The Pennsylvania State University

The Graduate School

Department of Civil and Environmental Engineering

SERVICEABILITY ANALYSIS FOR DEFLECTION OF REINFORCED CONCRETE FLOOR SLABS IN MULTI-STORY HIGH-RISE BUILDINGS

A Thesis in

Civil Engineering

by

Christopher J. Motter

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2009

The thesis of Christopher J. Motter was reviewed and approved* by the following:

Andrew Scanlon
Professor of Civil Engineering
Thesis Advisor

Andrea Schokker
Adjunct Professor of Civil Engineering

Eric Donnell
Assistant Professor of Civil Engineering

Peggy Johnson
Professor of Civil Engineering
Head of Civil and Environmental Engineering

*Signatures are on file in the graduate school.

ABSTRACT

An analytical algorithm is developed to compute the mid-panel deflection-time history for two-way reinforced concrete floor slabs in multi-story high-rise buildings. Mid-panel deflection is determined using the crossing beam analogy, superimposing column strip and middle strip deflections. The analysis procedure incorporates the effects of early-age construction loading. The algorithm determines the loading-time history of the slab based upon the details of construction schemes involving one level of shoring in conjunction with two, three, or four levels of reshoring. Alternatively, the algorithm may operate based upon a prescribed loading-time history. The analytical algorithm is implemented as an interactive C++ computer program using Microsoft Visual Basic C++ Standard Edition 2003. A parametric study determines the sensitivity of the algorithm to variation in individual input parameters. Comparison of program results with published experimental results justifies the program as a valid means of computing two-way slab deflection-time histories.

TABLE OF CONTENTS

List of Tables

List of Figures

LIST OF SYMBOLS

b = section width

b_1 = long span (column strip) width

b_2 = short span (middle strip) width

$C(t_i, t_j)$ = time-dependent creep coefficient

C_u = average ultimate creep coefficient

cycle = number of days comprising the slab casting schedule

d = a constant used to define the time-ratio component for a given member shape and size, measured in days

DL = self-weight dead load of slab (psf)

E = modulus of elasticity

E_c = modulus of elasticity of concrete

f'_c = compressive strength of concrete

$f'_{c(28\text{-day})}$ = 28-day compressive strength of concrete

H = relative humidity as a percent

h = slab thickness

I = moment of inertia

I_{crx} = x-direction cracked moment of inertia

I_{cry} = y-direction cracked moment of inertia

I_e = effective moment of inertia

$I_{e\text{-}col}$ = effective moment of inertia for the column strip

$I_{e\text{-}mid}$ = effective moment of inertia for the middle strip

I_{ex} = x-direction effective moment of inertia

I_{ey} = y-direction effective moment of inertia

I_g = gross moment of inertia

$I_{g\text{-}col}$ = gross moment of inertia for the column strip

$I_{g\text{-}mid}$ = gross moment of inertia for the middle strip

K = stiffness coefficient for shoring

k = overall correction factor

k_1 = allowance for error in theoretical load ratio R

k_2 = allowance for weight of formwork

k_{BC} = boundary condition correction factor

k_{BC-col} = boundary condition correction factor for column strip

k_{BC-mid} = boundary condition correction factor for middle strip

k_{DP} = drop panel correction factor

k_r = correction factor used to compute long-time creep deflections

k_{SS} = slab strip moment intensity correction factor

k_{SS-col} = slab strip moment intensity correction factor for column strip

k_{SS-mid} = slab strip moment intensity correction factor for middle strip

k_{WF} = strip width correction factor

k_{WF-col} = strip width correction factor for column strip

k_{WF-mid} = strip width correction factor for middle strip

l = unbraced clear span

l_1 = long direction clear span

l_2 = short direction clear span

LL = live load (psf)

maximum load = maximum load (psf)

M_{cr} = cracking moment

M_x = x-direction moment calculated based upon uncracked analysis

M_y = y-direction moment calculated based upon uncracked analysis

R = load ratio

reshoring = number of levels of reshoring being used

SDL = superimposed dead load (psf)

stripping = number of days after slab casting at which formwork removal takes place

sustained load = sustained load (psf)

t = time since slab casting

t_i = time increment of interest

t_j = time increment of load application

t_{la} = time of load application, taken with respect to slab casting

w = distributed load

$w_{construction}$ = construction load as a uniform distributed load (lb/in)

$w_{maximum}$ = maximum load as a uniform distributed load (lb/in)

w_{LL} = live load as a uniform distributed load (lb/in)

w_{SDL} = superimposed dead load as a uniform distributed load (lb/in)

w_{slab} = slab dead load (self-weight not including formwork) as a uniform distributed load (lb/in)

$w_{sustained}$ = sustained load as a uniform distributed load (lb/in)

α = adjustment factor applied to gross moment of inertia to generate effective moment of inertia

α_c = adjustment factor applied to gross moment of inertia to generate effective moment of inertia for column strips

α_m = adjustment factor applied to gross moment of inertia to generate effective moment of inertia for middle strips

Δ = total deflection (mid-panel or mid-span)

Δ_c = column strip deflection

Δ_e = instantaneous (elastic) deflection

$\Delta_{e,cum.}$ = cumulative instantaneous (elastic) deflection

Δ_{cr} = long-time creep deflection

$\Delta_{cr,cum.}$ = cumulative long-time creep deflection

Δ_m = middle strip deflection

Δ_{mp} = mid-panel deflection

Δw = loading change

γ_α = creep correction factor to account for entrained air content

γ_λ = relative humidity creep correction factor

γ_ψ = creep correction factor used to account for the ratio of fine to course aggregate

γ_c = overall creep correction factor

γ_{conc} = unit weight of concrete

γ_{la} = loading age creep correction factor

γ_{S} = creep correction factor to account for slump

γ_{VS} = creep correction factor to account for volume to surface area ratio

λ = the long-time multiplier

λ_c = average creep multiplier

λ_u = ultimate creep coefficient

ρ' = reinforcement ratio

ψ = a constant defining the time-ratio component for a given member shape and size

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Accurate analysis of construction loads in floor slabs of multi-story high-rise concrete buildings is essential to understanding the structural effects created via the construction process. Construction loads, if not designed for properly, can lead to detrimental effects, such as slab cracking and excessive deflection. Because construction loads act upon a structure long before it has entered into the "service" phase of its life cycle, construction loads have the potential to significantly impact the service life of the structure.

Because the long-term effects of construction loads are significant, construction loading distribution and corresponding slab deflection computations are present topics of debate within the structural engineering community.

1.2 OBJECTIVE AND SCOPE

The objective of this study was to develop a simplified analysis procedure to evaluate the effects of construction loading on long-time deflections of two-way reinforced concrete floor slabs. The analysis procedure accounts for various shoring and reshoring scenarios. The objective was achieved within the following scope:

1. A literature review was conducted to determine the primary factors affecting long-time slab deflections including methods to account for construction loading.
2. A method of analysis suitable for incorporation in a spreadsheet or computer program was developed.
3. The analysis procedure was evaluated by comparing computed results with those obtained from slab tests available in the literature.
4. The effects of various parameters on the computed results were evaluated by conducting a parametric study.
5. Based on the results of the study recommendations were made for design and construction of two way slabs to ensure adequate control of long-time deflections. It is anticipated that the analysis procedure will be helpful to engineers and contractors evaluating proposed shoring and reshoring schemes.

1.3 THESIS LAYOUT

The thesis is divided into six chapters. Chapter 1 is the introduction, detailing the scope and objective of the thesis. Chapter 2 provides a review of literature on the topics of construction load analysis and two-way slab analysis.

Chapter 3 uses the methodology outlined in Chapter 2 to perform load ratio analyses for construction scenarios involving one level of shoring with two levels of reshoring, one level of shoring with three levels of reshoring, and one level of shoring with four levels of reshoring. Chapter 3 also contains a step-by-step outline of the computer algorithm written to compute the deflection-time history for two-way concrete floor slabs.

Chapter 4 provides verification of the computer algorithm by comparing algorithmic results to experimental results, while Chapter 5 provides a parametric study in which input parameters are varied one-by-one to determine parameter sensitivity. Chapter 6 provides overall conclusions and recommendations.

The main body of the report is followed by a list of references for all works referenced within the thesis. An appendix also accompanies the report. The appendix contains the text file for the C++ computer program used to perform the analysis procedures detailed within the main body of the report. The C++ computer program was written and implemented using Microsoft Visual Basic C++ Standard Edition 2003.

CHAPTER 2
LITERATURE REVIEW

2.1 INTRODUCTION

This chapter presents a review of literature on analysis procedures for computing load ratios and performing two-way slab analysis. Section 2.2 presents an analysis procedure used to compute load ratios for specific shoring/reshoring construction scenarios. Section 2.3 is devoted to early-age concrete properties, while section 2.4 details two-way slab analysis.

2.2 CONSTRUCTION LOAD ANALYSIS: THE SIMPLIFIED METHOD

When constructing multi-story high-rise buildings with reinforced concrete floor slabs, a step-by-step procedure is performed. Formwork and shoring is set up upon the most recently cast floor slab, and the next floor is cast. Typically, shoring is left in place for an extended period of time, leading to construction schemes involving two, three, and four levels of shoring. (Ghosh, 1997)

Grundy and Kabaila (1963) documented a methodology for computing construction loads on concrete floor slabs in multi-story high-rise buildings based upon the shoring scenario. The procedure has become known as the simplified method. More refined procedures (Liu et al. 1985 and Aguinaga-Zapata and Bazant, 1986) provide good correlation with the simplified method.

2.2.1 ASSUMPTIONS

A number of simplifying assumptions are necessary in order to perform a construction load analysis using the Grundy and Kabaila (1963) simplified method. The requisite assumptions are as follows:

1. The slabs behave elastically.
2. Initially, the slabs are supported by a rigid foundation.
3. Shoring is spaced close enough that shore reactions may be treated as a distributed load. Therefore, the shores supporting the slabs and formwork may be taken as a continuous uniform elastic support with a stiffness coefficient K (K being the load required to generate a unit displacement).
4. Because shores are rigid, all slabs connected by shores deflect identically. Any additional loading applied to the slab-shoring system is distributed amongst the slabs in the support network based upon their relative flexural stiffnesses.
5. The shoring stiffness, K, is infinite with respect to slab stiffness.

2.2.2 CONSTRUCTION LOAD DISTRIBUTION

The distribution of load amongst supporting slabs during the construction cycle can best be understood by example. Assuming a constant flexural stiffness within slabs (ignoring the effects of varying concrete compressive strength and modulus of elasticity) and a seven day casting cycle with formwork stripping at five days, the overall load distribution process assuming three levels of shores may be illustrated by Figure 2.1 (Grundy & Kabaila, 1963).

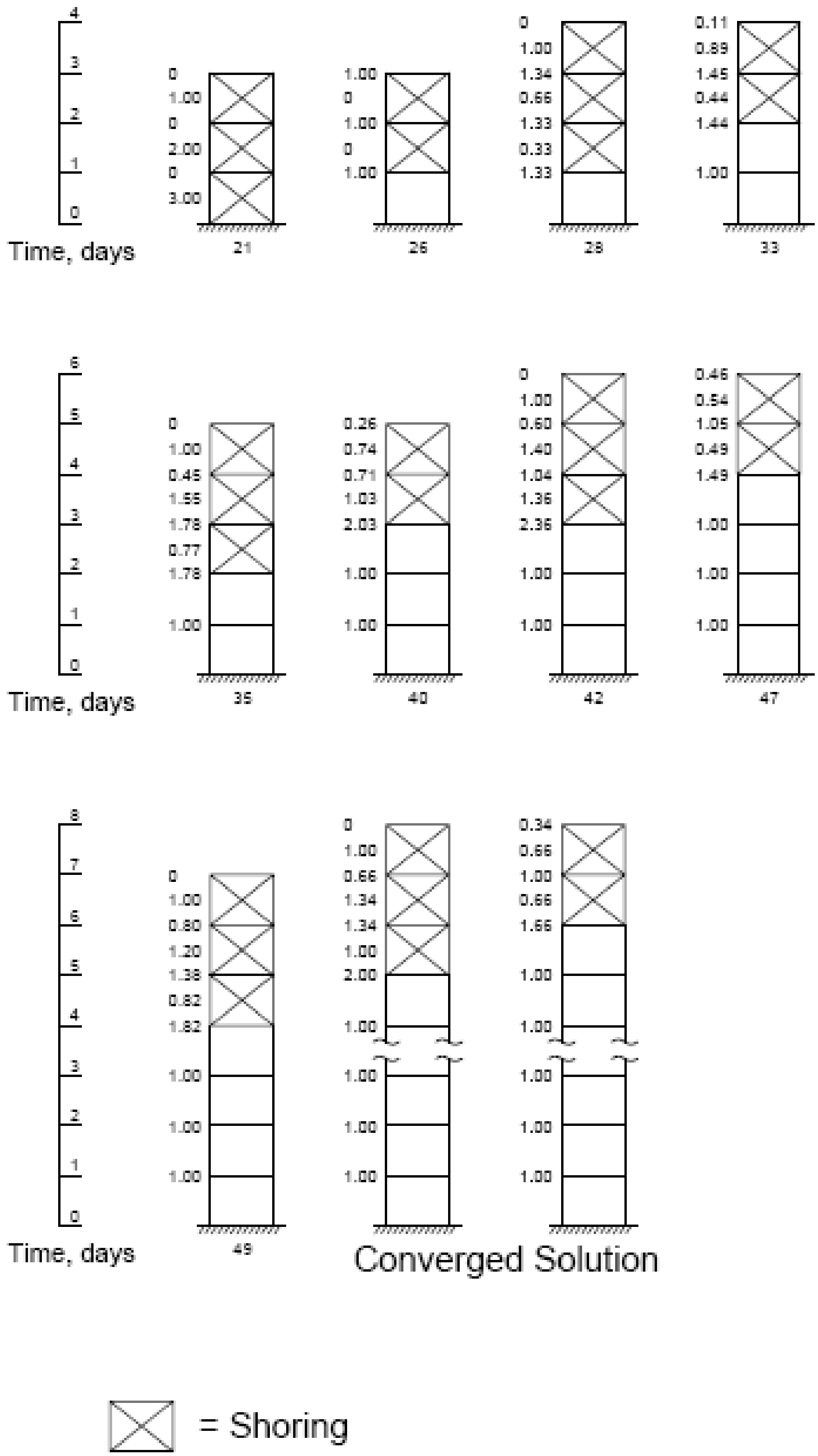

Figure 2.1 – Construction Load Distribution Using the Simplified Method (Grundy & Kabaila, 1963)

The numbers in Figure 2.1 are referred to as "load ratios." Construction loads carried by slabs and shoring are typically expressed in terms of load ratios. Load ratios are defined as follows:

$$R = \text{load carried by the slab / slab weight} \tag{2–1}$$

Load ratios may be multiplied by the total self-weight of a floor in order to yield the actual load magnitude.

Until the formwork gets off the ground, all loads are transferred through the shores into the rigid foundation. At 21 days the floor slabs at levels one, two, and three are supported from the ground by stiff shores, meaning that these slabs cannot deflect and do not carry any load. At 26 days the lowest level of shoring is removed, which enables all three slabs to deflect and carry their self-weight. Because the shores are assumed to be infinitely stiff with respect to slabs, the slab deflections will be identical. The removed shores are then placed upon the slab at level three in order to construct a new slab at level four. The weight of the fresh slab at level four is distributed amongst the slabs at levels one, two, and three in proportion to their relative flexural stiffnesses. Therefore, at 26 days the total force in the shoring (3.00) is distributed evenly between slabs at floor levels one, two, and three.

At 28 days the weight of the freshly cast level four slab (1.00) is distributed evenly among slabs at levels one, two, and three. The increase in load at each level of shoring reflects the magnitude of the load being transferred to the slab directly underneath. At 33 days the shoring above level one is removed. The force within this level of shoring is distributed evenly amongst slabs at levels two, three, and four. The distribution process continues in this manner until convergence has been reached.

2.2.3 ANALYTICAL RESULTS

Grundy and Kabaila (1963) performed loading distribution computations in precisely this manner for construction scenarios involving two levels of shoring, three levels of shoring, and four levels of shoring. The maximum loading ratio for two levels of shoring was found to be 2.25 at the overall 28-day mark. The maximum loading ratio for three levels of shoring was found to be 2.36 at the overall 42-day mark, while the maximum loading ratio for four levels of shoring was found to be 2.43 at the overall 56-day mark. The maximum loads are always carried by the last slab cast before the removal of ground-level shores. Although the magnitude of the maximum load ratio increases slightly with an increase in the number of levels of shoring, the age of the slab at the time at which it is supporting the maximum load ratio increases significantly with an increase in the number of levels of shoring.

Grundy and Kabaila (1963) performed identical analyses for slabs with varying flexural stiffnesses (E_c increasing with age). The procedure differs only in the distribution of load amongst slabs. Rather than distributing load evenly amongst all slabs connected with shoring, the load is distributed based upon relative flexural stiffnesses, which in turn are dependent upon E_c values for all slabs in question. The results obtained using varying flexural stiffnesses do not differ significantly from those obtained assuming constant flexural stiffnesses. Although it is certainly more realistic to work with varying flexural stiffnesses, constant flexural stiffnesses may be used with very minimal detriment.

Beresford (1964) performed analyses assuming finite rather than infinite shoring stiffnesses, K. The results did not change significantly. Sbarounis (1984) suggests that the maximum load ratios will be reduced by approximately 10% by incorporating the effects of concrete cracking into the computation of load distribution factors. It should be mentioned that both

Beresford (1964) and Sbarounis (1984) obtained maximum load ratios greater than 2.0.

2.2.4 CONSTRUCTION SCHEMES INVOLVING RESHORING

Due to economic considerations, reshores are typically utilized when employing construction schemes otherwise involving multiple levels of shoring. When this technique is used, multiple levels of shoring may be used in conjunction with multiple levels of reshoring (Ghosh, 1997). A typical construction cycle utilizing shoring and reshoring operates based upon the following sequence of steps (Stivaros & Halvorsen, 1992):

1. Shoring and formwork is installed upon the uppermost floor in preparation for the casting of the next floor.
2. The lowest level of reshores are removed.
3. The lowest level of shores are removed.
4. The reshores are inserted into the void created by the removal of the shores, making the reshores the uppermost level of reshores.

When shoring is removed from beneath a slab, the slab is allowed to deflect and carry its own weight. Reshores are then inserted in place of the shoring. Similar to shores, reshores allow the load of the next freshly cast floor to be distributed amongst all floors in the supporting assembly. The major difference between shoring and reshoring lies in the fact that reshores carry no load at the time of installation. (Ghosh, 1997)

2.3 EARLY AGE CONCRETE PROPERTIES

Structural failures may take two forms. Strength failures refer to the collapse of a structure or a structural member, while serviceability failures refer to the distress that causes a significant reduction in the functionality of the structure. Strength failures do not occur often but are typically catastrophic. Serviceability failures occur more frequently. Serviceability failures are typically not life threatening but are generally responsible for significant financial losses. Common examples of serviceability failures include excessive floor slab deflections or significant concrete cracking. (Ghosh, 1997)

The application of construction loads to structures at early age can contribute to the likelihood of both structural and serviceability failures. Early age concrete properties play a vital role in properly analyzing the effects created by the application of construction loads. Understanding strength-related consequences requires the consideration of concrete compressive strength. To properly consider serviceability-based consequences it is imperative to also assess concrete modulus of elasticity and creep. (Ghosh, 1997)

2.3.1 CONCRETE COMPRESSIVE STRENGTH

Temperature and moisture have a significant effect on the development of concrete strength (Price, 1951). Concrete cured at higher temperature develops greater 28-day strength. However, concrete cast and cured at high temperature has lower strength at later ages than concrete cured at low temperature. (Ghosh, 1997)

Klieger (1958) performed research on concrete test specimens in an effort to determine the effects of temperature on concrete strength. Klieger's research led him to the following set of conclusions:

1. At the 1-day, 3-day, and 7-day marks concrete compressive strength increases with an increase in initial temperature and curing temperature.
2. An increase in the initial and curing temperature of concrete corresponds to a considerably lower compressive strength at the 90-day and 1-year marks.
3. Compressive strength of concrete mixed using ASTM Types I, II, and III cements were influenced similarly by temperature effects. Differences were insignificant.
4. At specific degrees of hydration there may be an optimum curing temperature during the early life of concrete that leads to higher compressive strength at later ages. This temperature is dependent upon cement type. For Types I and II cements the optimum temperature appeared to be 55 degrees Fahrenheit, while the optimum temperature for Type III cement appeared to be 40 degrees Fahrenheit.
5. For concrete containing a calcium chloride additive, increases in compressive strength were proportionally greater at early ages and lower temperatures.

ACI Committee 209 (2008) recommends using the following equation for the computation of the time-dependent compressive strength of concrete:

$$f'_c(t) = [t / (4.0 + 0.85\, t)] * f'_{c(28\text{-day})} \qquad (2\text{–}2)$$

where,

t is the time since casting, measured in days

$f'_{c(28\text{-}day)}$ is the 28-day compressive strength of concrete

$f'_c(t)$ is the time-dependent compressive strength of concrete at the time increment in question

2.3.2 ELASTIC MODULUS

ACI Committee 209 (2008) recommends using the following equation for the computation of the time-dependent modulus of elasticity of concrete:

$$E_c(t) = 33 * \gamma_{conc}^{1.5} * f'_c(t)^{0.5} \qquad (2\text{–}3)$$

where,

γ_{conc} is the unit weight of concrete (pcf)

$f'_c(t)$ is the time-dependent compressive strength of concrete (psi)

$E_c(t)$ is the time-dependent modulus of elasticity of concrete (psi)

ACI 318-05 provides the following equation for computing the time-dependent modulus of elasticity of normal weight concrete:

$$E_c(t) = 57000*[f'_c(t)^{0.5}] \qquad (2\text{–}4)$$

where,

$f'_c(t)$ is the time-dependent compressive strength of concrete (psi)

$E_c(t)$ is the time-dependent modulus of elasticity of concrete (psi)

This equation is based on the previous equation, assuming a unit weight of concrete of approximately 144.0 pounds per cubic foot (pcf).

2.3.3 CREEP

The basic creep property of concrete is represented by a time-dependent creep coefficient. ACI Committee 209 recommends computing time-dependent creep coefficients based upon the following equation:

$$C(t_i, t_j) = \{(t_i - t_j)^{\psi} / [d + (t_i - t_j)^{\psi}]\} * C_u \qquad (2\text{–}5)$$

d and ψ are considered constants defining the time-ratio component for a given member shape and size. ψ is typically taken between 0.40 and 0.80, while d is typically taken between 6 and 30 days. ACI-209R-92 recommends using an average value of 0.6 for ψ and an average value of 10 days for d. The simplified equation to compute time-dependent creep coefficients appears as follows:

$$C(t_i, t_j) = \{(t_i - t_j)^{0.6} / [10 + (t_i - t_j)^{0.6}]\} * C_u \qquad (2\text{–}6)$$

where,

$C(t_i, t_j)$ is the time-dependent creep coefficient

$(t_i - t_j)$ is the time since load application, measured in days

C_u is the ultimate creep coefficient

Under standard conditions, ACI committee 209 recommends an average ultimate creep coefficient of 2.35. For non-standard conditions, ACI-209R-92 recommends multiplying the average ultimate creep coefficient by an overall creep correction factor comprised of six individual creep correction factors as follows:

$$C_u = 2.35 * \gamma_c \qquad (2\text{–}7)$$

γ_c collectively represents the effects of all six creep correction factors that may be included within creep computations. ACI-209R-92 suggests computing γ_c as follows:

$$\gamma_c = \gamma_{la} * \gamma_\lambda * \gamma_{VS} * \gamma_S * \gamma_\psi * \gamma_\alpha \qquad (2–8)$$

γ_{la}, γ_λ, γ_{VS}, γ_S, γ_ψ, γ_α are a series of creep correction factors used to modify the average ultimate creep coefficient. γ_{la} is the loading age correction factor, while γ_S is the correction factor to account for relative humidity. For the purposes of this report the other four correction factors will be ignored. γ_{VS} is the correction factor to account for the volume to surface area ratio, γ_S is the correction factor to account for slump, γ_ψ is the correction factor used to account for the ratio of fine to coarse aggregate, and γ_α is the correction factor to account for entrained air content. (ACI 209.2R-08)

The ACI Committee 209 recommended correction factor for loading age, γ_{la}, is computed as follows:

$$\gamma_{la} = 1.25(t_{la})^{-0.118} \qquad \text{(moist cured)} \qquad (2–9)$$

$$\gamma_{la} = 1.13(t_{la})^{-0.094} \qquad \text{(steam cured)} \qquad (2–10)$$

t_{la} is the time after slab casting of loading application measured in days. Ghosh (1997) recommends using the following similarly-formatted correction factor, citing better correlation with experimental results:

$$\gamma_{la} = 2.3(t_{la})^{-0.25} \qquad (2–11)$$

The correction factor for relative humidity, γ_λ, is computed as follows (ACI 209.2R-08):

$$\gamma_\lambda = 1.27 - 0.0067H \qquad (2–12)$$

H is the relative humidity expressed as a percent. When choosing to include this correction factor, H must be taken greater than 40%.

Thus, the final equation used to compute time-dependent creep coefficients appears as follows:

$$C(t_i, t_j) = \{(t_i - t_j)^{0.6} / [10 + (t_i - t_j)^{0.6}]\} * 2.35 * \gamma_{la} * \gamma_{\lambda} \quad (2\text{–}13)$$

2.4 TWO-WAY SLAB SERVICEABILITY ANALYSIS

2.4.1 LOADS

During the construction cycle, deflections are computed based upon construction loads. Construction loads include only the effects of self-weight and are computed as follows (ACI 435.9R-91):

$$w_{construction} = k_1 * k_2 * R * w_{slab} \quad (2\text{–}14)$$

where,

k_1 is the allowance for error in theoretical load ratio R

k_2 is the allowance for weight of formwork

R = (applied load)/(slab dead load) = load ratio calculated by Grundy and Kabaila procedure

w_{slab} is the slab dead load (self-weight not including formwork)

$w_{construction}$ is the construction load

Gardner (1985) recommends using $k_1 = 1.1$ and $k_2 = 1.1$.

Once the construction cycle is complete, loads are taken as sustained loads. Sustained loads include the effects of service loads such as a superimposed dead load and a fraction of the design live load (taken as 10% for the purposes of this study). Sustained loads are computed as follows:

$$w_{sustained} = R * (w_{slab} + w_{SDL} + 0.1w_{LL}) \quad (2\text{–}15)$$

where,
R = 1.0 for sustained loads since the construction cycle has ended
w_{SDL} is the superimposed dead load
w_{LL} is the live load
$w_{sustained}$ is the sustained load

After five years time the maximum additional live load is applied to the slab. The maximum load is computed using the full live load as follows:

$$w_{maximum} = R * (w_{slab} + w_{SDL} + w_{LL}) \quad (2\text{–}16)$$

$w_{maximum}$ is the maximum load and R is taken as 1.0 for maximum loads since the construction cycle has ended.

2.4.2 CROSSING BEAM METHOD

Analysis approaches have been developed in which the two-way slab system is broken down into orthogonal one-way systems. The one-way systems may then be analyzed using beam analysis, relying upon superposition to render the results applicable to two-way systems. Scanlon and Murray (1982) describe a crossing-beam method, in which column and middle strips are treated as beam elements. The middle strip in the short

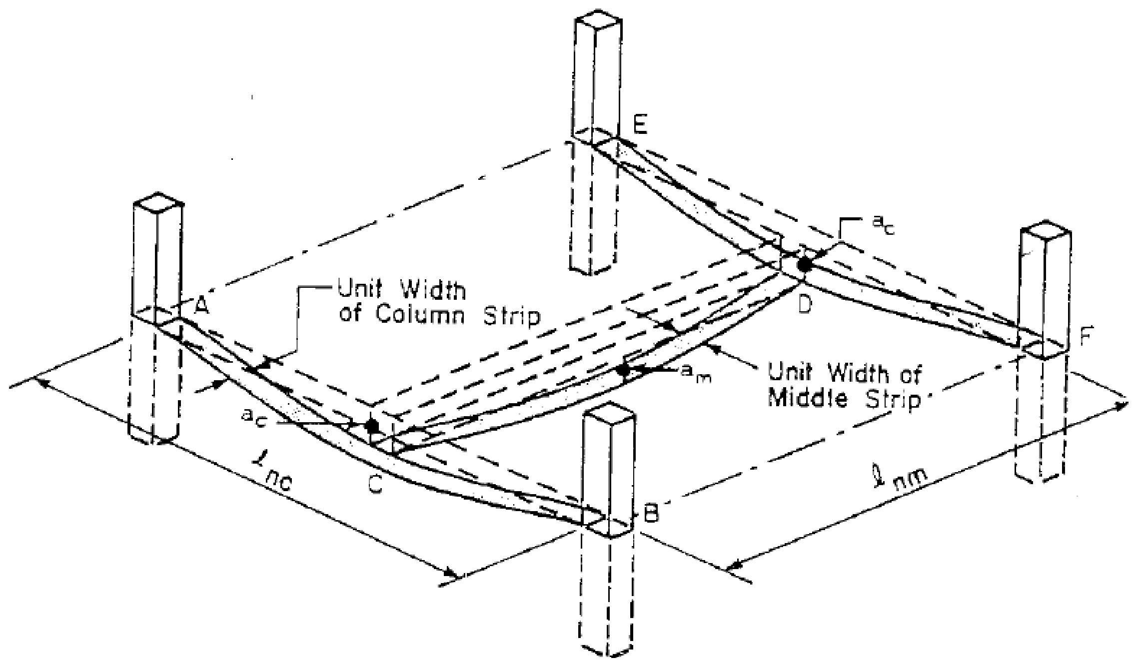

Figure 2.2 – Two-Way Slab Analysis Using the Crossing Beam Method

direction is supported by adjacent column strips running perpendicular in the long direction as shown in Figure 2.2. Computing the deflection at the center of the two-way slab requires the superposition of the column strip deflection and the perpendicular middle strip deflection:

$$\Delta_{mp} = \Delta_c + \Delta_m \tag{2–17}$$

where,

Δ_c is the column strip deflection

Δ_m is the middle strip deflection

Δ_{mp} is the mid-panel deflection

Chang and Hwang (1996) verify that the crossing-beam analogy accurately approximates two-way slab deflections.

2.4.3 BEAM DEFLECTION EQUATION

The standard beam-deflection equation used to compute the deflection at mid-span of a prismatic beam element is as follows:

$$\Delta = (k/384) * (w * l^4) / (E * I) \qquad (2\text{–}18)$$

where,

$k = k_{BC}$, a boundary condition correction factor

w is the distributed load applied over the entire clear span length

l is the clear span between the beam supports

E is the modulus of elasticity of the beam material

I is the moment of inertia of the beam cross-section

Δ is the beam deflection at mid-span

For the column strip or middle strip of a two-way slab, instantaneous (elastic) deflection caused by a loading increment $\Delta w(t)$ at time t is computed using the basic beam-deflection equation as follows:

$$\Delta_e(t) = k/384 * \Delta w(t) * l^4 / [E_c(t) * I_e] \qquad (2\text{–}19)$$

where,

t is the time since slab casting

k is the overall correction factor

$\Delta w(t)$ is the change in distributed load at time t

l is the clear span length of the slab strip

$E_c(t)$ is the modulus of elasticity of concrete at time t

I_e is the effective moment of inertia of the slab strip

$\Delta_e(t)$ is the instantaneous (elastic) deflection for the loading change taking place at time t

2.4.4 MODIFIED MOMENT OF INERTIA

A modification of the moment of inertia for the section may be included to account for the effects of cracking. The concept of an effective moment of inertia, I_e, was originally proposed by Branson (1963). Scanlon and Murray (1982) applied Branson's equation to orthogonal column and middle strips in the x and y directions, respectively:

$$I_{ex} = (M_{cr}/M_x)^3 * I_g + [1 - (M_{cr}/M_x)^3] * I_{crx} \qquad (2\text{–}20)$$

$$I_{ey} = (M_{cr}/M_y)^3 * I_g + [1 - (M_{cr}/M_y)^3] * I_{cry} \qquad (2\text{–}21)$$

where,

M_{cr} is the cracking moment

I_{crx} is the x-direction moment of inertia of the cracked section

I_{cry} is the y-direction moment of inertia of the cracked section

M_x is the x-direction moment calculated based upon an uncracked analysis

M_y is the y-direction moment calculated based upon an uncracked analysis

I_{ex} is the x-direction effective moment of inertia

I_{ey} is the y-direction effective moment of inertia

The effective moment of inertia, I_e, may alternatively be expressed in terms of the gross moment of inertia as αI_g. I_g for rectangular sections is $I_g = bh^3/12$. α depends upon the beam dimensions, reinforcement ratio, and the applied moment. For the purposes of the algorithm α is approximated as $\alpha_c = 0.4$ for column strips and approximated as $\alpha_m = 0.8$ for middle strips. Rangan (1976) suggests using the cracked moment of inertia for the column strip effective moment of inertia. Using $0.4*I_g$ for the column strip effective moment of inertia yields a value slightly higher than the cracked moment of inertia, accounting for the additional stiffness contribution provided by the concrete in tension.

For flat plates and slabs, the intensity of cracking in column strips is more severe than in middle strips (Rangan, 1976). The reinforcement ratio for column strips is typically greater than the reinforcement ratio for middle strips. The reinforcement ratio for two-way slabs generally falls between 0.2 and 0.6. The value of the column strip reinforcement ratio typically falls at the higher end of this range, while the value of the middle strip reinforcement ratio typically falls at the lower end of this range. Based on Scanlon and Bischoff (2008), using $\alpha_c = 0.4$ for column strips and $\alpha_m = 0.8$ for middle strips appears reasonable.

2.4.5 CORRECTION FACTORS

Computing two-way slab deflections requires the inclusion of a number of correction factors. The overall correction factor, k, is the product of all correction factors. Thus, k is computed using the following equation:

$$k = k_{BC} * k_{DP} * k_{WF} * k_{SS} \qquad (2\text{–}22)$$

where,

k_{BC} is the boundary condition factor

k_{DP} is the drop panel correction factor

k_{WF} is the strip width correction factor

k_{SS} is the slab strip moment intensity factor

k will differ for column and middle strips. Individual correction factors are described in the following subsections.

2.4.5.1 BOUNDARY CONDITION FACTOR

The boundary conditions of the slab strip affect the bending behavior of the strip. A simply supported strip, for example, deflects to a much greater extent than an equivalent continuous strip. To account for this discrepancy the following boundary condition factors are applied to the strip in question:

k_{BC} = 5.0 for a simply supported strip (2–23)

k_{BC} = 2.0 for a strip with one end continuous (2–24)

k_{BC} = 1.4 for a strip with both ends continuous (2–25)

k_{BC} = 1.0 for a fixed end strip (2–26)

It is important to note that the boundary conditions may very well differ for the column strip and the middle strip.

2.4.5.2 DROP PANEL CORRECTION FACTOR

If drop panels are present in flat slabs an additional factor is used to correct for the additional stiffness provided by the drop panels. ACI 318-05 requires drop panel thickness to be 1.25 times slab thickness. The average of the positive and negative region I_g values for the section containing drop panels becomes $1.35*I_g$, where I_g is based upon the slab thickness in the positive moment region of the column strip. Therefore, the moment of inertia is modified by a factor of 1.35 if drop panels are present. Because the moment of inertia appears in the denominator of deflection equations, this modification is implemented as a correction factor using $k_{DP} = (1 / 1.35)$.

2.4.5.3 STRIP WIDTH CORRECTION FACTOR

Despite the fact that the two-way system has been decomposed into orthogonal one-way systems, a correction factor is needed to accommodate the effects of two-way action. A strip width modification factor is included to account for the discrepancy between column strip width and middle strip width. Because the column strip spans the long direction and the middle strip spans the short direction, the middle strip will never be narrower than the column strip. The strip width modification factor for a column strip is unity, while the strip width modification factor for the middle strip is computed as follows:

$$k_{WF} = (l_1/2) / (l_1 - l_2/2) \quad (2\text{–}27)$$

where,

l_1 is the long span dimension

l_2 is the short span dimension

k_{WF} is the strip width modification factor

Therefore, the strip width modification factors are as follows:

$$k_{WF\text{-}col} = 1.0 \quad (2\text{–}28)$$

$$k_{WF\text{-}mid} = (l_1/2) / (l_1 - l_2/2) \quad (2\text{–}29)$$

2.4.5.4 SLAB STRIP MOMENT INTENSITY FACTOR

ACI 318-05 Chapter 13 provides factors for the distribution of total static moment between column and middle strips. For the interior panel of a flat plate without supporting beams, 75% of the negative moment and 60% of the positive moment are allotted to the column strip. Conversely, 25% of the negative moment and 40% of the positive moment are allotted to the middle strip (ACI 318-05). Using an average value yields the following moment magnification factors to account for the manner in which load is distributed between column and middle strips:

$$k_{SS\text{-}col} = 1.35 \quad (2\text{–}30)$$

$$k_{SS\text{-}mid} = 0.65 \quad (2\text{–}31)$$

Although these values are moment magnification factors, the algorithm operates based upon loading values rather than computed moment values. These slab strip intensity correction factors are applied directly to loading values.

2.4.6 LONG-TIME MUTIPLIERS

Long-time deflections may be approximated by applying a long-time multiplier to the computed instantaneous deflections. ACI 318-05 specifies the value of long-time multipliers to be computed as follows:

$$\lambda = 2/(1 + 50\rho') \quad (2\text{–}32)$$

where,

ρ' is the reinforcement ratio

λ is the long-time multiplier

The long-time multiplier is multiplied by the instantaneous (elastic) deflection to compute the additional long-time deflection.

Alternatively, long-time creep deflections may be computed using the following ACI Committee 209 equation (Graham & Scanlon, 1984):

$$\Delta_{cp} = [k_r * C(t_i, t_j)] * \Delta_e \tag{2–33}$$

where,

k_r is a correction factor accounting for a neutral axis shift, taken as 0.85

$C(t_i, t_j)$ is the time-dependent long-time creep coefficient

Δ_e is the instantaneous (elastic) deflection

Δ_{cp} is the long-time creep deflection

The quantity $[k_r * C(t_i, t_j)]$ in the preceding equation may be represented by the long-time multiplier, λ. Substituting using equations (2-6), (2-7), and (2-8) and substituting 0.85 for k_r leads to the following expression for the long-time multiplier:

$$\lambda(t_i, t_j) = \{(t_i - t_j)^{0.6} / [10 + (t_i - t_j)^{0.6}]\} * 2.0 * \gamma_{la} * \gamma_{\lambda} \tag{2–34}$$

where,

t_i is the time increment in question

t_j is the time of load application

2.0 is the average creep multiplier, $\lambda_c = 0.85 * 2.35$

γ_{la} is the creep correction factor for loading age

γ_{λ} is the creep correction factor for relative humidity

$\lambda(t_i, t_j)$ is the long-time multiplier and is a function of the time increment in question and the time of load application

2.4.7 CREEP RECOVERY FACTOR

Once loading is removed from a member (representing a negative loading change), instantaneous deflection recovery takes place. Instantaneous deflection recovery is followed by time-dependent creep recovery. Creep is not 100% recoverable, meaning that only a fraction of the total creep deformation is recovered via creep recovery. Residual deformation remains as a result of creep (Neville, 1981). The value of any long-time multiplier associated with a reduction in loading (a negative loading change) is fractionally reduced to account for the fact that creep recovery is incomplete. The fractional multiplier applied to long-time multipliers associated with loading reduction is known as the creep recovery factor. (Ghosh, 1997)

CHAPTER 3
METHOD OF ANALYSIS

3.1 INTRODUCTION

This chapter presents the algorithm used to compute the deflection-time history for two-ways slabs under a time-varying load history. The algorithm includes development of a loading-time history based upon a given construction procedure involving shoring and reshoring and the construction schedule. A prescribed loading-time history may alternatively be offered. Based on the loading-time history, time-dependent deflections are computed at mid-panel for a given slab panel.

3.2 LOADING-TIME HISTORY DURING SHORING/RESHORING

This section relies on the methodology of the Grundy and Kabaila (1963) simplified method to perform load ratio computations for construction scenarios involving one level of shoring with two levels of reshoring, one level of shoring with three levels of reshoring, and one level of shoring with four levels of reshoring. Each sub-section consists both of a timetable providing details of the construction procedures performed and a schematic diagram showing the specifics of the load ratios each step of the way. The analyses are based upon a construction scenario utilizing a seven day casting cycle with stripping at five days. However, the resulting load ratios (R-values) would be equally applicable for construction schemes utilizing alternative casting cycles and stripping times. The analyses are carried out until the point of convergence. The results of the analyses are used to generate the array of load ratios (R-values) to be used in the computer algorithm analysis procedure presented in section 3.2. The step-by-step process of computing load ratios due to the shoring/reshoring procedure is outlined in the following sub-sections.

3.2.1 ONE LEVEL OF SHORING WITH TWO LEVELS OF RESHORING

Table 3.1 shows the construction procedures and load sharing associated with a construction scheme utilizing one level of shoring in conjunction with two levels of reshoring. Figure 3.1 shows the distribution of construction load amongst slabs, shores, and reshores for this construction scheme. Construction load distribution is based upon the Grundy and Kabaila (1963) simplified method.

Table 3.1 – Construction Procedures and Load Sharing for One Level of Shoring with Two Levels of Reshoring

Time (days)	Operation	Load Sharing
0	shore 1st	fresh slab cannot support its weight; shores support the load and transfer to the foundation
5	remove 1st shores	slab supports its weight
5	reshore 1st	slab supports its weight; reshores carry no load
7	shore 2nd	fresh slab cannot support its weight; 1st floor slab supports its weight; shores transfer load to reshores then to foundation
12	remove 2nd shores	1st and 2nd floor slabs support their own weight; reshores carry no load

12	reshore 2nd	1st and 2nd floor slabs support their own weight; reshores carry no load
14	shore 3rd	fresh slab cannot support its weight; 1st and 2nd floor slabs support their own weight; shores transfer load to reshores then to foundation
19	remove 3rd shores	1st, 2nd, and 3rd floor slabs support their own weight; reshores carry no load
19	remove 1st reshores	1st, 2nd, and 3rd floor slabs support their own weight; reshores carry no load
19	reshore 3rd	1st, 2nd, and 3rd floor slabs support their own weight; reshores carry no load
21	shore 4th	fresh slab cannot support its weight; the load is evenly distributed amongst the 1st, 2nd and 3rd floor slabs; shores carry full load to 3rd floor; 3rd floor reshores carry 2/3 of load to 2nd floor; 2nd floor reshores carry 1/3 of load to 1st floor
26	remove 4th shores	all slabs support their own weight; reshores carry no load
26	remove 2nd reshores	all slabs support their own weight; reshores carry no load
26	reshore 4th	all slabs support their own weight; reshores carry no load
28	shore 5th	fresh slab cannot support its weight; the load is evenly distributed amongst the 2nd, 3rd, and 4th floor slabs; shores carry full load to 4th floor; 4th floor reshores carry 2/3 of load to 3rd floor; 3rd floor reshores carry 1/3 of load to 2nd floor

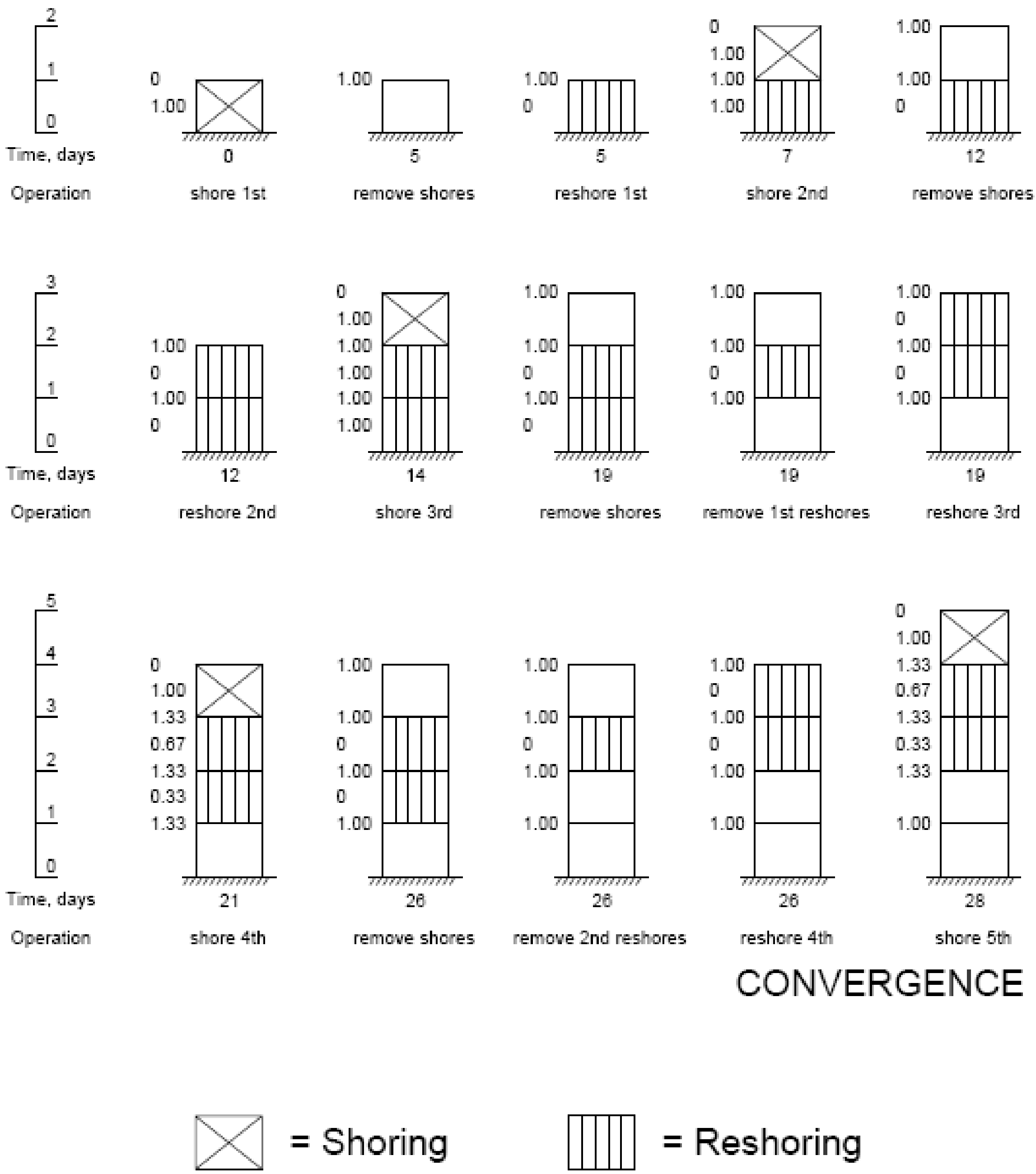

Figure 3.1 – Construction Load Distribution for One Level of Shoring with Two Levels of Reshoring

3.2.2 ONE LEVEL OF SHORING WITH THREE LEVELS OF RESHORING

Table 3.2 shows the construction procedures and load sharing associated with a construction scheme utilizing one level of shoring in conjunction with three levels of reshoring. Figure 3.2 shows the distribution of construction load amongst slabs, shores, and reshores for this construction scheme. Construction load distribution is based upon the Grundy and Kabaila (1963) simplified method.

Table 3.2 – Construction Procedures and Load Sharing for One Level of Shoring with Three Levels of Reshoring

Time (days)	Operation	Load Sharing
0	shore 1st	fresh slab cannot support its weight; shores support the load and transfer to the foundation
5	remove 1st shores	slab supports its weight
5	reshore 1st	slab supports its weight; reshores carry no load
7	shore 2nd	fresh slab cannot support its weight; 1st floor slab supports its weight; shores transfer load to reshores then to foundation
12	remove 2nd shores	1st and 2nd floor slabs support their own weight; reshores carry no load
12	reshore 2nd	1st and 2nd floor slabs support their own weight; reshores carry no load
14	shore 3rd	fresh slab cannot support its weight; 1st and 2nd floor slabs support their own weight; shores transfer load to reshores then to foundation
19	remove 3rd shores	1st, 2nd, and 3rd floor slabs support their own weight; reshores carry no load

19	reshore 3rd	1st, 2nd, and 3rd floor slabs support their own weight; reshores carry no load
21	shore 4th	fresh slab cannot support its weight; 1st, 2nd, and 3rd floor slabs support their own weight; shores transfer load to reshores then to foundation
26	remove 4th shores	all slabs support their own weight; reshores carry no load
26	remove 1st reshores	all slabs support their own weight; reshores carry no load
26	reshore 4th	all slabs support their own weight; reshores carry no load
28	shore 5th	fresh slab cannot support its weight; The load is evenly distributed amongst the 1st, 2nd, 3rd, and 4th floor slabs; shores carry full load to 4th floor; 4th floor reshores carry 3/4 of load to 3rd floor; 3rd floor reshores carry 1/2 of load to 2nd floor; 2nd floor reshores carry 1/4 of load to 1st floor
33	remove 5th shores	all slabs support their own weight; reshores carry no load
33	remove 2nd reshores	all slabs support their own weight; reshores carry no load
33	reshore 5th	all slabs support their own weight; reshores carry no load
35	shore 6th	fresh slab cannot support its weight; The load is evenly distributed amongst the 2nd, 3rd, 4th, and 5th floor slabs; shores carry full load to 5th floor; 5th floor reshores carry 3/4 of load to 4th floor; 4th floor reshores carry 1/2 of load to 3rd floor; 3rd floor reshores carry 1/4 of load to 2nd floor

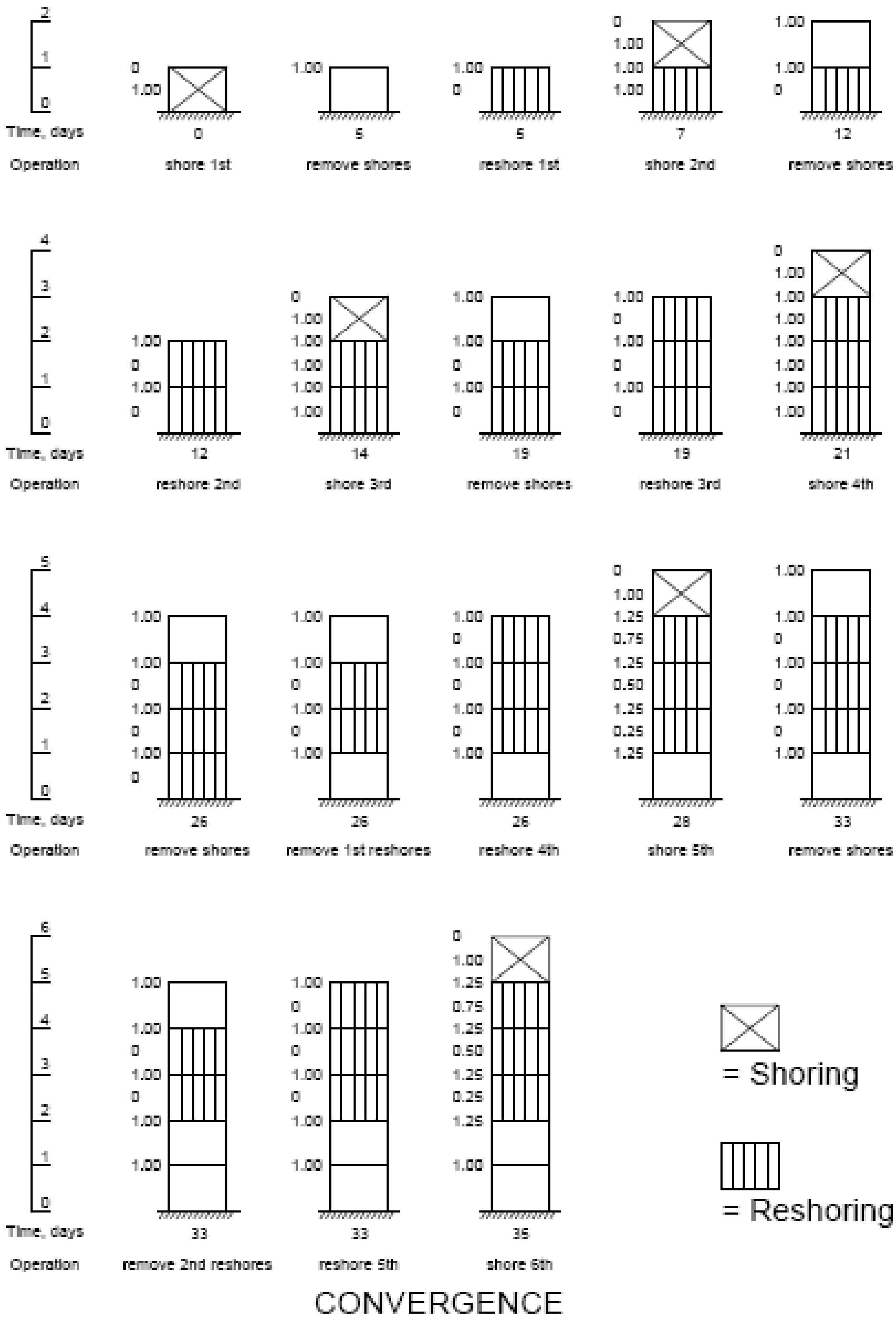

Figure 3.2 – Construction Load Distribution for One Level of Shoring with Three Levels of Reshoring

3.2.3 ONE LEVEL OF SHORING WITH FOUR LEVELS OF RESHORING

Table 3.3 shows the construction procedures and load sharing associated with a construction scheme utilizing one level of shoring in conjunction with four levels of reshoring. Figure 3.3 shows the distribution of construction load amongst slabs, shores, and reshores for this construction scheme. Construction load distribution is based upon the Grundy and Kabaila (1963) simplified method.

Table 3.3 – Construction Procedures and Load Sharing for One Level of Shoring with Four Levels of Reshoring

Time (days)	Operation	Load Sharing
0	shore 1st	Fresh slab cannot support its weight; shores support the load and transfer to the foundation
5	remove 1st shores	Slab supports its weight
5	reshore 1st	Slab supports its weight; reshores carry no load
7	shore 2nd	fresh slab cannot support its weight; 1st floor slab supports its weight; shores transfer load to reshores then to foundation
12	remove 2nd shores	1st and 2nd floor slabs support their own weight; reshores carry no load
12	reshore 2nd	1st and 2nd floor slabs support their own weight; reshores carry no load
14	shore 3rd	fresh slab cannot support its weight; 1st and 2nd floor slabs support their own weight; shores transfer load to reshores then to foundation
19	remove 3rd shores	1st, 2nd, and 3rd floor slabs support their own weight; reshores carry no load
19	reshore 3rd	1st, 2nd, and 3rd floor slabs support their own weight; reshores carry no load

21	shore 4th	fresh slab cannot support its weight; 1st, 2nd, and 3rd floor slabs support their own weight; shores transfer load to reshores then to foundation
26	remove 4th shores	all slabs support their own weight; reshores carry no load
26	reshore 4th	all slabs support their own weight; reshores carry no load
28	shore 5th	fresh slab cannot support its weight; 1st, 2nd, 3rd, and 4th floor slabs support their own weight; shores transfer load to reshores then to foundation
33	remove 5th shores	all slabs support their own weight; reshores carry no load
33	remove 1st reshores	all slabs support their own weight; reshores carry no load
33	reshore 5th	all slabs support their own weight; reshores carry no load
35	shore 6th	fresh slab cannot support its weight; The load is evenly distributed amongst the 1st, 2nd, 3rd, 4th, and 5th floor slabs; shores carry full load to 5th floor; 5th floor reshores carry 4/5 of load to 4th floor; 4th floor reshores carry 3/5 of load to 3rd floor; 3rd floor reshores carry 2/5 of load to 2nd floor; 2nd floor reshores carry 1/5 of load to 1st floor
40	remove 6th shores	all slabs support their own weight; reshores carry no load
40	remove 2nd reshores	all slabs support their own weight; reshores carry no load
40	reshore 6th	all slabs support their own weight; reshores carry no load
42	shore 7th	fresh slab cannot support its weight; The load is evenly distributed amongst the 2nd, 3rd, 4th, 5th, and 6th floor slabs; shores carry full load to 6th floor; 6th floor reshores carry 4/5 of load to 5th floor; 5th floor reshores carry 3/5 of load to 4th floor; 4th floor reshores carry 2/5 of load to 3rd floor; 3rd floor reshores carry 1/5 of load to 2nd floor

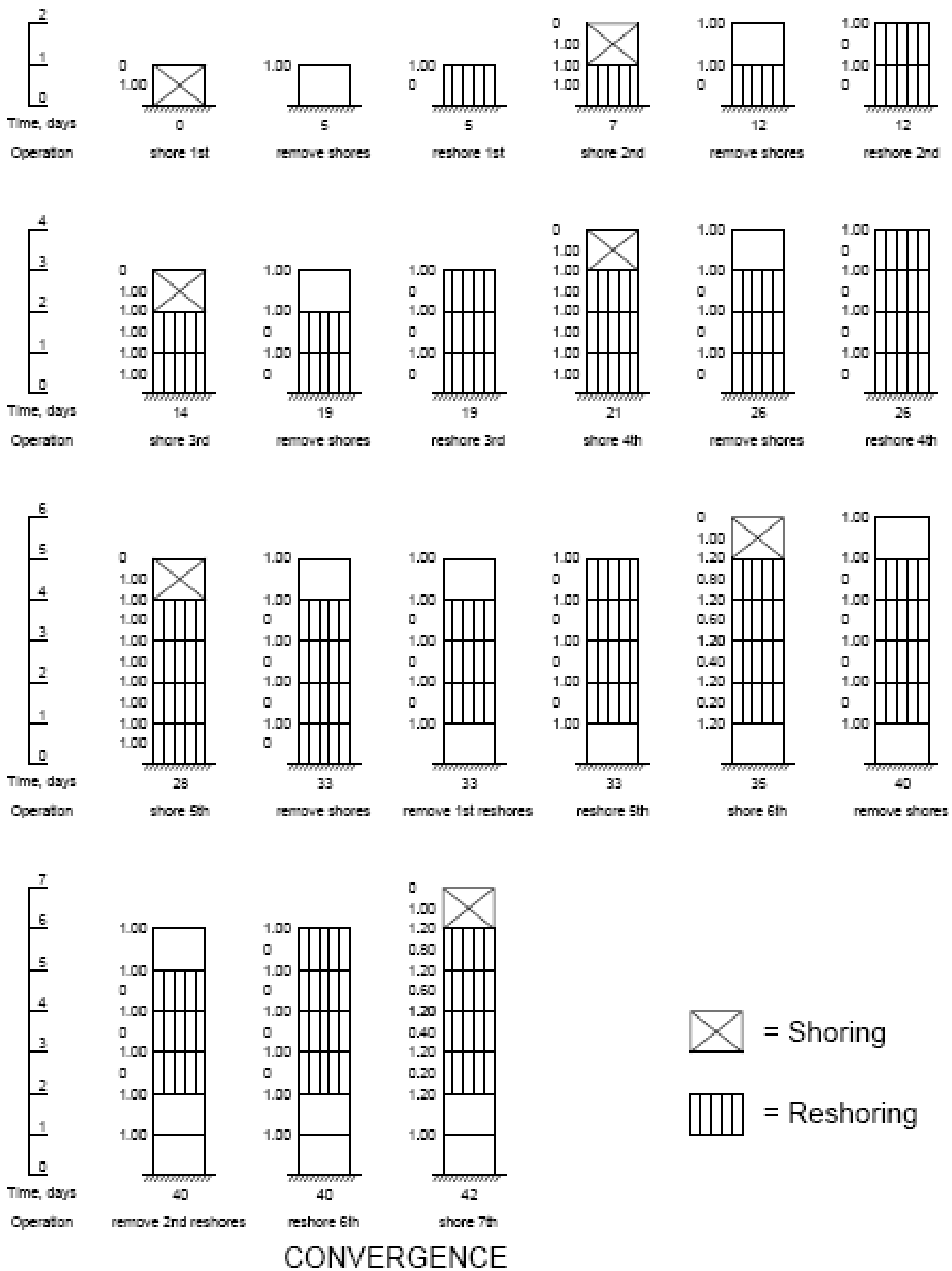

Figure 3.3 – Construction Load Distribution for One Level of Shoring with Four Levels of Reshoring

3.3 LONG-TIME DEFLECTIONS

When focusing on a specific floor slab in question, long-time multipliers are computed at as many time intervals as desired for each application (or change) of loading. The cumulative effects of all loading changes determine the overall deflection-time history of the slab in question.

Each individual long-time multiplier produces an associated time-dependent long-time creep deflection as follows:

$$\Delta_{cr}(t_i, t_j) = \Delta w(t_j) * \lambda(t_i, t_j) * (k/384) * l^4 / [E_c(t_j) * I_e] \qquad (3\text{–}1)$$

where,

$\Delta w(t_j)$ is the change in loading at time t_j

$\lambda(t_i, t_j)$ is the time-dependent long-time multiplier at time t_i for a loading change taking place at time t_j

k is the overall correction factor

l is the unbraced length

$E_c(t_j)$ is the concrete modulus of elasticity at time t_j

I_e is the effective moment of inertia of the slab strip

$\Delta_{cr}(t_i, t_j)$ is the time-dependent long-time creep deflection for the slab strip at time t_i for a loading change taking place at time t_j

Based on a generalization of the principle of creep superposition (McHenry, 1943) the cumulative time-dependent long-time creep deflection, $\Delta_{cr,cum.}(t_i)$, for the slab strip at time t_i for the series of all loading changes, $\sum^{i}_{j=0}\Delta w(t_j)$, taking place prior to and including those at time t_i can be computed as follows:

$$\Delta_{cr,cum}(t_i) = \sum^{i}_{j=0}\{\Delta w(t_j) * \lambda(t_i, t_j) * (k/384) * l^4 / [E_c(t_j) * I_e]\} \qquad (3\text{–}2)$$

The previous equation (3–2) excludes the effects of instantaneous deflections. For a specific loading change, $\Delta w(t_j)$, taking place at time t_j, the deflection taking place at a later time increment of interest, t_i, is computed by adding the initial (elastic) deflection, $\Delta_e(t_j)$, at time t_j to the long-term deflection, $\Delta_{cr}(t_i, t_j)$, at time t_i caused by the loading change at time t_j. The total (instantaneous + creep) time-dependent deflection, $\Delta(t_i)$, for the slab strip at time t_i for the series of all loading changes, $\sum^{i}_{j=0}\Delta w(t_j)$, taking place prior to and including those at time t_i can be computed as follows:

$$\Delta(t_i) = \sum^{i}_{j=0}\{\Delta w(t_j) * [1 + \lambda(t_i, t_j)] * (k/384) * l^4 / [E_c(t_j) * I_e]\} \qquad (3\text{–}3)$$

3.4 STEP-BY-STEP OUTLINE OF THE COMPUTER ALGORITHM

Loading-Time History:

Step 1: (USER INPUT) Determine whether the program will utilize a deflection-time history input by the user or whether the program will utilize a deflection-time history computed based upon user input regarding shoring/reshoring and the construction scheme.

Time-Independent Concrete Properties:

Step 2: (USER INPUT) Specify the 28-day compressive strength of concrete, $f'_{c(28\text{-day})}$. This value should be input in units of psi.

Slab Dimensions:

Step 3: (USER INPUT) Specify the slab thickness, h. This value should be input in units of inches.

Step 4: (USER INPUT) Specify the long direction clear span dimension, l_1. This dimension should be the clear span length between supports and should be input in units of feet.

Step 5: (USER INPUT) Specify the short direction clear span dimension, l_2. This dimension should be the clear span length between supports and should be input in units of feet.

Step 6: Compute the long span width, $b_1 = l_2/2$.

Step 7: Compute the short span width, $b_2 = l_1 – l_2/2$.

Cross-Sectional Properties:

Step 8: Compute the gross moment of inertia for the column strip, $I_{g\text{-}col} = 1/12 * (b_1 * 12) * h^3$. Note that it is also possible to compute the moment of inertia on a per foot basis. For the purposes of this algorithm, however, all moment of inertia computations will be based on the width of the strip.

Step 9: Compute the gross moment of inertia for the middle strip, $I_{g\text{-}mid} = 1/12 * (b_2 * 12) * h^3$.

Step 10: Compute the effective moment of inertia, $I_{e\text{-}col} = \alpha I_{g\text{-}col}$, for the column strip. $\alpha = 0.4$ for a column strip.

Step 11: Compute the effective moment of inertia, $I_{e\text{-}mid} = \alpha I_{g\text{-}mid}$, for the middle strip. $\alpha = 0.8$ for a middle strip.

Correction Factors:

Step 12: (USER INPUT) Specify the boundary condition factor for the column strip, $k_{BC\text{-}col}$. The boundary condition factor will be 5.0 for a simply supported strip, 2.0 for a strip with one end continuous, 1.4 for a strip with both ends continuous, and 1.0 for a fixed end strip.

Step 13: (USER INPUT) Specify the boundary condition factor for the middle strip, $k_{BC\text{-}mid}$.

Step 14: (USER INPUT) Determine whether drop panels are present.

Step 15: $k_{DP} = (1/\ 1.35)$ if drop panels are present. $k_{DP} = 1.0$ if drop panels are not present.

Step 16: $k_{WF} = 1.0$ for a column strip.

Step 17: $k_{SS} = 1.35$ for a column strip.

Step 18: Compute the correction factor for the column strip, $k = k_{BC\text{-}col} * k_{DP} * k_{WF} * k_{SS}$.

If the user chooses to input information regarding reshoring and the casting cycle rather than choosing to input information regarding the loading-time history, then follow steps A1 through A15 and ignore steps B1 through B3:

Loads:

Step A1: (USER INPUT) Specify the superimposed dead load, SDL, applied to the slab. This value should be input in units of psf.

Step A2: (USER INPUT) Specify the live load, LL, applied to the slab. This value should be input in units of psf.

Step A3: (USER INPUT) Specify the unit weight of concrete, γ_{conc}. This value should be input in units of pcf.

Step A4: Compute the self-weight dead load of the concrete in units of psf, DL = γ_{conc} * (h / 12).

Step A5: Compute the sustained load applied to the slab,
sustained load = DL + SDL + (0.1 * LL).

Step A6: Compute the maximum load applied to the slab,
maximum load = DL + SDL + LL.

Step A7: Compute the distributed construction load on the column strip in units of pounds per linear inch, $w_{construction}$ = (DL * b_1) / 12.

Step A8: Compute the distributed sustained load on the column strip in units of pounds per linear inch, $w_{sustained}$ = (sustained load * b_1) / 12.

Step A9: Compute the distributed maximum load on the column strip in units of pounds per linear inch, $w_{maximum}$ = (maximum load * b_1) / 12.

Construction Scheme:

Step A10: (USER INPUT) Specify the number of levels of reshoring being used.

Step A11: (USER INPUT) Specify the casting cycle in days.

Step A12: (USER INPUT) Specify the number of days after casting at which stripping takes place.

Load Ratios & Loading-Time History:

Step A13: Set 25 pairings of time and R values. For two levels of reshoring use Table 3.4. For three levels of reshoring use Table 3.5, and for four levels of reshoring use Table 3.6. These tables are based upon information obtained from Figure 3.1, Figure 3.2, and Figure 3.3.

Table 3.4 – Time Values and Load Ratios for One Level of Shoring with Two Levels of Reshoring

Day[0] =	0	R[0] =	0.0
Day[1] =	Stripping	R[1] =	1.0
Day[2] =	Cycle	R[2] =	1.0
Day[3] =	Cycle	R[3] =	1.33
Day[4] =	Cycle + stripping	R[4] =	1.33
Day[5] =	Cycle + stripping	R[5] =	1.0
Day[6] =	2*cycle	R[6] =	1.0
Day[7] =	2*cycle	R[7] =	1.33
Day[8] =	2*cycle + stripping	R[8] =	1.33
Day[9] =	2*cycle + stripping	R[9] =	1.0
Day[10] =	3*cycle	R[10] =	1.0
Day[11] =	3*cycle	R[11] =	1.33
Day[12] =	3*cycle + stripping	R[12] =	1.33
Day[13] =	3*cycle + stripping	R[13] =	1.0
Day[14] =	round Day[13] up to the nearest (or next) multiple of 30	R[14] =	1.0
Day[15] =	Day[14] + 30	R[15] =	1.0
Day[16] =	Day[14] + 60	R[16] =	1.0
Day[17] =	Day[14] + 90	R[17] =	1.0
Day[18] =	Day[14] + 120	R[18] =	1.0
Day[19] =	365	R[19] =	1.0
Day[20] =	730	R[20] =	1.0
Day[21] =	1095	R[21] =	1.0
Day[22] =	1460	R[22] =	1.0
Day[23] =	1825	R[23] =	1.0
Day[24] =	1825	R[24] =	1.0

Table 3.5 – Time Values and Load Ratios for One Level of Shoring with Three Levels of Reshoring

Day[0] =	0	R[0] =	0.0
Day[1] =	Stripping	R[1] =	1.0
Day[2] =	Cycle	R[2] =	1.0
Day[3] =	Cycle	R[3] =	1.25
Day[4] =	Cycle + stripping	R[4] =	1.25
Day[5] =	Cycle + stripping	R[5] =	1.0
Day[6] =	2*cycle	R[6] =	1.0
Day[7] =	2*cycle	R[7] =	1.25
Day[8] =	2*cycle + stripping	R[8] =	1.25
Day[9] =	2*cycle + stripping	R[9] =	1.0
Day[10] =	3*cycle	R[10] =	1.0
Day[11] =	3*cycle	R[11] =	1.25
Day[12] =	3*cycle + stripping	R[12] =	1.25
Day[13] =	3*cycle + stripping	R[13] =	1.0
Day[14] =	4*cycle	R[14] =	1.0
Day[15] =	4*cycle	R[15] =	1.25
Day[16] =	4*cycle + stripping	R[16] =	1.25
Day[17] =	4*cycle + stripping	R[17] =	1.0
Day[18] =	round Day[17] up to nearest (or next) multiple of 40	R[18] =	1.0
Day[19] =	Day[18] + 40	R[19] =	1.0
Day[20] =	Day[19] + 40	R[20] =	1.0
Day[21] =	365	R[21] =	1.0
Day[22] =	1095	R[22] =	1.0
Day[23] =	1825	R[23] =	1.0
Day[24] =	1825	R[24] =	1.0

Table 3.6 – Time Values and Load Ratios for One Level of Shoring with Four Levels of Reshoring

Day[0] = 0	R[0] = 0.0
Day[1] = Stripping	R[1] = 1.0
Day[2] = Cycle	R[2] = 1.0
Day[3] = Cycle	R[3] = 1.20
Day[4] = Cycle + stripping	R[4] = 1.20
Day[5] = Cycle + stripping	R[5] = 1.0
Day[6] = 2*cycle	R[6] = 1.0
Day[7] = 2*cycle	R[7] = 1.20
Day[8] = 2*cycle + stripping	R[8] = 1.20
Day[9] = 2*cycle + stripping	R[9] = 1.0
Day[10] = 3*cycle	R[10] = 1.0
Day[11] = 3*cycle	R[11] = 1.20
Day[12] = 3*cycle + stripping	R[12] = 1.20
Day[13] = 3*cycle + stripping	R[13] = 1.0
Day[14] = 4*cycle	R[14] = 1.0
Day[15] = 4*cycle	R[15] = 1.20
Day[16] = 4*cycle + stripping	R[16] = 1.20
Day[17] = 4*cycle + stripping	R[17] = 1.0
Day[18] = 5*cycle	R[18] = 1.0
Day[19] = 5*cycle	R[19] = 1.20
Day[20] = 5*cycle + stripping	R[20] = 1.20
Day[21] = 5*cycle + stripping	R[21] = 1.0
Day[22] = 365	R[22] = 1.0
Day[23] = 1825	R[23] = 1.0
Day[24] = 1825	R[24] = 1.0

Step A14: At each time increment of interest, compute all 25 effective loading values. Select $w_{construction}$, $w_{sustained}$, or $w_{maximum}$ based upon Table 3.7. For $w_{construction}$ use $w = 1.1 * 1.1 * R * w_{construction}$ to compute effective loading values. For $w_{sustained}$ use $w = R * w_{sustained}$ to compute effective loading values, and for $w_{maximum}$ use $w = R * w_{maximum}$ to compute effective loading values.

Table 3.7 – Selection of $w_{construction}$, $w_{sustained}$, or $w_{maximum}$ for Computation of Effective Loading Values

If reshoring =2

Day[0]	$w_{construction}$
Day[1]	$w_{construction}$
Day[2]	$w_{construction}$
Day[3]	$w_{construction}$
Day[4]	$w_{construction}$
Day[5]	$w_{construction}$
Day[6]	$w_{construction}$
Day[7]	$w_{construction}$
Day[8]	$w_{construction}$
Day[9]	$w_{construction}$
Day[10]	$w_{construction}$
Day[11]	$w_{construction}$
Day[12]	$w_{construction}$
Day[13]	$w_{sustained}$
Day[14]	$w_{sustained}$
Day[15]	$w_{sustained}$
Day[16]	$w_{sustained}$
Day[17]	$w_{sustained}$
Day[18]	$w_{sustained}$
Day[19]	$w_{sustained}$
Day[20]	$w_{sustained}$
Day[21]	$w_{sustained}$
Day[22]	$w_{sustained}$
Day[23]	$w_{sustained}$
Day[24]	$w_{maximum}$

If reshoring = 3

Day[0]	$w_{construction}$
Day[1]	$w_{construction}$
Day[2]	$w_{construction}$
Day[3]	$w_{construction}$
Day[4]	$w_{construction}$
Day[5]	$w_{construction}$
Day[6]	$w_{construction}$
Day[7]	$w_{construction}$
Day[8]	$w_{construction}$
Day[9]	$w_{construction}$
Day[10]	$w_{construction}$
Day[11]	$w_{construction}$
Day[12]	$w_{construction}$
Day[13]	$w_{construction}$
Day[14]	$w_{construction}$
Day[15]	$w_{construction}$
Day[16]	$w_{construction}$
Day[17]	$w_{sustained}$
Day[18]	$w_{sustained}$
Day[19]	$w_{sustained}$
Day[20]	$w_{sustained}$
Day[21]	$w_{sustained}$
Day[22]	$w_{sustained}$
Day[23]	$w_{sustained}$
Day[24]	$w_{maximum}$

If reshoring = 4

Day[0]	$w_{construction}$
Day[1]	$w_{construction}$
Day[2]	$w_{construction}$
Day[3]	$w_{construction}$
Day[4]	$w_{construction}$
Day[5]	$w_{construction}$
Day[6]	$w_{construction}$
Day[7]	$w_{construction}$
Day[8]	$w_{construction}$
Day[9]	$w_{construction}$
Day[10]	$w_{construction}$
Day[11]	$w_{construction}$
Day[12]	$w_{construction}$
Day[13]	$w_{construction}$
Day[14]	$w_{construction}$
Day[15]	$w_{construction}$
Day[16]	$w_{construction}$
Day[17]	$w_{construction}$
Day[18]	$w_{construction}$
Day[19]	$w_{construction}$
Day[20]	$w_{construction}$
Day[21]	$w_{sustained}$
Day[22]	$w_{sustained}$
Day[23]	$w_{sustained}$
Day[24]	$w_{maximum}$

Step A15: Compute the loading change between time increments, $\Delta w(t_i) = w_i - w_{i-1}$. In the form of computer program arrays, the formula appears as follows: $\Delta w[0] = 0$, and $\Delta w[x] = w[x] - w[x-1]$ for integers $x = 1$ to 24.

If the user chooses to input information regarding the loading-time history rather than choosing to input information regarding reshoring and the casting cycle, then follow steps B1 through B3 and ignore steps A1 through A15:

Prescribed Loading-Time History:

Step B1: (USER INPUT) Specify the loading-time history in the form of 25 loading-time pairings. Loading will be input in psf and time will be input in days.

Step B2: At each time increment compute the distributed load on the column strip in units of pounds per linear inch, w = (input loading * b_1) / 12.

Step B3: Compute the loading change between time increments, $\Delta w(t_i) = w_i - w_{i-1}$. In the form of computer program arrays, the formula appears as follows: Δw[0] = 0, and Δw[x] = w[x] – w[x – 1] for integers x = 1 to 24.

Time-Dependent Concrete Properties:

Step 19: At each time increment compute the compressive strength of concrete as a function of time. $f'_c(t_i) = f'_{c(28\text{-day})} * t_i / [4.00 + (0.85 * t_i)]$, where t_i is in days and compressive strength is in units of psi. In the form of computer program arrays, the formula appears as follows: $f'_c[x] = f'_{c(28\text{-day})} * day[x] / [4.00 + (0.85 * day[x])]$ for integers x = 0 to 24.

Step 20: At each time increment compute the modulus of elasticity of concrete as a function of time. $E_c(t_i) = 57000 * ([f'_c(t_i)])^{0.5}$, where compressive strength and modulus are in units of psi. In the form of computer program arrays, the formula appears as follows: $E_c[x] = 57000 * (f'_c[x])^{0.5}$ for integers x = 0 to 24.

Instantaneous Deflection:

Step 21: Compute the instantaneous (elastic) deflection at each time increment for the column strip. Use the equation $\Delta_e(t_i) = (k/384) * \Delta w(t_i) * (l_1 * 12)^4 / [E_c(t_i) * I_{e\text{-col}}]$. l_1 is multiplied by 12 because l_1 must be converted to units of inches. In the form of computer program arrays, the formula appears as follows: $\Delta_e[x] = (k/384) * \Delta w[x] * (l_1 * 12)^4 / (E_c[x] * I_{e\text{-col}})$ for x = 0 to 24.

Step 22: Compute the cumulative instantaneous deflection at each time increment for the column strip. In the form of computer program arrays, the formula appears as follows: $\Delta_{e,cum.}[x] = \Delta_e[x]$ for x = 0, and $\Delta_{e,cum.}[x] = \Delta_{e,cum.}[x - 1] + \Delta_e[x]$ for x = 1 to 24.

Long-Time Multiplier:

Step 23: (USER INPUT) Specify the average creep multiplier, λ_c.

Step 24: (USER INPUT) Specify the creep recovery factor.

Step 25: (USER INPUT) Select which equation to use in computing the creep correction factor for loading age. Select between the ACI 209 recommended equation and the Ghosh recommended equation. If using the ACI 209 recommended equation, select between the moist cured and steam cured version of the equation. In the following equation γ_{la} is the creep correction factor for loading age and t_{la} is the time of application of the loading:

$\gamma_{la} = 1.25(t_{la})^{-0.118}$ (ACI 209 moist cured)

$\gamma_{la} = 1.13(t_{la})^{-0.094}$ (ACI 209 steam cured)

$\gamma_{la} = 2.3(t_{la})^{-0.25}$ (Ghosh)

Step 26: (USER INPUT) Determine whether to include the correction factor for relative humidity. If including the correction factor for relative humidity, specify the relative humidity as a percent.

Step 27: If including the correction factor for relative humidity, compute the correction factor for relative humidity using the following equation: $\gamma_\lambda = 1.27 - 0.0067H$, where H is the relative humidity as a percent. If excluding the correction factor for relative humidity set $\gamma_\lambda = 1.0$.

Step 28: Compute the loading age correction factor, γ_{la}, at each time increment. Be sure to use the appropriate equation (ACI 209 moist cured, ACI 209 steam cured, or Ghosh).

Step 29: Compute a set of long-time multipliers for each time increment of interest. Use the following time-dependent equation: $\lambda(t_i, t_j) = (t_i - t_j)^{0.6} / [10 + (t_i - t_j)^{0.6}] * \lambda_c * \gamma_{la} * \gamma_\lambda$. Time t_i represents the time increment of interest, while time t_j represents the time increment of loading application. Thus, $t_i > t_j$. The loading change at each time increment will generate long-time multiplier values for each time increment beyond the time increment of the loading change. In the form of computer program arrays, the formula appears as follows: $\lambda[x, y] = (day[x] - day[y])^{0.6} / [10 + (day[x] - day[y])^{0.6}] * \lambda_c * \gamma_{la} * \gamma_\lambda$, where x corresponds to the time increment of interest and y corresponds to the time increment of loading change. Thus, for each individual integer in the range y = 0 to 24, x = (y + 1) to 24. Additionally, for each individual integer in the range y = 0 to 24, the long-time multipliers corresponding to the range x = 0 to y will be set to zero.

Step 30: Modify all long-time multipliers based upon whether the long-time multiplier corresponds to a positive or negative loading change. For each set of multipliers corresponding to a positive loading change (or zero loading change), no change is necessary. For each set of multipliers corresponding to a negative loading change, all multipliers within the set must be multiplied by the creep recovery factor to obtain modified multipliers.

Long-Time Deflection:

Step 31: Compute the long-time deflection associated with each and every long-time multiplier. Use the following time-dependent equation: $\Delta_{cr}(t_i, t_j) = [k * (l * 12)^4 * \Delta w(t_j) * \lambda(t_i, t_j)] / [384 * I_e * E_c(t_j)]$, where t_j is the time increment of loading change and t_i is the time increment of interest. l will be l_1 for a column strip and l_2 for a middle strip. Similarly, I_e will be $I_{e\text{-}col}$ for a column strip and $I_{e\text{-}mid}$ for a middle strip. Each loading change taking place at time t_j

generates long-time deflections computed at each time increment after time t_j. In the form of computer program arrays, the formula appears as follows: $\Delta_{cr}[x, y] = \{k * (l * 12)^4 * \Delta w[y] * \lambda[x, y]\} / \{384 * I_e * E_c[y]\}$, where x corresponds to the time increment of interest and y corresponds to the time increment of loading change. Thus, for each individual integer in the range y = 0 to 24, x = (y + 1) to 24. Additionally, for each individual integer in the range y = 0 to 24, the long-time creep deflections corresponding to the range x = 0 to y will be set to zero (in keeping with the computation procedure for long-time multipliers).

Step 32: Compute the total long-time creep deflection taking place at each time increment of interest, $\Delta_{cr,cum.}(t_i)$. For a given time increment, this will be the summation of each individual time-dependent deflection at that time increment. Each and every previous loading change will produce one corresponding deflection value at a given time increment. The cumulative long-time creep deflection taking place at a given time increment, t_i, may be computed as follows: $\Delta_{cr,cum.}(t_i) = \sum^{i}_{j=0}[\Delta_{cr}(t_i, t_j)]$. In the form of computer program arrays, the formula appears as follows: $\Delta_{cr,cum.}[x] = \sum^{x}_{y=0}(\Delta_{cr}[x, y])$. As this step and the preceding steps may be a bit confusing in written format, Table 3.8 may assist in understanding the process. In the form of computer program arrays the subscripts in Table 3.8 stand for x, y. Table 3.8 shows only the first five time increments and load increments, but the procedure is the same for all 25.

Table 3.8 – Clarification on the Computation of the Cumulative Long-Time Creep Deflection, $\Delta_{cr,cum.}$

	t[0]		t[1]		t[2]		t[3]		t[4]	
	Mult.	Defl.	mult.	Defl.	mult.	defl.	mult.	defl.	mult.	defl.
Δw[0]	0	0	$\lambda_{1,0}$	$\Delta_{1,0}$	$\lambda_{2,0}$	$\Delta_{2,0}$	$\lambda_{3,0}$	$\Delta_{3,0}$	$\lambda_{4,0}$	$\Delta_{4,0}$
Δw[1]	0	0	0	0	$\lambda_{2,1}$	$\Delta_{2,1}$	$\lambda_{3,1}$	$\Delta_{3,1}$	$\lambda_{4,1}$	$\Delta_{4,1}$
Δw[2]	0	0	0	0	0	0	$\lambda_{3,2}$	$\Delta_{3,2}$	$\lambda_{4,2}$	$\Delta_{4,2}$
Δw[3]	0	0	0	0	0	0	0	0	$\lambda_{4,3}$	$\Delta_{4,3}$
Δw[4]	0	0	0	0	0	0	0	0	0	0
Total long-time defl.	0		$\Delta_{1,0}$		$\Delta_{2,0} + \Delta_{2,1}$		$\Delta_{3,0} + \Delta_{3,1} + \Delta_{3,2}$		$\Delta_{4,1} + \Delta_{4,1} + \Delta_{4,2} + \Delta_{4,3}$	

Column Strip Deflection:

Step 33: Compute the total deflection for the column strip at each time increment by summing the cumulative instantaneous deflection and the cumulative long-time creep deflection at each time increment, $\Delta_c(t_i) = \Delta_{e,cum.}(t_i) + \Delta_{cr,cum.}(t_i)$.

Modified Middle-Strip Correction Factors:

Step 34: Modify k_{SS}. $k_{SS} = 0.65$ for a middle strip.

Step 35: Modify k_{WF}. $k_{WF} = (l_1/2) / (l_1 - l_2/2)$ for a middle strip.

Step 36: Compute k for the middle strip. $k = k_{BC\text{-}mid} * k_{DP} * k_{WF} * k_{SS}$.

If the user chooses to input information regarding reshoring and the casting cycle rather than choosing to input information regarding the loading-time history, then follow steps A16 through A20 and ignore steps B4 through B5:

Modified Middle Strip Loads:

Step A16: Compute the distributed construction load on the middle slab strip in units of pounds per linear inch, $w_{construction} = (DL * b_2) / 12$.

Step A17: Compute the distributed sustained load on the middle slab strip in units of pounds per linear inch, $w_{sustained} = (\text{sustained load} * b_2) / 12$.

Step A18: Compute the distributed maximum load on the middle slab strip in units of pounds per linear inch, $w_{maximum} = (\text{maximum load} * b_2) / 12$.

Modified Middle Strip Loading-Time History:

Step A19: Compute effective loading values for the middle strip. This procedure is identical to the procedure performed in step A14.

Step A20: Compute the loading change between time increments, $\Delta w(t_i) = w_i - w_{i-1}$. In the form of computer program arrays, the formula appears as follows: $\Delta w[0] = 0$, and $\Delta w[x] = w[x] - w[x - 1]$ for integers $x = 1$ to 24.

If the user chooses to input information regarding the loading-time history rather than choosing to input information regarding reshoring and the casting cycle, then follow steps B4 through B5 and ignore steps A16 through A20:

Prescribed Loading-Time History:

Step B4: At each time increment compute the distributed load on the middle strip in units of pounds per linear inch, w = (input loading * b_2) / 12.

Step B5: Compute the loading change between time increments, $\Delta w(t_i) = w_i - w_{i-1}$. In the form of computer program arrays, the formula appears as follows: Δw[0] = 0, and Δw[x] = w[x] – w[x – 1] for integers x = 1 to 24.

Middle Strip Deflection:

Step 37: Perform step 20 and step 21 for the middle strip to compute instantaneous deflections. Then perform step 27 through step 32 for the middle strip. This will ultimately result in the computation of the total deflection at the mid-span of the middle strip, $\Delta_m(t_i)$, for each time increment. Be sure to substitute the effective moment of inertia for the middle strip, to use the modified correction factor for the middle strip, and to use the l_2 clear span dimension.

Total (Mid-Panel) Deflection:

Step 38: Compute the total mid-panel deflection for the slab for each time increment using the crossing-beam method, $\Delta_{mp}(t_i) = \Delta_c(t_i) + \Delta_m(t_i)$.

Step 39: (OUTPUT) Report the final mid-panel deflection-time results as 25 pairings of time and corresponding mid-panel deflection (t, $\Delta_{mp}(t_i)$).

3.5 SUMMARY

The algorithm outlined in this chapter was implemented as a C++ computer program using Microsoft Visual Basic C++ Standard Edition 2003. The C++ computer code is provided in Appendix A, preceded by a summary of the requisite computer program input and corresponding computer program output.

CHAPTER 4
VERIFICATION OF THE METHOD OF ANALYSIS

4.1 INTRODUCTION

The purpose of this chapter is to validate the analytical methodology outlined in Chapter 3 by comparing results generated using the computer algorithm to experimental results. Specifically, the results generated using the computer algorithm will be compared to experimental results presented in Guo and Gilbert (2002).

Section 4.2 presents the data input into the computer algorithm. The data is based upon information contained within Guo and Gilbert (2002). Section 4.3 presents the results of the computer algorithm analysis. Section 4.4 presents the Guo and Gilbert (2002) experimental results. Section 4.5 compares the results of the computer algorithm analysis and the Guo and Gilbert (2002) experimental analysis.

Table 4.1 – Algorithmic Input Parameters for Guo & Gilbert (2002) S1 Slab Analysis

		Metric		U.S
$f'_{c(28\text{-day})}$	=	39.2 Mpa	=	5690 psi
h	=	100 mm	=	3.94 in
l_1	=	2800 mm	=	9.19 ft
l_2	=	2800 mm	=	9.19 ft
$k_{BC\text{-}col}$	=	2.0	=	2.0
$k_{BC\text{-}mid}$	=	2.0	=	2.0
drop panel	=	No	=	No
humidity	=	Exclude	=	Exclude
λ_c	=	2.0	=	2.0

4.2 INPUT DATA

Guo and Gilbert (2002) performed analyses for seven different slabs, designated S1, S2, S3, S4, S5, S6, and S7. Comparison within this report focuses upon the S1 slab. The S1 slab consists of four square corner panels supported by columns.

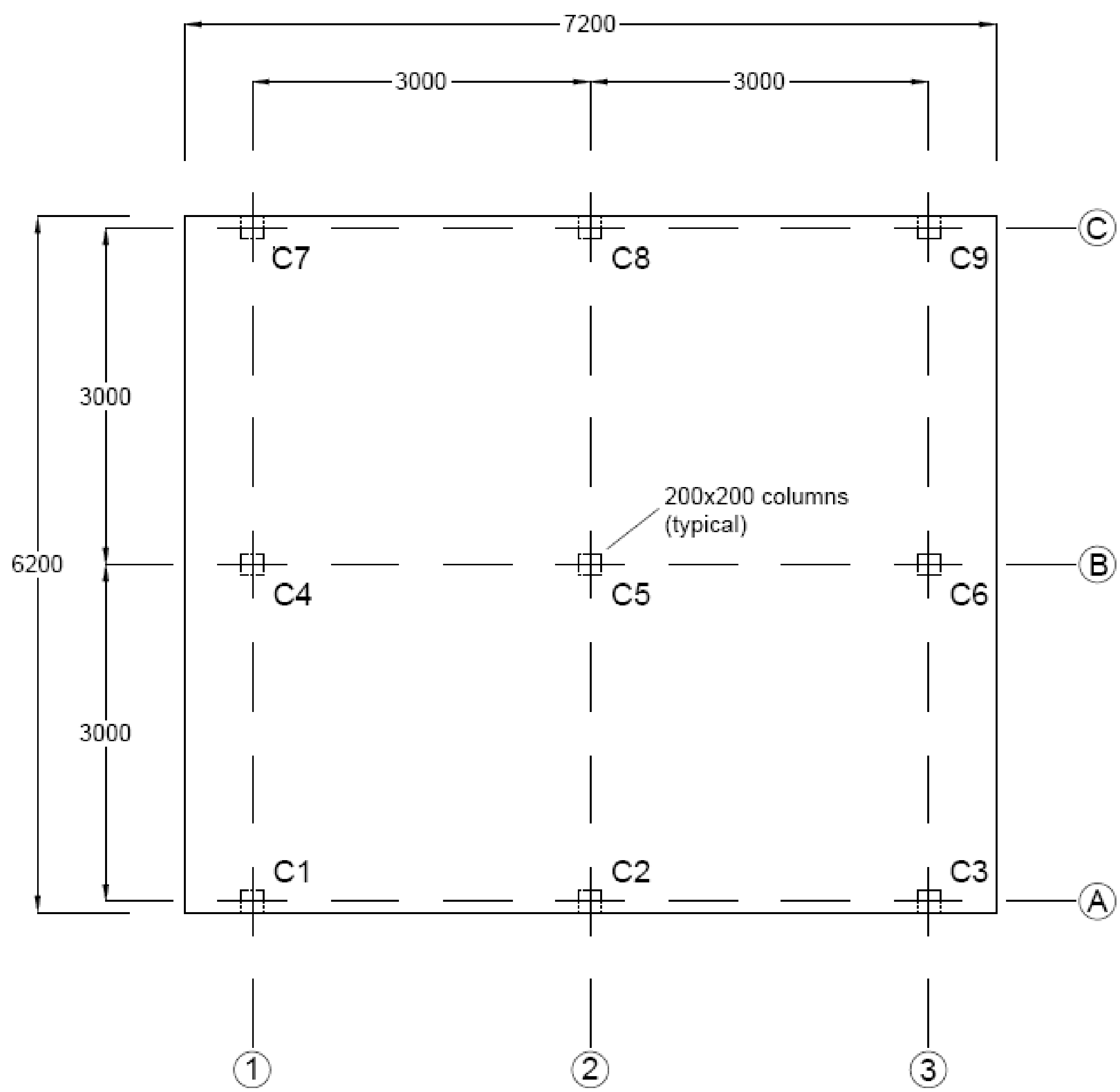

Figure 4.1 – Guo & Gilbert (2002) S1 Slab Plan (Dimensions in mm)

There is a slight cantilever overhang which may be neglected since the panel will still behave as a corner panel. Figure 4.1 (Guo & Gilbert, 2002) shows the S1 slab plan with all dimensions in millimeters (mm). Dimensions and material properties within the Guo and Gilbert (2002) report were provided in metric units. Algorithmic input parameters obtained from the Guo and Gilbert (2002) report are converted to U.S. units and are shown in Table 4.1. A plot of the loading-time history for the S1 slab obtained from the Guo and Gilbert (2002) report is shown in Figure 4.2 converted to U.S. units. Table 4.2 provides the loading-time input for the computer algorithm.

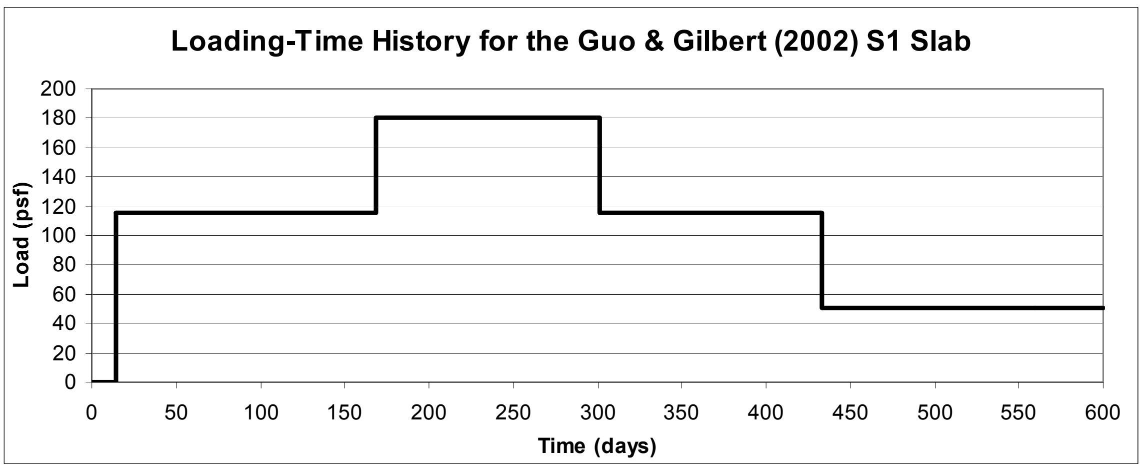

Figure 4.2 – Loading-Time History for the Guo & Gilbert (2002) S1 Slab in U.S. Units

Table 4.2 – Algorithmic Loading-Time Input for Guo & Gilbert (2002) S1 Slab Analysis

Time (days)	Load (psf)		Time (days)	Load (psf)		Time (days)	Load (psf)		Time (days)	Load (psf)		Time (days)	Load (psf)
0	0		120	115.914		240	180.868		360	115.914		480	50.125
14	0		160	115.914		280	180.868		400	115.914		512	50.125
14	115.914		169	115.914		301	180.868		433	115.914		520	50.125
40	115.914		169	180.868		301	115.914		433	50.125		560	50.125
80	115.914		200	180.868		320	115.914		440	50.125		600	50.125

The computer algorithm is run for six different scenarios, generated by combining creep recover factors of 0.5, 0.7, and 0.9 with either the Ghosh equation for computing the loading age correction factor or the ACI moist-cured equation for computing the loading age correction factor. All other input parameters remain constant for the six scenarios.

4.3 COMPUTER ALGORITHM RESULTS

The deflection-time output for the computer algorithm analysis is shown in Table 4.3 and Table 4.4. Deflections are provided in metric units (millimeters) to provide ease of comparison with Guo and Gilbert (2002) experimental results.

Table 4.3 provides algorithmic results obtained using the Ghosh equation to compute the loading age creep correction factor, while Table 4.4 provides results obtained using the ACI moist-cured equation to compute the loading age creep correction factor. Both tables provide results for creep recovery factors of 0.5, 0.7, and 0.9.

Table 4.3 – Computer Algorithm Results for the Guo & Gilbert (2002) S1 Slab Using the Ghosh Equation

	Ghosh Equation Creep Recover = 0.5	Ghosh Equation Creep Recover = 0.7	Ghosh Equation Creep Recover = 0.9
Time	Deflection	Deflection	Deflection
(days)	(mm)	(mm)	(mm)
0	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000
14	3.2108	3.2108	3.2108
40	6.3714	6.3714	6.3714
80	7.4303	7.4303	7.4303
120	7.9556	7.9556	7.9556
160	8.2918	8.2918	8.2918
169	8.3524	8.3524	8.3524
169	9.9305	9.9305	9.9305
200	10.9959	10.9959	10.9959
240	11.4290	11.4290	11.4290
280	11.7056	11.7056	11.7056
301	11.8200	11.8200	11.8200
301	10.2514	10.2514	10.2514
320	10.0227	9.8948	9.7669
360	10.0413	9.8556	9.6699
400	10.1100	9.8981	9.6861
433	10.1703	9.9445	9.7186
433	8.5853	8.3594	8.1336
440	8.4034	8.0974	7.7914
480	8.2660	7.8658	7.4656
512	8.2560	7.8239	7.3919
520	8.2572	7.8190	7.3809
560	8.2732	7.8111	7.3490
600	8.2979	7.8184	7.3389

Table 4.4 – Computer Algorithm Results for the Guo & Gilbert (2002) S1 Slab Using the ACI Moist Cured Equation

	ACI, Moist Cured Creep Recover = 0.5	ACI, Moist Cured Creep Recover = 0.7	ACI, Moist Cured Creep Recover = 0.9
Time	Deflection	Deflection	Deflection
(days)	(mm)	(mm)	(mm)
0	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000
14	3.2108	3.2108	3.2108
40	5.6444	5.6444	5.6444
80	6.4597	6.4597	6.4597
120	6.8641	6.8641	6.8641
160	7.1230	7.1230	7.1230
169	7.1697	7.1697	7.1697
169	8.7477	8.7477	8.7477
200	9.8335	9.8335	9.8335
240	10.2415	10.2415	10.2415
280	10.4934	10.4934	10.4934
301	10.5960	10.5960	10.5960
301	9.0273	9.0273	9.0273
320	8.7393	8.5916	8.4440
360	8.7163	8.5020	8.2876
400	8.7582	8.5135	8.2688
433	8.8007	8.5400	8.2793
433	7.2157	6.9549	6.6942
440	6.9894	6.6317	6.2740
480	6.7907	6.3195	5.8484
512	6.7553	6.2460	5.7367
520	6.7512	6.2346	5.7181
560	6.7448	6.1996	5.6544
600	6.7513	6.1855	5.6195

4.4 EXPERIMENTAL RESULTS (GUO & GILBERT)

The Guo and Gilbert (2002) experimental deflection-time results for the S1 slab are shown in Table 4.5.

Table 4.5 – Guo & Gilbert (2002) Experimental Deflection-Time Results for the S1 Slab

Time (days)	Deflection (mm)	Time (days)	Deflection (mm)	Time (days)	Deflection (mm)	Time (days)	Deflection (mm)
14	1.96	76	5.70	198	8.40	300	9.82
14.7	2.27	80	5.73	204	8.52	301	9.86
15	2.38	84	5.92	210	8.45	301.1	8.87
16	2.48	98	6.28	213	8.37	302	8.85
17	2.73	107	6.43	217	8.39	303	8.86
20	3.19	113	6.49	220	8.40	306	8.80
21	3.33	122	6.41	224	8.40	308	8.84
22	3.52	126	6.45	227	8.44	321	8.76
23	3.55	127	6.50	231	8.56	325	8.87
24	3.63	129	6.58	234	8.54	328	8.86
27	3.84	132	6.66	238	8.49	335	8.89
28	3.97	136	6.55	243	8.64	343	9.17
29	4.19	142	6.67	250	8.64	346	9.17
30	4.23	150	6.71	258	8.82	349	9.26
31	4.24	157	6.64	267	8.75	356	9.33
34	4.38	162	6.88	273	8.77	363	9.11
36	4.45	164	6.82	279	8.79	373	9.20
38	4.48	167	6.83	279.02	8.62	377	9.26
41	4.54	168	6.90	279.05	8.56	387	9.49
43	4.68	168.3	6.91	279.2	8.33	394	9.47
45	4.62	169	6.94	279.96	7.72	409	9.63
49	4.81	169.3	8.01	280	7.69	416	9.68
51	4.97	169.4	8.05	281	8.37	421	9.60
52	5.01	170	8.11	282	8.90	433	9.61
55	5.15	171	8.15	285	9.31	447	8.49
58	5.18	174	8.17	286	9.38	457	8.44
62	5.53	175	8.27	288	9.45	468	8.55
65	5.53	181	8.53	292	9.56	512	8.17
70	5.60	184	8.43	294	9.69		
73	5.65	191	8.52	296	9.76		

4.5 GRAPHICAL COMPARISON OF ALGORITHMIC AND EXPERIMENTAL RESULTS

This section provides a graphical comparison of the computer algorithm deflection-time results (provided in section 4.3) and the Guo & Gilbert (2002) experimental deflection-time results (provided in section 4.4). Figure 4.3 corresponds to algorithmic results computed using a creep recovery factor of 0.5. Figure 4.4 corresponds to algorithmic results computed using a creep recovery factor of 0.7, while Figure 4.5 corresponds to algorithmic results computed using a creep recovery factor of 0.9.

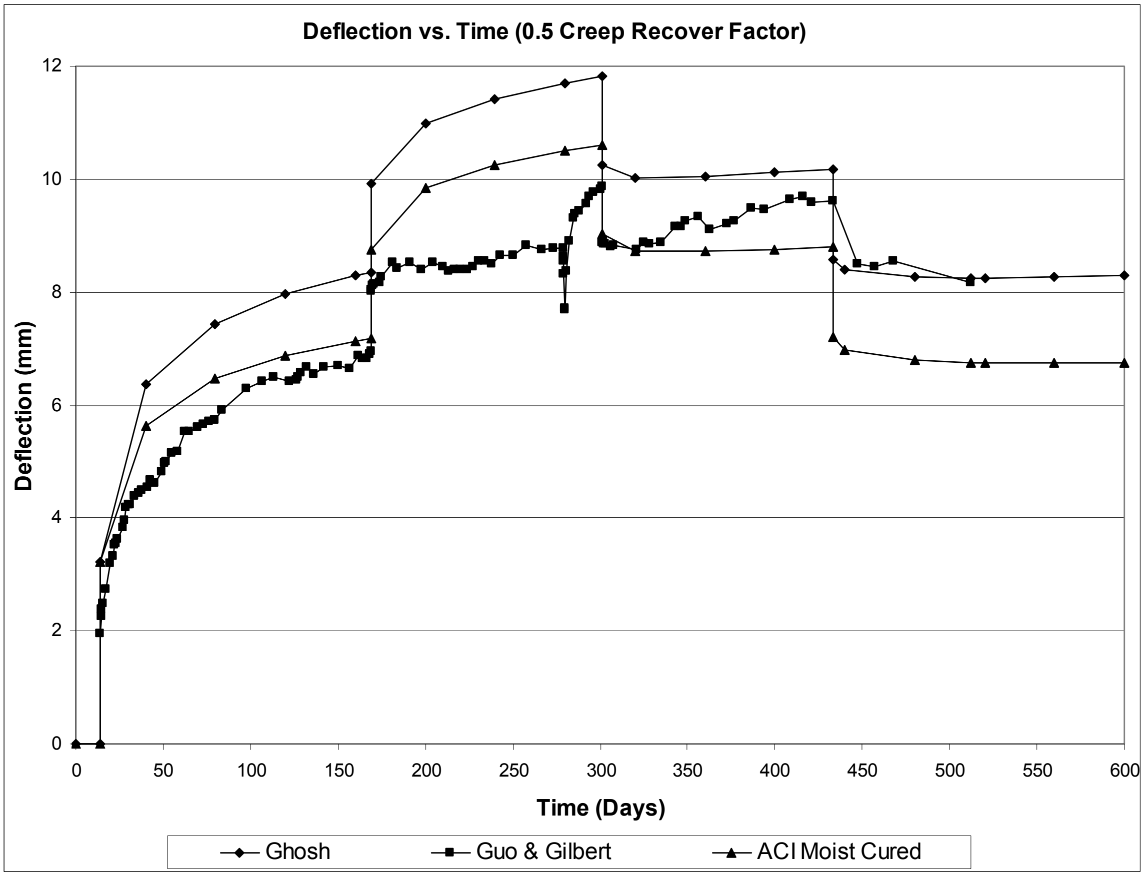

Figure 4.3 – Comparison of Algorithmic (0.5 Creep Recovery Factor) and Experimental Results

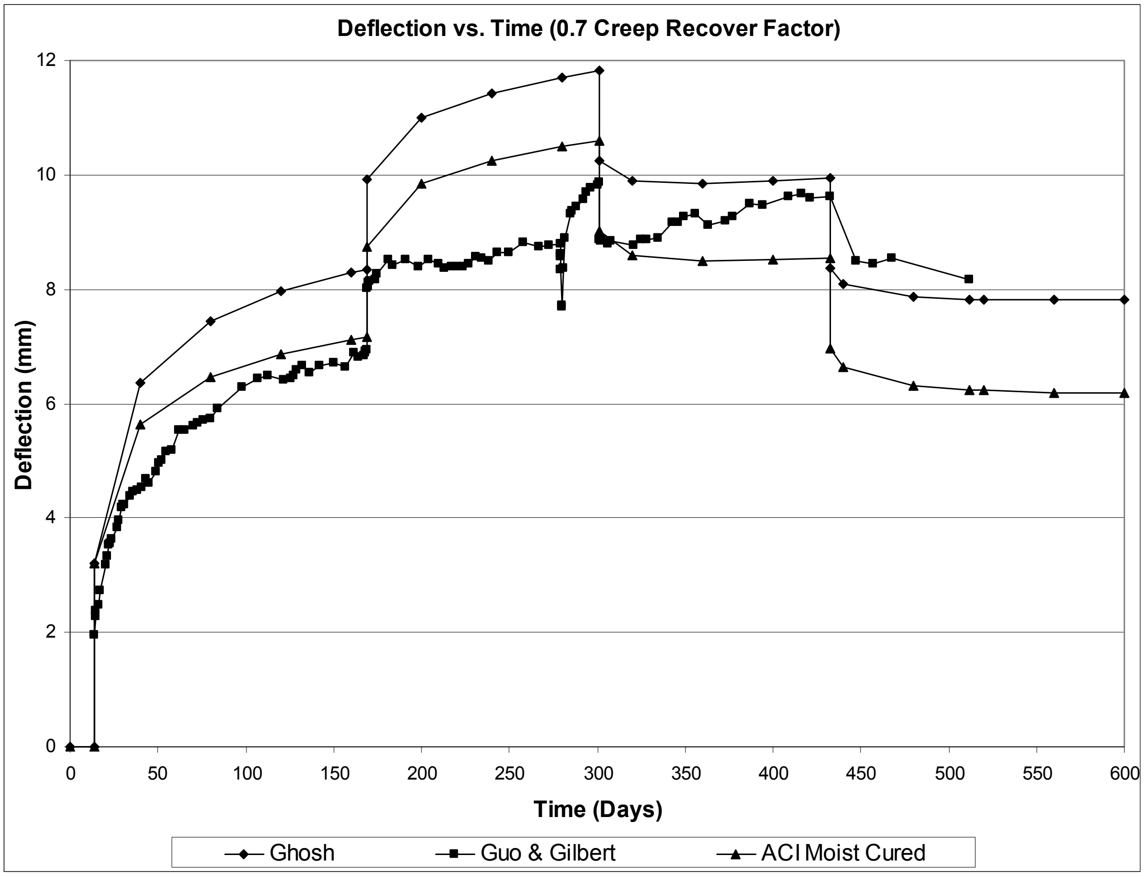

Figure 4.4 – Comparison of Algorithmic (0.7 Creep Recovery Factor) and Experimental Results

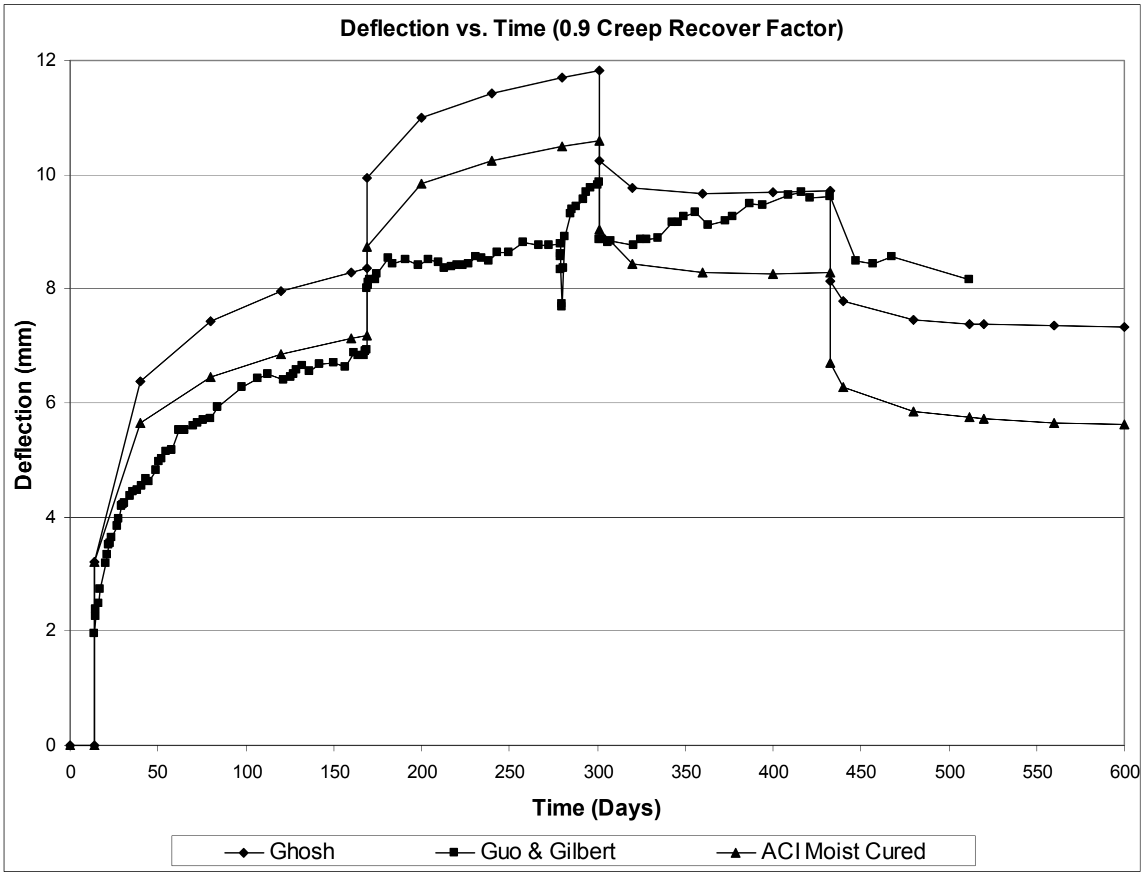

Figure 4.5 – Comparison of Algorithmic (0.9 Creep Recovery Factor) and Experimental Results

4.6 CONCLUSION

The graphical comparison of algorithmic and experimental results presented in the previous section indicates that a creep recovery factor of 0.5 provides the best correlation with experimental results for the Guo and Gilbert (2002) S1 slab. Because creep recovery is only applicable for loading decrements, the effect of this parameter is irrelevant up to the 301-day mark at which point in time the first loading decrement occurs. From the 301-day mark forward, the 0.5 creep recovery factory shows the best correlation with the Guo and Gilbert (2002) experimental data.

The ACI moist cured equation provides better correlation with the Guo and Gilbert (2002) experimental data up to the first loading decrement. Neither equation accurately captures the trend taking place in the experimental data between the first and second loading decrements. The Ghosh equation provides better correlation with the tail end of the Guo and Gilbert (2002) experimental data.

Overall the computer algorithm output shows strong correlation with the experimental data. The algorithmic results capture the major trends displayed within the experimental data. When compared to the experimental data, the algorithmic output appears slightly conservative. Because the algorithm represents an approximate analysis method, a slightly conservative estimate of experimental data is appropriate.

CHAPTER 5
PARAMETRIC STUDY

5.1 INTRODUCTION

This chapter presents a parametric study conducted using the computer algorithm presented in Chapter 3. All computer algorithm input parameters are set to constant values as specified in section 5.2. Specific input parameters are varied one-by-one in an effort to determine parameter sensitivity. The input parameters studied within the parametric study consist of the following: average creep multiplier, λ_c; 28-day concrete compressive strength, $f'_{c(28\text{-day})}$; casting cycle, stripping time, and shoring/reshoring scheme; short-direction clear span, l_2; slab thickness, h; creep recovery factor; loading age creep correction equation; and panel location. Section 5.3 provides an overview of the parametric study, specifying the value of all input parameters used for each step of the process. Sections 5.4 through 5.11 individually focus upon algorithmic sensitivity to variation in one specific parameter. Section 5.12 compares an algorithmic analysis performed using the constants of section 5.2 to an algorithmic analysis that excludes construction load effects. This section is included in addition to the parametric study to demonstrate the significance of construction load analysis.

5.2 CONSTANTS (UNLESS OTHERWISE VARIED)

For the purposes of the parametric study, all computer algorithm input parameters are set to the constant values shown in Table 5.1. Each parameter is varied one-by-one in order to evaluate parameter sensitivity. For the purposes of the parametric study, the creep correction factor to account for relative humidity is excluded throughout.

Table 5.1 – Constant Values for the Parametric Study

$f'_{c(28\text{-day})}$	=	4000 psi
h	=	7 inches
l_1	=	19 feet
l_2	=	19 feet
$k_{BC\text{-}col}$	=	1.4
$k_{BC\text{-}mid}$	=	1.4
drop panels	=	No
SDL	=	20 psf
LL	=	80 psf
unit weight	=	150 pcf
reshoring levels	=	2 levels
casting cycle	=	7 days
stripping	=	5 days
creep equation	=	Ghosh
λ_c	=	2
creep recovery factor	=	0.5

5.3 OUTLINE OF THE PARAMETRIC STUDY

Table 5.2 presents an overview of the parametric study, detailing the values of the variables and constants associated with the study of each individual parameter.

Table 5.2 – Outline of the Parametric Study

Parameter	Variables	Constants
Average Creep Multiplier	$\lambda_c = 1.0$ $\lambda_c = 2.0$ $\lambda_c = 3.0$	$f'_{c(28\text{-day})} = 4000$ psi h = 7 inches $l_1 = 19$ feet $l_2 = 19$ feet $k_{BC\text{-}col} = 1.4$ $k_{BC\text{-}mid} = 1.4$ drop panels = no SDL = 20 psf LL = 80 psf unit weight = 150 pcf reshoring levels = 2 casting cycle = 7 days stripping = 5 days creep equation = Ghosh creep recover factor = 0.5
28-Day Compressive Strength	$f'_{c(28\text{-day})} = 3000$ psi $f'_{c(28\text{-day})} = 4000$ psi $f'_{c(28\text{-day})} = 5000$ psi $f'_{c(28\text{-day})} = 6000$ psi $f'_{c(28\text{-day})} = 7000$ psi $f'_{c(28\text{-day})} = 8000$ psi	h = 7 inches $l_1 = 19$ feet $l_2 = 19$ feet $k_{BC\text{-}col} = 1.4$ $k_{BC\text{-}mid} = 1.4$ drop panels = no SDL = 20 psf LL = 80 psf unit weight = 150 pcf reshoring levels = 2 casting cycle = 7 days stripping = 5 days creep equation = Ghosh avg. creep multiplier, $\lambda_c = 2.0$ creep recover factor = 0.5

Casting Cycle, Stripping Time, Shoring Levels	7 days, 5 days, 1 level 7 days, 5 days, 2 levels 7 days, 5 days, 3 levels 4 days, 3 days, 1 level 4 days, 3 days, 2 levels 4 days, 3 days, 3 levels	$f'_{c(28\text{-day})}$ = 4000 psi h = 7 inches l_1 = 19 feet l_2 = 19 feet $k_{BC\text{-}col}$ = 1.4 $k_{BC\text{-}mid}$ = 1.4 drop panels = no SDL = 20 psf LL = 80 psf unit weight = 150 pcf creep equation = Ghosh avg. creep multiplier, λ_c = 2.0 creep recover factor = 0.5
Short Direction Clear Span, l_2	l_2 = 19' l_2 = 14' l_2 = 9'	$f'_{c(28\text{-day})}$ = 4000 psi h = 7 inches l_1 = 19 feet $k_{BC\text{-}col}$ = 1.4 $k_{BC\text{-}mid}$ = 1.4 drop panels = no SDL = 20 psf LL = 80 psf unit weight = 150 pcf reshoring levels = 2 casting cycle = 7 days stripping = 5 days creep equation = Ghosh avg. creep multiplier, λ_c = 2.0 creep recover factor = 0.5

Slab Thickness, h	h = 6" h = 7" h = 8"	$f'_{c(28\text{-}day)}$ = 4000 psi l_1 = 19 feet l_2 = 19 feet $k_{BC\text{-}col}$ = 1.4 $k_{BC\text{-}mid}$ = 1.4 drop panels = no SDL = 20 psf LL = 80 psf unit weight = 150 pcf reshoring levels = 2 casting cycle = 7 days stripping = 5 days creep equation = Ghosh avg. creep multiplier, λ_c = 2.0 creep recover factor = 0.5
Creep Recovery Factor	creep recov. factor = 0.5 creep recov. factor = 0.7 creep recov. factor = 0.9	$f'_{c(28\text{-}day)}$ = 4000 psi h = 7 inches l_1 = 19 feet l_2 = 19 feet $k_{BC\text{-}col}$ = 1.4 $k_{BC\text{-}mid}$ = 1.4 drop panels = no SDL = 20 psf LL = 80 psf unit weight = 150 pcf reshoring levels = 2 casting cycle = 7 days stripping = 5 days creep equation = Ghosh avg. creep multiplier, λ_c = 2.0

Long-Time Creep Correction Equation	Ghosh ACI Moist Cured	$f'_{c(28\text{-day})}$ = 4000 psi h = 7 inches l_1 = 19 feet l_2 = 19 feet $k_{BC\text{-}col}$ = 1.4 $k_{BC\text{-}mid}$ = 1.4 drop panels = no SDL = 20 psf LL = 80 psf unit weight = 150 pcf reshoring levels = 2 casting cycle = 7 days stripping = 5 days avg. creep multiplier, λ_c = 2.0 creep recover factor = 0.5
Panel Location	Corner: $k_{BC\text{-}col}$ = 2.0 $k_{BC\text{-}mid}$ = 2.0 Edge: $k_{BC\text{-}col}$ = 2.0 $k_{BC\text{-}mid}$ = 1.4 Edge: $k_{BC\text{-}col}$ = 1.4 $k_{BC\text{-}mid}$ = 2.0 Interior: $k_{BC\text{-}col}$ = 1.4 $k_{BC\text{-}mid}$ = 1.4	$f'_{c(28\text{-day})}$ = 4000 psi h = 7 inches l_1 = 19 feet l_2 = 19 feet drop panels = no SDL = 20 psf LL = 80 psf unit weight = 150 pcf reshoring levels = 2 casting cycle = 7 days stripping = 5 days creep equation = Ghosh avg. creep multiplier, λ_c = 2.0 creep recover factor = 0.5

5.4 PARAMETER: AVERAGE CREEP MULTIPLIER, λ_c

Within the parametric study, Table 5.3 and Figure 5.1 present the sensitivity to variation in the average creep multiplier, λ_c. Figure 5.2 focuses upon the sensitivity of this parameter during the construction phase.

Table 5.3 – Sensitivity to Variation in the Average Creep Multiplier, λ_c

Average creep multiplier, λ_c = 1.0			Average creep multiplier, λ_c = 2.0 ** constant state **			Average creep multiplier, λ_c = 3.0	
Time (days)	Deflection (inches)		Time (days)	Deflection (inches)		Time (days)	Deflection (inches)
0	0.0000		0	0.0000		0	0.0000
5	0.3781		5	0.3781		5	0.3781
7	0.4547		7	0.5312		7	0.6078
7	0.5717		7	0.6482		7	0.7248
12	0.6710		12	0.8469		12	1.0228
12	0.5643		12	0.7401		12	0.9160
14	0.5782		14	0.7679		14	0.9577
14	0.6827		14	0.8725		14	1.0623
19	0.7442		19	0.9954		19	1.2466
19	0.6431		19	0.8943		19	1.1456
21	0.6520		21	0.9121		21	1.1722
21	0.7521		21	1.0122		21	1.2723
26	0.8014		26	1.1108		26	1.4203
26	0.7299		26	1.0393		26	1.3488
30	0.7485		30	1.0766		30	1.4046
60	0.8484		60	1.2764		60	1.7043
90	0.9025		90	1.3846		90	1.8667
120	0.9385		120	1.4565		120	1.9746
150	0.9649		150	1.5093		150	2.0536
365	1.0552		365	1.6900		365	2.3247
730	1.1089		730	1.7973		730	2.4857
1095	1.1339		1095	1.8474		1095	2.5608
1460	1.1491		1460	1.8777		1460	2.6064
1825	1.1596		1825	1.8986		1825	2.6377
1825	1.3444		1825	2.0834		1825	2.8225

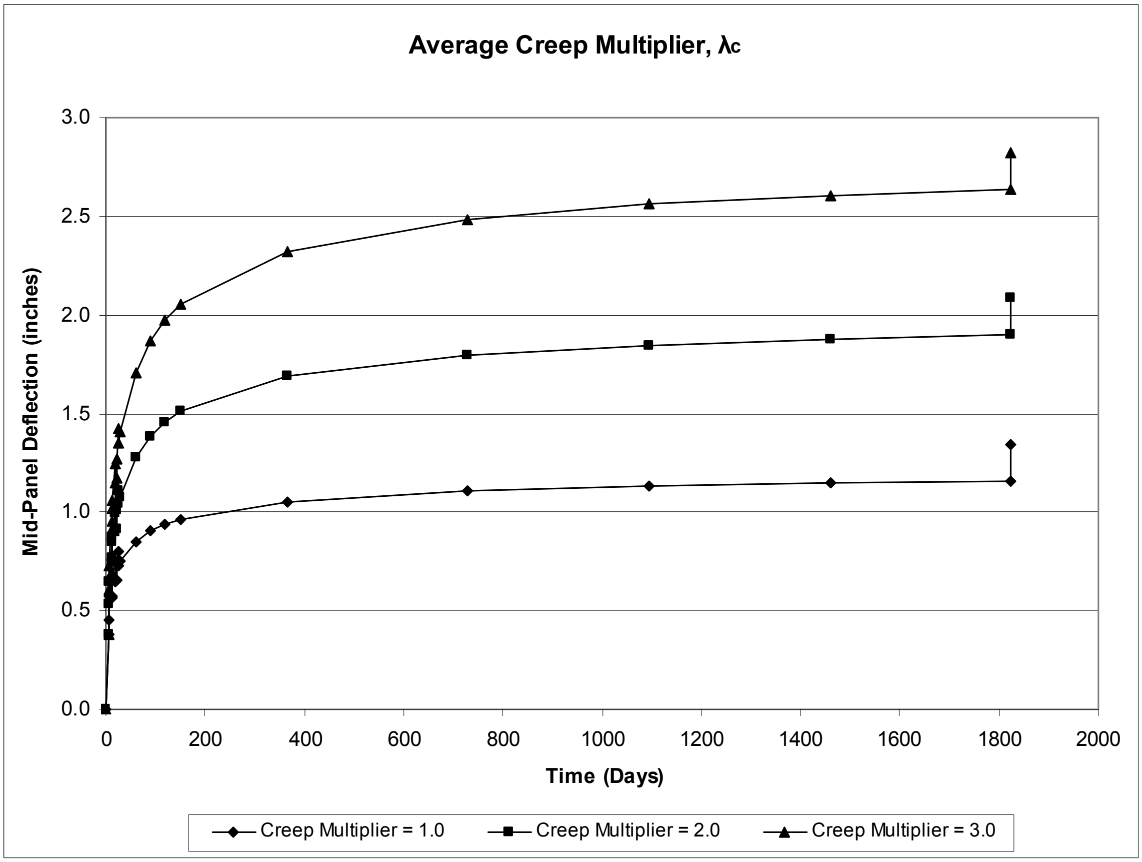

Figure 5.1 – Sensitivity to Variation in the Average Creep Multiplier, λ_c

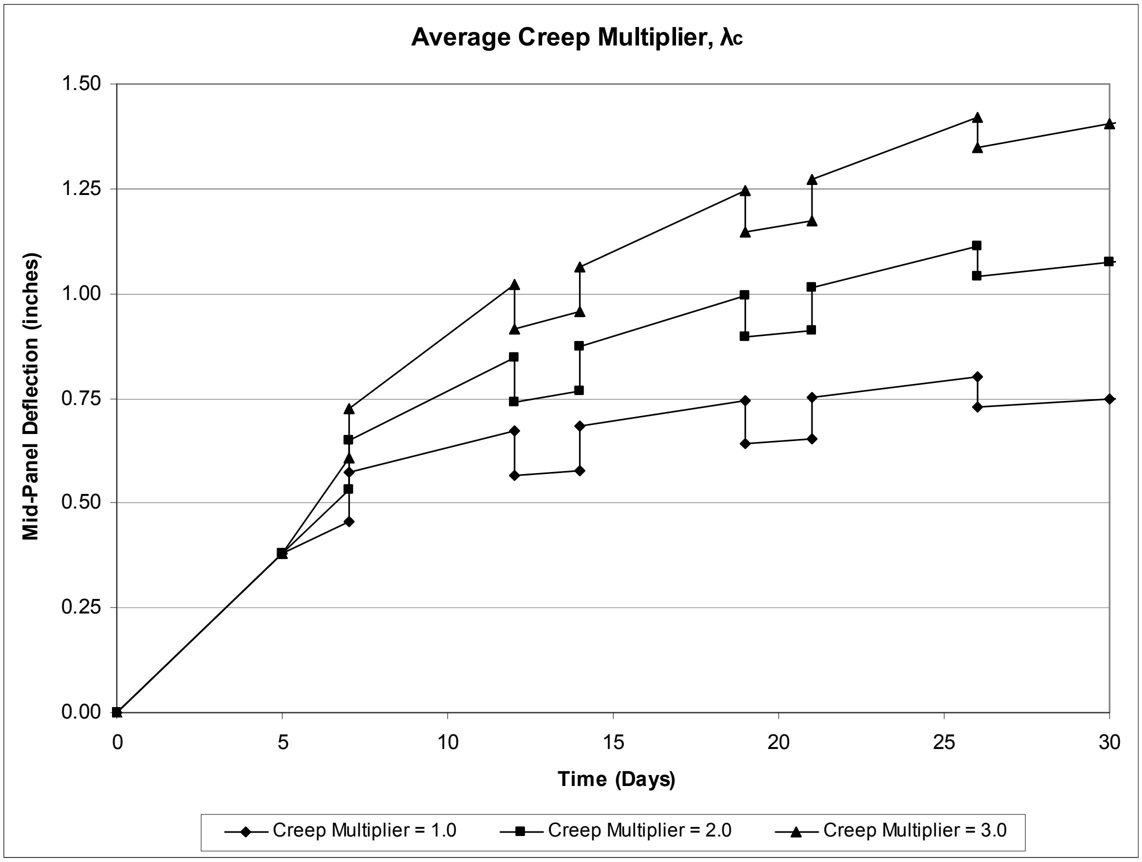

Figure 5.2 – Average Creep Multiplier, λ_c, During the Construction Phase

5.5 PARAMETER: 28-DAY COMPRESSIVE STRENGTH, $f'_{c(28\text{-day})}$

Within the parametric study, Table 5.4 and Figure 5.3 present the sensitivity to variation in the average 28-day concrete compressive strength, $f'_{c(28\text{-day})}$. Figure 5.4 focuses upon the sensitivity of this parameter during the construction phase.

Table 5.4 – Sensitivity to Variation in the 28-Day Compressive Strength, $f'_{c(28\text{-day})}$

$f'_{c(28\text{-day})}$ = 3000 psi			$f'_{c(28\text{-day})}$ = 4000 psi ** constant state **			$f'_{c(28\text{-day})}$ = 5000 psi	
Time (days)	Deflection (inches)		Time (days)	Deflection (inches)		Time (days)	Deflection (inches)
0	0.0000		0	0.0000		0	0.0000
5	0.4366		5	0.3781		5	0.3382
7	0.6134		7	0.5312		7	0.4752
7	0.7485		7	0.6482		7	0.5798
12	0.9779		12	0.8469		12	0.7575
12	0.8546		12	0.7401		12	0.6620
14	0.8867		14	0.7679		14	0.6869
14	1.0075		14	0.8725		14	0.7804
19	1.1494		19	0.9954		19	0.8903
19	1.0327		19	0.8943		19	0.7999
21	1.0532		21	0.9121		21	0.8158
21	1.1688		21	1.0122		21	0.9053
26	1.2827		26	1.1108		26	0.9936
26	1.2001		26	1.0393		26	0.9296
30	1.2431		30	1.0766		30	0.9629
60	1.4739		60	1.2764		60	1.1416
90	1.5988		90	1.3846		90	1.2384
120	1.6819		120	1.4565		120	1.3028
150	1.7427		150	1.5093		150	1.3499
365	1.9514		365	1.6900		365	1.5116
730	2.0754		730	1.7973		730	1.6076
1095	2.1331		1095	1.8474		1095	1.6523
1460	2.1682		1460	1.8777		1460	1.6795
1825	2.1924		1825	1.8986		1825	1.6982
1825	2.4057		1825	2.0834		1825	1.8635

Table 5.4 (Continued) – Sensitivity to Variation in the 28-Day Compressive Strength, $f'_{c(28\text{-}day)}$

$f'_{c(28\text{-}day)}$ = 6000 psi			$f'_{c(28\text{-}day)}$ = 7000 psi			$f'_{c(28\text{-}day)}$ = 8000 psi	
Time (days)	Deflection (inches)		Time (days)	Deflection (inches)		Time (days)	Deflection (inches)
0	0.0000		0	0.0000		0	0.0000
5	0.3087		5	0.2858		5	0.2674
7	0.4338		7	0.4016		7	0.3756
7	0.5293		7	0.4900		7	0.4584
12	0.6915		12	0.6402		12	0.5988
12	0.6043		12	0.5595		12	0.5234
14	0.6270		14	0.5805		14	0.5430
14	0.7124		14	0.6596		14	0.6170
19	0.8127		19	0.7524		19	0.7039
19	0.7302		19	0.6761		19	0.6324
21	0.7447		21	0.6895		21	0.6449
21	0.8264		21	0.7651		21	0.7157
26	0.9070		26	0.8397		26	0.7855
26	0.8486		26	0.7857		26	0.7349
30	0.8790		30	0.8138		30	0.7612
60	1.0422		60	0.9649		60	0.9025
90	1.1305		90	1.0467		90	0.9791
120	1.1893		120	1.1010		120	1.0299
150	1.2323		150	1.1409		150	1.0672
365	1.3799		365	1.2775		365	1.1950
730	1.4675		730	1.3586		730	1.2709
1095	1.5084		1095	1.3965		1095	1.3063
1460	1.5332		1460	1.4194		1460	1.3278
1825	1.5502		1825	1.4352		1825	1.3425
1825	1.7011		1825	1.5749		1825	1.4732

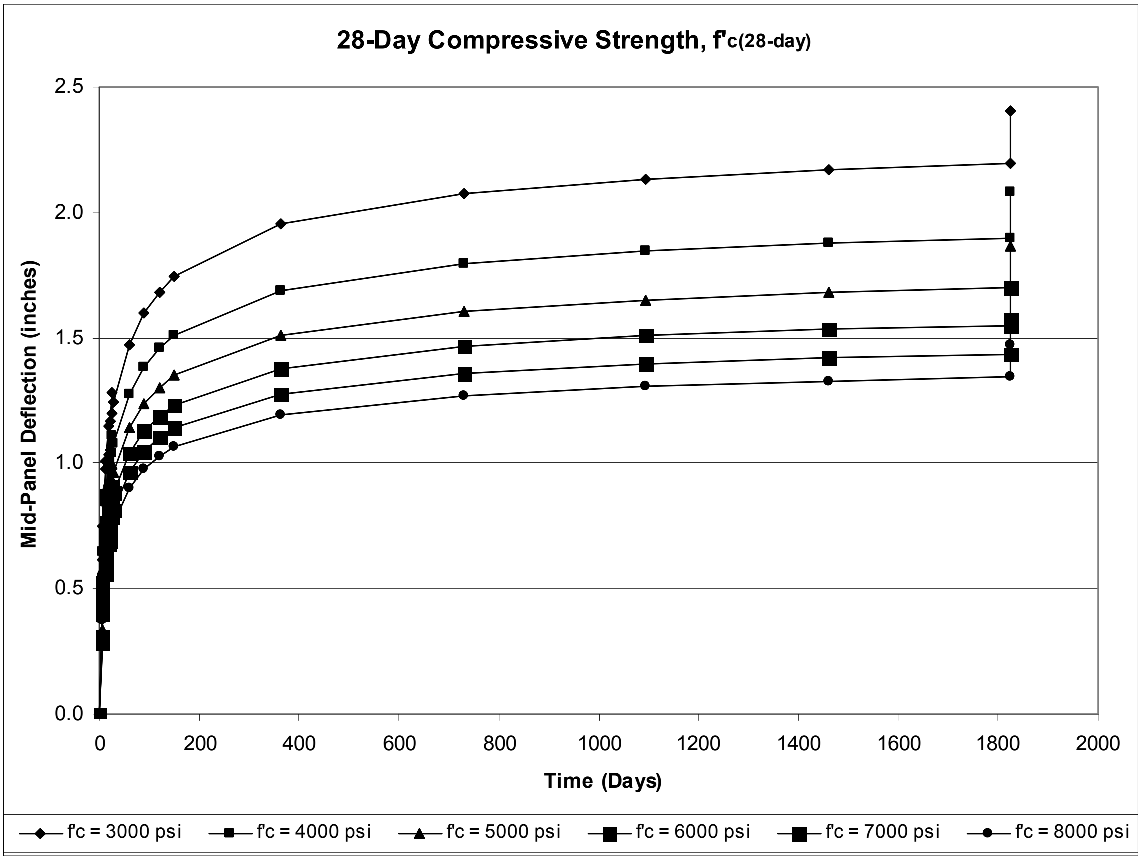

Figure 5.3 – Sensitivity to Variation in the 28-Day Compressive Strength, $f'_{c(28\text{-day})}$

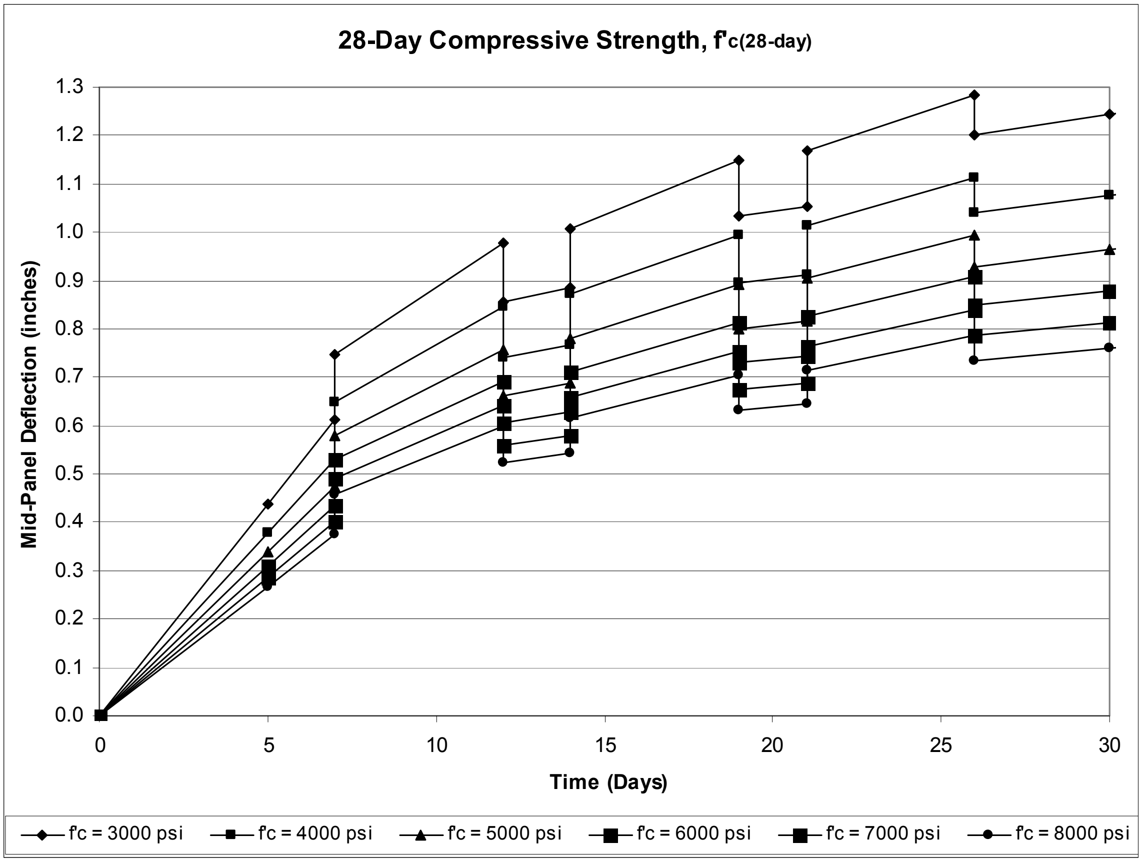

Figure 5.4 – 28-Day Compressive Strength, $f'_{c(28\text{-}day)}$, During the Construction Phase

5.6 PARAMETERS: CASTING CYLCE, STRIPPING TIME, SHORING/RESHORING SCHEME

Within the parametric study, Table 5.5 and Figure 5.5 present the sensitivity to variation in casting cycle, formwork stripping time, and shoring/reshoring scheme. Figure 5.6 focuses upon the sensitivity of these parameters during the construction phase.

Table 5.5 – Sensitivity to Variation in Casting Cycle, Stripping Time, & Shoring/Reshoring Scheme

7 day cycle, stripping at 5 days; 1 level of shoring, 2 levels of reshoring ** constant state **			7 day cycle, Stripping at 5 days; 1 level of shoring, 3 levels of reshoring			7 day cycle, Stripping at 5 days; 1 level of shoring, 4 levels of reshoring	
Time (days)	Deflection (inches)		Time (days)	Deflection (inches)		Time (days)	Deflection (inches)
0	0.0000		0	0.0000		0	0.0000
5	0.3781		5	0.3781		5	0.3781
7	0.5312		7	0.5312		7	0.5312
7	0.6482		7	0.6190		7	0.6014
12	0.8469		12	0.8004		12	0.7726
12	0.7401		12	0.7204		12	0.7085
14	0.7679		14	0.7496		14	0.7386
14	0.8725		14	0.8280		14	0.8013
19	0.9954		19	0.9363		19	0.9009
19	0.8943		19	0.8606		19	0.8403
21	0.9121		21	0.8791		21	0.8592
21	1.0122		21	0.9541		21	0.9193
26	1.1108		26	1.0391		26	0.9961
26	1.0393		26	0.9654		26	0.9371
30	1.0766		28	0.9793		28	0.9513
60	1.2764		28	1.0527		28	1.0099
90	1.3846		33	1.1254		33	1.0747
120	1.4565		33	1.0792		33	1.0167
150	1.5093		40	1.1333		35	1.0282
365	1.6900		80	1.3261		35	1.0860
730	1.7973		120	1.4275		40	1.1433
1095	1.8474		365	1.6569		40	1.1120
1460	1.8777		1095	1.8109		365	1.6328
1825	1.8986		1825	1.8611		1825	1.8338
1825	2.0834		1825	2.0459		1825	2.0186

Table 5.5 (Continued) – Sensitivity to Variation in Casting Cycle, Stripping Time, & Shoring/Reshoring Scheme

4 day cycle, stripping at 3 days; 1 level of shoring, 2 levels of reshoring			4 day cycle, Stripping at 3 days; 1 level of shoring, 3 levels of reshoring			4 day cycle, stripping at 3 days; 1 level of shoring, 4 levels of reshoring	
Time (days)	Deflection (inches)		Time (days)	Deflection (inches)		Time (days)	Deflection (inches)
0	0.0000		0	0.0000		0	0.0000
3	0.4350		3	0.4350		3	0.4350
4	0.5732		4	0.5732		4	0.5732
4	0.7067		4	0.6733		4	0.6533
7	0.9228		7	0.8719		7	0.8413
7	0.8058		7	0.7841		7	0.7711
8	0.8338		8	0.8131		8	0.8008
8	0.9478		8	0.8986		8	0.8692
11	1.0834		11	1.0194		11	0.9811
11	0.9752		11	0.9383		11	0.9162
12	0.9945		12	0.9580		12	0.9362
12	1.1013		12	1.0381		12	1.0002
15	1.2108		15	1.1337		15	1.0875
15	1.1354		15	1.0559		15	1.0253
30	1.3774		16	1.0715		16	1.0409
60	1.6204		16	1.1487		16	1.1026
90	1.7554		19	1.2308		19	1.1767
120	1.8462		19	1.1825		19	1.1160
150	1.9133		40	1.4442		20	1.1291
365	2.1451		80	1.6778		20	1.1894
730	2.2840		120	1.8044		23	1.2550
1095	2.3490		365	2.0964		23	1.2225
1460	2.3884		1095	2.2951		365	2.0614
1825	2.4156		1825	2.3601		1825	2.3202
1825	2.6004		1825	2.5449		1825	2.5050

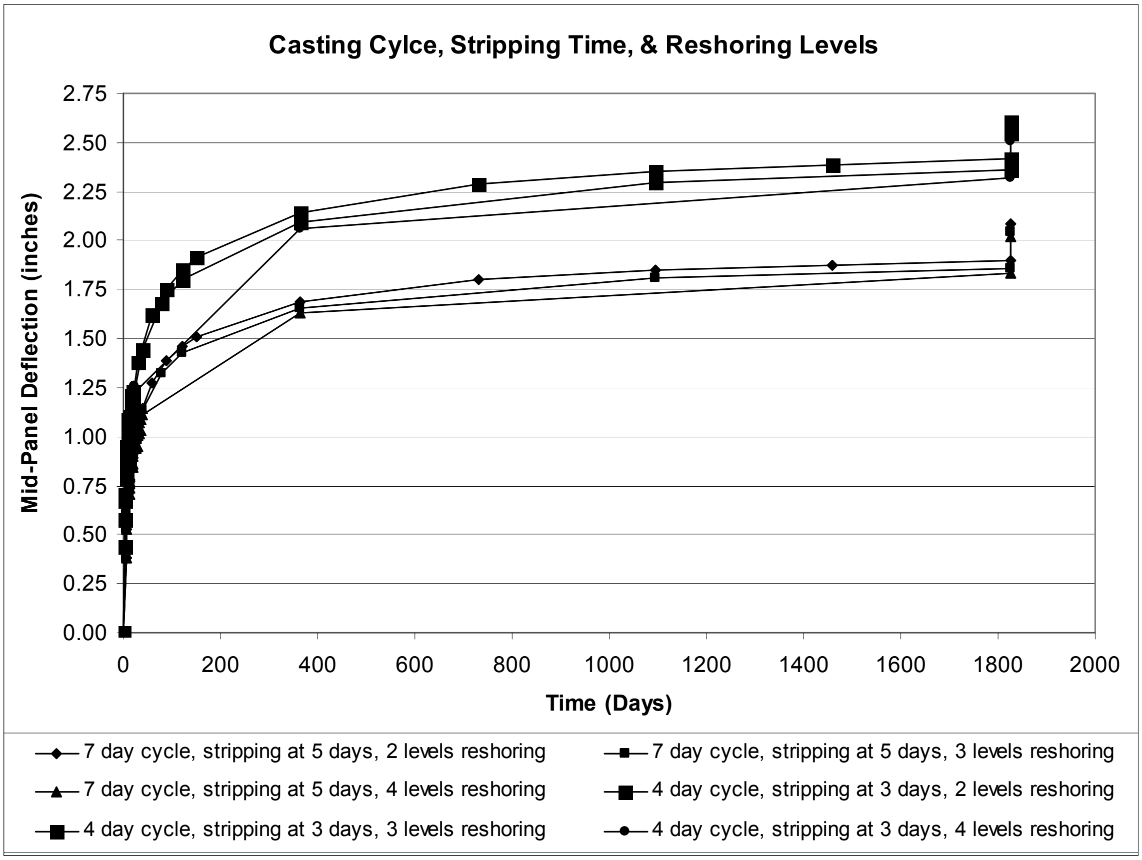

Figure 5.5 – Sensitivity to Variation in Casting Cycle, Stripping, & Reshoring

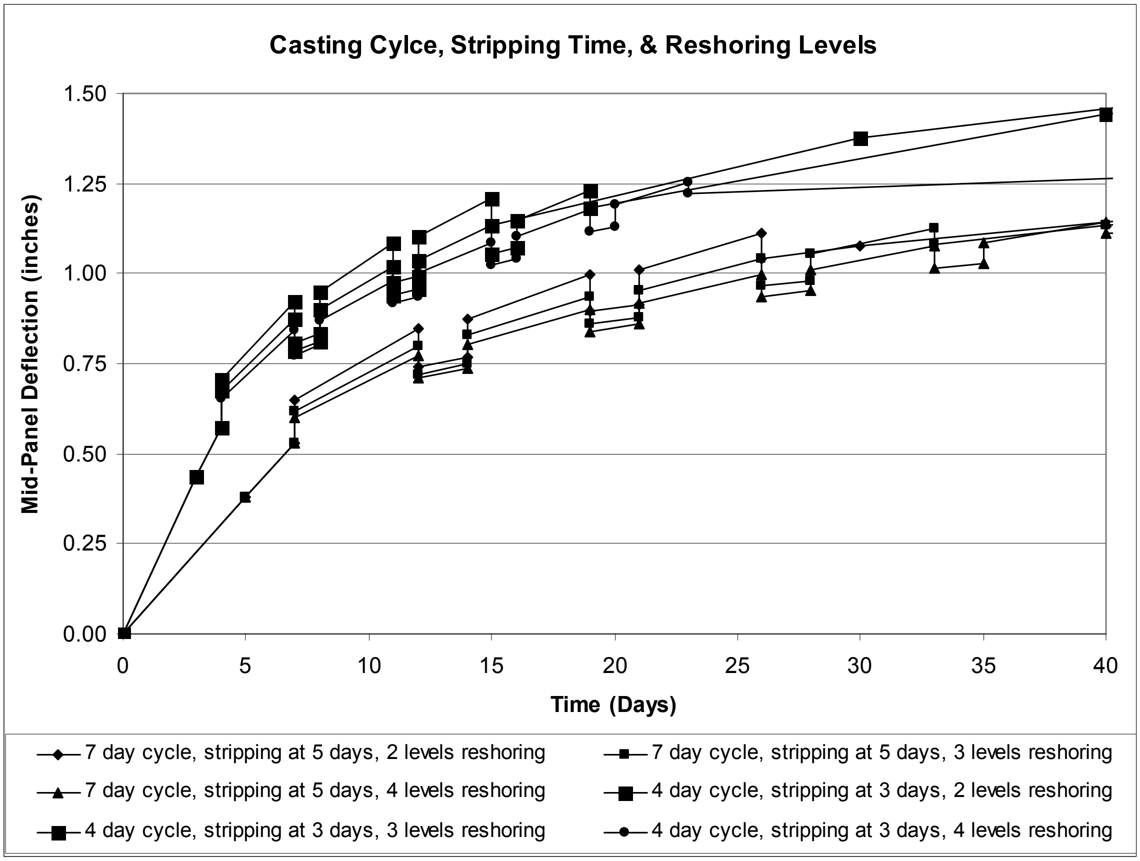

Figure 5.6 – Casting Cycle, Stripping, & Reshoring During the Construction Phase

5.7 PARAMETER: SHORT DIRECTION CLEAR SPAN, l_2

Within the parametric study, Table 5.6 and Figure 5.7 present the sensitivity to variation in the short direction clear span, l_2. Figure 5.8 focuses upon the sensitivity of this parameter during the construction phase.

Table 5.6 – Sensitivity to Variation in the Short Direction Clear Span, l_2

$l_1 = 19, l_2 = 19$ ** constant state **			$l_1 = 19, l_2 = 14$			$l_1 = 19, l_2 = 9$	
Time (days)	Deflection (inches)		Time (days)	Deflection (inches)		Time (days)	Deflection (inches)
0	0.0000		0	0.0000		0	0.0000
5	0.3781		5	0.3219		5	0.3072
7	0.5312		7	0.4522		7	0.4316
7	0.6482		7	0.5518		7	0.5266
12	0.8469		12	0.7209		12	0.6880
12	0.7401		12	0.6300		12	0.6013
14	0.7679		14	0.6537		14	0.6239
14	0.8725		14	0.7427		14	0.7088
19	0.9954		19	0.8473		19	0.8086
19	0.8943		19	0.7613		19	0.7265
21	0.9121		21	0.7764		21	0.7409
21	1.0122		21	0.8616		21	0.8223
26	1.1108		26	0.9456		26	0.9024
26	1.0393		26	0.8847		26	0.8443
30	1.0766		30	0.9164		30	0.8746
60	1.2764		60	1.0865		60	1.0369
90	1.3846		90	1.1786		90	1.1248
120	1.4565		120	1.2399		120	1.1832
150	1.5093		150	1.2848		150	1.2261
365	1.6900		365	1.4386		365	1.3729
730	1.7973		730	1.5300		730	1.4601
1095	1.8474		1095	1.5726		1095	1.5007
1460	1.8777		1460	1.5984		1460	1.5254
1825	1.8986		1825	1.6162		1825	1.5424
1825	2.0834		1825	1.7735		1825	1.6925

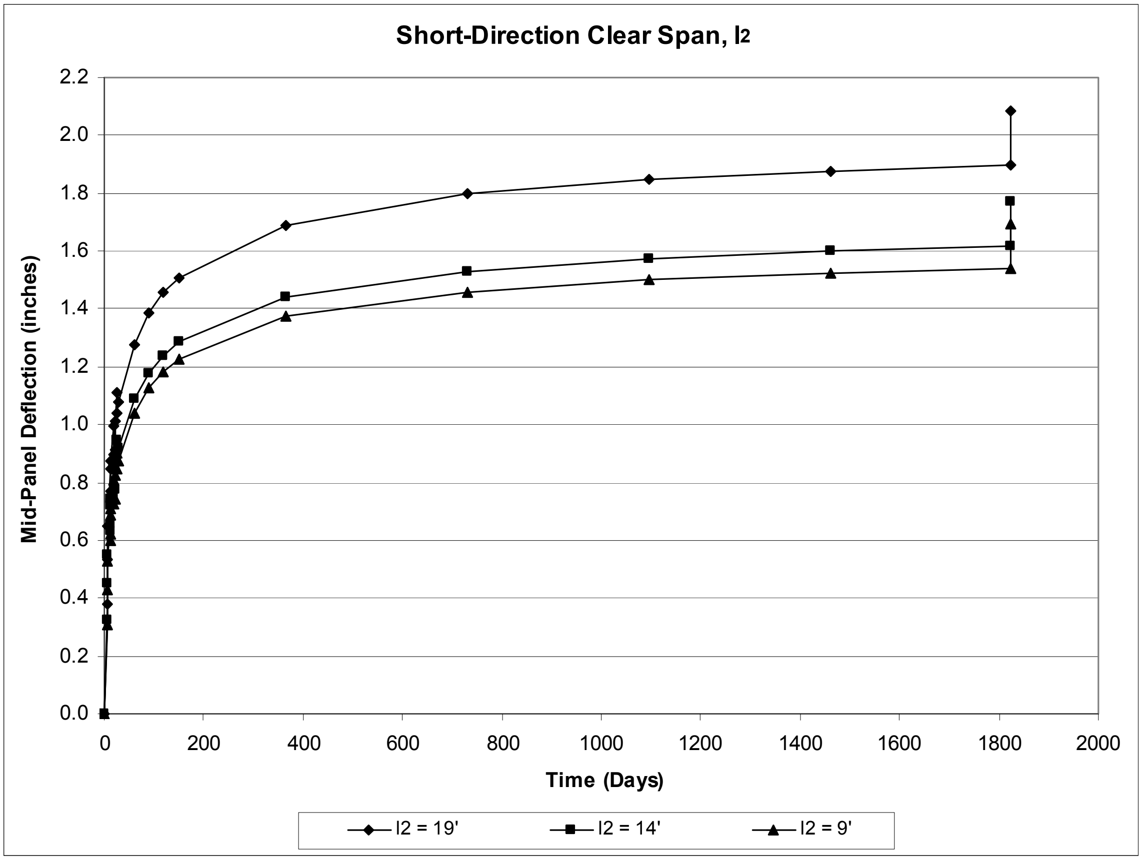

Figure 5.7 – Sensitivity to Variation in the Short-Direction Clear Span, l_2

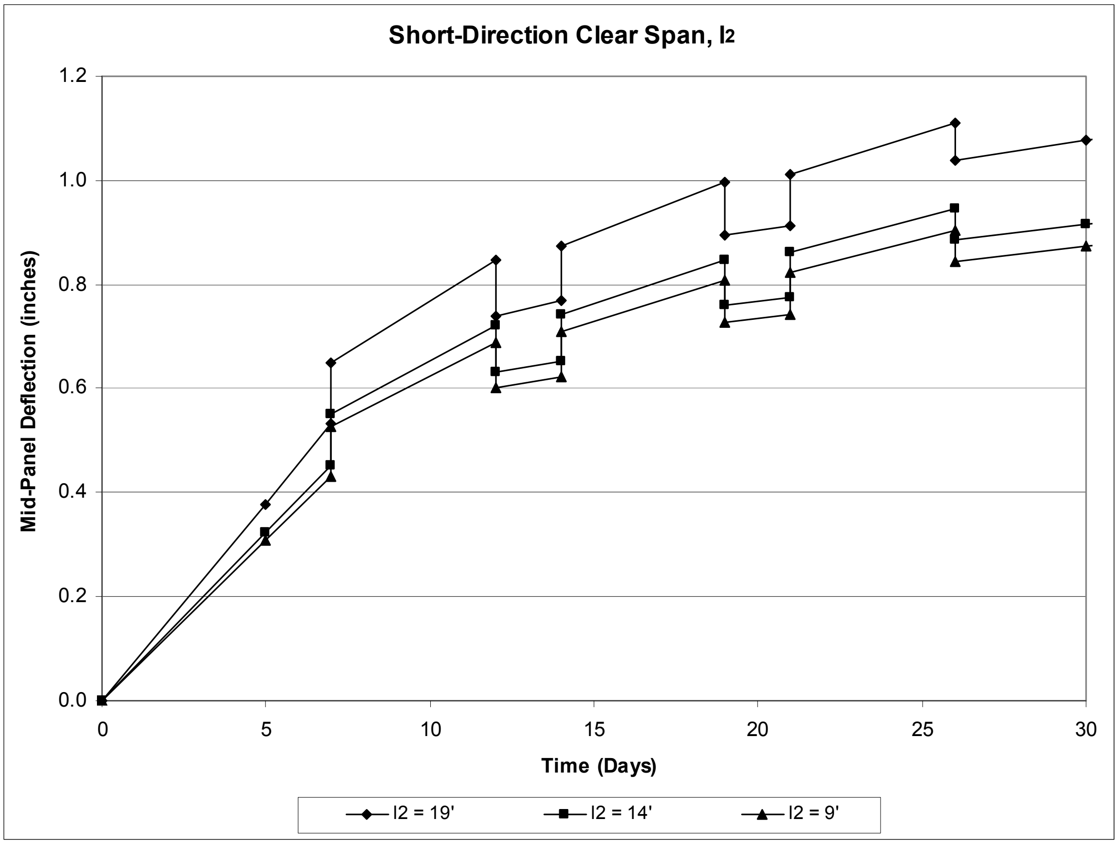

Figure 5.8 – Short-Direction Clear Span, l_2, During the Construction Phase

5.8 PARAMETER: SLAB THICKNESS, h

Within the parametric study, Table 5.7 and Figure 5.9 present the sensitivity to variation in the slab thickness, h. Figure 5.10 focuses upon the sensitivity of this parameter during the construction phase.

Table 5.7 – Sensitivity to the Variation in the Slab Thickness, h

h = 6"			h = 7" ** constant state **			h = 8"	
Time (days)	Deflection (inches)		Time (days)	Deflection (inches)		Time (days)	Deflection (inches)
0	0.0000		0	0.0000		0	0.0000
5	0.5147		5	0.3781		5	0.2895
7	0.7231		7	0.5312		7	0.4067
7	0.8823		7	0.6482		7	0.4963
12	1.1527		12	0.8469		12	0.6484
12	1.0074		12	0.7401		12	0.5667
14	1.0453		14	0.7679		14	0.5880
14	1.1876		14	0.8725		14	0.6680
19	1.3548		19	0.9954		19	0.7621
19	1.2173		19	0.8943		19	0.6847
21	1.2415		21	0.9121		21	0.6983
21	1.3777		21	1.0122		21	0.7749
26	1.5120		26	1.1108		26	0.8505
26	1.4323		26	1.0393		26	0.7883
30	1.4864		30	1.0766		30	0.8154
60	1.7632		60	1.2764		60	0.9663
90	1.9122		90	1.3846		90	1.0485
120	2.0111		120	1.4565		120	1.1031
150	2.0836		150	1.5093		150	1.1432
365	2.3318		365	1.6900		365	1.2806
730	2.4791		730	1.7973		730	1.3622
1095	2.5478		1095	1.8474		1095	1.4003
1460	2.5895		1460	1.8777		1460	1.4234
1825	2.6182		1825	1.8986		1825	1.4393
1825	2.9116		1825	2.0834		1825	1.5631

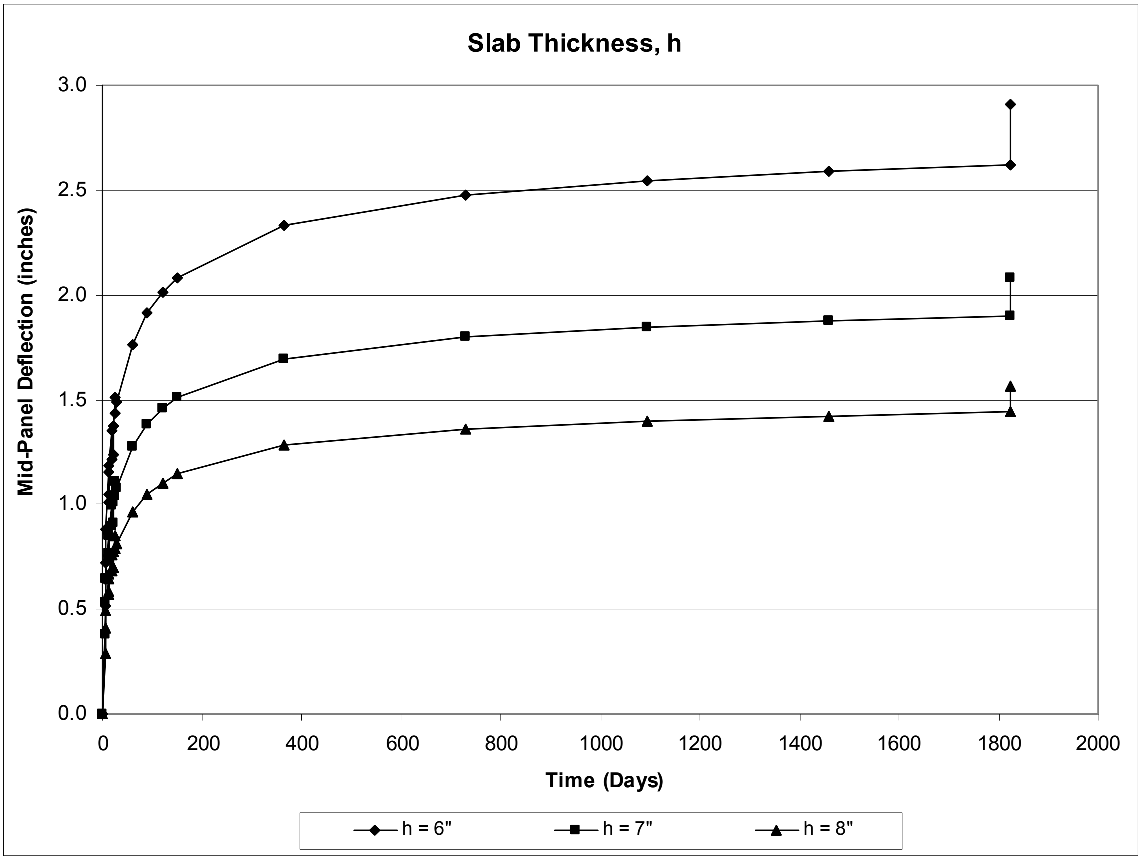

Figure 5.9 – Sensitivity to Variation in the Slab Thickness, h

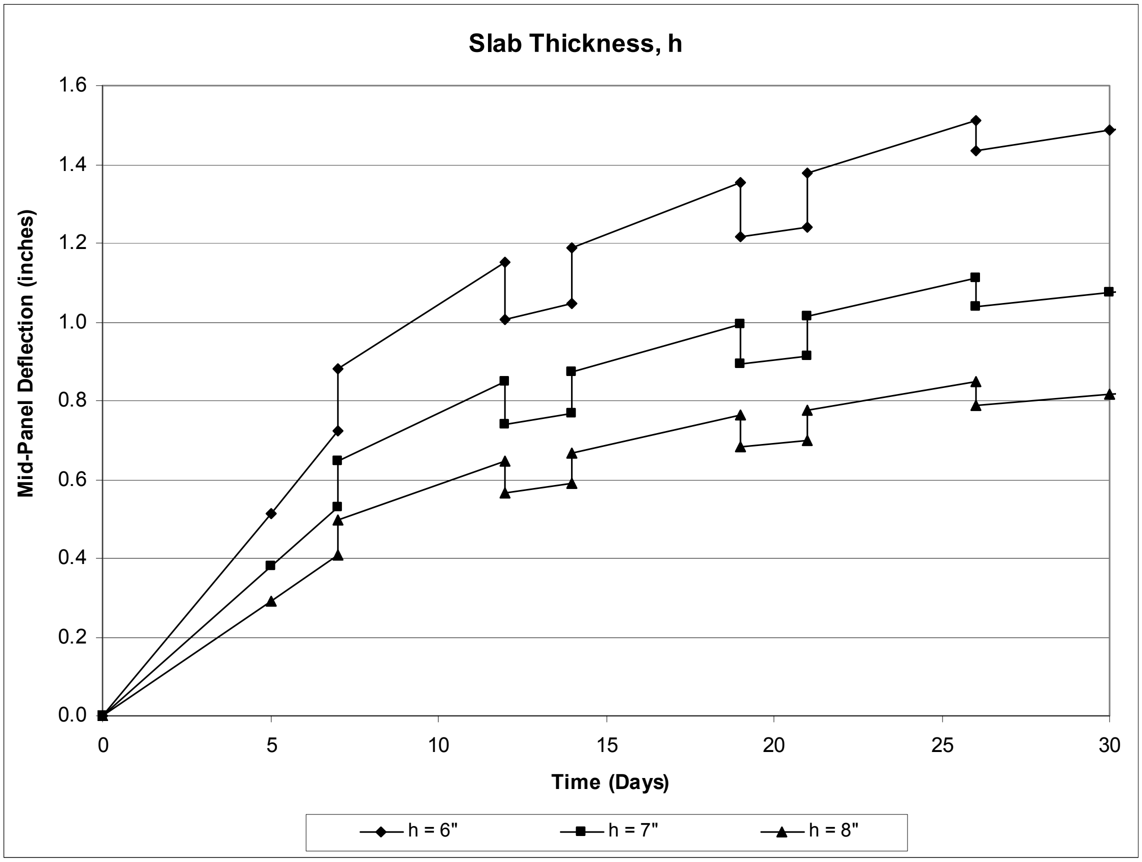

Figure 5.10 – Slab Thickness, h, During the Construction Phase

5.9 PARAMETER: CREEP RECOVERY FACTOR

Within the parametric study, Table 5.8 and Figure 5.11 present the sensitivity to variation in the creep recovery factor. Figure 5.12 focuses upon the sensitivity of this parameter during the construction phase.

Table 5.8 – Sensitivity to Variation in the Creep Recovery Factor

creep recovery = 0.5 ** constant state **			creep recovery = 0.7			creep recovery = 0.9	
Time (days)	Deflection (inches)		Time (days)	Deflection (inches)		Time (days)	Deflection (inches)
0	0.0000		0	0.0000		0	0.0000
5	0.3781		5	0.3781		5	0.3781
7	0.5312		7	0.5312		7	0.5312
7	0.6482		7	0.6482		7	0.6482
12	0.8469		12	0.8469		12	0.8469
12	0.7401		12	0.7401		12	0.7401
14	0.7679		14	0.7610		14	0.7541
14	0.8725		14	0.8656		14	0.8586
19	0.9954		19	0.9826		19	0.9697
19	0.8943		19	0.8815		19	0.8687
21	0.9121		21	0.8919		21	0.8717
21	1.0122		21	0.9920		21	0.9717
26	1.1108		26	1.0827		26	1.0546
26	1.0393		26	1.0112		26	0.9831
30	1.0766		30	1.0388		30	1.0011
60	1.2764		60	1.2151		60	1.1538
90	1.3846		90	1.3131		90	1.2416
120	1.4565		120	1.3786		120	1.3007
150	1.5093		150	1.4268		150	1.3444
365	1.6900		365	1.5927		365	1.4953
730	1.7973		730	1.6914		730	1.5856
1095	1.8474		1095	1.7376		1095	1.6278
1460	1.8777		1460	1.7656		1460	1.6534
1825	1.8986		1825	1.7849		1825	1.6711
1825	2.0834		1825	1.9697		1825	1.8559

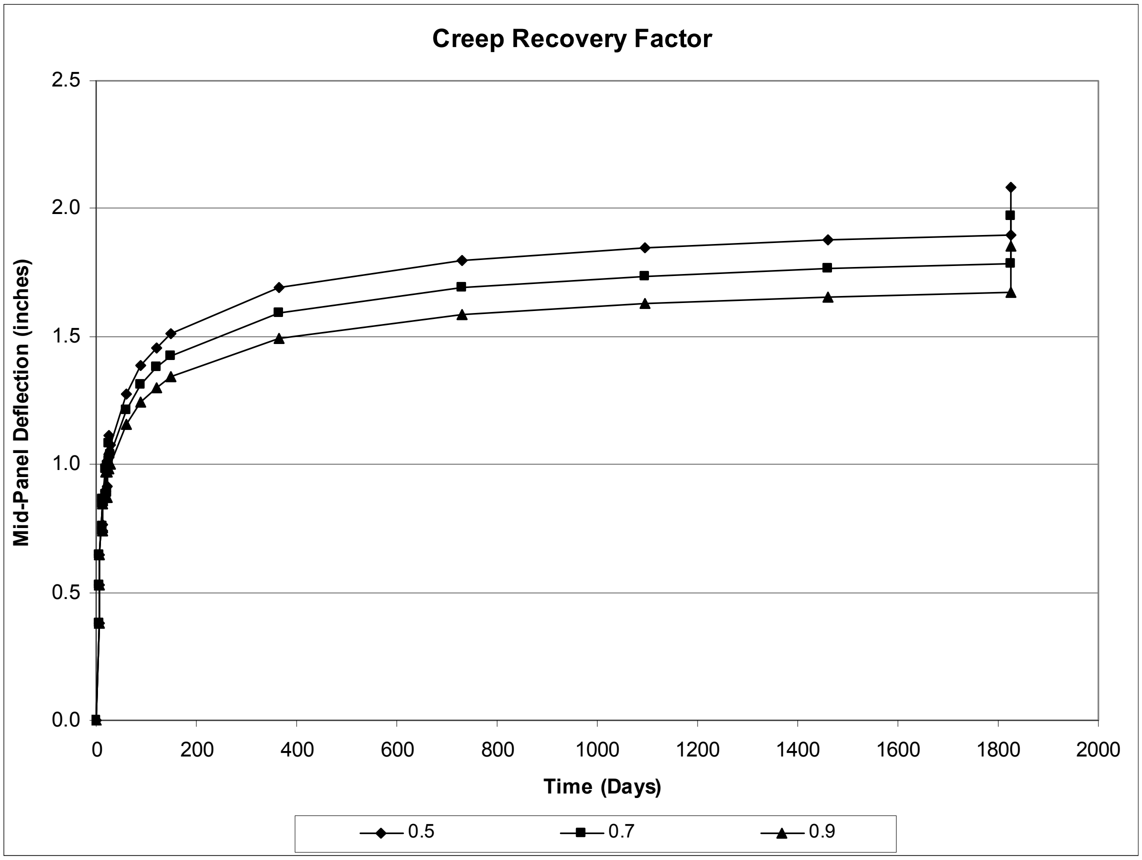

Figure 5.11 – Sensitivity to Variation in the Creep Recovery Factor

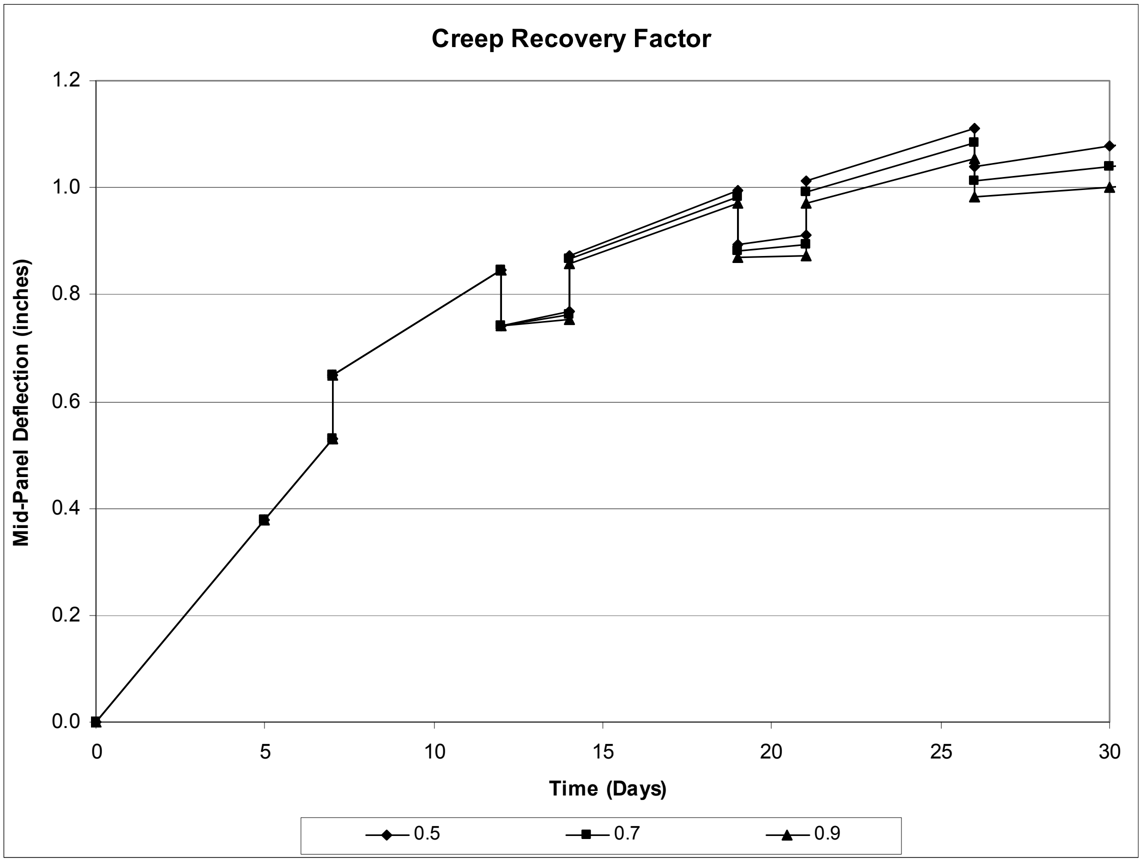

Figure 5.12 – Creep Recover Factor During the Construction Phase

5.10 PARAMETER: LOADING AGE CREEP CORRECTION EQUATION

Within the parametric study, Table 5.9 and Figure 5.13 present the sensitivity to variation in the selection of the loading age creep correction equation. Figure 5.14 focuses upon the sensitivity of this parameter during the construction phase.

Table 5.9 – Sensitivity to Variation in the Loading Age Creep Correction Equation

Ghosh ** constant state **			ACI 209 Moist Cured	
Time (days)	Deflection (inches)		Time (days)	Deflection (inches)
0	0.0000		0	0.0000
5	0.3781		5	0.3781
7	0.5312		7	0.4810
7	0.6482		7	0.5980
12	0.8469		12	0.7336
12	0.7401		12	0.6269
14	0.7679		14	0.6445
14	0.8725		14	0.7491
19	0.9954		19	0.8362
19	0.8943		19	0.7351
21	0.9121		21	0.7459
21	1.0122		21	0.8460
26	1.1108		26	0.9183
26	1.0393		26	0.8468
30	1.0766		30	0.8716
60	1.2764		60	1.0096
90	1.3846		90	1.0843
120	1.4565		120	1.1339
150	1.5093		150	1.1703
365	1.6900		365	1.2949
730	1.7973		730	1.3688
1095	1.8474		1095	1.4033
1460	1.8777		1460	1.4243
1825	1.8986		1825	1.4386
1825	2.0834		1825	1.6235

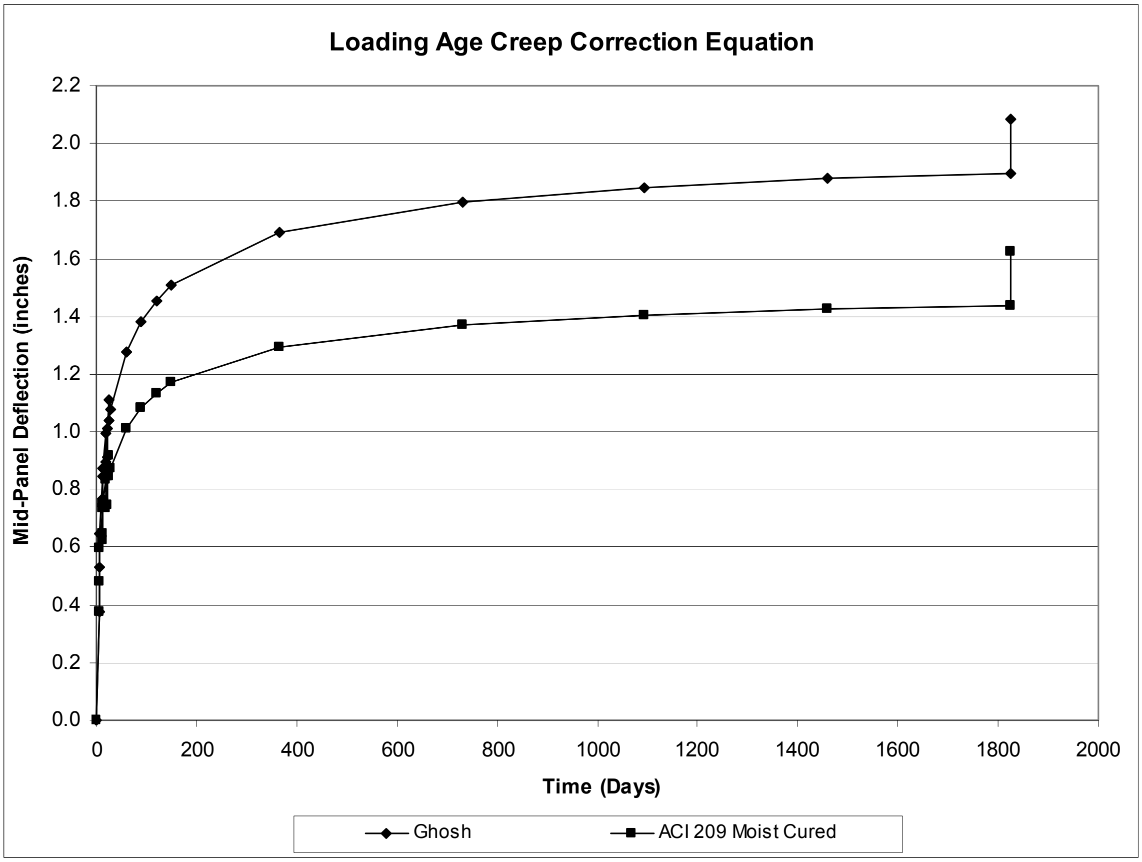

Figure 5.13 – Sensitivity to Variation in the Loading Age Creep Correction Equation

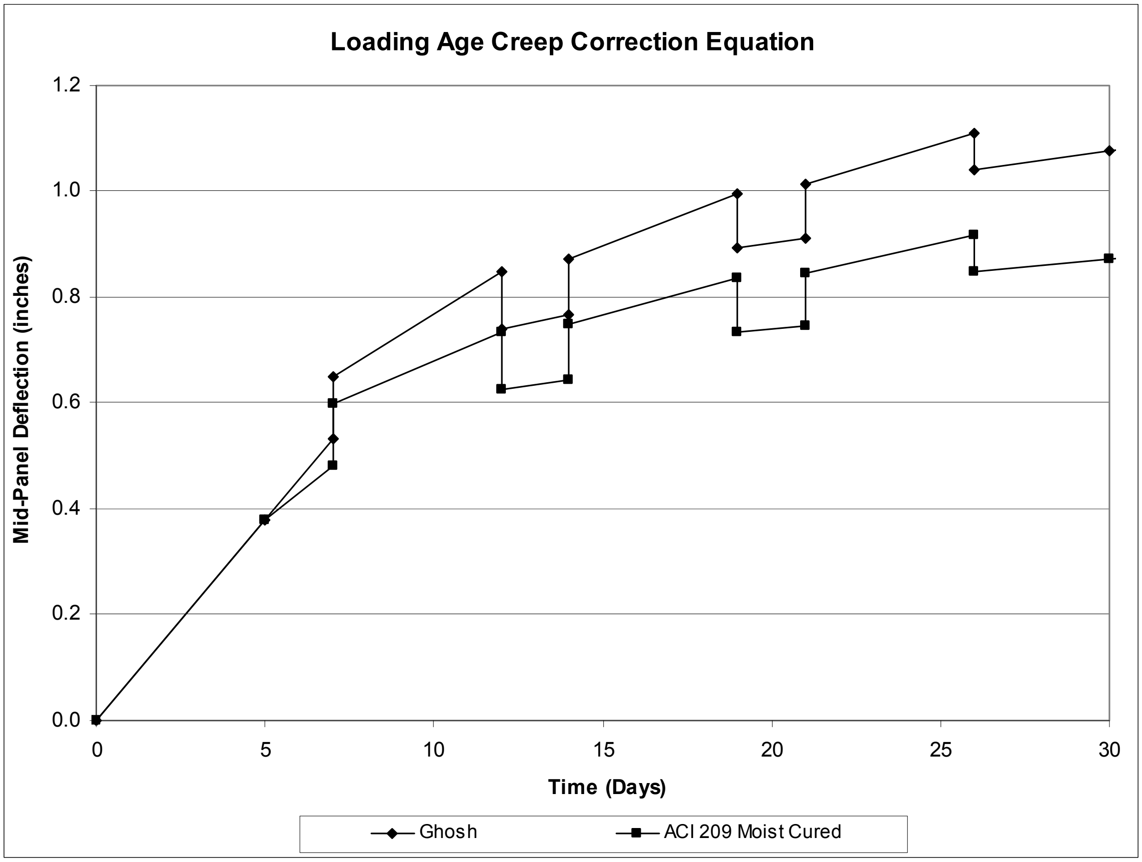

Figure 5.14 – Loading Age Creep Correction Equation During the Construction Phase

5.11 PARAMETER: PANEL LOCATION

Within the parametric study, Table 5.10 and Figure 5.15 present the sensitivity to variation in the panel location. Figure 5.16 focuses upon the sensitivity of this parameter during the construction phase.

Table 5.10 – Sensitivity to Variation in the Panel Location

Interior Panel kbc col = 1.4 kbc mid = 1.4 * constant state *		Edge Panel kbc col = 2.0 kbc mid = 1.4		Edge Panel kbc col = 1.4 kbc mid = 2.0		Corner Panel kbc col = 2.0 kbc mid = 2.0	
Time (days)	Deflection (inches)	Time (days)	Deflection (inches)	Time (days)	Deflection (inches)	Time (days)	Deflection (inches)
0	0	0	0	0	0	0	0
5	0.3781	5	0.5087	5	0.4096	5	0.5402
7	0.5312	7	0.7147	7	0.5754	7	0.7589
7	0.6482	7	0.8721	7	0.7021	7	0.926
12	0.8469	12	1.1394	12	0.9173	12	1.2098
12	0.7401	12	0.9958	12	0.8017	12	1.0573
14	0.7679	14	1.0332	14	0.8318	14	1.0971
14	0.8725	14	1.1739	14	0.9451	14	1.2465
19	0.9954	19	1.3392	19	1.0782	19	1.422
19	0.8943	19	1.2033	19	0.9687	19	1.2776
21	0.9121	21	1.2271	21	0.9879	21	1.303
21	1.0122	21	1.3618	21	1.0963	21	1.446
26	1.1108	26	1.4945	26	1.2032	26	1.5869
26	1.0393	26	1.3983	26	1.1258	26	1.4848
30	1.0766	30	1.4484	30	1.1661	30	1.5379
60	1.2764	60	1.7173	60	1.3825	60	1.8234
90	1.3846	90	1.8629	90	1.4997	90	1.978
120	1.4565	120	1.9596	120	1.5777	120	2.0808
150	1.5093	150	2.0306	150	1.6348	150	2.1561
365	1.69	365	2.2737	365	1.8305	365	2.4143
730	1.7973	730	2.4181	730	1.9468	730	2.5676
1095	1.8474	1095	2.4855	1095	2.001	1095	2.6391
1460	1.8777	1460	2.5263	1460	2.0339	1460	2.6825
1825	1.8986	1825	2.5545	1825	2.0565	1825	2.7123
1825	2.0834	1825	2.8031	1825	2.2567	1825	2.9763

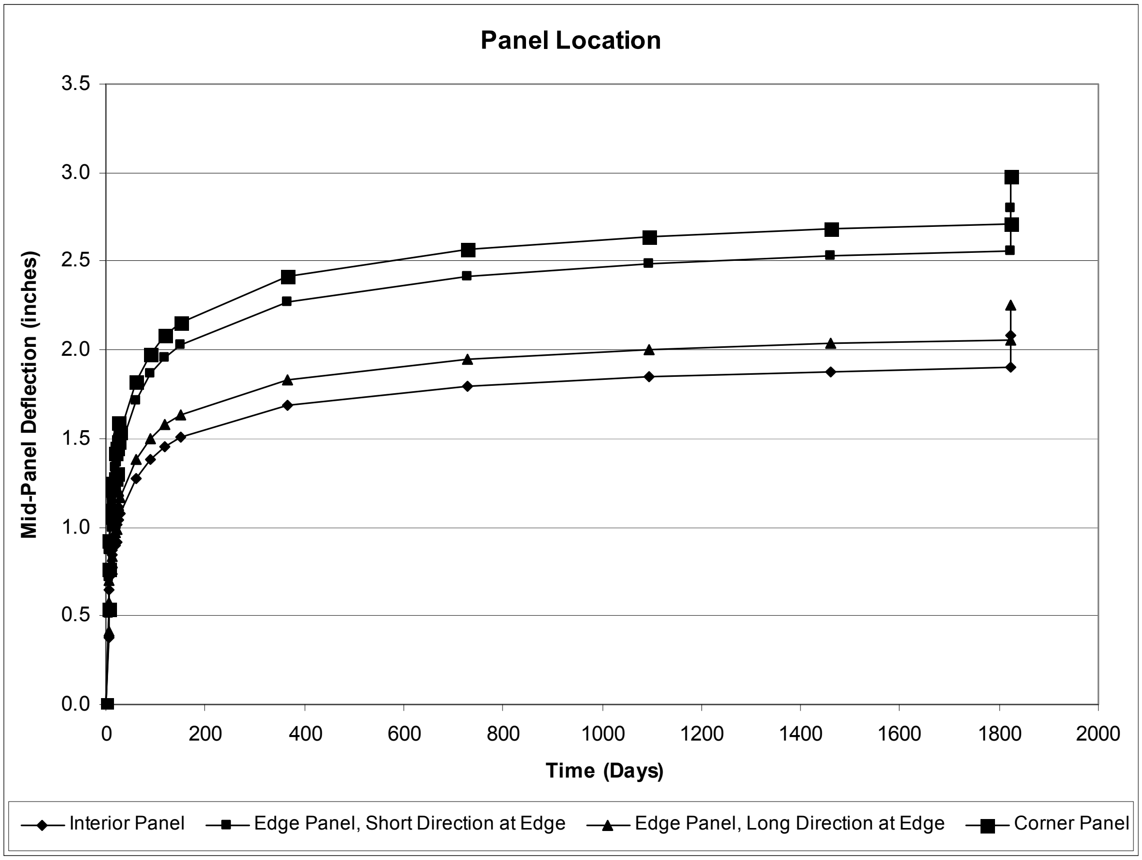

Figure 5.15 – Sensitivity to Variation in the Panel Location

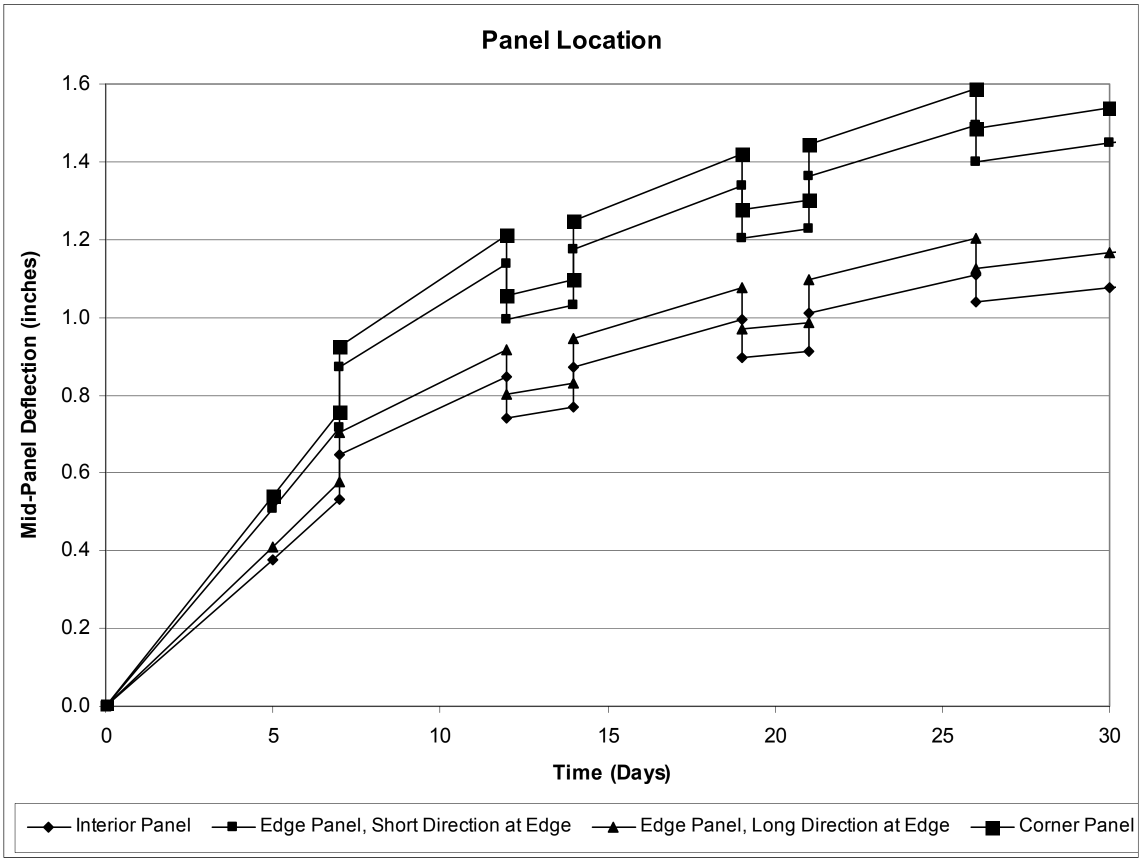

Figure 5.16 – Panel Location During the Construction Phase

5.12 SIGNIFICANCE OF CONSTRUCTION LOAD ANALYSIS

This section demonstrates the effect that construction load analysis has on deflections. The results of an analysis for the constant state (detailed within section 5.2) are compared to the results of an analysis that excludes the effects of construction loading. Because the computer algorithm was written to incorporate the effects of construction loading, a specific loading-time history is input to exclude the effects of construction loading. For this loading-time history, construction loads are set to zero. The sustained load is applied at the 28-day mark and is computed using equation (2-15) with a superimposed dead load of 20psf, a live load of 80psf, and a slab self-weight of 87.5psf (corresponding to a unit weight of concrete of 150pcf and a slab thickness of seven inches). The superimposed dead load, live load, slab thickness, and concrete unit weight values are consistent with the constants specified in section 5.2. The maximum load is computed using equation (2-16) and is applied at the 5-year mark. Table 5.11 shows the loading-time history associated with the analysis excluding the effects of construction loading. This loading-time history serves as 25 pairings of loading-time input for the computer program. Table 5.12 and Figure 5.17 provide the results of the comparison.

Table 5.11 – Loading-Time History for the Exclusion of Construction Load Effects

Time (days)	Loading (psf)	Time (days)	Loading (psf)	Time (days)	Loading (psf)
0	0.0	140	115.5	320	115.5
20	0.0	160	115.5	365	115.5
28	0.0	180	115.5	730	115.5
28	115.5	200	115.5	1095	115.5
40	115.5	220	115.5	1460	115.5
60	115.5	240	115.5	1825	115.5
80	115.5	260	115.5	1825	187.5
100	115.5	280	115.5		
120	115.5	300	115.5		

Table 5.12 – Sensitivity to the Inclusion or Exclusion of Construction Load Effects

Including Construction Load Effects ** constant state **			Excluding Construction Load Effects	
Time (days)	Deflection (inches)		Time (days)	Deflection (inches)
0	0.0000		0	0.0000
5	0.3781		20	0.0000
7	0.5312		28	0.0000
7	0.6482		28	0.3200
12	0.8469		40	0.5168
12	0.7401		60	0.6044
14	0.7679		80	0.6508
14	0.8725		100	0.6818
19	0.9954		120	0.7047
19	0.8943		140	0.7226
21	0.9121		160	0.7371
21	1.0122		180	0.7492
26	1.1108		200	0.7596
26	1.0393		220	0.7685
30	1.0766		240	0.7764
60	1.2764		260	0.7834
90	1.3846		280	0.7897
120	1.4565		300	0.7953
150	1.5093		320	0.8005
365	1.6900		365	0.8106
730	1.7973		730	0.8550
1095	1.8474		1095	0.8752
1460	1.8777		1460	0.8874
1825	1.8986		1825	0.8957
1825	2.0834		1825	1.0805

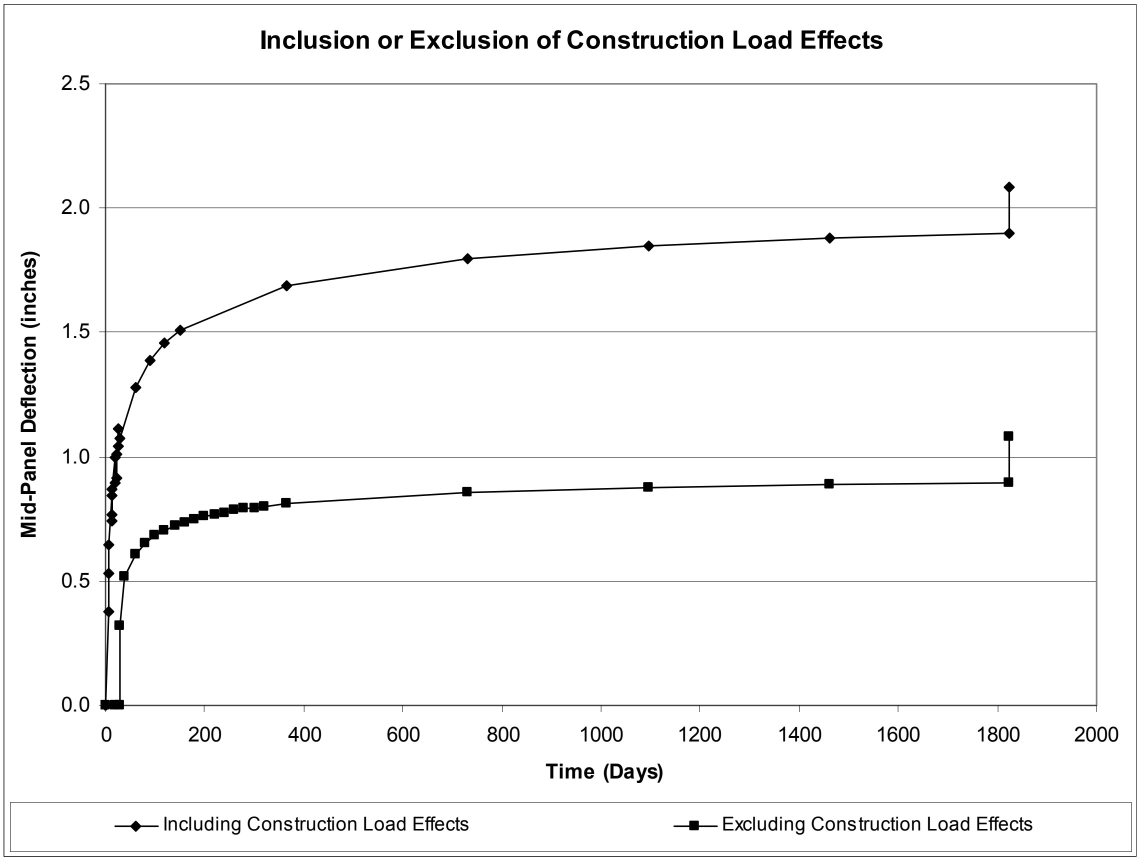

Figure 5.17 – Sensitivity to the Inclusion or Exclusion of Construction Load Effects

5.13 CONCLUSION

Within the scope of the parametric study, the algorithm is extremely sensitive to alterations in the average creep multiplier, λ_c. Changing the average creep multiplier from 1.0 to 3.0 corresponds to one-year and three-year deflections that more than double.

Changing the 28-day concrete compressive strength from 3000psi to 8000psi causes an approximate 39% decrease in deflections. Changing the compressive strength from 4000psi to 6000psi causes an approximate 18% decrease in deflections. The algorithm displays moderate sensitivity to 28-day concrete compressive strength.

The algorithm shows extremely minimal sensitivity to the number of levels of reshoring being used. Increasing the number of levels of reshoring from two to four decreases one-year and three-year deflections by less than 5%. Using a four-day casting cycle with stripping at three days rather than a seven-day casting cycle with stripping at five days serves to reduce one-year and three-year deflections by approximately 20%.

Changing l_2 from 19 feet to nine feet reduces deflections by approximately 19%. l_1 is held constant at 19 feet. Therefore, an l_2 value of 19 feet represents a square slab. An l_2 value of nine feet approximately represents the limiting case of two-way slab behavior.

The algorithm is very sensitive to changes in slab thickness, h. Increasing the slab thickness from six inches to eight inches reduces deflections by approximately 45%.

The algorithm displays minor sensitivity to variation in the value of the creep recovery factor. Increasing the creep recovery factor from 0.5 to 0.9 leads to an 11% decrease in one-year and three-year deflections.

Using the Ghosh equation to compute loading age creep correction factors rather than the ACI 209 moist cured equation leads to larger deflections. The algorithm displays minor-to-moderate sensitivity to this parameter. Using the ACI 209 moist cured equation rather than the Ghosh equation causes deflections in the 60-day to five-year range to decrease between 20% and 25%.

The algorithm also appears moderately sensitive to variations in panel location. Interior panel deflections are approximately 30% less than edge panel deflections.

The algorithm is extremely sensitive to the inclusion or exclusion of construction load effects. Including rather than excluding the effects of construction loading more than doubles one-year and three-year deflections. This indicates the importance of including construction load effects in the design of two-way slabs.

CHAPTER 6
CONCLUSIONS & RECOMMEDATIONS

6.1 SUMMARY

A thorough literature review was conducted on the topics of construction load analysis and two-way slab analysis. An analytical algorithm was generated based upon information garnered within the literature review. The analytical algorithm was written as a computer program using Microsoft Visual Basic C++ Standard Edition 2003. Algorithmic results were compared to experimental results to provide validity to the established analytical methodology. A parametric study was conducted to determine the sensitivity level for each of the major algorithmic input parameters involved.

6.2 CONSTRUCTION LOAD ANALYSIS & TWO-WAY SLAB ANALYSIS

Construction loads are applied during the construction phase. Load ratios may be determined using the Grundy and Kabaila (1963) simplified method. Sustained loads include 10% of the live load and are applied once the construction cycle is complete. Maximum loads include the full live load and are applied at the five-year mark.

Two-way slab serviceability analysis relies upon the decomposition of the slab into orthogonal column and middle strips in accordance with the crossing beam method outlined by Scanlon and Murray (1982). Correction factors are included to accommodate the effects of boundary conditions, drop panels if present, strip width, and slab strip moment intensity. A modified moment of inertia is used to accommodate the effects of concrete cracking and subsequent sectional resistance to bending.

Time-dependent concrete properties are used in place of standard 28-day values. The effects of concrete creep are used in the computation of long-time multipliers. A creep recovery factor is applied to any long-time multiplier associated with a negative loading change. Long-time multipliers are applied to instantaneous deflections to compute long-time deflections. Instantaneous deflections are computed using the basic beam deflection equation obtained based upon elastic beam theory.

6.3 METHOD OF ANALYSIS

The principles of the Grundy and Kabaila (1963) simplified method are used to compute load ratios for construction schemes involving one level of shoring with two levels of reshoring, one level of shoring with three levels of reshoring, and one level of shoring with four levels of reshoring. The computer program either allows the user to input a prescribed loading-time history or allows the program to generate a loading-time history based upon the specified shoring/reshoring scheme. The computer program is interactive and relies upon user input to then output the mid-panel deflection-time history of the two-way reinforced concrete slab.

6.4 VALIDITY OF THE ALGORITHM

Algorithmic results were compared to experimental research conducted by Guo and Gilbert (2002). Good correlation exists between the algorithmic results and the experimental results, suggesting that the algorithm is capable of performing two-way slab serviceability analysis. The findings suggest that the C++ computer program reasonably estimates the deflection-time history for reinforced concrete floor slabs.

6.5 PARAMETRIC STUDY

The parametric study suggests that the algorithm is highly sensitive to alteration in the average creep multiplier, λ_c, and the slab thickness, h. The algorithm appears moderately sensitive to 28-day concrete compressive strength, $f'_{c(28\text{-day})}$, and panel location. The algorithm displays minor-to-moderate sensitivity to the selection of the loading age creep correction equation. Minor sensitivity exists for the short-direction clear span, l_2, and the creep recovery factor. Minor sensitivity also exists for casting cycle and stripping time, while very minimal sensitivity exists for the number of levels of reshoring. Therefore, the parametric study suggests that two-way slab deflections are most influenced by the average creep multiplier, λ_c, and the slab thickness, h.

The algorithm is highly sensitive to the inclusion or exclusion of construction load effects, thereby demonstrating the overall significance of performing construction load analysis procedures.

6.6 RECOMMENDATIONS

The computer algorithm outlined in this study can be used to estimate the effects of construction procedures on the long-time deflections of two-way slabs. The input can be adjusted to accommodate specific data on concrete properties when available. Otherwise standard properties based on ACI recommendations can be used.

This study focuses upon the development of a method of analysis capable of computing the mid-panel deflection-time history of two-way reinforced concrete floor slab in multi-story high-rise buildings. The method of analysis developed within this study is intended to serve as an approximate analysis procedure. The associated computer program is intended as a design aid that may be used to assist an engineer or construction team in analyzing and designing two-way slabs.

More refined analysis procedures could be used to perform two-way slab analysis. Experimental testing procedures performed for two-way slab deflections display significant inherent variability in measured results. Such laboratory procedures are conducted under controlled conditions, suggesting that two-way slab deflection behavior is associated with significant inherent variability. For this reason, the scope of this study is limited to the development of an approximate analysis procedure. More refined analysis procedures could be performed but arguably would be ill-suited for modeling a process associated with such a significant degree of inherent variability.

The method of analysis developed within this study is implemented as a deterministic computer program, meaning that the same input will always produce the same output. It is possible to incorporate the effects of inherent variability into a non-deterministic computer program. One such way to model this variability would be to utilize Monte Carlo simulation in order to incorporate random elements of chance.

REFERENCES

ACI Committee 209, 1992, "Prediction of Creep, Shrinkage, and Temperature Effects in Concrete Structures (ACI 209R-92)," American Concrete Institute, Farmington Hills, MI, 47 pp.

ACI Committee 209, 2008, "Guide for Modeling and Calculating Shrinkage and Creep in Hardened Concrete (ACI 209.2R-08)," American Concrete Institute, Farmington Hills, MI, 44 pp.

ACI Committee 318, 2005, "Building Code Requirements for Structural Concrete (ACI 318-05) and Commentary (ACI 318R-05)," American Concrete Institute, Farmington Hills, MI, 430 pp.

ACI Committee 435, 1991, "State-of-the-Art Report on Control of Two-Way Slab Deflections (ACI 435.9R-91)," American Concrete Institute, Farmington Hills, MI, 14 pp.

Aguinaga-Zapata, Manuel, and Bazant, Zdenek P., Sept-Oct 1986, "Creep Deflections in Slab Buildings and Forces in Shores during Construction," ACI Journal, *Proceedings* V. 83, No. 5, pp. 719-726.

Beresford, F.D., 1964, "An Analytical Examination of Propped Floors in Multistory Flat Plate Construction," *Constructional Review*, V. 37, No. 11, pp. 16-20.

Branson, D.E., Aug. 1963, "Instantaneous and Time-Dependent Deflections of Simple and Reinforced Concrete Beams," *HPR Publication* No. 7, Part 1, Alabama Highway Department, Bureau of Public Records, pp. 1-78.

Chang, Kuan-Yung, and Hwang, Shyh-Jiann, Feb. 1996, "Practical Estimation of Two-Way Slab Deflections," *Journal of Structural Engineering*, ASCE, New York, NY, V. 122, No. 2, pp. 150-159.

Gardner, N.J., Apr. 1985, "Shoring, Reshoring, and Safety," *Concrete International: Design & Construction*, V. 7, No. 4, pp. 28-34.

Ghosh, S.K., 1997, "Construction Loadings in High Rise Buildings," *Concrete Construction Engineering Handbook*, V. 8, No. 1, 60 pp.

Graham, Cameron J., and Scanlon, A., 1984, "Deflection of Reinforced Concrete Slabs under Construction Loading," *Deflections of Concrete Structures*, SP-86, American Concrete Institute, Detroit, MI, pp. 167-184.

Grundy, Paul, and Kabaila, A., Dec. 1963, "Construction Loads on Slabs with Shored Formwork in Multi-story Buildings," ACI Journal, *Proceedings* V. 60, No. 12, pp. 1729-1738.

Guo, X.H., and Gilbert, R.I., Feb. 2002, "An Experimental Study of Reinforced Concrete Flat Slabs under Sustained Service Loads," The University of New South Wales, Sydney, Australia, No. R-407.

Klieger, P., 1958. "Effect of Mixing and Curing Temperature on Concrete Strength," ACI Journal, *Proceedings* V. 29, No. 12, pp. 1063-1081.

Liu, Xila; Chen, Wai-Fah; and Bowman, Mark D., May 1985, "Construction Load Analysis for Concrete Structures," *Journal of Structural Engineering*, ASCE, V. 111, No. 5, pp. 1019-1036.

McHenry, D., 1943, "A New Aspect of Creep in Concrete and its Application to Design," ASTM, *Proceedings* V. 43, pp. 1069-1087.

Neville, A.M., 1981, *Properties of Concrete*, third edition, Pitman, London and Marshfield, 779 pp.

Price, W.H., 1951, "Factors Influencing Concrete Strength," ACI Journal, *Proceedings* V. 22, No. 6, pp. 417-432.

Rangan, B. Vijaya, Apr. 1976, "Prediction of Long-Term Deflections of Flat Plates and Slabs," ACI Journal, *Proceedings* V. 73, No. 4, pp. 223-226.

Sbarounis, J.A., 1984, "Multistory Flat Plate Buildings: Construction Loads and Immediate Deflections," *Concrete International: Design & Construction*, V. 6, No. 2, pp. 70-77.

Scanlon, Andrew, and Bischoff, Peter, July-Aug 2008, "Shrinkage Restraint and Loading History Effects on Deflections of Flexural Members," ACI Structural Journal, *Proceedings* V. 105, No. 4, pp. 498-506.

Scanlon, Andrew, and Murray, David W., Nov. 1982, "Practical Calculation of Two-Way Slab Deflections," *Concrete International: Design & Construction*, pp. 43-50.

Stivaros, P.C., and Halvorsen, G.T., Aug. 1992, "Construction Load Analysis of Slabs and Shores Using Microcomputers," *Concrete International: Design & Construction*, V. 14, No. 8, pp. 27-32.

APPENDIX
C++ COMPUTER PROGRAM TEXT FILE

USER INPUT:

Main:

- Select whether to allow the program to compute the deflection-time history based upon input regarding shoring/reshoring and the casting cycle (Option A) or to input a prescribed deflection-time history (Option B).
- Input the nominal 28-day compressive strength of concrete in units of psi.
- Input the slab thickness in units of inches.
- Input the long direction clear span of the panel in units of feet.
- Input the short direction clear span of the panel in units of feet.
- Input the boundary condition factor for the long direction (column strip).
- Input the boundary condition factor for the short direction (middle strip).
- Select whether or not drop panels are present.

Option A:

- Input the superimposed dead load applied to the slab in units of psf.
- Input the live load applied to the slab in units of psf.
- Input the unit weight of concrete being used in units of pcf.
- Input the number of levels of reshoring being used.
- Input the length of the casting cycle in days.

- Input the number of days after slab casting at which formwork stripping takes place.

Option B:

- Input the loading-time history for the slab in the form of 25 loading-time pairings. Time is input in days, while slab loading is input in units of psf.

Return to Main:

- Input the average creep multiplier.
- Input the creep recovery factor.
- Select whether to use the ACI 209 recommended equation or the Ghosh (1997) recommended equation to compute the creep correction factor for loading age.
    - If selecting the ACI 209 recommended equation to compute the creep correction factor for loading age, select whether the concrete was moist cured or steam cured.
- Select whether or not to include the creep correction factor for relative humidity.
    - If selecting to include the creep correction factor for relative humidity, input the relative humidity as a percent.

OUTPUT:

- The mid-panel deflection-time history for the slab appears as output in the form of 25 deflection-time pairings. Time is measured in days, while deflection is reported in units of inches.

C++ TEXT FILE:

```
// The purpose of this program is to compute the deflection-time history for any
// slab in a multi-story high rise building.  The program is applicable to
// construction schemes utilizing one level of shoring with two, three, or four
// levels of reshoring.  The results can be applied to any slab in the building,
// provided the slab is at a floor level greater than or equal to the number of
// levels of reshoring.  For example, if one level of shoring is being used in
// conjunction with four levels of reshoring, the results put forth by this
// program would apply to slabs on the fourth floor and beyond.  When using this
// program, bear in mind that a time of zero days corresponds to the time of
// casting of the slab in question.  Thus, time is kept with respect to a
// specific slab.



#include <iostream>
#include <fstream>
#include <cmath>

using namespace std;



int main()

{

        // Variables:

        int inputselector;            // variable used to determine the method of
                                      // determining the loading-time history

        float fc;                     // variable representing the nominal 28-day
                                      // compressive strength of concrete (psi)

        float thickness;              // variable representing the slab thickness in
                                      // inches

        float longdimension;          // variable representing the slab clear span in
                                      // the long direction (feet)

        float shortdimension;         // variable representing the slab clear span in
                                      // the short direction (feet)

        float b1;                     // variable representing the width of the column
                                      // strip (feet)

        float b2;                     // variable representing the width of the middle
                                      // strip (feet)

        float mominertiagrosscol;     // variable indicating the gross moment of
                                      // inertia of the column slab strip (in^4)

        float mominertiagrossmid;     // variable indicating the gross moment of
                                      // inertia of the middle slab strip (in^4)

        float alpha;                  // variable used to modify the gross moment of
                                      // inertia

        float mominertiacol;          // variable indicating the modified moment of
                                      // inertia of the column slab strip (in^4)

        float mominertiamid;          // variable indicating the modified moment of
                                      // inertia of the middle slab strip (in^4)
```

```
float kbccol;                 // boundary condition correction factor used for
                              // column strip deflection computations

float kbcmid;                 // boundary condition correction factor used for
                              // middle strip deflection computations

int droppanel;                // variable used to determine whether drop
                              // panels are present

float kdp;                    // correction factor to account for the presence
                              // of drop panels

float kwf;                    // strip width modification factor

float kss;                    // moment intensity correction factor based upon
                              // whether the analysis is for a column or
                              // middle strip

float k;                      // overall correction factor

float superimposeddl;         // variable representing the superimposed dead
                              // load (psf)

float liveload;               // variable representing the live load (psf)

float unitweight;             // variable representing the unit weight of
                              // concrete (pcf)

float deadload;               // variable representing the dead load
                              // corresponding to the slab self weight (psf)

float sustainedload;          // variable representing the sustained
                              // loading (psf)

float maxload;                // variable representing the maximum
                              // loading (psf)

float wconstruction;          // variable to indicate the distributed load
                              // on the slab strip due to construction
                              // load (lb/in)

float wsustained;             // variable to indicate the distributed load
                              // on the slab strip due to sustained
                              // load (lb/in)

float wmax;                   // variable to indicate the distributed load
                              // on the slab strip due to the maximum
                              // load (lb/in)

int reshoring;                // variable to indicate the number of levels of
                              // reshoring

int castingcycle;             // variable to indicate the length of the
                              // casting cycle in days

int shoreremoval;             // variable to indicate the number of days
                              // after casting at which formwork is removed

float avgcreepmult;           // variable used to represent the average
                              // creep multiplier when computing lambda

float creeprecovery;          // variable used to represent the creep recovery
                              // factor for negative load changes

int loadingage;               // variable used to determine whether ACI 209 or
                              // Ghosh will be used for computing the loading
                              // age correction factor

int moistorsteam;             // variable used to determine whether moist or
                              // steam curing was undertaken
```

```
float lacorrection;          // creep correction coefficient used to adjust
                             // for loading age

int includehumidity;         // variable used to determine whether to include
                             // the relative humidity creep correction factor

float relhumidity;           // variable representing the relative humidity

float rhcorrection;          // creep correction coefficient used to adjust
                             // for relative humidity

int counter;                 // variable used to fill arrays

int loadcounter;             // variable used to fill time-dependent creep
                             // deflection arrays for the loading change of
                             // interest

int timecounter;             // variable used to fill time-dependent creep
                             // deflection arrays for the time increment of
                             // interest


// Arrays:

float R[25];                 // array containing R values (floor loading
                             // ratios) at time increments of interest

int day[25];                 // array containing time periods of interest,
                             // measured in days

float w[25];                 // array containing w values of interest (lb/in)

float deltaw[25];            // array containing values for changes in w
                             // between subsequent time intervals (lb/in)

float loadinput[25];         // array containing load input values in units
                             // of psf

float fprimec[25];           // array containing values for concrete
                             // compressive strength at time increments of
                             // interest (psi)

float E[25];                 // array containing values for concrete modulus
                             // of elasticity at time increments of
                             // interest (psi)

float delta[25];             // array containing incremental deflection
                             // values for time increments of
                             // interest (inches)

float sumdelta[25];          // array containing total deflection values for
                             // time increments of interest (inches)

float moistcured[25];        // array used for the power function in order
                             // for the program to compile correctly

float steamcured[25];        // array used for the power function in order
                             // for the program to compile correctly

float ghosh[25];             // array used for the power function in order
                             // for the program to compile correctly

float raisepower[25];        // array used for the power function in order
                             // for the program to compile correctly

float lambda[25];            // array containing time-dependent long-time
                             // multipliers for time increments of interest
```

```
float creepdefl[25];          // array containing incremental time-dependent
                              // creep deflections for time increments of
                              // interest (inches)

float cumcreepdefl[25];       // array containing cumulative time-dependent
                              // creep deflections for time increments of
                              // interest (inches)

float colstripdefl[25];       // array containing total long direction column
                              // strip deflections for time increments of
                              // interest (inches)

float midstripdefl[25];       // array containing total short direction middle
                              // strip deflections for time increments of
                              // interest (inches)

float totaldefl[25];          // array containing total mid-panel deflections
                              // for time increments of interest (inches)


// Message to user:

cout << "The purpose of this program will be to evaluate the " << endl;
cout << "deflection behavior of reinforced concrete slabs in " << endl;
cout << "multi-story high rise buildings.  Specifically, the " << endl;
cout << "deflection versus time will be output for a slab of " << endl;
cout << "interest." << endl << endl;



// Loading-Time History:

cout << "This program allows the user to select from two different " << endl;
cout << "forms of input.  The user may input information regarding " << endl;
cout << "shoring and reshoring, allowing the program to compute " << endl;
cout << "information regarding the loading-time history.  Or, the " << endl;
cout << "user may choose to input the specifics of the " << endl;
cout << "loading-time history.  If you would like to input the " << endl;
cout << "specifics of the loading-time history enter the number 1 " << endl;
cout << "from the keypad.  Enter the number 0 from the keypad if " << endl;
cout << "you would like the program to compute the loading-time " << endl;
cout << "history based on information regarding shoring and " << endl;
cout << "reshoring.  Please enter the appropriate number at this " << endl;
cout << "time." << endl << endl;

cin >> inputselector;
cout << endl << endl;



// Time-Independent Concrete Properties:

cout << "Please enter the nominal 28-day compressive strength of " << endl;
cout << "the concrete being used.  This value should be in units " << endl;
cout << "of psi and should be a whole number." << endl << endl;

cin >> fc;
cout << endl << endl;



// Slab Dimensions:

cout << "Please enter the slab thickness.  This value should be in " << endl;
cout << "units of inches." << endl << endl;

cin >> thickness;
cout << endl << endl;
```

```
cout << "Please enter the long direction clear span slab " << endl;
cout << "dimension.  This dimension should be the length between " << endl;
cout << "supports and should be in units of feet." << endl << endl;

cin >> longdimension;
cout << endl << endl;

cout << "Please enter the short direction clear span slab " << endl;
cout << "dimension.  This dimension should be the length between " << endl;
cout << "supports and should be in units of feet." << endl << endl;

cin >> shortdimension;
cout << endl << endl;

b1 = shortdimension/2;
b2 = longdimension - shortdimension/2;



// Cross-Sectional Properties:

mominertiagrosscol = b1 * thickness * thickness * thickness;
mominertiagrossmid = b2 * thickness * thickness * thickness;

alpha = 0.4;

mominertiacol = alpha * mominertiagrosscol;

alpha = 0.8;

mominertiamid = alpha * mominertiagrossmid;



// Correction Factors:

cout << "Please enter the k(bc) value based on the support " << endl;
cout << "conditions.  k(bc) is the boundary condition factor.  " << endl;
cout << "Enter 5.0 for a simply supported member.  Enter 1.0 for " << endl;
cout << "a fixed end member.  Enter 1.4 for a member with both " << endl;
cout << "ends continuous.  Enter 2.0 for a member with one end " << endl;
cout << "continuous.  Please enter the k(bc) value for the column " << endl;
cout << "strip at this time." << endl << endl;

cin >> kbccol;
cout << endl << endl;

cout << "Please enter the k(bc) value for the middle strip at this " << endl;
cout << "time." << endl << endl;

cin >> kbcmid;
cout << endl << endl;


cout << "If drop panels are present please input a numerical 1.  " << endl;
cout << "If drop panels are not present please input a numerical " << endl;
cout << "2.  Please input the proper number at this time." << endl << endl;

cin >> droppanel;
cout << endl << endl;

if (droppanel == 1)
       kdp = (1 / 1.35);

if (droppanel == 2)
       kdp = 1.0;

kwf = 1.0;

kss = 1.35;
```

```
k = kbccol * kss * kwf * kdp;


if (inputselector == 0)
{

        // Loads:

        cout << "Please enter the superimposed dead load applied to " << endl;
        cout << "the slab.  This value should be in units of psf." << endl;
        cout << endl;

        cin >> superimposeddl;
        cout << endl << endl;

        cout << "Please enter the live load applied to the slab.  This " << endl;
        cout << "value should be in units of psf." << endl << endl;

        cin >> liveload;
        cout << endl << endl;

        cout << "Please enter the unit weight of concrete being used.  " << endl;
        cout << "This value should be in units of pcf and should be a " << endl;
        cout << "whole number." << endl << endl;

        cin >> unitweight;
        cout << endl << endl;

        deadload = unitweight * thickness/12;
        sustainedload = deadload + superimposeddl + 0.1*liveload;
        maxload = deadload + superimposeddl + liveload;

        wconstruction = (deadload * b1)/12;
        wsustained = (sustainedload * b1)/12;
        wmax = (maxload * b1)/12;



        // Construction Scheme:

        cout << "This program assumes that one level of shoring is " << endl;
        cout << "being used.  Please enter the number of levels of " << endl;
        cout << "reshoring being used.  This program accommodates " << endl;
        cout << "construction scenarios involving two, three, or " << endl;
        cout << "four levels of reshoring.  Be sure to enter a 2, " << endl;
        cout << "3, or 4 from the keypad at this time." << endl << endl;

        cin >> reshoring;
        cout << endl << endl;


        cout << "Please enter the number of days between the casting " << endl;
        cout << "of floors (the length of the casting cycle).  Be sure " << endl;
        cout << "to enter a whole number." << endl << endl;

        cin >> castingcycle;
        cout << endl << endl;


        cout << "Please enter the number of days after casting at " << endl;
        cout << "which formwork is removed.  Be sure to enter a whole " << endl;
        cout << "number." << endl << endl;

        cin >> shoreremoval;
        cout << endl << endl;
```

```
// Load Ratios & Loading-Time History:

R[0] = 0.0;

for (counter = 1; counter < 25; counter++)
	R[counter] = 1.0;

for (counter = 4; counter <= (4 + 4*reshoring); counter = counter + 4)
{
	R[counter - 1] = 1.0 + 1.0 / (float(reshoring) + 1.0);
	R[counter] = R[counter - 1];
}

day[0] = 0;
day[1] = shoreremoval;
day[2] = castingcycle;

for (counter = 4; counter <= (4*reshoring); counter = counter + 4)
{
	day[counter - 1] = castingcycle * (counter/4);
	day[counter] = day[counter - 1] + shoreremoval;
	day[counter + 1] = day[counter - 1] + shoreremoval;
	day[counter + 2] = day[counter - 1] + castingcycle;
}

day[counter - 1] = day[counter - 2];
day[counter] = day[counter - 2] + shoreremoval;
day[counter + 1] = day[counter - 2] + shoreremoval;

if (reshoring==2)
{
	day[14] = 30 * int((3*castingcycle + 30) / 30);
	for (counter = 15; counter < 19; counter++)
		day[counter] = day[counter - 1] + 30;

	day[19] = 365;			// one year
	day[20] = 730;			// two years
	day[21] = 1095;			// three years
	day[22] = 1460;			// four years
	day[23] = 1825;			// five years
	day[24] = 1825;			// five years
}

if (reshoring==3)
{
	day[18] = 40 * int((4*castingcycle + 40) / 40);
	for (counter = 19; counter < 21; counter++)
		day[counter] = day[counter - 1] + 40;

	day[21] = 365;			// one year
	day[22] = 1095;			// three years
	day[23] = 1825;			// five years
	day[24] = 1825;			// five years
}

if (reshoring==4)
{
	day[22] = 365;			// one year
	day[23] = 1825;			// five years
	day[24] = 1825;			// five years
}

w[0] = R[0] * 1.1 * 1.1 * wconstruction;
deltaw[0] = 0.0;

for (counter = 1; counter <= (4 + 4*reshoring); counter = counter + 1)
	w[counter] = R[counter] * 1.1 * 1.1 * wconstruction;

for(counter = (5 + 4*reshoring); counter < 24; counter++)
	w[counter] = R[counter] * wsustained;
```

```
        w[24] = R[24] * wmax;

        for(counter = 1; counter < 25; counter++)
                deltaw[counter] = w[counter] - w[counter - 1];
}



if (inputselector == 1)
{

        // Prescribed Loading-Time History:

        cout << "The program is designed to develop a loading-time " << endl;
        cout << "history based upon 25 pairings of loading and time " << endl;
        cout << "input.  Time must be input in days, while loading is " << endl;
        cout << "input in units of pounds per square foot." << endl << endl;

        cout << "Please input the first time increment." << endl << endl;

        cin >> day[0];
        cout << endl << endl;

        cout << "Now input the corresponding load magnitude." << endl << endl;

        cin >> loadinput[0];
        cout << endl << endl;
        w[0] = (loadinput[0]*b1)/12;

        for(counter = 1; counter < 24; counter++)
        {
                cout << "Input the next time increment." << endl << endl;

                cin >> day[counter];
                cout << endl << endl;

                cout << "Input the next load increment." << endl << endl;

                cin >> loadinput[counter];
                cout << endl << endl;
                w[counter] = (loadinput[counter]*b1)/12;
        }

        cout << "Please input the last time increment." << endl << endl;

        cin >> day[24];
        cout << endl << endl;

        cout << "Now input the corresponding load magnitude." << endl << endl;

        cin >> loadinput[24];
        cout << endl << endl;
        w[24] = (loadinput[24]*b1)/12;

        deltaw[0] = 0.0;
        for(counter = 1; counter < 25; counter++)
        {
                deltaw[counter] = w[counter] - w[counter - 1];
        }
}



// Time-Dependent Concrete Properties:

for(counter = 1; counter < 25; counter++)
{
        fprimec[counter] = (day[counter] / (4.00 + 0.85 * day[counter])) * fc;

        E[counter] = 57000 * sqrt(fprimec[counter]);
}
```

```
// Instantaneous Deflection:

delta[0] = 0.0;
sumdelta[0] = delta[0];

for(counter = 1; counter < 25; counter++)
{
        delta[counter] = k / 384.0 * deltaw[counter] / E[counter] *
                              longdimension * 12 * longdimension * 12 *
                              longdimension * 12 * longdimension * 12 /
                              mominertiacol;

        sumdelta[counter] = sumdelta[counter - 1] + delta[counter];
}



// Long-Time Multiplier:

cout << "Please input the average creep multiplier." << endl;
cout << endl;

cin >> avgcreepmult;

cout << endl << endl;

cout << "Please input the creep recovery factor.  This value " << endl;
cout << "should be input as a decimal and will be used to modify " << endl;
cout << "the values of long-time multipliers computed for negative " << endl;
cout << "loading changes." << endl << endl;

cin >> creeprecovery;

cout << endl << endl;

cout << "The creep correction factor for loading age may either be " << endl;
cout << "computed using the ACI 209 recommended equation or the " << endl;
cout << "Ghosh recommended equation.  If you would like to use the " << endl;
cout << "ACI equation input a numerical 1.  Input a numerical 2 " << endl;
cout << "for the Ghosh equation.  Please enter the appropriate " << endl;
cout << "number now." << endl << endl;

cin >> loadingage;
cout << endl << endl;

if (loadingage == 1)
{
        cout << "Enter a numerical 1 if the concrete was moist cured.  " << endl;
        cout << "Enter a numerical 2 if the concrete was steam cured.  " << endl;
        cout << "Please enter the appropriate number now." << endl << endl;

        cin >> moistorsteam;
        cout << endl << endl;
}

cout << "If you would like to include the creep correction factor " << endl;
cout << "for relative humidity please input a numerical 1.  To " << endl;
cout << "exclude the relative humidity creep correction factor " << endl;
cout << "please input a numerical 2.  Please input the appropriate " << endl;
cout << "number now." << endl << endl;

cin >> includehumidity;
cout << endl << endl;

if (includehumidity == 1)
{
        cout << "Please input the relative humidity as a percent." << endl;
        cout << endl;
```

```
        cin >> relhumidity;
        cout << endl << endl;

        rhcorrection = 1.27 - (67/10000 * relhumidity);
}

if (includehumidity == 2)
        rhcorrection = 1.0;

for(counter = 0; counter < 25; counter++)
{
        lambda[counter] = 0.0;
        cumcreepdefl[counter] = 0.0;
}

for(counter = 0; counter < 25; counter++)
{
        moistcured[counter] = -0.118;
        steamcured[counter] = -0.094;
        ghosh[counter] = -0.25;
        raisepower[counter] = 0.6;
}

for(loadcounter = 1; loadcounter < 25; loadcounter++)
{
        for(timecounter = loadcounter+1; timecounter < 25; timecounter++)
        {
                if (loadingage == 1 && moistorsteam == 1)
                lacorrection =
                1.25 * pow(float(day[loadcounter]),moistcured[timecounter]);

                if (loadingage == 1 && moistorsteam == 2)
                lacorrection =
                1.13 * pow(float(day[loadcounter]),steamcured[timecounter]);

                if (loadingage == 2)
                lacorrection =
                2.30 * pow(float(day[loadcounter]),ghosh[timecounter]);

                lambda[timecounter] = avgcreepmult * lacorrection * rhcorrection *
                               pow(float(day[timecounter] - day[loadcounter]),
                               raisepower[timecounter]) / (10 +
                               (pow(float(day[timecounter] - day[loadcounter]),
                               raisepower[timecounter])));

                if (deltaw[loadcounter] <= 0)
                        lambda[timecounter] = creeprecovery * lambda[timecounter];

                if (day[timecounter] == day[loadcounter])
                        lambda[timecounter] = 0.0;



                // Long-Time Deflection:

                creepdefl[timecounter] = (k * longdimension * 12 *
                                       longdimension * 12 * longdimension * 12 *
                                       longdimension * 12 * deltaw[loadcounter] *
                                       lambda[timecounter]) /
                                       (384*mominertiacol*E[loadcounter]);

                cumcreepdefl[timecounter] = cumcreepdefl[timecounter] +

creepdefl[timecounter];
        }
}
```

```
// Column Strip Deflection:

for(counter = 0; counter < 25; counter++)
        colstripdefl[counter] = sumdelta[counter] + cumcreepdefl[counter];



// Modified Middle Strip Correction Factors:

kss = 0.65;
kwf = (longdimension/2) / (longdimension - shortdimension/2);


k = kbcmid * kss * kwf * kdp;


if (inputselector == 0)
{

        // Modified Middle Strip Loads:

        wmax = (maxload * b2)/12;
        wsustained = (sustainedload * b2)/12;
        wconstruction = (deadload * b2)/12;



        // Modified Middle Strip Loading-Time History:

        for (counter = 1; counter <= (4 + 4*reshoring); counter = counter + 1)
                w[counter] = R[counter] * 1.1 * 1.1 * wconstruction;

        for(counter = (5 + 4*reshoring); counter < 24; counter++)
                w[counter] = R[counter] * wsustained;

        w[24] = R[24] * wmax;

        for(counter = 1; counter < 25; counter++)
                deltaw[counter] = w[counter] - w[counter - 1];
}



if (inputselector == 1)
{

        // Prescribed Loading-Time History:

        for(counter = 0; counter < 25; counter++)
        {
                w[counter] = (loadinput[counter]*b2)/12;
        }

        deltaw[0] = 0.0;
        for(counter = 1; counter < 25; counter++)
        {
                deltaw[counter] = w[counter] - w[counter - 1];
        }
}
```

```
// Middle Strip Deflection:

delta[0] = 0.0;
sumdelta[0] = delta[0];

for(counter = 1; counter < 25; counter++)
{
        delta[counter] = k / 384.0 * deltaw[counter] / E[counter] *
                              shortdimension * 12 * shortdimension * 12 *
                              shortdimension * 12 * shortdimension * 12 /
                              mominertiamid;

        sumdelta[counter] = sumdelta[counter - 1] + delta[counter];
}

for(counter = 0; counter < 25; counter++)
{
        lambda[counter] = 0.0;
        cumcreepdefl[counter] = 0.0;
}

for(loadcounter = 1; loadcounter < 25; loadcounter++)
{
        for(timecounter = loadcounter+1; timecounter < 25; timecounter++)
        {
                if (loadingage == 1 && moistorsteam == 1)
                lacorrection =
                1.25 * pow(float(day[loadcounter]),moistcured[timecounter]);

                if (loadingage == 1 && moistorsteam == 2)
                lacorrection =
                1.13 * pow(float(day[loadcounter]),steamcured[timecounter]);

                if (loadingage == 2)
                lacorrection =
                2.30 * pow(float(day[loadcounter]),ghosh[timecounter]);

                lambda[timecounter] = avgcreepmult * lacorrection * rhcorrection *
                              pow(float(day[timecounter] - day[loadcounter]),
                              raisepower[timecounter]) / (10 +
                              (pow(float(day[timecounter] - day[loadcounter]),
                              raisepower[timecounter])));


                if (deltaw[loadcounter] <= 0)
                        lambda[timecounter] = creeprecovery * lambda[timecounter];

                if (day[timecounter] == day[loadcounter])
                        lambda[timecounter] = 0.0;

                creepdefl[timecounter] = (k * shortdimension * 12 *
                                      shortdimension * 12 * shortdimension * 12 *
                                      shortdimension * 12 * deltaw[loadcounter] *
                                      lambda[timecounter]) /
                                      (384*mominertiamid*E[loadcounter]);

                cumcreepdefl[timecounter] = cumcreepdefl[timecounter] +
                                                      creepdefl[timecounter];
        }
        cout << endl;
}

for(counter = 0; counter < 25; counter++)
        midstripdefl[counter] = sumdelta[counter] + cumcreepdefl[counter];
```

```
	// Total (Mid-Panel) Deflection:

	for(counter = 0; counter < 25; counter++)
		totaldefl[counter] = colstripdefl[counter] + midstripdefl[counter];



	// Output:

	cout << "Here are the final results for total deflection versus time:";
	cout << endl << endl << endl;
	cout << "Time          Deflection (inches)" << endl << endl;

	for(counter = 0; counter < 25; counter++)
	{
		cout << day[counter] << "             " << totaldefl[counter] << endl;
	}

	cout << endl << endl;



	return 1;
}
```