE,J)
   DO 90 LY=1,NINT
     A=XG(LY,NINT)
   DO 80 LZ=1,NINT
      B=XG(LZ,NINT)
     C=1.0
   D=-1.0
     IF (ICODE.EQ.1) CALL JAC20(C,A,B,XX,P,XJ)
     IF (ICODE.EQ.2) CALL JAC20(D,A,B,XX,P,XJ)
     IF (ICODE.EQ.3) CALL JAC20(A,C,B,XX,P,XJ)
     IF (ICODE.EQ.4) CALL JAC20(A,D,B,XX,P,XJ)
   90 CONTINUE
  80 CONTINUE
100 CONTINUE
IF (ICODE.EQ.5) CALL JAC20(A,B,C,XX,P,XJ)
IF (ICODE.EQ.6) CALL JAC20(A,B,D,XX,P,XJ)
   WGT=WT(LY,NINT)*WT(LZ,NINT)
   I3=I*3
   I2=I3-1
   I1=I3-2
   IA=1
   IB=2
   IF (((ICODE.EQ.1).OR.(ICODE.EQ.2)) THEN
      IA=2
      IB=3
   END IF
   IF (((ICODE.EQ.3).OR.(ICODE.EQ.4)) THEN
      IA=3
      IB=1
   END IF
   RJ1=(XJ(IA,2)*XJ(IB,3))-(XJ(IA,3)*XJ(IB,2))
   RJ2=(XJ(IA,3)*XJ(IB,1))-(XJ(IA,1)*XJ(IB,3))
   RJ3=(XJ(IA,1)*XJ(IB,2))-(XJ(IA,2)*XJ(IB,1))

C
RE(I1)=RE(I1)+AN(I)*SIG*RJ1*WGT
RE(I2)=RE(I2)+AN(I)*SIG*RJ2*WGT
RE(I3)=RE(I3)+AN(I)*SIG*RJ3*WGT

80 CONTINUE
90 CONTINUE
100 CONTINUE
C
RETURN
END
C
C======== DOT =======
C
FUNCTION DOT(A,B,N)
C
C CALLED BY: SKYLIN
C CALLS : NONE
C
C FUNCTION TO CALCULATE THE DOT PRODUCT OF TWO VECTORS
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A( 172000),B(6000)
DOT=0.0
DO 100 I=1,N
   DOT=DOT+A(I)*B(I)
100 CONTINUE
RETURN
END
SUBROUTINE ELAST(TF,IA,ICODE)

CALLED BY: MAIN
CALLS : NONE

SUBROUTINE TO CALCULATE THE VALUE OF THE MODULUS OF ELASTICITY
AT THE END OF A TIME INTERVAL.

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NEELDL
COMMON/MDATA/COOD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBDF(2000,3),VSFB(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/MDATA/COOD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBDF(2000,3),VSFB(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/MDATA/COOD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBDF(2000,3),VSFB(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/CEBCS/SHCEB1,SHCEB2,THK1,THK2,PB1,PB2,PB3,PB4
COMMON/BPEC/PH1,PH2,E1,E2
COMMON/BPC2S/PH1,PH2,E1,E2,R1,R2
C
IF (IA.EQ.4) GO TO 1000
C
DO 100 I=1,NUMMAT
IF (ICODE.EQ.0) THEN
   IF (TF.NE.TL(I)) GO TO 100
   GO TO 75
END IF
EPR(I)=RMAT(I,1)
IF (TF.LE.TL(I)) GO TO 100
50 TFF=TF+CURLEN(I)-TL(I)
C
IF (ICUR(I).EQ.1) CONST=TFF/(4.0+0.85*TFF)
IF (ICUR(I).EQ.2) CONST=TFF/(2.3+0.92*TFF)
IF (ICUR(I).EQ.3) CONST=TFF/(1.0+0.95*TFF)
IF (ICUR(I).EQ.4) CONST=TFF/(0.7+0.98*TFF)
FPC=CONST*FPC28(I)
CONST=33.0*(WCONC(I)**1.5)
RMAT(I,1)=CONST*DSQRT(FPC)
C
75 IF (ICODE.EQ.0) EPR(I)=RMAT(I,1)
C
100 CONTINUE
C
WRITE(6,1010)
DO 800 I=1,NUMMAT
   WRITE(6,1020) I,EPR(I),RMAT(I,1)
800 CONTINUE
1000 RETURN
1010 FORMAT(//,5X,'MODULUS OF ELASTICITY :','//,'
*T8','MATERIAL NO.','T28','AT PREVIOUS TIME','T48','AT NEXT TIME','//)
1020 FORMAT(T11,I2,T27,F15.4,T47,F15.4)
SUBROUTINE ELCD(N, XX, DR, D)

C CALLED BY: NEWPOS, TENEPS
C CALLS : NONE
C
C SUBROUTINE TO SET UP ELEMENT NODAL COORDINATE ARRAY AND
C DISPLACEMENT VECTOR

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION D(100000), XX(20,3), DR(60)
COMMON/MDATA/COOD(25000,4), NELC(2500,22), RMAT(20,5), IEL(2500),
* VBDF(2000,3), VSBF(1000,4), ELDL(500,3), COLD(25000,4), ILEV(2500)

DO 20 J = 3, 22
   J1 = J - 2
   L = NELC(N, J)
   LL = L*3
   LLM2 = LL - 2
   LLM1 = LL - 1
   JJ = J1*3
   JJM2 = JJ - 2
   JJM1 = JJ - 1
   DR(JJM2) = D(LLM2)
   DR(JJM1) = D(LLM1)
   DR(JJ) = D(LL)
   DO 10 K = 1, 3
      K1 = K + 1
      XX(J1, K) = COLD(L, K1)
   10 CONTINUE
20 CONTINUE
C
RETURN
END
C
C________________ GESCP ________________
C
SUBROUTINE GESCP(A,B,N)
C
CALLED BY: AGING
CALLS : NONE
C
GAUSSIAN ELIMINATION WITH SCALED COLUMN PIVOTING
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(4,4),B(4),C(4,4),S(4),NROW(4),X(4)
C
NM1=N-1
DO 10 I=1,N
   SMAX=DABS(A(I,1))
   DO 5 J=2,N
      IF (DABS(A(I,J)).GT.SMAX) SMAX=DABS(A(I,J))
  5 CONTINUE
   S(I)=SMAX
   IF (SMAX.EQ.0.0) THEN
      WRITE(6,2000)
      STOP
   END IF
   NROW(I)=I
  10 CONTINUE
C
DO 50 I=1,NM1
   RMAX=DABS(A(NROW(I),I))/S(NROW(I))
   IP=I
   DO 30 J=I+1,N
      R=DABS(A(NROW(J),I))/S(NROW(J))
      IF (R.GT.RMAX) THEN
         IP=J
         RMAX=R
      END IF
   30 CONTINUE
C
IF (IP.NE.I) THEN
   NCOPY=NROW(I)
   NROW(I)=NROW(IP)
   NROW(IP)=NCOPY
END IF
DO 40 J=I+1,N
   C(NROW(J),I)=A(NROW(J),I)/A(NROW(I),I)
   DO 35 K=1,N
      A(NROW(J),K)=A(NROW(J),K)-C(NROW(J),I)*A(NROW(I),K)
   35 CONTINUE
   B(NROW(J))=B(NROW(J))-C(NROW(J),I)*B(NROW(I))
  40 CONTINUE
C
BACK SUBSTITUTION
X(N)=B(NROW(N))/A(NROW(N),N)
DO 90 I=NM1,1,-1
C
214
SUM=0.0
DO 80 J=I+1,N
    SUM=SUM+A(NROW(I),J)*X(J)
80    CONTINUE
    X(I)=(B(NROW(I))-SUM)/A(NROW(I),I)
90    CONTINUE
C
DO 100 I=1,N
100   B(I)=X(I)
C
2000 FORMAT(15X,'NO UNIQUE SOLUTION EXISTS !')
C
RETURN
END
@PROCESS DC(MDATA,TEMPS)
C
C=---------------- HEAT =----------------
C
SUBROUTINE HEAT(ICODE,DEG,ISLAB,TI)
C
CALLED BY: MAIN
CALLS : NONE
C
SUBROUTINE TO ASSIGN TEMPERATURES TO INTEGRATION POINTS.
CODES ARE ASSIGNED TO EACH INTEGRATION POINT (ITCOD(I,J)) TO
DETERMINE WHETHER IT EXPERIENCES A NEW MAXIMUM TEMPERATURE
OR NOT.
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/DEPTH/DEP(6),ADTEMP(4,15)
COMMON/TEMPS/TEMP(2500,15,3),ITCOD(2500,15)
COMMON/MOD/EPR(20),FPC28(20),WCONC(20),TL(20),ICUR(20),CURLEN(20)
COMMON/MDATA/COORD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBD(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NBSF,NSBF,NHBW,NEQ,NELDL
C
DO 500 N=1,NUMEL
  KK=NELC(N,2)
  IF (TL(KK).GT.TI) GO TO 400
  DO 100 IP=1,15
    IF (ICODE.EQ.0) THEN
      TEMP(N,IP,1)=DEG
      TEMP(N,IP,3)=DEG
      IF (DEG.LE.20.) TEMP(N,IP,3)=20.0
    END IF
    TEMP(N,IP,1)=TEMP(N,IP,2)
  50    K=ILEV(N)
    IF (K.LT.3) THEN
      TEMP(N,IP,2)=DEG
    GO TO 75
    END IF
  75    KM2=K-2
    TEMP(N,IP,2)=DEG+ADTEMP(KM2,IP)
    JJ=0
    IF (TEMP(N,IP,2).GT.TEMP(N,IP,3)) THEN
      JJ=1
      TEMP(N,IP,3)=TEMP(N,IP,2)
    END IF
    IF (ITCOD(N,IP).EQ.0) THEN
      IF (JJ.EQ.1) ITCOD(N,IP)=2
      GO TO 100
    END IF
    IF (ITCOD(N,IP).EQ.1) THEN
      IF (TEMP(N,IP,2).GE.TEMP(N,IP,1)) ITCOD(N,IP)=3
    END IF
  100   CONTINUE
GO TO 500
C

400 IF (ISLAB.EQ.1) THEN
    DO 420 IP=1,15
        DO 410 J=1,3
            TEMP(N,IP,J)=DEG
            IF (TEMP(N,IP,3).LE.20.0) TEMP(N,IP,3)=20.0
    420 CONTINUE
    END IF

C

500 CONTINUE

C

RETURN

END
SUBROUTINE INIT(ICODE, LEN)

CALLED BY: MAIN
CALLS : NONE

SUBROUTINE TO INITIALIZE GLOBAL STIFFNESS AND FORCE MATRICES. ALSO CALCULATES THE HALF-BANDWIDTH OF THE GLOBAL STIFFNESS MATRIX.

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/STIFF/GSTIF(20000000)
COMMON/GFV/GF(1000000)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/MDATA/COORD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),*VBDVF(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)

IF (ICODE.NE.0) GO TO 15

C ----- COMPUTE HALF BAND WIDTH -------

NHBW=0
DO 10 N=1,NUMEL
NPE=20
IF ((IEL(N).EQ.1).OR.(IEL(N).EQ.2)) NPE=8
DO 10 I=1,NPE
DO 10 J=1,NPE
   I2=I+2
   J2=J+2
   NW=(IABS(NELC(N,I2)-NELC(N,J2))+1)*3
10 IF (NHBW.LT.NW) NHBW=NW
WRITE(8,*) ' NO. OF EQUATIONS = ',NEQ
WRITE(8,*) ' HALF BAND WIDTH = ',NHBW
C ----- INITIALIZE GLOBAL STIFFNESS MATRIX AND FORCE VECTOR ----
15 DO 30 I=1,NEQ
   GF(I)=0.0
30 CONTINUE
   DO 20 J=1,LEN
      GSTIF(J)=0.0
20 CONTINUE

RETURN

END
SUBROUTINE JACOB(P, XX, XJ, DET, NEL, NN)

CALLED BY: STQB
CALLS: NONE

THIS SUBROUTINE CALCULATES THE JACOBIAN, IT'S INVERSE
AND ITS DETERMINANT FOR ANY POINT IN A 3-D ISOPARAMETRIC
ELEMENT.

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION P(3,NN),XX(NN,3),XJ(3,3),XJI(3,3)

DO 20 I=1,3
DO 10 J=1,3
   SUM=0.0
   DO 5 K=1,NN
      SUM=SUM+P(I,K)*XX(K,J)
   5 CONTINUE
   XJ(I,J)=SUM
10 CONTINUE
20 CONTINUE

RJ1=(XJ(2,2)*XJ(3,3))-(XJ(2,3)*XJ(3,2))
RJ2=(XJ(2,3)*XJ(3,1))-(XJ(2,1)*XJ(3,3))
RJ3=(XJ(2,1)*XJ(3,2))-(XJ(2,2)*XJ(3,1))
RJ4=(XJ(1,3)*XJ(3,3))-(XJ(1,2)*XJ(3,3))
RJ5=(XJ(1,1)*XJ(3,3))-(XJ(1,3)*XJ(3,1))
RJ6=(XJ(1,2)*XJ(3,1))-(XJ(1,1)*XJ(3,3))
RJ7=(XJ(1,2)*XJ(2,3))-(XJ(1,3)*XJ(2,2))
RJ8=(XJ(2,1)*XJ(1,3))-(XJ(1,1)*XJ(2,3))
RJ9=(XJ(1,1)*XJ(2,2))-(XJ(1,2)*XJ(2,1))

DET=XJ(1,1)*RJ1+XJ(1,2)*RJ2+XJ(1,3)*RJ3
IF (DET.LE.0.0) THEN
   WRITE(6,2000) NEL,DET
2000 FORMAT(5X,'ERROR IN ELEMENT ',I5,', DETERMINANT = ',D15.8)
   STOP
END IF
DUM=1.0/DET

XJI(1,1)=RJ1*DUM
XJI(1,2)=RJ4*DUM
XJI(1,3)=RJ7*DUM
XJI(2,1)=RJ2*DUM
XJI(2,2)=RJ5*DUM
XJI(2,3)=RJ8*DUM
XJI(3,1)=RJ3*DUM
XJI(3,2)=RJ6*DUM
XJI(3,3)=RJ9*DUM

RETURN
SUBROUTINE JAC20 (R, S, T, XX, P, YJ)

CALLED BY: DIST20
CALLS: NONE

SUBROUTINE TO CALCULATE THE JACOBIAN ON ELEMENT FACES, TO BE USED BY SUBROUTINE 'DIST20'.

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION YJ(3,3), P(3,20), XX(20,3)
COMMON/SHP20/AN(20)

A1 = 1. - R*R
A2 = 1. - R
A3 = 1. + R
B1 = 1. - S*S
B2 = 1. - S
B3 = 1. + S
C1 = 1. - T*T
C2 = 1. - T
C3 = 1. + T

C ------ SHAPE FUNCTIONS -------

AN(9) = 0.25 * A1 * B3 * C3
AN(10) = 0.25 * A2 * B1 * C3
AN(11) = 0.25 * A1 * B2 * C3
AN(12) = 0.25 * A3 * B1 * C3
AN(13) = 0.25 * A1 * B3 * C2
AN(14) = 0.25 * A2 * B1 * C2
AN(15) = 0.25 * A1 * B2 * C2
AN(16) = 0.25 * A3 * B1 * C2
AN(17) = 0.25 * A1 * B3 * C1
AN(18) = 0.25 * A2 * B3 * C1
AN(19) = 0.25 * A2 * B2 * C1
AN(20) = 0.25 * A3 * B2 * C1
AN(1) = 0.125 * A3 * B3 * C3 - 0.5 * (AN(9) + AN(12) + AN(17))
AN(2) = 0.125 * A3 * B3 * C3 - 0.5 * (AN(9) + AN(2) + AN(18))
AN(3) = 0.125 * A2 * B2 * C3 - 0.5 * (AN(10) + AN(11) + AN(19))
AN(4) = 0.125 * A3 * B2 * C3 - 0.5 * (AN(11) + AN(12) + AN(20))
AN(5) = 0.125 * A3 * B3 * C2 - 0.5 * (AN(13) + AN(16) + AN(17))
AN(6) = 0.125 * A2 * B3 * C2 - 0.5 * (AN(13) + AN(14) + AN(18))
AN(7) = 0.125 * A2 * B2 * C2 - 0.5 * (AN(14) + AN(15) + AN(19))
AN(8) = 0.125 * A3 * B2 * C2 - 0.5 * (AN(15) + AN(16) + AN(20))

C ------- DERIVATIVES OF SHAPE FUNCTIONS -------

P(1,9) = -0.5 * R * B3 * C3
P(2,9) = 0.25 * A2 * A3 * C3
P(3,9) = 0.25 * A2 * A3 * B3
P(1,10) = -0.25 * B2 * B3 * C3
P(2,10) = -0.5 * A2 * S * C3
P(3,10) = 0.25 * A2 * B3 * B2
P(1,11) = -0.5 * R * B2 * C3
\[
\begin{align*}
P(2,11) &= -P(2,9) \\
P(3,11) &= 0.25 \times A_1 \times B_2 \\
P(1,12) &= P(1,10) \\
P(2,12) &= -0.5 \times A_3 \times S \times C_3 \\
P(3,12) &= 0.25 \times A_3 \times B_1 \\
P(1,13) &= -0.5 \times B_3 \times C_2 \times R \\
P(2,13) &= 0.25 \times A_1 \times C_2 \\
P(3,13) &= -P(3,9) \\
P(1,14) &= -0.25 \times B_1 \times C_2 \\
P(2,14) &= -0.5 \times A_2 \times C_2 \times S \\
P(3,14) &= -P(3,10) \\
P(1,15) &= -0.5 \times B_2 \times C_2 \times R \\
P(2,15) &= -0.25 \times A_1 \times C_2 \\
P(3,15) &= -P(3,11) \\
P(1,16) &= 0.25 \times B_1 \times C_2 \\
P(2,16) &= 0.5 \times A_3 \times S \times C_2 \\
P(3,16) &= -P(3,12) \\
P(1,17) &= 0.25 \times B_3 \times C_2 \times C_3 \\
P(2,17) &= 0.25 \times A_3 \times C_2 \times C_3 \\
P(3,17) &= 0.5 \times A_3 \times B_3 \times T \\
P(1,18) &= -P(1,17) \\
P(2,18) &= 0.25 \times A_2 \times C_1 \\
P(3,18) &= -0.5 \times A_2 \times B_3 \times T \\
P(1,19) &= -0.25 \times B_2 \times C_1 \\
P(2,19) &= -P(2,18) \\
P(3,19) &= -0.5 \times A_2 \times B_2 \times T \\
P(1,20) &= 0.25 \times B_2 \times C_1 \\
P(2,20) &= -P(2,17) \\
P(3,20) &= -0.5 \times A_3 \times B_2 \times T \\
\end{align*}
\]
\[ P(1,7) = -0.125B_2C_2 - 0.5(P(1,14) + P(1,15) + P(1,19)) \]
\[ P(2,7) = -0.125A_2C_2 - 0.5(P(2,14) + P(2,15) + P(2,19)) \]
\[ P(3,7) = -0.125A_2B_2 - 0.5(P(3,14) + P(3,15) + P(3,19)) \]
\[ P(1,8) = 0.125B_2C_2 - 0.5(P(1,15) + P(1,16) + P(1,20)) \]
\[ P(2,8) = -0.125A_3C_2 - 0.5(P(2,15) + P(2,16) + P(2,20)) \]
\[ P(3,8) = -0.125A_3B_2 - 0.5(P(3,15) + P(3,16) + P(3,20)) \]

\begin{verbatim}
C
DO 20 I=1,3
  DO 10 J=1,3
    SUM=0.0
    DO 5 K=1,20
      SUM=SUM+P(I,K)*XX(K,J)
      CONTINUE
    YJ(I,J)=SUM
  CONTINUE
20 CONTINUE
C
RETURN
END
\end{verbatim}
SUBROUTINE LOCAL(N,XX,XP,R,S,T)

CALLED BY: TENEPS
CALLS : JACOB

SUBROUTINE TO PERFORM FIXED-POINT ITERATION TO OBTAIN NORMALIZED COORDINATES FROM GLOBAL COORDINATES.

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION XX(20,3),XP(3),P(3,20),XJINV(3,3),XJTINV(3,3),XG(60)
DIMENSION XOLD(3),XNEW(3),AA(20),SHP(3,60),XPP(3),DIFF(3)
LOGICAL YES

NN=20
XOLD(1)=R
XOLD(2)=S
XOLD(3)=T

C -- SET UP VECTOR OF COORDINATES OF NODES --
DO 20 I=1,20
   DO 10 J=1,3
      K=(I-1)*3+J
      XG(K)=XX(I,J)
10 CONTINUE
20 CONTINUE

C -- INITIALIZE SHAPE FUNCTION MATRIX --
DO 40 I=1,3
   DO 30 J=1,60
      SHP(I,J)=0.0
30 CONTINUE
40 CONTINUE

C -- BEGIN ITERATIONS --
TOL=1.D-5
DO 1000 IT=1,30

C -- SET UP SHAPE FUNCTION AND DERIVATIVES --
A1=1.-R*R
A2=1.-R
A3=1.+R
B1=1.-S*S
B2=1.-S
B3=1.+S
C1=1.-T*T
C2=1.-T
C3=1.+T

C ------ SHAPE FUNCTIONS ------
AA(9)=0.25*A1*B3*C3
AA(10)=0.25*A2*B1*C3
AA(11)=0.25*A1*B2*C3
AA(12)=0.25*A3*B1*C3
AA(13)=0.25*A1*B3*C2
AA(14)=0.25*A2*B1*C2
AA(15)=0.25*A1*B2*C2
AA(16)=0.25*A3*B1*C2
\( \text{AA(17)}=0.25\text{A3*B3*C1} \)
\( \text{AA(18)}=0.25\text{A2*B3*C1} \)
\( \text{AA(19)}=0.25\text{A2*B2*C1} \)
\( \text{AA(20)}=0.25\text{A3*B2*C1} \)
\( \text{AA(1)}=0.125\text{A3*B3*C3-0.5*(AA(9)+AA(12)+AA(17))} \)
\( \text{AA(2)}=0.125\text{A2*B3*C3-0.5*(AA(9)+AA(10)+AA(18))} \)
\( \text{AA(3)}=0.125\text{A2*B2*C3-0.5*(AA(10)+AA(11)+AA(19))} \)
\( \text{AA(4)}=0.125\text{A3*B2*C3-0.5*(AA(11)+AA(12)+AA(20))} \)
\( \text{AA(5)}=0.125\text{A3*B3*C2-0.5*(AA(13)+AA(16)+AA(17))} \)
\( \text{AA(6)}=0.125\text{A2*B3*C2-0.5*(AA(13)+AA(14)+AA(18))} \)
\( \text{AA(7)}=0.125\text{A2*B2*C2-0.5*(AA(14)+AA(15)+AA(19))} \)
\( \text{AA(8)}=0.125\text{A3*B2*C2-0.5*(AA(15)+AA(16)+AA(20))} \)

**C ------- DERIVATIVES OF SHAPE FUNCTIONS ---------**

\( \text{P(1,9)}=-0.5\text{R*B3*C3} \)
\( \text{P(2,9)}=0.25\text{A2*A3*C3} \)
\( \text{P(3,9)}=0.25\text{A2*A3*B3} \)
\( \text{P(1,10)}=-0.25\text{B2*B3*C3} \)
\( \text{P(2,10)}=-0.5\text{A2*S*C3} \)
\( \text{P(3,10)}=0.25\text{A2*B3*B2} \)
\( \text{P(1,11)}=-0.5\text{B2*C3*R} \)
\( \text{P(2,11)}=-\text{P(2,9)} \)
\( \text{P(3,11)}=0.25\text{A1*B2} \)
\( \text{P(1,12)}=-\text{P(1,10)} \)
\( \text{P(2,12)}=-0.5\text{A3*C3*S} \)
\( \text{P(3,12)}=0.25\text{A3*B1} \)
\( \text{P(1,13)}=-0.5\text{B3*C2*R} \)
\( \text{P(2,13)}=0.25\text{A1*C2} \)
\( \text{P(3,13)}=-\text{P(3,9)} \)
\( \text{P(1,14)}=-0.25\text{B1*C2} \)
\( \text{P(2,14)}=-0.5\text{A2*C2*S} \)
\( \text{P(3,14)}=-\text{P(3,10)} \)
\( \text{P(1,15)}=-0.5\text{B2*C2*R} \)
\( \text{P(2,15)}=0.25\text{A1*C2} \)
\( \text{P(3,15)}=-\text{P(3,11)} \)
\( \text{P(1,16)}=0.25\text{B1*C2} \)
\( \text{P(2,16)}=-0.5\text{A3*C2*S} \)
\( \text{P(3,16)}=-\text{P(3,12)} \)
\( \text{P(1,17)}=0.25\text{B3*C2*C3} \)
\( \text{P(2,17)}=-0.25\text{A3*C2*C3} \)
\( \text{P(3,17)}=-0.5\text{A3*B3*T} \)
\( \text{P(1,18)}=-\text{P(1,17)} \)
\( \text{P(2,18)}=0.25\text{A2*C1} \)
\( \text{P(3,18)}=-0.5\text{A2*B3*T} \)
\( \text{P(1,19)}=-0.25\text{B2*C1} \)
\( \text{P(2,19)}=-\text{P(2,18)} \)
\( \text{P(3,19)}=-0.5\text{A2*B2*T} \)
\( \text{P(1,20)}=0.25\text{B2*C1} \)
\( \text{P(2,20)}=-\text{P(2,17)} \)
\( \text{P(3,20)}=-0.5\text{A3*B2*T} \)

\( \text{P(1,1)}=0.125\text{B3*C3-0.5*(P(1,9)+P(1,12)+P(1,17))} \)
\( \text{P(2,1)}=0.125\text{A3*C3-0.5*(P(2,9)+P(2,12)+P(2,17))} \)
\( \text{P(3,1)}=0.125\text{A3*B3-0.5*(P(3,9)+P(3,12)+P(3,17))} \)
\[
P(1,2) = -0.125B_3C_3 - 0.5(P(1,10) + P(1,9) + P(1,18)) \\
P(2,2) = 0.125A_2C_3 - 0.5(P(2,10) + P(2,9) + P(2,18)) \\
P(3,2) = -0.125A_2B_3 - 0.5(P(3,10) + P(3,9) + P(3,18)) \\
P(1,3) = -0.125B_2C_3 - 0.5(P(1,10) + P(1,11) + P(1,19)) \\
P(2,3) = -0.125A_2C_3 - 0.5(P(2,10) + P(2,11) + P(2,19)) \\
P(3,3) = -0.125A_2B_2 - 0.5(P(3,10) + P(3,11) + P(3,19)) \\
P(1,4) = -0.125B_3C_2 - 0.5(P(1,13) + P(1,16) + P(1,17)) \\
P(2,4) = -0.125A_3C_2 - 0.5(P(2,13) + P(2,16) + P(2,17)) \\
P(3,4) = -0.125A_3B_3 - 0.5(P(3,13) + P(3,16) + P(3,17)) \\
P(1,5) = -0.125B_3C_2 - 0.5(P(1,13) + P(1,14) + P(1,18)) \\
P(2,5) = -0.125A_3C_2 - 0.5(P(2,13) + P(2,14) + P(2,18)) \\
P(3,5) = -0.125A_3B_3 - 0.5(P(3,13) + P(3,14) + P(3,18)) \\
P(1,6) = -0.125B_2C_2 - 0.5(P(1,14) + P(1,15) + P(1,19)) \\
P(2,6) = -0.125A_2C_2 - 0.5(P(2,14) + P(2,15) + P(2,19)) \\
P(3,6) = -0.125A_2B_3 - 0.5(P(3,14) + P(3,15) + P(3,19)) \\
P(1,7) = -0.125B_2C_2 - 0.5(P(1,14) + P(1,15) + P(1,20)) \\
P(2,7) = -0.125A_2C_2 - 0.5(P(2,14) + P(2,15) + P(2,20)) \\
P(3,7) = -0.125A_2B_2 - 0.5(P(3,14) + P(3,15) + P(3,20)) \\
P(1,8) = -0.125B_2C_2 - 0.5(P(1,15) + P(1,16) + P(1,20)) \\
P(2,8) = -0.125A_3C_2 - 0.5(P(2,15) + P(2,16) + P(2,20)) \\
P(3,8) = -0.125A_3B_2 - 0.5(P(3,15) + P(3,16) + P(3,20))
\]

C -- SET UP SHP(3,60) --
DO 60 I=1,3
   DO 50 J=1,20
      K=I+(J-1)*3
      SHP(I,K)=AA(J)
50 CONTINUE
60 CONTINUE
C --
DO 80 I=1,3
   SUM=0.0
   DO 70 J=1,60
      SUM=SUM+SHP(I,J)*XG(J)
70 XPP(I)=SUM
80 CONTINUE
DO 90 I=1,3
   XPP(I)=XPP(I)-XP(I)
90 CONTINUE
C -- CALCULATE JACOBIAN AND ITS INVERSE --
CALL JACOB(P,XX,XJINV,DET,N,20)
C -- GET TRANSPOSE --
DO 110 I=1,3
   DO 100 J=1,3
      XJTINV(I,J)=XJINV(J,I)
100 CONTINUE
110 CONTINUE
C PERFORM XJTINV*XPP
DO 130 I=1,3
SUM=0.0
DO 120 J=1,3
  SUM=SUM+XJTINV(I,J)*XPP(J)
  DIFF(I)=SUM
120  CONTINUE
C --  CALCULATE XNEW
DO 140 I=1,3
  XNEW(I)=XOLD(I)-DIFF(I)
140  CONTINUE
C --  CHECK FOR CONVERGENCE --
SUM=0.0
DO 150 I=1,3
  SUM=SUM+DIFF(I)*DIFF(I)
  SUM=DSQRT(SUM)
  IF (SUM.LT.TOL) GO TO 1500
150  CONTINUE
XOLD(I)=XNEW(I)
R=XOLD(1)
S=XOLD(2)
T=XOLD(3)
1000 CONTINUE
WRITE(6,*)'LOCAL COORDINATES CANNOT BE FOUND FOR :
WRITE(6,*)' POINT IN ELEMENT : ',N
STOP
1500  R=XNEW(1)
  S=XNEW(2)
  T=XNEW(3)
C
RETURN
END
SUBROUTINE NEWLDS(TF)

CALLED BY: MAIN
CALLS : NONE

NEW LOADINGS ARE READ BY THIS SUBROUTINE FROM UNIT 9.

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/MDATA/COORD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBD(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)

WRITE(6,1000) TF
1000 FORMAT(///,T6,'NEW LOADING APPLIED AT ','F9.3,' DAYS',///)
READ(9,*) NELDL
WRITE(6,1010) NELDL
1010 FORMAT(T6,'NUMBER OF ELEMENT DISTRIBUTED LOADS = ','I3,/) 
IF (NELDL.EQ.0) GO TO 100
WRITE(6,1020)
1020 FORMAT(T6,'ELEMENT',T26,'FACE NO.',T36,'LOAD',/) 
DO 10 1=1,NELDL
   READ(9,*) (ELDL(I,J),J=1,3)
   K=ELDL(I,1)
   L=ELDL(I,2)
   WRITE(6,1030) K,L,ELDL(I,3)
1030 FORMAT(10,I3,28,I3,37,D15.8)
10 CONTINUE

100 READ(9,*) NSBF
WRITE(6,1040) NSBF
1040 FORMAT(///,T6,'NUMBER OF SPECIFIED NODAL FORCES = ','I3,/) 
IF (NSBF.EQ.0) GO TO 200
WRITE(6,1050)
1050 FORMAT(5X,'NODE',2X,'X',5X,'Y',5X,'Z',/) 
DO 110 I=1,NSBF
   READ(9,*) (VSBF(I,J),J=1,4)
   L=VSBF(I,1)
   WRITE(6,1060) L,VSBF(I,2),VSBF(I,3),VSBF(I,4)
1060 FORMAT(5X,14,I4,1X,3(2X,D15.8))
110 CONTINUE

200 WRITE(6,1070)
1070 FORMAT(///,T6,'NEW MATERIAL PROPERTIES : ','///)
WRITE(6,1080) NUMMAT
1080 FORMAT(T6,'NUMBER OF MATERIALS = ','I3) 
WRITE(6,1090)
1090 FORMAT(5X,'MAT. NO.',5X,'E',5X,'PR',5X,'WT.-X',5X,'WT.-Y',
*5X,'WT.-Z',/) 
DO 210 I=1,NUMMAT
   READ(9,*) KCODE
210 CONTINUE
C---- IF KCODE = 0, DO NOT READ NEW MODULUS OF ELASTICITY
   IF (KCODE.EQ.0) THEN
      READ(9,*) RMAT(1,3),RMAT(1,4),RMAT(1,5)
   GO TO 205
   END IF
   READ(9,*) (RMAT(I,J),J=1,5)
205 WRITE(6,1110) I,(RMAT(I,J),J=1,5)
1110 FORMAT(5X,I5,5(2X,D10.4))
210 CONTINUE
C
RETURN
END
@PROCESS DC(MDATA,PRESTR,GFV,STDISP)
C
C============= NEW POS =====-========================================
C
SUBROUTINE NEWPOS(ICODE)
C
CALLED BY: MAIN
CALLS : ELCD
C
SUBROUTINE TO CALCULATE THE NEW POSITION OF STRAND NODES
C
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION SHP(3,60),AA(20),ELD(60),DD(3),CC(20,3),D(100000)
COMMON/GFV/GF(100000)
COMMON/ SIZE/NUMNP,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/STRAND/NOTEN,NOTSG,ISTRAN
COMMON/PRESTR/NCTEN(500,5),PCOOD(2000,3),FS(500,2),
*CFS(500,2),FI(50),AR(50),SEG(500,3,3)
COMMON/MDATA/COOD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBD(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/STDISP/DS(100000)
C
DO 10 I=1,NEQ
10 D(I)=GF(I)
   IF (ICODE.EQ.3) THEN
      DO 20 I=1,NEQ
20     D(I)=D(I)+DS(I)
   END IF
C
IP=0
DO 1000 IS=1,NOTSG
   N=NCTEN(IS,4)
   CALL ELCD(N,CC,ELD,D)
C -- INITIALIZE SHAPE FUNCTION ARRAY --
   DO 40 I=1,3
      DO 30 J=1,60
           SHP(I,J)=0.0
30    CONTINUE
40 CONTINUE
C -- FIND NEW COORDINATES FOR EACH POINT ON THE SEGMENT --
   DO 500 IN=1,3
500   ID=NCTEN(IS,IN)
IF (ID.EQ.IP) GO TO 500
R=SEGC(IS,IN,1)
S=SEGC(IS,IN,2)
T=SEGC(IS,IN,3)

C -- EVALUATE ELEMENT SHAPE FUNCTION MATRIX --
A1=1.-R*R
A2=1.-R
A3=1.+R
B1=1.-S*S
B2=1.-S
B3=1.+S
C1=1.-T*T
C2=1.-T
C3=1.+T

C ------ SHAPE FUNCTIONS -------
AA(9)=0.25*A1*B3*C3
AA(10)=0.25*A2*B1*C3
AA(11)=0.25*A1*B2*C3
AA(12)=0.25*A3*B1*C3
AA(13)=0.25*A1*B3*C2
AA(14)=0.25*A2*B1*C2
AA(15)=0.25*A1*B2*C2
AA(16)=0.25*A3*B1*C2
AA(17)=0.25*A3*B3*C1
AA(18)=0.25*A2*B3*C1
AA(19)=0.25*A1*B2*C1
AA(20)=0.25*A3*B2*C1
AA(1)=0.125*A3*B3*C3-0.5*(AA(9)+AA(12)+AA(17))
AA(2)=0.125*A2*B3*C3-0.5*(AA(9)+AA(10)+AA(18))
AA(3)=0.125*A2*B2*C3-0.5*(AA(10)+AA(11)+AA(19))
AA(4)=0.125*A3*B2*C3-0.5*(AA(11)+AA(12)+AA(20))
AA(5)=0.125*A3*B3*C2-0.5*(AA(13)+AA(16)+AA(17))
AA(6)=0.125*A2*B3*C2-0.5*(AA(13)+AA(14)+AA(18))
AA(7)=0.125*A2*B2*C2-0.5*(AA(14)+AA(15)+AA(19))
AA(8)=0.125*A3*B2*C2-0.5*(AA(15)+AA(16)+AA(20))

DO 60 I=1,3
   DO 50 J=1,20
      K=I+(J-1)*3
      SHP(I,K)=AA(J)
   50 CONTINUE
60 CONTINUE

C -- FIND DISPLACEMENTS AT TENDON SEGMENT NODES --
DO 80 I=1,3
   SUM=0.0
   DO 70 J=1,60
      SUM=SUM+SHP(I,J)*ELD(J)
   70 DD(I)=SUM
80 CONTINUE

C
   DO 100 I=1,3
   PCOOD(ID,I)=PCOOD(ID,I)+DD(I)
   IP=ID
100 CONTINUE

C
500 CONTINUE
1000 CONTINUE
C
RETURN
END
C
C================================ OUTNOD ==================================
C
SUBROUTINE OUTNOD
C
CALLED BY: MAIN
CALLS : NONE
C
SUBROUTINE TO IDENTIFY NODES AT WHICH DISPLACEMENT AND
STRESS OUTPUT IS DESIRED. INPUT TO THIS SUBROUTINE IS
READ FROM UNIT 5.
C
COMMON/NOUT/NSETS,NUM(50,2)
C
READ (5,*) NSETS
IF (NSETS.EQ.0) GO TO 1000
DO 100 I=1,NSETS
   READ(5,*) ISN,LSN
   NUM(I,1)=ISN
   NUM(I,2)=LSN
100 CONTINUE
C
1000 RETURN
END
SUBROUTINE PRESKY(MSIZE)

CALLED BY: MAIN
CALLS : NONE

SUBROUTINE TO IDENTIFY ADDRESSES OF DIAGONAL STIFFNESS TERMS
FOR USE IN SUBROUTINE SKYLIN.

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/ADDRESS/JDIAG(40000)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/MDATA/COOD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBDF(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)

NEQ=NUMNP*3
DO 10 I=1,NEQ
10  JDIAG(I)=0

NPE=20
DO 200 N=1,NUMEL
   DO 190 IA=1,NPE
      IA2=IA+2
      NN1=NELC(N,IA2)
      DO 180 I=1,3
         L=3*(NN1-1)+I
         DO 170 JB=1,NPE
            JB2=JB+2
            NN2=NELC(N,JB2)
            DO 160 J=1,3
               M=3*(NN2-1)+J
               KIJ=L-M+1
               IF (KIJ.LT.1) GO TO 160
               JDIAG(L)=MAX0(JDIAG(L),KIJ)
            160         CONTINUE
   170    CONTINUE
180    CONTINUE
190    CONTINUE
200    CONTINUE

DO 300 I=2,NEQ
   JDIAG(I)=JDIAG(I-1)+JDIAG(I)
300   CONTINUE

WRITE(6,600) JDIAG(NEQ)
600 FORMAT(///,5X,'SIZE OF STIFFNESS MATRIX = ',I8,///)
   IF (JDIAG(NEQ).GT.MSIZE) THEN
      WRITE(6,*') 'DIMENSIONS TOO SMALL !!!'
      STOP
   END IF
SUBROUTINE PRTDIS(TF)

CALL BY: MAIN

CALLS : NONE

NODAL DISPLACEMENTS ARE OUTPUT BY THIS SUBROUTINE IN UNIT 6.

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CUMDIS/DIC100000)
COMMON/size/NUMNP, NUMEL, NUMMAT, NSDF, NSBF, NHBW, NEQ, NELDL
COMMON/NOUT/NSETS, NUM(50,2)

WRITE(6,3000) TF
3000 FORMAT(///,T6,'RESULTS AFTER ',F9.3,T32,'DAYS',/, 
* T6,'----------------------------------------',///)

WRITE(6,*) ' '*WRITE(6,*) ' 'WRITE(6,*)' GLOBAL DISPLACEMENTS (CUMULATIVE) :
WRITE(6,*) ' 'WRITE(6,*)' NODE X Y Z'
WRITE(6,*) ' '

IF (NSETS.EQ.0) THEN
  DO 50 I=1,NUMNP
    WRITE(6,501) I,DI(J-2),DI(J-1),DI(J)
50    CONTINUE
  GO TO 1000
END IF

DO 200 II=1,NSETS
  ISN=NUM(II,1)
  LSN=NUM(II,2)
  DO 100 I=ISN,LSN
    J=I*3
    WRITE(6,501) I,DI(J-2),DI(J-1),DI(J)
100  CONTINUE

200  CONTINUE

501 FORMAT(5X,I5,2X,3(D15.8,2X))

C

1000 RETURN
END

SUBROUTINE PRTLOS(TI)

CALL BY: MAIN

CALLS : NONE
SUBROUTINE TO OUTPUT P/S LOSSES IN UNIT 11

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/PRESTR/NCTEN(500,5),PCOOD(2000,3),FS(500,2),
*CF5(500,2),FI(50),AR(50),SEGC(500,3,3)
COMMON/STRAND/NOTEN,NOTSG,ISTRAN

IO=11
WRITE(IO,1010) TI
WRITE(IO,1020)

DO 100 I=1,NOTSG
   SIG=0.0
   DO 50 J=1,2
      SIG=SIG+0.5*FS(I,J)
   50 CONTINUE
   K=NCTEN(I,5)
   PLOSS=100.0*(FI(K)-SIG)/FI(K)
   WRITE(IO,1030) K,I,SIG,PLOSS
100 CONTINUE

1010 FORMAT(//,5X,'AVERAGE PRESTRESS LOSS AFTER ',F9.3,' DAYS')
1020 FORMAT(/,T8,'TENDON NO.',T23,'SEGMENT NO.',
      *T37,'AVERAGE',T55,'PERCENTAGE LOSS',
      */T37,'CURRENT STRESS',T55,'FROM INITIAL STRESS',/)
1030 FORMAT(T11,I3,T25,I3,T35,F15.4,T58,F6.3)

RETURN
END

C
C============= PRTSTR ==============================================
C
SUBROUTINE PRTSTR(TI)
C
C  CALLED  BY:  MAIN
C  CALLS :  NONE
C
C STRESSES AT NODES ARE OUTPUT BY THIS SUBROUTINE IN UNIT 10.
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RNSTR(20,6),GNSTR(25000,6)
DIMENSION ICOUNT(25000)
COMMON/STREXT/EX(8,15)
COMMON/SIGMA/GPCUM(25000,6,15),DELGP(2500,6,15)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/MDATA/COOD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBDF(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/NOUT/NSETS,NUM(50,2)

IO=10
DO 15 I=1,NUMNP
   DO 10 J=1,6
GNSTR(I,J)=0.0
10 CONTINUE
ICOUNT(I)=0
15 CONTINUE
C
DO 100 N=1,NUMEL
KM=NELC(N,2)
E=RMAT(KM,1)
IF (E.LT.10.0) GO TO 100
DO 40 I=1,8
DO 30 J=1,6
SUM=0.0
DO 20 K=1,15
SUM=SUM+EX(I,K)*GPCUM(N,J,K)
RNSTR(I,J)=SUM
20 CONTINUE
30 CONTINUE
DO 60 J=1,6
DO 45 I=9,20
IF (I.GT.16) THEN
I1=I-16
I2=I1+4
GO TO 45
END IF
I1=I-8
I2=I1+1
IF (I1.EQ.4) I2=1
IF (I1.EQ.8) I2=5
45 RNSTR(I,J)=0.5*(RNSTR(I1,J)+RNSTR(I2,J))
60 CONTINUE
DO 80 J=3,22
J1=J-2
L=NELC(N,J)
DO 70 K=1,6
GNSTR(L,K)=GNSTR(L,K)+RNSTR(J1,K)
70 CONTINUE
ICOUNT(L)=ICOUNT(L)+1
80 CONTINUE
100 CONTINUE
C
DO 120 I=1,NUMNP
IF (ICOUNT(I).EQ.0) ICOUNT(I)=1
RC=REAL(ICOUNT(I))
DO 110 K=1,6
GNSTR(I,K)=GNSTR(I,K)/RC
110 CONTINUE
120 CONTINUE
WRITE(IO,3800) TI
WRITE(IO,3900)
IF (NSETS.EQ.0) THEN
DO 125 I=1,NUMNP
125 WRITE(IO,4000) I,(GNSTR(I,K),K=1,6)
GO TO 5000
END IF
DO 200 II=1,NSETS
    ISN=NUM(II,1)
    LSN=NUM(II,2)
    DO 150 I=ISN,LSN
 150    WRITE(10,4000) I,(GNSTR(I,K),K=1,6)
 200   CONTINUE
C
3800 FORMAT(//,T2,'NODAL STRESSES AT ',F9.3,' DAYS')
3900 FORMAT(/,T3,'NODE',T12,'SIGX',T29,'SIGY',T46,'SIGZ',
     *T63,'SIGXY',T80,'SIGYZ',T97,'SIGXZ',/)
4000 FORMAT(1X,15,3X,6(D15.8,2X))
C
5000 RETURN
END

PROCESS DC(MDATA,PRESTR,GFV,STIFF,ADDR)
C
C------------------------ PSLOAD --------------------------
C
SUBROUTINE PSLOAD(ICODE)
C
C CALLED BY: MAIN
C CALLS : DCS, STQB
C
SUBROUTINE TO CALCULATE THE LOAD VECTOR DUE TO PRESTRESS
LOADS ARE APPLIED AS GLOBAL NODAL FORCES
C
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION BS(60),B(6,60),GP(2),GPS(2,3),DIR(6)
DIMENSION ELF(60),CC(20,3),BSE(60),SP(60,60)
COMMON/GFV/GF(100000)
COMMON/STRAIN/NOTEN,NOTSG,ISTRAN
COMMON/STIFF/GSTIF(2000000)
COMMON/ADDRS/JDIAG(40000)
DATA GP/-0.577350269189626D0,0.577350269189626D0/
C
ES=28.5D6
DO 2000 ISG=1,NOTSG
    N=NCTEN(ISG,4)
    MAT=NELC(N,2)
    EC=RMAT(MAT,1)
    EST=ES-EC
C -- FIND (R,S,T) COORDINATES OF GAUSS POINTS --
    DO 4 I=1,60
        ELF(I)=0.0
        DO 6 J=1,60
            DO 5 I=1,60
                SP(I,J)=0.0
            5 CONTINUE
        6 CONTINUE
2000 CONTINUE
DO 10 I1=1,2
  I2=I1+1
  A1=SEGC(ISG,I2,1)-SEGC(ISG,I1,1)
  B1=SEGC(ISG,I2,2)-SEGC(ISG,I1,2)
  C1=SEGC(ISG,I2,3)-SEGC(ISG,I1,3)
  CON=GP(2)
  IF (I1.EQ.1) CON=1.0-GP(2)
  GPS(I1,1)=SEGC(ISG,I1,1)+CON*A1
  GPS(I1,2)=SEGC(ISG,I1,2)+CON*B1
  GPS(I1,3)=SEGC(ISG,I1,3)+CON*C1
10  CONTINUE
C -- SET UP COORDINATES OF ELEMENT NODES --
DO 40  J=3,22
  L=NELC(N,J)
  DO 30  K=1,3
    J1=J-2
    K1=K+1
    CC(J1,K)=COORD(L,K1)
30  CONTINUE
DO 40  CONTINUE
C -- PERFORM INTEGRATION --
DO 500  IG=1,2
  R=GPS(IG,1)
  S=GPS(IG,2)
  T=GPS(IG,3)
C -- SET UP B-MATRIX OF PARENT ELEMENT --
  CALL STQB(CC,B,DET,R,S,T,N)
C -- CALCULATE [BS] AT THE GAUSS POINT --
  CALL DCS(DIR,RJST,GP,ISG,IG,N)
DO 50  I=1,60
  BS(I)=0.0
50  DO 70  I=1,60
    SUM=0.0
    DO 60  J=1,6
      SUM=SUM+DIR(J)*B(J,I)
      BS(I)=SUM
60  CONTINUE
DO 70  I=1,60
70  BSE(I)=BS(I)*EST
  CONST=AR(NCTEN(ISG,5))*RJST
DO 78  I=1,60
    DO 75  J=1,60
      SP(I,J)=BSE(I)*BS(J)*CONST+SP(I,J)
    CONTINUE
78  DO 72  I=1,60
72  SIGMA=CFS(ISG,IG)
  IF (ICODE.EQ.0) SIGMA=FS(ISG,IG)
  CONST=SIGMA*CONST
DO 80  I=1,60
  ELF(I)=ELF(I)-CONST*BS(I)
80  CONTINUE
C -- SET UP GLOBAL LOAD VECTOR AND STIFFNESS MATRIX CONTRIBUTIONS
DO 1000  I=1,20
  I2=I+2
  NR=(NELC(N,I2)-1)*3
DO 990 J=1,3
NR=NR+1
L=(I-1)*3+J
GF(NR)=GF(NR)+ELF(L)
DO 980 K=1,20
K2=K+2
NNO=NELC(N,K2)
DO 970 KK=1,3
M=(NNO-1)*3+KK
JBJ=3*(K-1)+KK
KIJ=M-NR
IF(KIJ.LT.0) GO TO 970
ND=JDIAG(M)-KIJ
GSTIF(ND)=GSTIF(ND)+SP(L,JBJ)

970 CONTINUE
980 CONTINUE
990 CONTINUE
1000 CONTINUE
2000 CONTINUE

C
RETURN
END

C
PROCESS DC(PRESTR,MDATA,STIFF,PSEPS,GFV)
C
C
SUBROUTINE PSTRES(ICOD)

C
CALLED BY: MAIN
C CALLS : NONE
C
SUBROUTINE TO EVALUATE THE STRESS AT STRAND INTEGRATION POINTS
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION EP(12),TP(10),FMAX(500,2)
COMMON/PRESTR/NCTEN(500,5),PCOOD(2000,3),FS(500,2),*
CFS(500,2),FI(50),AR(50),SEGC(500,3,3)
COMMON/STRAND/NOTEN,NOTSG,ISTRAN
COMMON/PSEPS/TEPS(500,2),DELEPS(500,2)

C
EINIT=28.5D6
IF (ICOD.EQ.0) THEN
DO 20 I=1,NOTSG
FINIT=FS(I,1)
DO 10 J=1,11,2
IF (FINIT.GT.EP(J)) JK=J

10 CONTINUE
DIFF=FINIT-EP(JK)
JK1=JK+1
JK2=JK+2
JK3=JK+3

C
TEPS(I,1) = EP(JK1) + DIFF * SLOPE
CFS(I,1) = 0.0
FMAX(I,1) = FS(I,1)
TEPS(I,2) = TEPS(I,1)
CFS(I,2) = 0.0
FMAX(I,2) = FMAX(I,1)

20 CONTINUE
GO TO 200
END IF

C
DO 100 I = 1, NOTSG
DO 90 K = 1, 2
DELSIG = EINIT * DELEPS(I,K)
FNEW = FS(I,K) + DELSIG
IF (FNEW .GT. FMAX(I,K)) THEN
DO 60 J = 1, 11, 2
   IF (FNEW .GT. EP(J)) JK = J
60 CONTINUE
DIFF = FNEW - EP(JK)
JK1 = JK + 1
JK2 = JK + 2
JK3 = JK + 3
TEPS(I,K) = EP(JK1) + DIFF * SLOPE
GO TO 80
END IF
TEPS(I,K) = TEPS(I,K) + DELEPS(I,K)

80 CFS(I,K) = DELSIG
FS(I,K) = FNEW
CONTINUE
CONTINUE
CONTINUE
CONTINUE
CONTINUE
RETURN
END
SUBROUTINE REDATA(IANCOD)

CALLED BY: MAIN
CALLS : NONE

SUBROUTINE TO READ INPUT DATA FROM UNIT 9

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION KC(2500)
CHARACTER*1 ABC
COMMON/SIZE/NUMP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NELQ,NEQLD
COMMON/MDATA/COOD(25000,4),NELC(25000,22),RMAT(20,5),IEL(2500),
VBDF(2000,3),VSDF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/PRESTR/NCTEN(500,5),PCOD(2000,3),FS(500,2),
*CFS(500,2),F1(50),AR(50),SEG(500,3,3)
COMMON/STRAND/NOTEN,NOTSG,ISTRAN
COMMON/YIELD/FY
COMMON/MD/EPR(20),FPC28(20),WCONC(20),TL(20),ICUR(20),CURLEN(20)
COMMON/DEPTH/DEP(6),ADTEMP(4,15)
COMMON/SPR/SPR(250,3),NSSF

C -------- READ NODAL POINT COORDINATES -----------
READ(9,*) NUMP
WRITE(8,*)  
WRITE(8,*)  
WRITE(8,*) ' NUMBER OF NODAL POINTS = ',NUMP
WRITE(8,505)
DO 10 I=1,NUMP
   READ(9,*) (COOD(I,J),J=1,4)
   L=COOD(I,1)
   WRITE(8,510) L,COOD(I,2),COOD(I,3),COOD(I,4)
10 CONTINUE

C -------- READ MATERIAL PROPERTIES -------------
READ(9,*) NUMMAT
WRITE(8,*)  
WRITE(8,*)  
WRITE(8,*) ' NUMBER OF MATERIALS = ',NUMMAT
WRITE(8,515)
DO 15 I=1,NUMMAT
   READ(9,*) (RMAT(I,J),J=1,5)
   WRITE(8,520) I,(RMAT(I,J),J=1,5)
   READ(9,*) FPC28(I),WCONC(I),TL(I),ICUR(I),CURLEN(I)
   WRITE(8,521) FPC28(I),WCONC(I),TL(I),ICUR(I),CURLEN(I)
15 CONTINUE

C -------- READ ELEMENT TYPE AND CONNECTIVITY -------
READ(9,*) NUMEL
WRITE(8,*)  
WRITE(8,*)  
WRITE(8,*) ' NUMBER OF ELEMENTS = ',NUMEL
WRITE(8,525)
DO 20 I=1,NUMEL
READ(9,*) IEL(I),ILEV(I)
IF (IEL(I).EQ.3) THEN
   READ(9,*) (NELC(I,J),J=1,22)
   WRITE(8,530) NELC(I,1),NELC(I,2),ILEV(I)
   WRITE(8,580) (NELC(I,J),J=3,22)
   IF (IANCOD.EQ.0) THEN
      KC(I)=0
      DO 17 K=4,22
         DO 16 J=3,K-1
            IF (NELC(I,K).EQ.NELC(I,J)) THEN
               KC(I)=1
               GO TO 20
            END IF
         END DO 16
      END DO 17
   END IF
   CONTINUE
   CONTINUE
   END IF
   GO TO 20
   END IF
   READ(9,*) (NELC(I,J),J=1,10)
   WRITE(8,530) NELC(I,1),NELC(I,2)
   WRITE(8,540) (NELC(I,J),J=3,10)
CONTINUE
READ (9,* ) NOLEV
WRITE(8,* ) ' '
WRITE(8,* ) 'NO. OF LEVELS = ' ,NOLEV
WRITE(8,* ) ' '
WRITE(8,541)
DO 22 I=1,NOLEV
   READ(9,* ) DEP(I)
   WRITE(8,542) I,DEP(I)
CONTINUE
C ----- READ DISTRIBUTED ELEMENT LOAD DATA -------
READ(9,* ) NELDL
WRITE(8,* ) ' '
WRITE(8,* ) ' '
WRITE(8,* ) ' NUMBER OF ELEMENT DISTRIBUTED LOADS = ' ,NELDL
IF (NELDL.EQ.0) GO TO 27
WRITE(8,* ) ' '
WRITE(8,* ) ' ELEMENT FACE NO. LOAD'
WRITE(8,* ) ' '
DO 25 I=1,NELDL
   READ(9,* ) (ELDL(I,J),J=1,3)
   K=ELDL(I,1)
   L=ELDL(I,2)
   WRITE(8,570) K,L,ELDL(I,3)
CONTINUE
C ----- READ DATA FOR SPECIFIED NODAL D.O.F. -------
READ(9,* ) NSDF
WRITE(8,* ) ' '
WRITE(8,* ) ' '
WRITE(8,* ) ' NUMBER OF SPECIFIED D.O.F. = ' ,NSDF
WRITE(8,* ) ' '
IF (NSDF.EQ.0) GO TO 32
WRITE(8,545)
DO 30 I=1,NSDF
    READ(9,*) (VBDF(I,J),J=1,3)
    K=VBDF(I,1)
    L=VBDF(I,2)
    WRITE(8,550) K,L,VBDF(I,3)
30    CONTINUE
C ------ READ NODAL FORCE DATA ------
32    READ(9,*) NSBF
    WRITE(8,*) ' '  
    WRITE(8,*) ' '  
    WRITE(3,*) ' NUMBER OF SPECIFIED NODAL FORCES = ',NSBF  
    WRITE(8,*) ' '  
    IF (NSBF.EQ.0) GO TO 36
    WRITE(8,555)
    DO 35 I=1,NSBF
      READ(9,*) (VSBF(I,J),J=1,4)
      L=VSBF(I,1)
      WRITE(8,560) L,VSBF(I,2),VSBF(I,3),VSBF(I,4)
35    CONTINUE
C ------ READ SUPPORT SPRING DATA ------
36    READ(9,*) NSSP
    WRITE(8,655) NSSP
    IF (NSSP.EQ.0) GO TO 39
    DO 37 I=1,NSSP
5    READ(9,*) (SPR(I,J),J=1,3)
      WRITE(8,660)
    DO 38 I=1,NSSP
      J=SPR(I,1)
      WRITE(6,665) J,SPR(I,2),SPR(I,3)
38    CONTINUE
C ------ READ PRESTRESS DATA ------
39    READ(9,*) NOTEN,NOTSG
    WRITE(8,585) NOTEN,NOTSG
    ISTRAN=1
    IF (NOTEN.EQ.0) THEN
      ISTRAN=0
      GO TO 500
    END IF
    WRITE(8,590)
    DO 40 I=1,NOTSG
      READ(9,*) (NCTE(I,J),J=1,5)
5    WRITE(8,600) (NCTE(I,J),J=1,5)
40    CONTINUE
    WRITE(8,610)
    NN=2*NOTSG+NOTEN
    DO 45 I=1,NN
      READ(9,*) (PCOOD(I,J),J=1,3)
    WRITE(8,620) I,(PCOOD(I,J),J=1,3)
45    CONTINUE
    WRITE(8,630)
    DO 50 I=1,NOTEN
      READ(9,*) FI(I),AR(I)
    WRITE(8,640) I,FI(I),AR(I)
50    CONTINUE
C READ(9,*) FY
WRITE(8,645) FY

C 500 IF (IANGCOD.EQ.0) THEN
  LC=0
  DO 60 I=1,NUMEL
  IF (KC(I).EQ.1) THEN
    WRITE(8,650) I
    LC=1
  END IF
  CONTINUE
  IF (LC.EQ.0) WRITE(8,*) 'NO PROBLEM WITH ELEMENT CONNECTIVITY!'
  STOP
END IF
C
RETURN
C ------------ FORMAT STATEMENTS ------------
505 FORMAT(/,5X,'NODE',3X,'X-COOD.',5X,'Y-COOD.',5X,'Z-COOD.',/)
510 FORMAT(T5,I5,T11,3(D10.4,1X))
515 FORMAT(/,5X,'MAT. NO.',5X,'E',5X,'PR',5X,'WT.-X',5X,'WT.-Y',
      *5X,'WT.-Z',/)
520 FORMAT(5X,I5,5(2X,D15.8))
521 FORMAT(/,5X,'STRENGTH AT 28 DAYS = ',D15.8,/,  
      *5X,'UNIT WEIGHT OF CONCRETE = ',D15.8,/,  
      *5X,'FIRST LOADING AT ',F9.3,' DAYS',/,  
      *5X,'CURING CODE = ',I1,/,  
      *5X,'LENGTH OF CURE = ',F9.3,' DAYS',/)  
525 FORMAT(/,5X,'ELEMENT NO.',5X,'ELEMENT MAT. NO.',/,  
      *5X,'ELEMENT CONNECTIVITY',/)
530 FORMAT(5X,I4,10X,I3,10X,I3)
540 FORMAT(5X,5(I5,2X))
541 FORMAT(T10,'LEVEL',T25,'DEPTH',/)
542 FORMAT(T14,I3,T23,F10.6,/)  
545 FORMAT(5X,'NODE',5X,'D.O.F.',5X,'DISP.',/)
550 FORMAT(5X,2(I5,2X),D10.4)
555 FORMAT(5X,'NODE',2X,'X',5X,'Y',5X,'Z',/)
560 FORMAT(5X,I5,1X,3(2X,D15.8))
570 FORMAT(6X,I4,5X,I4,4X,D15.8)
580 FORMAT(5X,10(I5,2X),/5X,10(I5,2X))
590 FORMAT(/,T6,'NO. OF TENDONS = ',T35,I3,/,  
      *T6,'NO. OF TENDON SEGMENTS = ',T35,I3)
600 FORMAT(/,T6,'TENDON SEGMENT POSITION :',/,,  
      *T5,'PT. 1',T15,'PT. 2',T25,'ELEM. 1',T35,'ELEM. 2',
      *T45,'TENDON NO.',/)  
610 FORMAT(/,T6,'COORDINATES OF SEGMENT END POINTS :',/,,  
      *T6,'POINT',T17,'X',T32,'Y',T47,'Z',/)  
615 FORMAT(T7,I4,T17,D15.8,T32,D15.8,T47,D15.8)
620 FORMAT(/,T6,'TENDON INITIAL STRESS :',/,,  
      *T5,'NO.',T15,'STRESS',T30,'AREA',/)  
625 FORMAT(T6,I2,T10,D15.8,T28,D15.8)
630 FORMAT(/,T6,'YIELD STRESS FOR TENDONS = ',D15.8,/)  
635 FORMAT(5X,'REPEATED NODE NUMBERS IN ELEMENT',2X,I5)
640 FORMAT(/,T6,'NO. OF SUPPORT SPRINGS = ',T35,I3,/)  
645 FORMAT(T7,'NODE',T15,'ANGLE',T30,'STIFFNESS',/)
SUBROUTINE RELAX(T1,T2,TREL)

CALLED BY: MAIN
CALLS : NONE

SUBROUTINE TO CALCULATE RELAXATION LOSSES IN P/S STRAND SEGMENTS

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/STRAND/NOTEN, NOTSG,ISTRAN
COMMON/YIELD/FY
COMMON/PRESTR/NCTEN(500,5),PCOOD(2000,3),FS(500,2),*
*CFS(500,2),FI(50),AR(50),SEG(500,3,3)

10=11
TR=TREL*24.0
TIN=T1*24.0
TFI=T2*24.0
Q=DLOG10(TFI)
P=DLOG10(TIN)
IF (TIN.EQ.TR) THEN
  R=P
  GO TO 8
END IF
R=DLOG10(TIN-TR)
8 DO 100 1=1,NOTSG
  K=NCTEN(1,5)
  REL=0.0
  SIG=0.0
  DO 50  J=1,2
    FCUM=FS(I,J)
    RK=R/10.0
    B=1.+0.55*RK
    C=4.*RK*FCUM/FY
    A=2.*RK/FY
    FINIT=(B-DSQRT(B*B-C))/A
    D=FINIT-0.55*FY
    IF (D.LT.0.0) D=0.0
    DELREL=FINIT*D*(P-Q)/(10.0*FY)
    FS(I,J)=FS(I,J)+DELREL
    CFS(I,J)=DELREL+CFS(I,J)
    REL=REL+DELREL*0.5
    SIG=SIG+FS(I,J)*0.5
 50  CONTINUE
100 CONTINUE
C
RETURN
END

@PROCESS DC(MDATA,PRESTR,STRAIN)
SUBROUTINE RELESE(TI,IA)

CALLED BY: MAIN
CALLS : ACISH, BP2SH, CEBSH

THIS SUBROUTINE CALCULATES PRESTRESSING FORCES AT RELEASE AND SHRINKAGE STRAINS PRIOR TO RELEASE.

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION DELREL(50)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/STRAND/NOTEN,NOTSG,ISTRAN
COMMON/YIELD/FY
COMMON/PRESTR/NCTEN(500,5),PCOOD(2000,3),FS(500,2),
*FS(500,2),FI(50),AR(50),SEGC(500,3)
COMMON/MOD/EPR(20),FPC28(20),WCONC(20),TL(20),ICUR(20),CURLEN(20)
COMMON/STRAIN/CSCUM(2500,6,15),DELCS(2500,6,15),HSV(2500,4,6,15)
COMMON/M DATA/CO0D(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBDF(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)

IO=11
IF (TI.EQ.0.0) GO TO 35
WRITE(IO,1010) TI
WRITE(IO,1020)
TIN=TI*24.
P=DLOG10(TIN)
DO 20 I=1,NOTEN
   A=FI(I)/FY-0.55
   B=P*A*0.1
   DELREL(I)=-FI(I)*B
   FN=FI(I)+DELREL(I)
   WRITE(IO,1030) I,DELREL(I),FN
20 CONTINUE
DO 30 I=1,NOTSG
   K=NCTEN(I,5)
   DO 25 J=1,2
      FS(I,J)=FI(K)+DELREL(K)
25 CONTINUE
30 CONTINUE

C ------- INITIAL SHRINKAGE -------
35 K=0
DO 400 N=1,NUMEL
   K1=K
   K=NELC(N,2)
   IF (TL(K).NE.TI) GO TO 400
   IF (K1.EQ.K) GO TO 50
   IF (IA.EQ.1) THEN
      T2=TI
      T1=0.D0
      CALL ACISH(T1,T2,K,EPSH)
   END IF
400 CONTINUE
IF (IA.EQ.2) THEN
    T1=0.0
    T2=TI
    CALL CEBSH(T1,T2,K,EPSH)
    WRITE(IO,*)'EPSHH = ',EPSH
END IF
IF (IA.EQ.3) THEN
    T2=TI
    T1=0.0
    CALL BP2SH(T1,T2,K,EPSH)
END IF
WRITE(6,1040) K,EPSH
50 DO 300 I=1,3
    DO 200 IP=1,15
        DELCS(N,I,IP)=EPSH
        CSCUM(N,I,IP)=EPSH
200 CONTINUE
300 CONTINUE
400 CONTINUE
C
C OUTPUT SHRINKAGE CURVE FOR EACH MATERIAL
C
DO 500 K=1,NUMMAT
    T0=TL(K)-CURLEN(K)
    WRITE(8,1050) K
    DO 450 I=1,30
        TN=I*25.0
        TNN=TN+TL(K)-CURLEN(K)
        IF (IA.EQ.1) CALL ACISH(T0,TNN,K,SHRINK)
        IF (IA.EQ.2) CALL CEBSH(T0,TNN,K,SHRINK)
        IF (IA.EQ.3) CALL BP2SH(T0,TNN,K,SHRINK)
        WRITE(8,1060) TN,SHRINK
450 CONTINUE
500 CONTINUE
C
1010 FORMAT(//,5X,'RELEASE OF PRESTRESS AT ',F9.3,' DAYS')
1020 FORMAT(/,T8,'TENDON NO. ',T23,'RELAXATION LOSS',
     *T44,'STRESS AT RELEASE',/)  
1030 FORMAT(T11,I3,T25,F10.4,T42,F15.4)
1040 FORMAT(//,5X,'INITIAL SHRINKAGE FOR MATERIAL NO. ',I2,
     *5X,'=',E15.8)
1050 FORMAT(//,3X,'SHRINKAGE STRAINS FOR MATERIAL NO. ',I2,/,
     *T5,'TIME',T25,'STRAIN',/)  
1060 FORMAT(T5,F10.5,T20,E15.8)
C
RETURN
END
SUBROUTINE SETEXT

CALLED BY: MAIN
CALLS : NONE

SUBROUTINE TO DEFINE NODAL STRESS EXTRAPOLATION MATRIX

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION EXT(8,15)
COMMON/STREXT/EX(8,15)
DATA EXT/0.666666667D-01,0.666666667D-01,0.666666667D-01,
   $ 0.666666667D-01,0.666666667D-01,0.666666667D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ -0.8281935983D-01,-0.8281935983D-01,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ -0.8281935983D-01,-0.8281935983D-01,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
   $ 0.2161526932D+00,-0.8281935983D-01,-0.8281935983D-01,
   $ -0.8281935983D-01,0.2161526932D+00,0.2161526932D+00,
C DO 20 I=1,8
    DO 10 J=1,15
        EX(I,J)=EXT(I,J)
    10 CONTINUE
    20 CONTINUE

C RETURN
END

C==================================================
C SUBROUTINE SETGPF
C
C CALLED BY: MAIN
C CALLS : NONE
C
C SUBROUTINE TO SET UP THE LOCATION OF QUADRATURE POINTS
C AND THE ASSOCIATED WEIGHTS.
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(4,4),B(4,4),IFC(6,8)
COMMON/FACES/IFACE(6,8)
COMMON/QUADR/XG(4,4),WT(4,4)
COMMON/INTEG/PTS(15,3),WTS(3)
DATA IFC/1,3,2,4,1,6,12,10,9,11,9,13,4,2,1,3,2,5,20,18,17,19,
*10,16,8,6,5,7,3,8,16,14,13,15,11,15,5,7,6,8,4,7,
*17,19,18,20,12,14/
DATA A/0.D0,0.D0,0.D0,0.D0,-0.5773502691896D0,
*0.5773502691896D0,0.D0,0.D0,-0.7745966692415D0,0.D0,
*0.7745966692415D0,0.D0,-0.8611363115941D0,-0.3399810435849D0,
*0.3399810435849D0,0.8611363115941D0/
DATA B/2.0D0,0.D0,0.D0,0.D0,1.0D0,1.0D0,0.D0,0.D0,
*0.555555555555556D0,0.88888888888889D0,0.555555555555556D0,0.D0,
*0.3478548451375D0,0.6521451548625D0,0.6521451548625D0,
*0.3478548451375D0/
DO 20 I=1,4
    DO 10 J=1,4
        XG(I,J)=A(I,J)
        WT(I,J)=B(I,J)
    10 CONTINUE
    20 CONTINUE
DO 40 I=1,6
    DO 30 J=1,8
        IFC(I,J)=IFC(I,J)
    30 CONTINUE
    40 CONTINUE

C DO 60 I=1,7
    DO 50 J=1,3
PTS(I,J)=0.0

50 CONTINUE
60 CONTINUE
C=0.848418011
D=0.727662441
PTS(2,1)=-C
PTS(3,1)=C
PTS(4,2)=-C
PTS(5,2)=C
PTS(6,3)=-C
PTS(7,3)=C
DO 70 I=8,14,2
   J=I+1
   PTS(I,1)=-D
   PTS(J,1)=D
70 CONTINUE
DO 80 I=8,9
   J=I+2
   PTS(I,2)=-D
   PTS(J,2)=D
   PTS(I,3)=-D
   PTS(J,3)=-D
80 CONTINUE
DO 90 I=12,13
   J=I+2
   PTS(I,2)=-D
   PTS(J,2)=D
   PTS(I,3)=D
   PTS(J,3)=D
90 CONTINUE
C
WTS(1)=0.712137436
WTS(2)=0.686227234
WTS(3)=0.396312395
C
RETURN
END

C
C================ SETUP ========================================================
C
SUBROUTINE SETUP(IA)
C
C CALLED BY: MAIN
C CALLS : ACI209, CEBFIP, BAPAN2
C
SUBROUTINE TO IDENTIFY THE CREEP AND SHRINKAGE MODEL IN USE
C
IMPLICIT REAL*8 (A-H,O-Z)
C
IF (IA.EQ.1) CALL ACI209
IF (IA.EQ.2) CALL CEBFIP
IF (IA.EQ.3) CALL BAPAN2
C
RETURN
C
C SUBROUTINE SKYLIN(A,B,JDIAG,NEQ,KK)
C
C CALLED BY: MAIN
C CALLS : DOT
C
C SUBROUTINE TO SOLVE EQUILIBRIUM EQUATIONS.
C SKYLINE STORAGE IS USED FOR THE GLOBAL STIFFNESS MATRIX.
C
C IMPLICIT REAL*8 (A-H,O-Z)
C DIMENSION A(1),B(1),JDIAG(1)
C FACTOR 'A' TO 'UT*D*U' AND REDUCE 'B'
J=0
DO 600 J=1,NEQ
JD=JDIAG(J)
JH=JD-JR
IS=J-JH+2
IF (JH-2) 600,300,100
100 IF (KKK.EQ.2) GO TO 500
IE=J-1
K=JR+2
ID=JDIAG(IS-1)
C REDUCE ALL EQUATIONS EXCEPT DIAGONAL
DO 200 I=IS,IE
IR=ID
ID=JDIAG(I)
IH=MIN0(ID-IR-1,I-IS+1)
IF (IH.GT.0) A(K)=A(K)-DOT(A(K-IH),A(ID-IH),IH)
200 K=K+1
C REDUCE THE DIAGONAL
300 IF (KKK.EQ.2) GO TO 500
IR=JR+1
IE=JD-1
K=J-JD
DO 400 I=IR,IE
ID=JDIAG(K+I)
IF (A(ID).EQ.0.0) GO TO 400
D=A(ID)
A(I)=A(I)/A(ID)
A(JD)=A(JD)-D*A(I)
400 CONTINUE
C REDUCE THE LOAD VECTOR
500 IF (KKK.NE.1) B(J)=B(J)-DOT(A(JR+1),B(IS-1),JH-1)
600 JR=JD
IF (KKK.EQ.1) RETURN
C DIVIDE BY DIAGONAL PIVOTS
DO 700 I=1,NEQ
ID=JDIAG(I)
IF (A(ID).NE.0.0) B(I)=B(I)/A(ID)
700 CONTINUE
C BACK SUBSTITUTION
J=NEQ
JD=JDIAG(J)

DO 800 J=J-1
IF (J.LE.0) RETURN
JR=JDIAG(J)
IF (JD-JR.LE.1) GO TO 1000
IS=J-JD+JR+2
K=JR-IS+1
DO 900 I=IS,J
900  B(I)=B(I)-A(I+K)*D

GO TO 800

C END

C PROCESS DC(SIGMA,MDATA,EPS,STRAIN,STIFF,GFV)

C

C = = = = = = = = = = = STRESS = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

C SUBROUTINE STRESS(MCODE,TIME)

C CALLED BY: MAIN
C CALLS : STQB

C THIS SUBROUTINE EVALUATES ELEMENT STRESSES AT GAUSS POINTS.

C IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION DR(60),CC(20,3),B(6,60),C(6,6),
*EPS(6),DB(6),GPSTR(6,15)
COMMON/SIGMA/GPCUM(2500,6,15),DELGF(2500,6,15)
COMMON/MDATA/COORD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),
*VBDF(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)
COMMON/QUADR/XG(4,4),WT(4,4)
COMMON/INTEGRAL/PTS(15,3),WTS(3)
COMMON/EPS/STRAIN(2500,6,15),DSTRN(2500,6,15)
COMMON/STRAIN/CSCUM(2500,6,15),DELCS(2500,6,15),HSV(2500,4,6,15)
COMMON/STIFF/GSTIF(20000000)
COMMON/GFV/GF(100000)
COMMON/NUMP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/MOD/EPR(20),FPC28(20),WCONC(20),TL(20),IGUR(20),CURLEN(20)

KM=0
DO 1000 N=1,NUMEL
K1M=KM
IF (MCODE.EQ.0) THEN
   DO 2 I=1,6
      DO 1 J=1,15
         GPCUM(N,I,J)=0.0
   1    CONTINUE
   2    CONTINUE
END IF

C DO 20 J=3,22
J1 = J - 2
L = NELC(N, J)
LL = L * 3
LLM2 = LL - 2
LLM1 = LL - 1
JJ = J1 * 3
JJM2 = JJ - 2
JJM1 = JJ - 1
DR(JJM2) = GF(LLM2)
DR(JJM1) = GF(LLM1)
DR(JJ) = GF(LL)
DO 10 K = 1, 3
   K1 = K + 1
   CC(J1, K) = COLD(L, K1)
10   CONTINUE
20   CONTINUE

C
KM = NELC(N, 2)
IF (MCODE.EQ.2).AND.(TIME.EQ.TL(KM)) THEN
   DO 24 I = 1, 6
   DO 23 J = 1, 15
      STRAIN(N, I, J) = 0.0
      CSCUM(N, I, J) = 0.0
23    CONTINUE
24    CONTINUE
END IF
IF (K1M.EQ.KM) GO TO 105
E = RMAT(KM, 1)
PR = RMAT(KM, 2)
DO 90 I = 1, 6
   DO 90 J = 1, 6
90    C(I, J) = 0.0
C
A = (1.0 - 2.0 * PR) / (1.0 - PR)
BC = E / (1.0 + PR)
F = PR / (1.0 - PR)
RT = BC / A
C(1, 1) = RT
C(2, 2) = RT
C(3, 3) = RT
C(1, 2) = F * RT
C(1, 3) = F * RT
C(2, 1) = F * RT
C(2, 3) = F * RT
C(3, 1) = F * RT
C(3, 2) = F * RT
C(4, 4) = 0.5 * A * RT
C(5, 5) = 0.5 * A * RT
C(6, 6) = 0.5 * A * RT
C
105   DO 150 IP = 1, 15
      RI = PTS(IP, 1)
      SI = PTS(IP, 2)
      TI = PTS(IP, 3)
CALL STQB(CC,B,DET,RI,S,T,N)
DO 115 I=1,6
   DB(I)=0.0
   DO 110 J=1,60
      DB(I)=DB(I)+B(I,J)*DR(J)
   110 CONTINUE
   DSTRN(N,I,IP)=DB(I)
   STRAIN(N,I,IP)=DB(I)+STRAIN(N,I,IP)
115 CONTINUE
DO 130 K=1,6
   GPSTR(K,IP)=0.0
   DO 120 L=1,6
      GPSTR(K,IP)=GPSTR(K,IP)+C(K,L)*(STRAIN(N,L,IP)-CSCUM(N,L,IP))
120 CONTINUE
130 CONTINUE
150 CONTINUE
DO 180 I=1,6
   DO 170 J=1,15
      DELGP(N,I,J)=GPSTR(I,J)-GPCUM(N,I,J)
   170 CONTINUE
180 CONTINUE
1000 CONTINUE
RETURN
END

C
C================================= STQB =======================================
C
C SUBROUTINE STQB(XX,B,DET,R,S,T,NEL)
C
C CALLED BY: BRQ, PSLOAD, STRESS, TENEPS
C CALLS : JACOB
C
C THIS SUBROUTINE EVALUATES THE STRAIN-DISPLACEMENT MATRIX (B) AT EACH INTEGRATION POINT.
C
C IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION XX(20,3),B(6,60),P(3,20),XJI(3,3),XJIC(3,3)
COMMON/SHP20/AN(20)
C
A1=1.-R*R
A2=1.-R
A3=1.+R
B1=1.-S*S
B2=1.-S
B3=1.+S
C1=1.-T*T
C2=1.-T
C3=1.+T

C --------- SHAPE FUNCTIONS ---------
AN(9)=0.25*A1*B3*C3
AN(10)=0.25*A2*B1*C3
AN(11)=0.25*A1*B2*C3
AN(12)=0.25*A3*B1*C3
\[ A_N(13) = 0.25A1B3C2 \]
\[ A_N(14) = 0.25A2B1C2 \]
\[ A_N(15) = 0.25A1B2C2 \]
\[ A_N(16) = 0.25A3B1C2 \]
\[ A_N(17) = 0.25A3B3C1 \]
\[ A_N(18) = 0.25A2B3C1 \]
\[ A_N(19) = 0.25A2B2C1 \]
\[ A_N(20) = 0.25A3B2C1 \]
\[ A_N(1) = 0.125A3B3C3 - 0.5(A_N(9) + A_N(12) + A_N(17)) \]
\[ A_N(2) = 0.125A2B3C3 - 0.5(A_N(9) + A_N(10) + A_N(18)) \]
\[ A_N(3) = 0.125A2B2C3 - 0.5(A_N(10) + A_N(11) + A_N(19)) \]
\[ A_N(4) = 0.125A3B2C3 - 0.5(A_N(11) + A_N(12) + A_N(20)) \]
\[ A_N(5) = 0.125A3B3C2 - 0.5(A_N(13) + A_N(16) + A_N(17)) \]
\[ A_N(6) = 0.125A2B3C2 - 0.5(A_N(13) + A_N(14) + A_N(18)) \]
\[ A_N(7) = 0.125A2B2C2 - 0.5(A_N(14) + A_N(15) + A_N(19)) \]
\[ A_N(8) = 0.125A3B2C2 - 0.5(A_N(15) + A_N(16) + A_N(20)) \]

C --------- DERIVATIVES OF SHAPE FUNCTIONS  ---------

\[ P(1,9) = -0.5R*B3*C3 \]
\[ P(2,9) = 0.25A2A3*C3 \]
\[ P(3,9) = 0.25A2A3*B3 \]
\[ P(1,10) = -0.25B2B3*C3 \]
\[ P(2,10) = -0.5A2S*C3 \]
\[ P(3,10) = 0.25A3B3*B2 \]
\[ P(1,11) = 0.5B2*C3*R \]
\[ P(2,11) = -P(2,9) \]
\[ P(3,11) = 0.25A1B2 \]
\[ P(1,12) = -P(1,10) \]
\[ P(2,12) = 0.5*A3*C3*S \]
\[ P(3,12) = 0.25A3*B1 \]
\[ P(1,13) = -0.5*B3*C2*R \]
\[ P(2,13) = -0.5*A1*C2 \]
\[ P(3,13) = -P(3,9) \]
\[ P(1,14) = 0.25B1*C2 \]
\[ P(2,14) = 0.5A2C2*S \]
\[ P(3,14) = -P(3,10) \]
\[ P(1,15) = 0.5B2*C2*R \]
\[ P(2,15) = -0.25A1*C2 \]
\[ P(3,15) = -P(3,11) \]
\[ P(1,16) = 0.25B1*C2 \]
\[ P(2,16) = 0.5A3*C2*S \]
\[ P(3,16) = -P(3,12) \]
\[ P(1,17) = 0.25B3*C2*C3 \]
\[ P(2,17) = 0.25A3C2*C3 \]
\[ P(3,17) = -0.5A3*B3*T \]
\[ P(1,18) = -P(1,17) \]
\[ P(2,18) = 0.25A2*C1 \]
\[ P(3,18) = 0.5A2B3*T \]
\[ P(1,19) = 0.25B2*C1 \]
\[ P(2,19) = -P(2,18) \]
\[ P(3,19) = 0.5A2B2*T \]
\[ P(1,20) = 0.25B2*C1 \]
\[ P(2,20) = -P(2,17) \]
\[ P(3,20) = 0.5A3B2*T \]
P(1,1) = 0.125 * B3 * C3 - 0.5 * (P(1,10) + P(1,12) + P(1,17))
P(2,1) = 0.125 * A3 * C3 - 0.5 * (P(2,10) + P(2,12) + P(2,17))
P(3,1) = 0.125 * A3 * B3 - 0.5 * (P(3,9) + P(3,12) + P(3,17))
P(1,2) = -0.125 * B3 * C3 - 0.5 * (P(1,10) + P(1,11) + P(1,19))
P(2,2) = 0.125 * A2 * C3 - 0.5 * (P(2,10) + P(2,11) + P(2,19))
P(3,2) = 0.125 * A2 * B2 - 0.5 * (P(3,9) + P(3,10) + P(3,19))
P(1,3) = -0.125 * B2 * C3 - 0.5 * (P(1,10) + P(1,11) + P(1,18))
P(2,3) = -0.125 * A2 * C2 - 0.5 * (P(2,10) + P(2,11) + P(2,18))
P(3,3) = 0.125 * A2 * B2 - 0.5 * (P(3,9) + P(3,10) + P(3,18))
P(1,4) = -0.125 * B2 * C2 - 0.5 * (P(1,11) + P(1,12) + P(1,20))
P(2,4) = -0.125 * A2 * C2 - 0.5 * (P(2,11) + P(2,12) + P(2,20))
P(3,4) = 0.125 * A2 * B2 - 0.5 * (P(3,11) + P(3,12) + P(3,20))
P(1,5) = 0.125 * B3 * C2 - 0.5 * (P(1,13) + P(1,14) + P(1,17))
P(2,5) = 0.125 * A3 * C2 - 0.5 * (P(2,13) + P(2,14) + P(2,17))
P(3,5) = 0.125 * A3 * B3 - 0.5 * (P(3,13) + P(3,14) + P(3,17))
P(1,6) = -0.125 * B3 * C2 - 0.5 * (P(1,13) + P(1,14) + P(1,18))
P(2,6) = -0.125 * A3 * C2 - 0.5 * (P(2,13) + P(2,14) + P(2,18))
P(3,6) = 0.125 * A3 * B3 - 0.5 * (P(3,13) + P(3,14) + P(3,18))
P(1,7) = -0.125 * B2 * C2 - 0.5 * (P(1,14) + P(1,15) + P(1,19))
P(2,7) = -0.125 * A2 * C2 - 0.5 * (P(2,14) + P(2,15) + P(2,19))
P(3,7) = -0.125 * A2 * B2 - 0.5 * (P(3,14) + P(3,15) + P(3,19))
P(1,8) = 0.125 * B2 * C2 - 0.5 * (P(1,15) + P(1,16) + P(1,20))
P(2,8) = -0.125 * A3 * C2 - 0.5 * (P(2,15) + P(2,16) + P(2,20))
P(3,8) = -0.125 * A3 * B2 - 0.5 * (P(3,15) + P(3,16) + P(3,20))

CALL JACOB(P,XX,XJI,DET,NEL,20)

DO 20 I = 1, 6
DO 10 J = 1, 60
B(I,J) = 0.0
10 CONTINUE
20 CONTINUE
K3 = 0
DO 40 K = 1, 20
   K3 = K3 + 3
   K2 = K3 - 1
   K1 = K3 - 2
DO 30 I = 1, 3
   B(1,K1) = B(1,K1) + XJI(1,I)*P(I,K)
   B(2,K2) = B(2,K2) + XJI(2,I)*P(I,K)
   B(3,K3) = B(3,K3) + XJI(3,I)*P(I,K)
30 CONTINUE
B(4,K1) = B(2,K2)
B(4,K2) = B(1,K1)
B(5,K2) = B(3,K3)
B(5,K3) = B(2,K2)
B(6,K1) = B(3,K3)
B(6,K3)=B(1,K1)
40 CONTINUE
C
RETURN
END
@PROCESS DC(STIFF,GFV,ADDR)
C
C================== SUPPOR ================================
C
SUBROUTINE SUPPOR
C
CALLED BY: ASSEM
CALLS : NONE
C
THIS SUBROUTINE ASSEMBLES THE STIFFNESSES OF SUPPORT SPRINGS INTO THE GLOBAL STIFFNESS MATRIX.
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/STIFF/GSTIF(20000000)
COMMON/GFV/GF(1000000)
COMMON/ADDR/JDIAG(40000)
COMMON/SPR/SPR(250,3),NSSP
C
DO 100 I=1,NSSP
  J=SPR(I,1)
  ANG=SPR(I,2)*3.141592654/180.0
  CS=DABS(DCOS(ANG))
  SN=DABS(DSIN(ANG))
  K=(J-1)*3
  NS1=JDIAG(K+1)
  NS3=JDIAG(K+3)
  GSTIF(NS1)=GSTIF(NS1)+SPR(I,3)*CS
  GSTIF(NS3)=GSTIF(NS3)+SPR(I,3)*SN
100 CONTINUE
RETURN
END
SUBROUTINE TEMPDIS

CALLED BY: MAIN
CALLS: NONE

SUBROUTINE TO ASSIGN A TEMPERATURE DISTRIBUTION ON THE CROSS SECTION.

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION T(3)
COMMON/DEPTH/DEP(6),ADTEMP(4,15)

E=0.848418011D0
F=0.727662441D0
DT=DEP(3)/2.*DEP(4)+DEP(5)
SLOPE=10./DT
T(1)=(DEP(3)+DEP(4))*5/DT
T(2)=(DEP(3)+2.*DEP(4)+DEP(5))*5/DT
T(3)=15.D0
SLOSL=10./DEP(6)
SE=SLOPE*E
SF=SLOPE*F
DO 10 I=1,11
10 ADTEMP(1,I)=0.0
ADTEMP(1,7)=SE
DO 20 I=12,15
20 ADTEMP(1,I)=SF

DO 100 J=1,3
IF (J.EQ.3) THEN
   SE=SLOSL*E
   SF=SLOSL*F
END IF
J1=J+1
DO 30 I=1,5
30 ADTEMP(J1,I)=T(J)
ADTEMP(J1,6)=T(J)-SE
ADTEMP(J1,7)=T(J)+SE
DO 40 I=8,11
40 ADTEMP(J1,I)=T(J)-SF
DO 50 I=12,15
50 ADTEMP(J1,I)=T(J)+SF
100 CONTINUE

FTOC=5./9.
DO 200 I=1,4
   DO 190 J=1,15
      ADTEMP(I,J)=FTOC*ADTEMP(I,J)
   190 CONTINUE
200 CONTINUE
SUBROUTINE TEMSFT(N, IP, F1, F2, TS1, TS2)

IMPLICIT REAL*8 (A-H,O-Z)
COMMON/TEMPS/TEMP(2500,15,3), ITCOD(2500,15)

TM1 = TEMP(N, IP, 1)
TM2 = TEMP(N, IP, 2)
TM3 = TEMP(N, IP, 3)
F1 = TM1 + 273.0
F2 = TM2 + 273.0
IF (ITCOD(N, IP) .EQ. 0) THEN
  FF1 = 0.0
  FF2 = 0.0
END IF
IF (ITCOD(N, IP) .EQ. 1) THEN
  FF1 = 5000.0 * (1.0 / 293.0 - 1.0 / F1)
  FF2 = 0
  ITCOD(N, IP) = 0
END IF
IF (ITCOD(N, IP) .EQ. 2) THEN
  FF1 = 0.0
  FF2 = 5000.0 * (1.0 / 293.0 - 1.0 / F2)
  ITCOD(N, IP) = 1
END IF
IF (ITCOD(N, IP) .EQ. 3) THEN
  FF1 = 5000.0 * (1.0 / 293.0 - 1.0 / F1)
  FF2 = 5000.0 * (1.0 / 293.0 - 1.0 / F2)
  ITCOD(N, IP) = 1
END IF

IF (FF1 .LE. 0.0) FF1 = 0.0
IF (FF2 .LE. 0.0) FF2 = 0.0
TS1 = DEXP(FF1)
TS2 = DEXP(FF2)

RETURN
END

SUBROUTINE TENEPS(ICODE)
CALLS: DCS, ELCD, STQB

SUBROUTINE TO CALCULATE THE STRAIN IN TENDON SEGMENTS

IMPLICIT REAL*8(A-H, O-Z)

DIMENSION D(100000), GP(2), EP(6), GPS(2, 3), CC(20, 3)

DIMENSION ELD(60), B(6, 60), DIR(6)

COMMON/ PRESTR/ NCTEN(500, 5), PCOOD(2000, 3), FS(500, 2),
  *CFS(500, 2), FI(50), AR(50), SEGC(500, 3, 3)

COMMON/ GVF/ GV(100000)

COMMON/ SIZE/ NUMP, NUMEL, NUMMAT, NSDF, NSBF, NHBW, NEQ, NELDL

COMMON/ MDATA/ COOD(25000, 4), NELC(2500, 22), RMAT(20, 5), IEL(2500),
  *VBDF(2000, 3), VSBF(1000, 4), ELDL(500, 3), COLD(25000, 4), ILEV(2500)

COMMON/ PEPS/ EPEP(500, 2), DELEPS(500, 2)

COMMON/ STRAND/ NOTEN, NOTSG, ISTRAN

COMMON/ STDISP/ DS(100000)

DATA GP/-0.577350269189626D0, 0.577350269189626D0/

DO 10 I=1, NEQ
  10 D(I)=GF(I)

IF (ICODE.EQ.3) THEN
  DO 20 I=1, NEQ
  20 D(I)=D(I)+DS(I)
END IF

DO 2000 I=1, NOTSG
  N=NCTEN(I, 4)
  DO 30 II=1, 2
    I2=II+1
    A1=SEGCG(I, I2, 1) - SEGCG(I, II, 1)
    B1=SEGCG(I, I2, 2) - SEGCG(I, II, 2)
    C1=SEGCG(I, I2, 3) - SEGCG(I, II, 3)
    CON=GP(2)
    IF (II.EQ.1) CON=1.0 - GP(2)
    GPS(II, 1)=SEGCG(I, II, 1) + CON*A1
    GPS(II, 2)=SEGCG(I, II, 2) + CON*B1
    GPS(II, 3)=SEGCG(I, II, 3) + CON*C1
  30 CONTINUE

C -- SET UP ELEMENT NODAL COORDINATE ARRAY AND
C DISPLACEMENT VECTOR --
CALL ELCD(N, CC, ELD, D)

C -- FIND STRAINS AT EACH GAUSS POINT ON TENDON --
DO 500 IG=1, 2
  R=GPS(IG, 1)
  S=GPS(IG, 2)
  T=GPS(IG, 3)

C -- SET UP B-MATRIX FOR THE PARENT ELEMENT --
CALL STQB(CC, B, DET, R, S, T, N)

C -- PERFORM B*ELD TO GET ELEMENT STRAINS --
DO 80 K=1, 6
  SUM=0.0
  DO 70 J=1, 60
SUM = SUM + B(K,J) * ELD(J)  
EP(K) = SUM

CONTINUE

C -- CALCULATE DIRECTION COSINES AT THE GAUSS POINT --  
CALL DCS(DIR,RJST,GP,I,IG,N)

C -- CALCULATE STRAIN AT THE GAUSS POINT --  
DELEPS(I,IG) = 0.0  
DO 90 K = 1,6  
DELEPS(I,IG) = DELEPS(I,IG) + DIR(K) * EP(K)  
TEPS(I,IG) = TEP(S(I,IG) + DELEPS(I,IG)

CONTINUE

CONTINUE

RETURN

END

@PROCESS DC(STIFF,GFV,ADDRESS)
@PROCESS DC(MDATA,PRESTR)

C

C = = = = = = = = = = = = = = = = = = = TENPOS = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

SUBROUTINE TENPOS

C CALLED BY: MAIN  
C CALLS : LOCAL  
C

C THIS SUBROUTINE CALCULATES THE POSITION OF STRAND NODES IN  
C NORMALIZED (R,S,T) COORDINATE SYSTEM.

IMPLICIT REAL*8(A-H,O-Z)  
DIMENSION XP(3),CC(20,3),P(3,20)  
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL  
COMMON/MDATA/COORD(25000,4),NELC(2500,22),RMAT(20,5),IEL(2500),  
*VBDF(2000,3),VSBF(1000,4),ELDL(500,3),COLD(25000,4),ILEV(2500)  
COMMON/STRAND/NOTEN,NOTS,ISTRAN  
COMMON/PRESTR/NCTEN(500,5),PCOOD(2000,3),FS(500,2),  
*CFS(500,2),FI(50),AR(50),SEGC(500,3,3)

C -- NCTEN - PT.#1,PT.#2,PT.#3,ELEMENT NO., TENDON NO. --  
C -- FS -- STRESS AT GAUSS POINTS 1 & 2 --  
C -- SEGC - (R,S,T) COORDINATES OF STRAND NODES --

DO 1000 I = 1,NOTS
N = NCTEN(I,4)  
DO 20 J = 3,22  
L = NELC(N,J)  
DO 10 K = 1,3  
J1 = J - 2  
K1 = K + 1  
CC(J1,K) = COOD(L,K1)
CONTINUE

CONTINUE

DO 500 IP = 1,3,2
NP = NCTEN(I,IP)
DO 40 J=1,3
  XP(J)=PCOOD(NP,J)
R=-0.5
S=0.0
T=-0.5
IF (IP.EQ.3) THEN
  R=0.5
  S=0.0
  T=0.5
END IF
CALL LOCAL(N,CC,XP,R,S,T)
  IF (DABS(S).LT.1.D-10) S=0.0
  SEGC(I,IP,1)=R
  SEGC(I,IP,2)=S
  SEGC(I,IP,3)=T
DO 50 KK=1,3
  IF (SEGC(I,IP,KK).GT.1.0) SEGC(I,IP,KK)=1.0
  IF (SEGC(I,IP,KK).LT.-1.) SEGC(I,IP,KK)=-1.
50  CONTINUE
500  CONTINUE
C
DO 650 J=1,3
  SEGC(I,2,J)=0.5*(SEGC(I,1,J)+SEGC(I,3,J))
C
1000  CONTINUE
C
RETURN
END
@PROCESS DC(STRAIN, EPS, STRN2, TEMPS, CEB1)
C
C================== ZEROEP ===============================================
C
SUBROUTINE ZEROEP
C
CALLED BY: MAIN
CALLS: NONE
C
SUBROUTINE TO INITIALIZE HIDDEN STATE VARIABLES AND STRAIN VECTORS
C
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/SIZE/NUMNP,NUMEL,NUMMAT,NSDF,NSBF,NHBW,NEQ,NELDL
COMMON/STRAIN/CSCUM(2500,6,15),DELCS(2500,6,15),HSV(2500,4,6,15)
COMMON/EPS/STRAIN(2500,6,15),DSTRN(2500,6,15)
COMMON/STRN2/HSV2(2500,6,15)
COMMON/TEMP/TEMP(2500,15,3),ITCOD(2500,15)
COMMON/CEB1/DSTOR(2500,6,15)
C
DO 50 N=1,NUMEL
  DO 40 I=1,15
    DO 30 J=1,6
      DO 20 K=1,4
        HSV(N,K,J,I)=0.0
        HSV2(N,J,I)=0.0
        CSCUM(N,J,I)=0.0
        DELCS(N,J,I)=0.0
        STRAIN(N,J,I)=0.0
        DSTRN(N,J,I)=0.0
        DSTOR(N,J,I)=0.0
      20 CONTINUE
      ITCOD(N,I)=3
    30 CONTINUE
  40 CONTINUE
  50 CONTINUE
C
RETURN
END
Appendix B
Input Instructions
and Listing of Program MESHGEN

Program MESHGEN can be used to generate the mesh and input file for program PCBRI
This appendix contains the input instructions to MESHGEN and a
listing of the program.

Input instructions for program MESHGEN

Program MESHGEN uses the values of character variables to perform opera-
tions required for the generation of the finite element mesh. All input to MESH-
GEN is format-free and character variables must be enclosed by single quotes.
Input variable names or their descriptions (bold letters) occupy separate input
lines. The input file must begin with a TITLE whose length is restricted to 60
characters. After the title, values must be given to a controlling character variable,
CHAR, which can assume the following distinct forms:

1. DEGOF
2. ELEMENTS
3. FINISH
4. LOADS
5. NODES
6. SUPPORTS
7. TENDONS
8. TIMES

The input required for each value of CHAR is now given. Program MESHGEN
prints error messages in file unit 6, for errors in input. The execution of MESHGEN
terminates on recognition of an error.

1. CHAR = 'DEGOF'
The value 'DEGOF' of CHAR is employed to generate data regarding speci-
fied degrees of freedom. It is followed by:

   ISUP  - no. of nodes at which displacements are specified

For each such node:
Figure B.1: Finite Element Mesh and Tendon Generation Example
Referring to the mesh example in Figure B.1(a), if only node 2 is restrained to zero displacements in the x and z directions, the input required is:

'DEGOF'
1
2 'X' 0.0 ' ' 0.0 'Z' 0.0

2. **CHAR = 'ELEMENTS'**
This option is used to generate input to PCB Ridge that describes element connectivity, element type, material type and the depths of elements on the cross section. Data pertaining to this option is given as:

- **NUMEL** - no. of elements
- **CHA1** - control character variable

Character variable CHA1 can assume the following values:

- **(a) CONNECTIVITY**
- **(b) DEPTH**
- **(c) MATERIAL**
- **(d) STOP**
- **(e) TYPE**

**CHA1 = 'CONNECTIVITY'**
Element connectivity is generated in one direction as follows:

\[ \text{CH1, I1, (NC(J), J=1,20)} \]
- **CH1** - 'ELE' or 'GEN'
- **I1** - first element no.
- **NC(J)** - connectivity

CH1 = 'ELE' is used for stand-alone elements.
At the end of each set of element generation input data, a character CH4 is required:

\[ \text{CH4} \quad - \quad \text{‘FINISH’ or ‘NEXT’} \]

To generate elements 1 to 2 in Figure B.1(a), the input data is given as:

\[ \begin{align*}
\text{‘GEN’} \quad 1 & \quad 69 & \quad 8 & \quad 6 & \quad 67 & \quad 64 & \quad 3 & \quad 1 & \quad 62 \\
 & \quad 47 & \quad 7 & \quad 46 & \quad 68 & \quad 45 & \quad 2 & \quad 44 & \quad 63 \\
 & \quad 66 & \quad 5 & \quad 4 & \quad 65 \\
\text{‘TO’} \quad 2 & \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 & \quad 61 \quad \text{‘BY’} & \quad 1 \\
\text{‘NEXT’ or ‘FINISH’}
\end{align*} \]

The ‘FINISH’ characters terminate element connectivity generation.

Note: A more powerful element generation scheme is available in the third option for node generation.

\text{CHA1 = ‘DEPTH’}

This option is used to specify the levels of elements in the mesh. Input required is:

\[ \text{NOLEV} \quad - \quad \text{no. of element levels (‘6’ in Figure B.1(a))} \]

For each level:

\[ \text{DEP} \quad - \quad \text{depth of level} \]

\[ \text{‘10.0’ for level 1 in Figure B.1(a)} \]

\text{CHA1 = ‘MATeRIAL’}

Material information is to be given as follows:

\[ \text{CH1, IS, IL, IMAT} \]

\text{CH1} \quad - \quad \text{‘ELEMENT’}

\text{IS} \quad - \quad \text{first element in a group}

\text{IL} \quad - \quad \text{last element in a group}

\text{IMAT} \quad - \quad \text{material no.} \]
NUMMAT - no. of different materials

For each material:

E, PR, WT-X, WT-Y, WT-Z

E - modulus of elasticity at first loading
   (E=0 if material is not loaded at start of analysis)
PR - Poisson's ratio
WT-X - unit weight in x-direction
WT-Y - unit weight in y-direction
WT-Z - unit weight in z-direction

FPC28, WCONC, TL, ICUR, CURLEN

FPC28 - 28-day concrete strength
WCONC - unit weight of concrete
TL - time at which concrete type is first loaded
ICUR - concrete cure type:
   '1': Type I cement, moist cured
   '2': Type I cement, steam cured
   '3': Type III cement, moist cured
   '4': Type III cement, steam cured
CURLEN - no. of days of curing

CHA1 = 'STOP'
This option terminates element and material data generation.

CHA1 = 'TYPE'
This option is used to specify element type and level numbers. Input under this option is as follows:

CH1, IS, IL, ITYP, LEV
CH1 - 'ELEMENT'
IS - first element in a group
IL - last element in a group
ITYP - element type ('3')
LEV - level no.

Elements 3 and 4 in Figure B.1(a) are at level 2. Therefore, to generate their level numbers, write:

'ELEMENT' 3 4 3 2

3. CHAR = 'FINISH'
This option terminates mesh generation.
4. **CHAR = 'LOADS'**
Distributed loads on element faces and nodal loads are specified in this option as follows:

- **NELDL** – no. of distributed loads

For each group of elements loaded with the same uniformly distributed load:
- **NE1, NE2, IFAC, FORCE**
  - **NE1** – first element
  - **NE2** – last element
  - **IFAC** – face no.
  - **FORCE** – distributed load magnitude and direction

For nodal loads:
- **NSBF** – no. of loaded nodes

For each loaded node:
- **Node no., x-force, y-force, z-force**

Consider, that in the mesh example of Figure B.1(a), elements 11 through 16 are loaded with a uniformly distributed load of 1000 psi in the negative local t-coordinate. Also, assume that node 17 is loaded with a force of 200 lbs. in the positive global x-direction. These loads can be specified as:

```
'LOADS'
  11 16  5  1000.0
  1
  17 200.0  0.0  0.0
```

5. **CHAR = 'NODES'**
The 'NODES' option is used for node generation and three ways to achieve nodal coordinate generation are available. Data for the 'NODES' option is given as:

- **NUMNP, KCODE**
  - **NUMNP** – no. of nodes in the mesh
  - **KCODE** – node generation option code

If **KCODE = '1'**: Generate nodes in lines.
In this option, the following statements should be given:

- **CH1, I1, x-cood., y-cood., z-cood.**
  - **CH1** – 'NODE' or 'GEN'
  - **I1** – node no.
Specifying CH1 = 'NODE' causes the input nodal coordinates to be reproduced 'as is' in the output file.

If CH1 = 'GEN':

CH2, I2, x-cood., y-cood., z-cood., CH3, INC
CH2 - 'TO'
I2 - last node no. in the line
CH3 - 'BY'
INC - node no. increment

To generate nodes 1, 62 and 123 (Figure B.1(a)), the input required is:

'NODES'
165 1
'GEN' 1 0.0 -12.0 0.0
'TO' 123 40.0 -12.0 0.0 'BY' 61

If KCODE = 2: Repeat cross sections of nodes.
This option is used to generate similar cross sections of nodes and input is as follows:

NSECT - no. of input sections
For each input section:

IST, NNS, IRS, IADD
IST - first node on input section
NNS - no. of nodes on the input section
IRS - no. of sections to be generated
IADD - node no. increment for each section
For each node on the input section:

Node no., x-cood., y-cood., z-cood.

As an example, the cross section of Figure B.1(a) in the y-z plane can be used to generate the sections at x = 20° and x = 40°. This is achieved by:

'NODES'
165 2
1
1 43 2 61
1 0.0 12.0 0.0
.. 
43 0.0 -25.0 60.0
If \( \text{KCODE} = 3 \): Generate horizontal or vertical stacks of nodes and elements. This is the most powerful option for node generation as well as element connectivity generation. In this option, a group of elements and nodes are generated by considering them to be enclosed by a master 20-node isoparametric element defined by master nodes. The isoparametric shape functions are used to generate intermediate nodes and elements. This option can be used to generate meshes with curved edges and is specific to meshes for the bridge type under investigation. Input is given as:

- \( \text{NSPANS} \) — no. of bridge spans
- \( \text{XINC}, \text{YINC}, \text{ZINC}, \text{NODINC}, \text{IELINC}, \text{IELADD} \)
  - \( \text{XINC} \) — increment of \( x \)-coordinate from one span to the next
  - \( \text{YINC} \) — increment of \( y \)-coordinate from one span to the next
  - \( \text{ZINC} \) — increment of \( z \)-coordinate from one span to the next
  - \( \text{NODINC} \) — increment in node no.
  - \( \text{IELINC} \) — increment in element no.
  - \( \text{IELADD} \) — increment in element no. when moving from one set of \( x \)-generated elements to a parallel set
- \( \text{NSETS} \) — no. of sets of horizontal or vertical stacks of elements

For each set:
- \( \text{NXL}, \text{NYL}, \text{NZL} \)
  - \( \text{NXL} \) — no. of elements in the \( x \)-direction
  - \( \text{NYL} \) — no. of elements in the \( y \)-direction
  - \( \text{NZL} \) — no. of elements in the \( z \)-direction

Note: Either \( \text{NYL} > 1 \) or \( \text{NZL} > 1 \).

- \( \text{ISTNOD}, \text{ISTEL}, \text{NINX1}, \text{NINX2}, \text{NINZ1}, \text{NINZ2}, \text{NINZ3} \)
  - \( \text{ISTNOD} \) — first node
  - \( \text{ISTEL} \) — first element
  - \( \text{NINX1} \) — first node increment in \( x \)-direction
  - \( \text{NINX2} \) — second node increment in \( x \)-direction
  - \( \text{NINZ1} \) — first node increment in \( z \)-direction
  - \( \text{NINZ2} \) — second node increment in \( z \)-direction
  - \( \text{NINZ3} \) — third node increment in \( z \)-direction
- \( \text{SS(NM)}, \text{NM} = 1, 1+2 \times \text{NYL} \)
  - \( \text{SS(NM)} \) — normalized coordinates of nodes in the \( y \)-direction

For 20 master nodes:
- \( x \)-coord., \( y \)-coord., \( z \)-coord.

If \( \text{NZL} > 1 \), subsequent to the first \( z \)-level of elements, only 12 master node coordinates need be given.

This option can best be explained by way of an example. In Figure B.1(a),
there is one horizontal stack of elements (11-16) and one vertical stack (1-10). Elements 9 and 10 have to be generated separately using the element generation option, but the coordinates of their nodes can be generated here. Coordinates of master nodes have to be given in the usual nodal connectivity order (Figure 2.2). Therefore, the mesh can be generated as:

'NODES'
  165  3
 0
 2

To generate elements 11 through 16 with the coordinates of their nodes:

2  3  1
26 11 18 33 7 4 4
-1.0  -0.6  -0.2 0.0  0.2  0.6  1.0

Note: NINX1 = 54-36 = 18
NINX2 = 87-54 = 33
NINZ1 = 33-26 = 7
NINZ2 = 37-33 = 4
NINZ3 = 58-54 = 4

Note: The width of the slab shown is 50". To obtain the normalized coordinate in the y-direction for node 42, perform the operation: -15/25 = -0.6

The master node coordinates are:

40.0  25.0  60.0 (Node 165)
 0.0  25.0  60.0 (Node 43)
 0.0 -25.0  60.0 (Node 37)
40.0 -25.0  60.0 (Node 159)
40.0  25.0  53.0 (Node 154)
 0.0  25.0  53.0 (Node 32)
 0.0 -25.0  53.0 (Node 26)
40.0 -25.0  53.0 (Node 148)
20.0  25.0  60.0 (Node 104)
 0.0  0.0  60.0 (Node 40)
20.0 -25.0  60.0 (Node 98)
40.0  0.0  60.0 (Node 162)
20.0  25.0  53.0 (Node 93)
 0.0  0.0  53.0 (Node 29)
20.0 -25.0  53.0 (Node 87)
40.0  0.0  53.0 (Node 151)
40.0  25.0  56.5 (Node 158)
 0.0  25.0  56.5 (Node 36)
 0.0 -25.0  56.5 (Node 33)
40.0 -25.0  56.5 (Node 155)
To generate the vertical stack of elements (1-10) and the nodal coordinates; NXL = 2, NYL = 1, NZL = 4. Therefore, the input required begins as:

\[
\begin{array}{ccc}
2 & 1 & 4 \\
1 & 1 & 43 & 18 & 3 & 2 & 2 \\
-1.0 & 0.0 & 1.0 \\
\end{array}
\]

followed by the coordinates of master nodes. The coordinates have to be given in the following order of master nodes:

- **Level 1:**
  - 130 8 6 128
  - 125 3 1 123
  - 68 7 67 129
  - 63 2 62 124
  - 127 5 4 126

- **Level 2:**
  - 135 13 11 133
  - 74 12 72 134
  - 132 10 9 131

- **Level 3:**
  - 140 18 16 138
  - 79 17 77 139
  - 137 15 14 136

- **Level 4:**
  - 145 23 21 143
  - 84 22 82 144
  - 142 20 19 141

- **Level 5:**
  - 152 30 28 150
  - 91 29 89 151
  - 147 25 24 146

Suggestion: The connectivity of elements 9 and 10 do not follow the pattern of elements 1 through 8. However, the coordinates of nodes at the mid-height of elements 9 and 10 (eg. 24 & 25) need to be generated. Therefore, it is prudent to first generate the coordinates for the vertical stack first, and to alter node numbering at the top of elements 9 and 10 by generating the horizontal stack of elements.

6. **CHAR = 'SUPPORTS'**
   This option is used to generate support spring data and is accessed by:

   \[
   \begin{array}{l}
   \text{NSSP} \quad \text{no. of support springs} \\
   \text{Node no., Direction ('1', '2' or '3'), Spring constant} \\
   \end{array}
   \]

7. **CHAR = 'TENDONS'**
   Prestress tendon data is generated using this option. The program automatically calculates the intersections of the tendon with element faces and divides it into strand segments. These segments are assigned to their parent elements.
elements. The coordinates of nodes on each strand segment are also generated. The input statements required by this option are:

**NTEN, NSEG**

**NTEN**  –  no. of continuous tendons  
**NSEG**  –  no. of tendon segments (straight-line pieces of tendons)

For each tendon segment:

**ITEN**  –  tendon no.  
**X1, Y1, Z1, X2, Y2, Z2**  
**X1**  –  x-cood. of end 1 of the segment  
**Y1**  –  y-cood. of end 1 of the segment  
**Z1**  –  z-cood. of end 1 of the segment  
**X2**  –  x-cood. of end 2 of the segment  
**Y2**  –  y-cood. of end 2 of the segment  
**Z2**  –  z-cood. of end 2 of the segment

For each tendon:

**FI, AR**  
**FI**  –  initial stress  
**AR**  –  cross sectional area

For all tendons:

**FY**  –  yield stress

The strand segments for the 2 tendons in Figure B.1(b) can be generated as follows:

‘TENDONS’

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>100.0</td>
</tr>
<tr>
<td>2</td>
<td>140.0</td>
</tr>
</tbody>
</table>

8. **CHAR = ‘TIMES’**

This option reproduces time step data. Required input is:

**NTIMES**  –  no. of time steps  
For each time step:

Time, Time code, Ambient temperature
Program Listing

The listing of MESHGEN is now given with subroutines arranged in alphabetical order. Table B.1 shows the data files required for the execution of MESHGEN.

Table B.1: Data Files for MESHGEN

<table>
<thead>
<tr>
<th>File Unit No.</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Input data for MESHGEN</td>
</tr>
<tr>
<td>6</td>
<td>Error statement(s)</td>
</tr>
<tr>
<td>8</td>
<td>Output from MESHGEN</td>
</tr>
<tr>
<td>9, 10, ...</td>
<td>Auxiliary files - One file for every additional span after the first</td>
</tr>
</tbody>
</table>
Program to generate the mesh for program PCBRIDGE

Controlling variables:

NELDL = No. of element distributed loads
NOLEV = No. of levels in the mesh
NOTEN = No. of prestress tendons
NOTSG = No. of tendon segments
NSBF = No. of loaded nodes
NSDF = No. of specified degrees of freedom
NSSP = No. of support springs
NTIMES = No. of time steps in the analysis

Implicit real*8(a-h,0-z)
Dimension Dep(10),Spr(50,3)
Common/One/Cc(14000,4),Vsbf(500,4),Eldl(1500,3),Nelc(1500,22),
     *Iel(1500),Rmat(20,5),Vbdf(2000,3),Ilev(1500)
Common/Two/Nunmp,Nunel,Nummat
Common/Three/Neldl,Nsbf,NSDF
Common/Four/Ncten(900,5),Pcood(5000,3),Fi(50),Ar(50)
Common/Five/Noten,Notsg
Common/Six/Fy
Common/Seven/Ti(40),Ic(40),Temp(40)
Common/Mat/Fpc28(20),Wconc(20),Tl(20),Icur(20),Curlen(20)
Logical Yes, Check
Character *5 CLIST(8)
Character *5 Char
Character *60 Title
Character *4 Char1
Data CLIST/'NODES','FINIS','ELEMENTS','LOADS','DEGOF',
     *'TENDO','TIMES','SUPPO'/

NSSP=0
NOLEV=0
NELDL=0
NSDF=0
NSBF=0
NOTEN=0
NOTSG=0
NTIMES=0

READ(4,*) TITLE
10 READ(4,*) CHAR
YES=.FALSE.
DO 15 I=1,8
IF (CHAR.EQ.CLIST(I)) THEN
  YES=.TRUE.
  GO TO 20
END IF

15 CONTINUE
WRITE(6,201) CHAR
201 FORMAT(5X,'INVALID CHARACTER INPUT : ',A5)
STOP

20 IF (CHAR.EQ.'FINIS') GO TO 999
IF (CHAR.EQ.'NODES') THEN
  READ(4,*) NUMNP,IKODE
  IF (IKODE.EQ.1) CALL GENNOD
  IF (IKODE.EQ.2) CALL GENN2
  IF (IKODE.EQ.3) CALL GENN3
  GO TO 10
END IF

IF (CHAR.EQ.'DEGOF') THEN
  READ(4,*) ISUP
  IF (ISUP.EQ.0) THEN
    NSDF=0
    GO TO 10
  END IF
  CALL DOF(ISUP)
  GO TO 10
END IF

IF (CHAR.EQ.'ELEME') THEN
  READ(4,*) NUMEL
  READ(4,*) CHA1
  CHECK=.FALSE.
  IF (CHA1.EQ.'TYPE') CHECK=.TRUE.
  IF (CHA1.EQ.'CONN') CHECK=.TRUE.
  IF (CHA1.EQ.'MATE') CHECK=.TRUE.
  IF (CHA1.EQ.'DEPT') CHECK=.TRUE.
  IF (CHA1.EQ.'STOP') CHECK=.TRUE.
  IF (CHECK) GO TO 33
  WRITE(6,202) CHA1
202 FORMAT(5X,'INVALID CHARACTERS IN ELEMENT SPECIFICATION : ',A4)
STOP

33 IF (CHA1.EQ.'TYPE') THEN
  CALL ELTYP
  GO TO 25
END IF

IF (CHA1.EQ.'CONN') THEN
  CALL GENELE
  GO TO 25
END IF

IF (CHA1.EQ.'MATE') THEN
  CALL MATL
  GO TO 25
END IF

IF (CHA1.EQ.'DEPT') THEN
  READ(4,*) NOLEV
  DO 41 I=1,NOLEV
41  READ(4,*) DEP(I)
GO TO 25
END IF
IF (CHAR.EQ.'STOP') GO TO 10
END IF

C
IF (CHAR.EQ.'SUPPO') THEN
READ(4,*) NSSP
DO 51 I=1,NSSP
51    READ(4,*) (SPR(I,J),J=1,3)
GO TO 10
END IF

C
IF (CHAR.EQ.'LOADS') THEN
CALL LOAD
GO TO 10
END IF

C
IF (CHAR.EQ.'TENDO') THEN
CALL PSTRES(NP)
NN=NP
GO TO 10
END IF

C
IF (CHAR.EQ.'TIMES') THEN
CALL TIME(NTIMES)
GO TO 10
END IF

C
999 N=NSDF
NSDF=N

C
1000 WRITE(8,*) TITLE
WRITE(8,*) NUMNP
DO 400 I=1,NUMNP
400    WRITE(8,901) (CC(I,J),J=1,4)
901    FORMAT(4(3X,F10.3))
WRITE(8,*) NUMMAT
DO 410 I=1,NUMMAT
410    WRITE(8,*) (RMAT(I,J),J=1,5)
WRITE(8,*) FPC28(I),WCONC(I),TL(I),ICUR(I),CURLEN(I)
WRITE(8,*) NUMEL
DO 420 I=1,NUMEL
    WRITE(8,*) IEL(I),ILEV(I)
420    CONTINUE
WRITE(8,*) NOLEV
DO 425 I=1,NOLEV
425    WRITE(8,*) DEP(I)
WRITE(8,*) NELDL
IF (NELDL.EQ.0) GO TO 501
DO 430 I=1,NELDL
430    WRITE(8,*) (ELDL(I,J),J=1,3)
445    CONTINUE
IF (NSDF.EQ.0) GO TO 445
501    WRITE(8,*) NSDF
DO 440 I=1,NSDF
440 WRITE(8,*)(VBDF(I,J),J=1,3)
445 WRITE(8,*), NSBF
   IF (NSBF.EQ.0) GO TO 480
   DO 450 I=1,NSBF
450 WRITE(8,*)(VSBF(I,J),J=1,4)
480 WRITE(8,*), NSSF
   IF (NSSF.EQ.0) GO TO 502
   DO 485 I=1,NSSF
485 WRITE(8,*)(VSBF(I,J),J=1,4)
480 WRITE(8,*), NSSF
   IF (NSSF.EQ.0) GO TO 502
   DO 485 I=1,NSSF
485 WRITE(8,*)(VSBF(I,J),J=1,4)
502 WRITE(8,*), NOTEN,NOTSG
   IF (NOTEN.EQ.0) GO TO 472
   DO 460 I=1,NOTSG
460 WRITE(8,*)(NCTEN(I,J),J=1,5)
   DO 465 I=1,NN
465 WRITE(8,921)(PCOOI(I,J),J=1,3)
921 FORMAT(3(3X,F10.3))
   DO 470 I=1,NOTEN
470 WRITE(8,*), FI(I),AR(I)
   WRITE(8,*), FY
472 IF (NTIMES.EQ.0) GO TO 2000
   DO 475 I=1,NTIMES
475 WRITE(8,*)(TI(I),IC(I),TEMP(I))
C
2000 STOP
END
C
C========== DET ===================================================
C
FUNCTION DET(A)
FUNCTION TO EVALUATE THE DETERMINANT OF A 3 X 3 MATRIX
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION A(3,3)

D1=A(2,2)*A(3,3)
D2=A(3,2)*A(2,3)
D3=A(2,1)*A(3,3)
D4=A(3,1)*A(2,3)
D5=A(2,1)*A(3,2)
D6=A(3,1)*A(2,2)

DET=A(1,1)*(D1-D2)-A(1,2)*(D3-D4)+A(1,3)*(D5-D6)

RETURN
END

PROCCESS DC(ONE)

C
C========== DOF ===================================================
C
SUBROUTINE DOF(ISUP)
SUBROUTINE TO GENERATE SPECIFIED DEGREES OF FREEDOM
IMPLICIT REAL*8(A-H,O-Z)
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NELC(1500,22),
*IEL(1500),RMAT(20,5),VBDF(2000,3),ILEV(1500)
COMMON/TWO/NUMNP,NUMEL,NUMMAT
COMMON/THREE/NELDI,NSBF,NSDF
CHARACTER *1 A,B,C

N=0
DO 20 I=1,ISUP
  READ(4,*) K,A,XA,B,XB,C,XC
  IF (A.EQ.'X') THEN
    N=N+1
    VBDF(N,1)=K
    VBDF(N,2)=1
    VBDF(N,3)=XA
  END IF
  IF (B.EQ.'Y') THEN
    N=N+1
    VBDF(N,1)=K
    VBDF(N,2)=2
    VBDF(N,3)=XB
  END IF
  IF (C.EQ.'Z') THEN
    N=N+1
    VBDF(N,1)=K
  END IF
20 CONTINUE

VBDF(N,2)=3
VBDF(N,3)=XC
END IF
20 CONTINUE
C
NSDF=N
RETURN
END
SUBROUTINE ELTYP

SUBROUTINE TO GENERATE ELEMENT TYPE AND LEVEL

IMPLICIT REAL*8(A-H,O-Z)
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NELC(1500,22),
*IEL(1500),RMAT(20,5),VBDF(2000,3),ILEV(1500)
COMMON/TWO/NUMNP,NUMEL,NUMMAT
CHARACTER *3 CH1

N=0
10 IF (N.EQ.NUMEL) GO TO 1000
READ(4,*) CH1,IS,IL,ITYP,LEV
IF (CH1.EQ.'ELE') GO TO 15
WRITE(6,703) CH1
STOP
703 FORMAT(A3)
WRITE(6,701) IS,IL
STOP
15 IF (CH1.EQ.'ELE') THEN
   DO 20 I=IS,IL
      IEL(I)=ITYP
      ILEV(I)=LEV
      N=N+1
20    CONTINUE
   GO TO 10
END IF
701 FORMAT(5X,'ERROR IN TYPE SPECIFICATION AT ELEMENTS : ',I4,1X,I4)
C
1000 RETURN
END
SUBROUTINE GENELE

SUBROUTINE TO GENERATE ELEMENTS IN A LINE

IMPLICIT REAL*8(A-H,O-Z)
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NELC(1500,22),
IEL(1500),RMAT(20,5),VBDF(2000,3),ILEV(1500)
COMMON/TWO/NUMNP,NUMEL,NUMMAT
DIMENSION INCR(20),NC(20)
CHARACTER *3 CHI
CHARACTER *2 CH2,CH3

N=0
10 READ(4,*) CH1,I1,(NC(J),J=1,20)
IF (CH1.EQ.'ELE') GO TO 14
IF (CH1.EQ.'GEN') GO TO 14
   WRITE(6,601) I1
   STOP
14 N=N+1
   DO 15 K=1,20
      K2=K+2
      NELC(I1,K2)=NC(K)
15 CONTINUE
NELC(I1,1)=I1
IF (CH1.EQ.'ELE') THEN
   READ(4,*) CH1
   IF (CH1.EQ.'FIN') GO TO 1000
   GO TO 10
END IF
IF (CH1.EQ.'GEN') THEN
   READ(4,*) CH2,I2,(INCR(J),J=1,20),CH3,INC
   IF ((CH2.NE.'TO').OR.(CH3.NE.'BY')) THEN
      WRITE(6,601) I2
      STOP
   END IF
   IST=I1+INC
   IB=I1
   DO 30 I=IST,I2,INC
      N=N+1
      NELC(I,1)=I
      DO 20 J=3,22
         J1=J-2
      20     NELC(I,J)=NELC(IB,J)+INCR(J1)
   IB=I
30 CONTINUE
   READ(4,*) CH1
   IF (CH1.EQ.'FIN') GO TO 1000
   GO TO 10
END IF
SUBROUTINE GENNOD

SUBROUTINE TO GENERATE NODES IN A LINE

IMPLICIT REAL*8(A-H,O-Z)
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NLC(1500,22),
*IEL(1500),RMAT(20,5),VBDF(2000,3),ILEV(1500)
COMMON/TWO/NUMNP,NUMEL,NUMMAT
DIMENSION RINC(3)
CHARACTER *3 CH1
CHARACTER *2 CH2,CH3

DO 777 I=1,NUMNP
   DO 666 J=1,4
      CC(I,J)=0.0
   666   CONTINUE
777   CONTINUE
DO 888 K=1,3
   RINC(K)=0.0
N=0
10   IF (N.EQ.NUMNP) GO TO 1000
   READ(4,*) CH1,I1,CC(I1,2),CC(I1,3),CC(I1,4)
   IF (CH1.EQ.'NOD') GO TO 14
   IF (CH1.EQ.'GEN') GO TO 14
   IF (CH1.EQ.'END') GO TO 1000
      WRITE(6,501) I1
   STOP
14   N=N+1
      CC(I1,1)=I1
      WRITE(6,*) (CC(I1,K),K=1,4)
   IF (CH1.EQ.'NOD') GO TO 10
   IF (CH1.EQ.'GEN') THEN
      READ(4,*) CH2,I2,(CC(I2,J),J=2,4),CH3,INC
      IF ((CH2.NE.'TO').OR.(CH3.NE.'BY')) THEN
         WRITE(6,501) I2
      STOP
      END IF
      CC(I2,1)=I2
      IDIFF=(I2-I1)/INC
      RDIFF=REAL(IDIFF)
      DO 20 J=1,3
         J1=J+1
         RINC(J)=(CC(I2,J1)-CC(I1,J1))/RDIFF
20 CONTINUE
   IST=I1+INC
   LAST=I2-INC
   IB=I1
   DO 40 I=IST,LAST,INC
      N=N+1
      CC(I,1)=I
      DO 30 J=1,3
         J1=J+1
      30 CC(I,J1)=CC(IB,J1)+RINC(J)
   IB=I
   40 CONTINUE
   N=N+1
   GO TO 10
END IF
C
501 FORMAT(5X,'ERROR IN INPUT AT NODE : ',I3)
C02 FORMAT(5X,'ERROR IN NODAL INPUT !!',/5X,'CHECK NO. OF NODES')
C
1000 RETURN
END
@PROCESS DC(ONE)
C
C======== GENN2 ==================================================================
C
SUBROUTINE GENN2
C
SUBROUTINES TO GENERATE NODES BY REPEATING CROSS SECTIONS
C
IMPLICIT REAL*8(A-H,O-Z)
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NELC(1500,22),
*IEL(1500),RMAT(20,5),VBDF(2000,3),ILEV(1500)
C
READ(4,*) NSECT
DO 500 NS=1,NSECT
   READ(4,*) IST,NNS,IRS,IADD
   READ(4,*) XINC,YINC,ZINC
   II=IST-1
   DO 10 I=1,NNS
      II=II+1
      READ(4,*) (CC(II,J),J=1,4)
   10 CONTINUE
   IF (IRS.EQ.0) GO TO 500
   DO 400 IS=1,IRS
      NNO=IST+IS*IADD
      DO 350 I=1,NNS
         NNP=NNO-IADD
         CC(NNO,1)=NNO
         CC(NNO,2)=CC(NNO,2)+XINC
         CC(NNO,3)=CC(NNO,3)+YINC
         CC(NNO,4)=CC(NNO,4)+ZINC
         NNO=NNO+1
      350 CONTINUE
   400 CONTINUE
SUBROUTINE GENN3

SUBROUTINE TO GENERATE NODES AND ELEMENTS BY MAPPING OF
NORMALIZED COORDINATES INTO GLOBAL COORDINATES USING
STANDARD ISOPARAMETRIC SHAPE FUNCTIONS FOR THE TRANSFORMATION

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION X(20,3),A(20),SS(30),KN(20)
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NELC(1500,22),
*IEL(1500),RMAT(20,5),VBDF(2000,3),ILEV(1500)

IO=4
IO1=9
READ(IO,*) NSPANS
XINC=0.0
YINC=0.0
ZINC=0.0
NODINC=0
IELINC=0
IELADD=0
IF (NSPANS.GT.1) READ(IO,*) XINC,YINC,ZINC,
*NODINC,IELINC,IELADD
DO 2000 NSP=1,NSPANS

READ(IO,*) NSETS
WRITE(IO1,*) NSETS
DO 1000 N=1,NSETS
READ(IO,*) NXL,NYL,NZL
WRITE(IO1,*) NXL,NYL,NZL
NSFX=1+2*NXL
NSFY=1+2*NYL
NSFZ=1+2*NZL
READ(IO,*) ISTNOD,ISTEL,NINX1,NINX2,NINZ1,NINZ2,NINZ3
IN1=ISTNOD+NODINC
IEL1=ISTEL+IELINC
WRITE(IO1,*) IN1,IEL1,NINX1,NINX2,NINZ1,NINZ2,NINZ3
READ(IO,*) (SS(NM),NM=1,NSFY)
WRITE(IO1,*) (SS(NM),NM=1,NSFY)
NEWWOD=ISTNOD
IFLAG=0

DO 900 IZ=1,NZL
IF (IELADD.EQ.0) IELADD=NXL
IF (IFLAG.EQ.0) THEN
   DO 5 I=1,20
      READ(IO,*) (XYZM(I,J),J=1,3)
A = XYZM(I,1) + X INC
B = XYZM(I,2) + Y INC
C = XYZM(I,3) + Z INC
WRITE(IO1,*) A, B, C
C
CONTINUE
GO TO 95
END IF
C
DO 20 I = 1, 4
   J = I + 4
   DO 10 K = 1, 3
      XYZM(J,K) = XYZM(I,K)
   L = I + 8
   M = L + 4
   DO 15 K = 1, 3
      XYZM(M,K) = XYZM(L,K)
   CONTINUE
   DO 25 I = 1, 4
      READ(IO,*) (XYZM(I,J), J = 1, 3)
      A = XYZM(I,1) + X INC
      B = XYZM(I,2) + Y INC
      C = XYZM(I,3) + Z INC
      WRITE(IO1,*) A, B, C
   CONTINUE
   DO 26 I = 9, 12
      READ(IO,*) (XYZM(I,J), J = 1, 3)
      A = XYZM(I,1) + X INC
      B = XYZM(I,2) + Y INC
      C = XYZM(I,3) + Z INC
      WRITE(IO1,*) A, B, C
   CONTINUE
   DO 27 I = 17, 20
      READ(IO,*) (XYZM(I,J), J = 1, 3)
      A = XYZM(I,1) + X INC
      B = XYZM(I,2) + Y INC
      C = XYZM(I,3) + Z INC
      WRITE(IO1,*) A, B, C
   CONTINUE
C
95
II = 0
III = 0
C
WRITE(6,*) 'NEWNOD = ', NEWNOD
NN = NEWNOD
DO 200 I = 1, NSFX
   R = 1.0 + 2.0*(I-1)/(2*NXL)
   II = II + 1
   III = III + 1
   IF (II.EQ.3) II = 1
   IJ = 0
   NSTOR1 = NN
   DO 190 K = 1, 3, II
      IF (((K.EQ.1).AND.(IFLAG.GT.0))) GO TO 99
      T = 1.0 + 2.0*(K-1)/2
I J = I J + 1
IF (I J .EQ. 2) I J = 2
IF (I J .EQ. 3) I J = 1
NSTOR2 = NN
DO 180 J = 1, NSF Y, I J
   IF ((K .EQ. 1) .AND. (IFLAG .GT. 0)) GO TO 159
   S = SS (J)
   CC (NN, 1) = NN
C  ------------------- EVALUATE SHAPE FUNCTIONS -------------------
   A 1 = 1. - R * R
   A 2 = 1. - R
   A 3 = 1. + R
   B 1 = 1. - S * S
   B 2 = 1. - S
   B 3 = 1. + S
   C 1 = 1. - T * T
   C 2 = 1. - T
   C 3 = 1. + T
   A N (9) = 0.25 * A 1 * B 3 * C 3
   A N (10) = 0.25 * A 2 * B 1 * C 3
   A N (11) = 0.25 * A 1 * B 2 * C 3
   A N (12) = 0.25 * A 3 * B 1 * C 3
   A N (13) = 0.25 * A 1 * B 3 * C 2
   A N (14) = 0.25 * A 2 * B 1 * C 2
   A N (15) = 0.25 * A 1 * B 2 * C 2
   A N (16) = 0.25 * A 3 * B 1 * C 2
   A N (17) = 0.25 * A 3 * B 3 * C 1
   A N (18) = 0.25 * A 2 * B 3 * C 1
   A N (19) = 0.25 * A 2 * B 2 * C 1
   A N (20) = 0.25 * A 3 * B 2 * C 1
   A N (1) = 0.125 * A 3 * B 3 * C 3 - 0.5 * (A N (9) + A N (12) + A N (17))
   A N (2) = 0.125 * A 2 * B 3 * C 3 - 0.5 * (A N (9) + A N (10) + A N (18))
   A N (3) = 0.125 * A 2 * B 2 * C 3 - 0.5 * (A N (10) + A N (11) + A N (19))
   A N (4) = 0.125 * A 3 * B 2 * C 3 - 0.5 * (A N (11) + A N (12) + A N (20))
   A N (5) = 0.125 * A 3 * B 3 * C 2 - 0.5 * (A N (13) + A N (16) + A N (17))
   A N (6) = 0.125 * A 2 * B 3 * C 2 - 0.5 * (A N (13) + A N (14) + A N (18))
   A N (7) = 0.125 * A 2 * B 2 * C 2 - 0.5 * (A N (14) + A N (15) + A N (19))
   A N (8) = 0.125 * A 3 * B 2 * C 2 - 0.5 * (A N (15) + A N (16) + A N (20))
C
   XXX = 0.0
   Y Y Y = 0.0
   Z Z Z = 0.0
DO 150 L = 1, 20
   XXX = XXX + A N (L) * XYZM (L, 1)
   Y Y Y = Y Y Y + A N (L) * XYZM (L, 2)
   Z Z Z = Z Z Z + A N (L) * XYZM (L, 3)
150 CONTINUE
   CC (NN, 2) = XXX
   CC (NN, 3) = Y Y Y
   CC (NN, 4) = Z Z Z
159 NN = NN + 1
180 CONTINUE
185 INCZ = NINZ2
IF (I J .EQ. 2) INCZ = NINZ2
IF (II.EQ.2) INCZ=NINZ3
     NN=NSTOR2+INCZ
190 CONTINUE
  IJK=III-(III/2*2)
  IF (IJK.EQ.1) THEN
    NN=ISTNOD+(I-1/2)*NINX1+(I-1)/2*NINX2+(IZ-1)*NINZ3
  END IF
  IF (IJK.EQ.0) THEN
    NN=ISTNOD+I*(NINX1+NINX2)/2+(IZ-1)*(NINZ1+NINZ2)
  END IF
200 CONTINUE
C
C ---------------- GENERATE ELEMENTS ----------------
C
N1=ISTEL+(IZ-1)*IELADD
IJUMP=NINX1+NINX2
KJUMP=NINZ1+NINZ2
N6=NEWNOD
C
GENERATE ELEMENTS BETWEEN SPANS
C
IF THERE IS MORE THAN ONE SPAN.
NXXL=NXL
IF (NSPANS.GT.1) THEN
  IF (NSP.LT.NSPANS) NXXL=NXL+1
END IF
DO 400 I=1,NXXL
  NN7=ISTNOD+(I-1)*IJUMP+(IZ-1)*KJUMP
  NN8=ISTNOD+I*IJUMP+(IZ-1)*KJUMP
  NN15=ISTNOD+I*NINX1+(I-1)*NINX2+(IZ-1)*NINZ3
DO 300 J=1,NYL
  NL=NL1+(J-1)*IELADD
  N7=NN7+(J-1)*2
  N8=NN8+(J-1)*2
  N15=NN15+(J-1)*1
C
  KN(7)=N7
  KN(14)=KN(7)+1
  KN(6)=KN(14)+1
  KN(19)=KN(7)+NINZ1-J+1
  KN(18)=KN(19)+1
  KN(3)=KN(7)+KJUMP
  KN(10)=KN(3)+1
  KN(2)=KN(10)+1
  KN(8)=N8
  KN(16)=KN(8)+1
  KN(5)=KN(16)+1
  KN(20)=KN(8)+NINZ1-J+1
  KN(17)=KN(20)+1
  KN(4)=KN(8)+KJUMP
  KN(12)=KN(4)+1
  KN(1)=KN(12)+1
  KN(15)=N15
  KN(13)=KN(15)+1
  KN(11)=KN(15)+NINZ3
  KN(9)=KN(11)+1
C
NELC(NL,1)=NL
DO 250 K=1,20
   K2=K+2
   NELC(NL,K2)=KN(K)
250   CONTINUE
C
300   CONTINUE
   N6=N1+IJUMP
   NL1=NL1+1
400   CONTINUE
C
   NEWNOD=ISTNOD+IZ*(NINZ1+NINZ2)
   IFLAG=IFLAG+1
900   CONTINUE
1000  CONTINUE
C
   REWIND(IO1)
   IO=IO1
   IO1=IO1+1
2000  CONTINUE
C
   RETURN
END
SUBROUTINE LOAD

SUBROUTINE TO GENERATE NODAL FORCES AND ELEMENT DISTRIBUTED LOADS

IMPLICIT REAL*8(A-H,O-Z)
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NELC(1500,22),
*IEL(1500),RMAT(20,5),VBDF(2000,3),ILEV(1500)
COMMON/TWO/NUMNP,NUMEL,NUMMAT
COMMON/THREE/NELDL,NSBF,NSDF

N=0
READ(4,*) NELDL
IF (NELDL.EQ.0) GO TO 500
10 READ(4,*) NE1,NE2,IFAC,FORCE
   DO 20 I=NE1,NE2
      N=N+1
      ELDL(N,1)=I
      ELDL(N,2)=IFAC
      ELDL(N,3)=FORCE
   20 CONTINUE
IF (N.NE.NELDL) GO TO 10
IF (N.GT.NELDL) THEN
   WRITE(6,*) 'ERROR IN DISTRIBUTED LOAD DATA !'
   STOP
END IF

500 READ(4,*) NSBF
IF (NSBF.EQ.0) GO TO 1000
   DO 30 I=1,NSBF
      30 READ(4,*) (VSBF(I,J),J=1,4)
1000 RETURN
END
@PROCESS DC(ONE)
C
C===============================================
C
SUBROUTINE MATL
C
SUBROUTINE TO SPECIFY MATERIAL INPUT
C
IMPLICIT REAL*8(A-H,O-Z)
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NELC(1500,22),
*IEL(1500),RMAT(20,5),VBDF(2000,3),ILEV(1500)
COMMON/TWO/NUMNP,NUMEL,NUMMAT
COMMON/MAT/FPC28(20),WCONC(20),TL(20),ICUR(20),CURLEN(20)
CHARACTER*3 CH1
C
N=0
10 IF (N.EQ.NUMEL) GO TO 30
READ(4,*) CH1,IS,IL,IMAT
IF ((CH1.EQ.'END').OR.(CH1.EQ.'ELE')) GO TO 15
WRITE(6,701) IS,IL
701 FORMAT(5X,'ERROR IN MATERIAL SPECIFICATION FOR ELEMENTS : ',2I3)
STOP
15 IF (CH1.EQ.'ELE') THEN
   DO 20 I=IS,IL
      NELC(I,2)=IMAT
      N=N+1
   20 CONTINUE
   WRITE(6,*) N
   GO TO 10
END IF
C
30 READ(4,*) NUMMAT
DO 40 K=1,NUMMAT
READ(4,*) (RMAT(K,L),L=1,5)
40 READ(4,*) FPC28(K),WCONC(K),TL(K),ICUR(K),CURLEN(K)
RETURN
END
SUBROUTINE PLINT(IFACE,N,T,TT,X21,Y21,Z21,X1,Y1,Z1)

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION JJ(4),IFAC(6,4),TF(4),NOFC(4),ARR(3,3),CD(3,3),M(3)
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NELC(1500,22),
*IEL(1500),RMAT(20,5),VBDF(2000,3),ILEV(1500)
DATA JJ/1, 2, 5, 6/
DATA IFAC/1, 2, 1, 3, 1, 5, 4, 6, 5, 7, 2, 8,
*8, 7, 6, 8, 3, 7, 5, 3, 2, 4, 4, 6/

C -- LOOP OVER 4 FACES --
DO 500 II=1,4

C -- DISCARD CURRENT FACE --
NOF=JJ(II)
IF (NOF.EQ.IFACE) THEN
  NOFC(II)=NOF
  TF(II)=2.0
  GO TO 500
END IF
NOFC(II)=NOF

C -- SET UP ARRAY OF THREE VERTICES ON THE FACE --
DO 10 I=1,3
  NN=IFAC(NOFC(I))+2
10  M(I)=NELC(N,NN)
DO 20 I=1,3
  DO 15 J=1,3
    J1=J+1
    CD(I,J)=CC(M(I),J1)
15  CONTINUE
20  CONTINUE

C -- SET UP COEFFS. OF THE EQUATION OF THE PLANE --
DO 30 I=1,3
  DO 25 J=1,3
    ARR(I,J)=CD(I,J)
25  CONTINUE
30  CONTINUE
D=-DET(ARR)
DO 35 I=1,3
  ARR(I,3)=1.D0
C=DET(ARR)
DO 40 I=1,3
  ARR(I,2)=CD(I,3)
B=-DET(ARR)
DO 45 I=1,3
  ARR(I,1)=CD(I,2)
A=DET(ARR)

C -- FIND THE POINT OF INTERSECTION OF THE LINE AND PLANE --
DEN=A*X21+B*Y21+C*Z21

IF (DEN.EQ.0.0) THEN
    TF(II)=2.0
    GO TO 500
END IF
RNUM=A*X1+B*Y1+C*Z1+D
TF(II)=-RNUM/DEN
IF (TF(II).LE.T) TF(II)=2.0

500 CONTINUE
C -- CHECK FOR THE LOWEST VALUE OF PARAMETER 'T' --
    TT=TF(1)
    IFACE=NOFC(1)
    DO 600 I=2,4
        IF (TF(I).LT.TT) THEN
            TT=TF(I)
            IFACE=NOFC(I)
        END IF
    END DO
600 CONTINUE
C
RETURN
END
@PROCESS DC(ONE)
C
C====== PSTRES ===============
C
SUBROUTINE PSTRES(NP)
C
SUBROUTINE TO GENERATE PRESTRESS STRAND DATA
C
IMPLICIT REAL*8(A-H,O-Z)
LOGICAL YES
DIMENSION END(50,6)
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NELC(1500,22),
*IEL(1500),RMAT(20,5),VBD(2000,3),ILEV(1500)
COMMON/TWO/NUMNP,NUMEL,NUMMAT
COMMON/THREE/NELDL,NSBF,NSDF
COMMON/FOUR/NCTEN(900,5),PCOOD(5000,3),FI(50),AR(50)
COMMON/FIVE/NOTEN,NOTSG
COMMON/SIX/FY
C
EPS=0.001
EPSM1=1.-EPS
EPSM1=1.+EPS
READ(4,*) NTEN,NSEG
WRITE(6,*) 'NO. OF TENDONS ',NTEN,NSEG
ISEG=1
IP=1
IOLD=0
DO 1000 I=1,NSEG
    READ(4,*) ITEN
    READ(4,*) (END(I,J),J=1,6)
    WRITE(6,*) ITEN
    WRITE(6,*) (END(I,J),J=1,6)
C -- SET UP PARAMETRIC EQUATIONS OF THE LINE --
    IF (ITEN.EQ.IOLD) THEN
IP=IP-1
X1=PCOOD(IP,1)
Y1=PCOOD(IP,2)
Z1=PCOOD(IP,3)
X21=END(I,4)-X1
Y21=END(I,5)-Y1
Z21=END(I,6)-Z1
GO TO 40
END IF
X1=END(I,1)
Y1=END(I,2)
Z1=END(I,3)
X21=END(I,4)-END(I,1)
Y21=END(I,5)-END(I,2)
Z21=END(I,6)-END(I,3)
40 OLD=ITEN
IFACE=2
T=0.0
PCOOD(IP,1)=X1
PCOOD(IP,2)=Y1
PCOOD(IP,3)=Z1
DO 50 N=1,NUMEL
   IF (ILEV(N).GT.5) GO TO 50
   YES=.FALSE.
   CALL SEARCH(N,X1,Y1,Z1,IFACE,YES)
   IF (YES) GO TO 100
50 CONTINUE
WRITE(6,*) 'POINT NOT FOUND',X1,Y1,Z1
100 CALL PRINT(IFACE,N,T,TT,X21,Y21,Z21,X1,Y1,Z1)
  X3=X1+TT*X21
  Y3=Y1+TT*Y21
  Z3=Z1+TT*Z21
  TM=(T+TT)*0.5
  X2=X1+TM*X21
  Y2=Y1+TM*Y21
  Z2=Z1+TM*Z21
  NCTEN(ISEG,1)=IP
  IP=IP+1
  PCOOD(IP,1)=X2
  PCOOD(IP,2)=Y2
  PCOOD(IP,3)=Z2
  NCTEN(ISEG,2)=IP
  IP=IP+1
  PCOOD(IP,1)=X3
  PCOOD(IP,2)=Y3
  PCOOD(IP,3)=Z3
  NCTEN(ISEG,3)=IP
  NCTEN(ISEG,4)=N
  NCTEN(ISEG,5)=ITEN
WRITE(6,*) ISEG,X3,Y3,Z3
C
IF ((TT.GE.EPSM1).AND.(TT.LE.EPSP1)) THEN
   IF (I.LT.NSEG) IP=IP+1
   IF (I.LT.NSEG) ISEG=ISEG+1
GO TO 1000
END IF
T=TT
IF (IFACE.EQ.6) IFACE=4
IFACE=IFACE+1
C
DO 200 N=1,NUMEL
   IF (ILEV(N).GT.5) GO TO 200
   YES=.FALSE.
   CALL SEARCH(N,X3,Y3,Z3,IFACE,YES)
   IF (YES) GO TO 300
200 CONTINUE
WRITE(6,*) 'POINT NOT FOUND ',X3,Y3,Z3
WRITE(6,*) 'INTERNAL',T,TT
STOP
300 ISEG=ISEG+1
GO TO 100
1000 CONTINUE
C
NOTEN=NTEN
NOTSG=ISEG
NP=IP
DO 2000 I=1,NOTEN
2000 READ(4,*) FI(I),AR(I)
   READ(4,*) FY
C
   WRITE(6,*) 'NP = ',NP
RETURN
END
SUBROUTINE SEARCH(N,X,Y,Z,NFACE,YES)

SUBROUTINE TO IDENTIFY THE ELEMENT NO. AND FACE NO. THAT A
POINT ON A PRESTRESS STRAND SEGMENT OCCUPIES

IMPLICIT REAL*8(A-H,O-Z)
DIMENSION IFAC(6,4),CD(4,3),M(4),ARR(3,3)
LOGICAL YES
COMMON/ONE/CC(14000,4),VSBF(500,4),ELDL(1500,3),NELC(1500,22),
*IEL(1500),RMAT(20,5),VBDF(2000,3),ILEV(1500)
DATA IFAC/1,2,1,3,1,5,4,6,5,7,2,8,8,7,6,8,3,7,5,3,2,4,4,6/

C -- SET UP ARRAY OF VERTICES OF QUADRILATERAL --
DO 10 I=1,4
   NN=IFAC(NFACE,I)+2
10  M(I)=NELC(N,NN)
   DO 20 I=1,4
      DO 15 J=1,3
          J1=J+1
          CD(I,J)=CC(M(I),J1)
15   CONTINUE
   CONTINUE
20  CONTINUE
C -- SET UP EQUATION OF THE PLANE --
DO 30 I=1,3
   DO 25 J=1,3
      ARR(I,J)=CD(I,J)
25   CONTINUE
30  CONTINUE
   D=-DET(ARR)
   DO 35 I=1,3
      ARR(I,3)=1.D0
35   C=DET(ARR)
   DO 40 I=1,3
      ARR(I,2)=CD(I,3)
40   B=-DET(ARR)
   DO 45 I=1,3
      ARR(I,1)=CD(I,2)
45   A=DET(ARR)
C -- CHECK IF POINT(X,Y,Z) LIES ON THE PLANE --
   EQ=A*X+B*Y+C*Z+D
   IF (DABS(EQ).GT.5.0D-1) GO TO 1000
C -- CHECK IF POINT(X,Y,Z) IS WITHIN THE QUADRILATERAL --
   EPSLON=-1.D-2
   DO 200 I1=1,4
      I2=I1+1
      IF (I2.EQ.5) I2=1
      XX=CD(I2,1)-CD(I1,1)
      YY=CD(I2,2)-CD(I1,2)
      ZZ=CD(I2,3)-CD(I1,3)
      DO 201 J=1,3
         EQ=XX*CD(I1,J)+YY*CD(I2,J)+ZZ*CD(I3,J)+D
         IF (DABS(EQ).GT.5.0D-1) GO TO 1000
201  CONTINUE
200  CONTINUE
XP=X-CD(I1,1)
YP=Y-CD(I1,2)
ZP=Z-CD(I1,3)
AA=YY*ZP-YP*ZZ
BB=XP*ZZ-XX*ZP
DD=XX*YP-XP*YY
IF (A*AA.LT.EPSLON) GO TO 1000
IF (B*BB.LT.EPSLON) GO TO 1000
IF (C*DD.LT.EPSLON) GO TO 1000
200 CONTINUE
YES=.TRUE.
WRITE(6,*) 'EQ = ',EQ
C
1000 RETURN
END
C
C============= TIME ==============
C
C SUBROUTINE TIME(N)
C
C SUBROUTINE TO INPUT TIME STEPS FOR THE ANALYSIS
C
C IMPLICIT REAL*8(A-H,O-Z)
COMMON/SEVEN/TI(40),IC(40),TEMP(40)
C
C READ(4,*) N
DO 10 I=1,N
10 READ(4,*) TI(I),IC(I),TEMP(I)
C
RETURN
END
Vita

Keith Joseph Rebello was born in London, England on September 15, 1959. He received a Bachelor of Technology degree in Civil Engineering in May 1983 from the Indian Institute of Technology, Bombay. He entered graduate school at Louisiana State University in August 1983. He graduated with a Master of Science degree in Civil Engineering in May 1986 and is presently a candidate for the degree of Doctor of Philosophy in Civil Engineering at Louisiana State University.
Candidate: Keith Joseph Rebello

Major Field: Civil Engineering

Title of Dissertation: Analysis for Creep, Shrinkage and Temperature Effects on Expansion Joint Movements in Composite Prestressed Concrete Bridges

Approved:

Vijaya K. A. Cepu
Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

R. Rebel Chart
Farsh Bajaj
Gregory T. Kueppel
David E. Thompson
George E. Vodnik

Date of Examination:
April 24, 1990